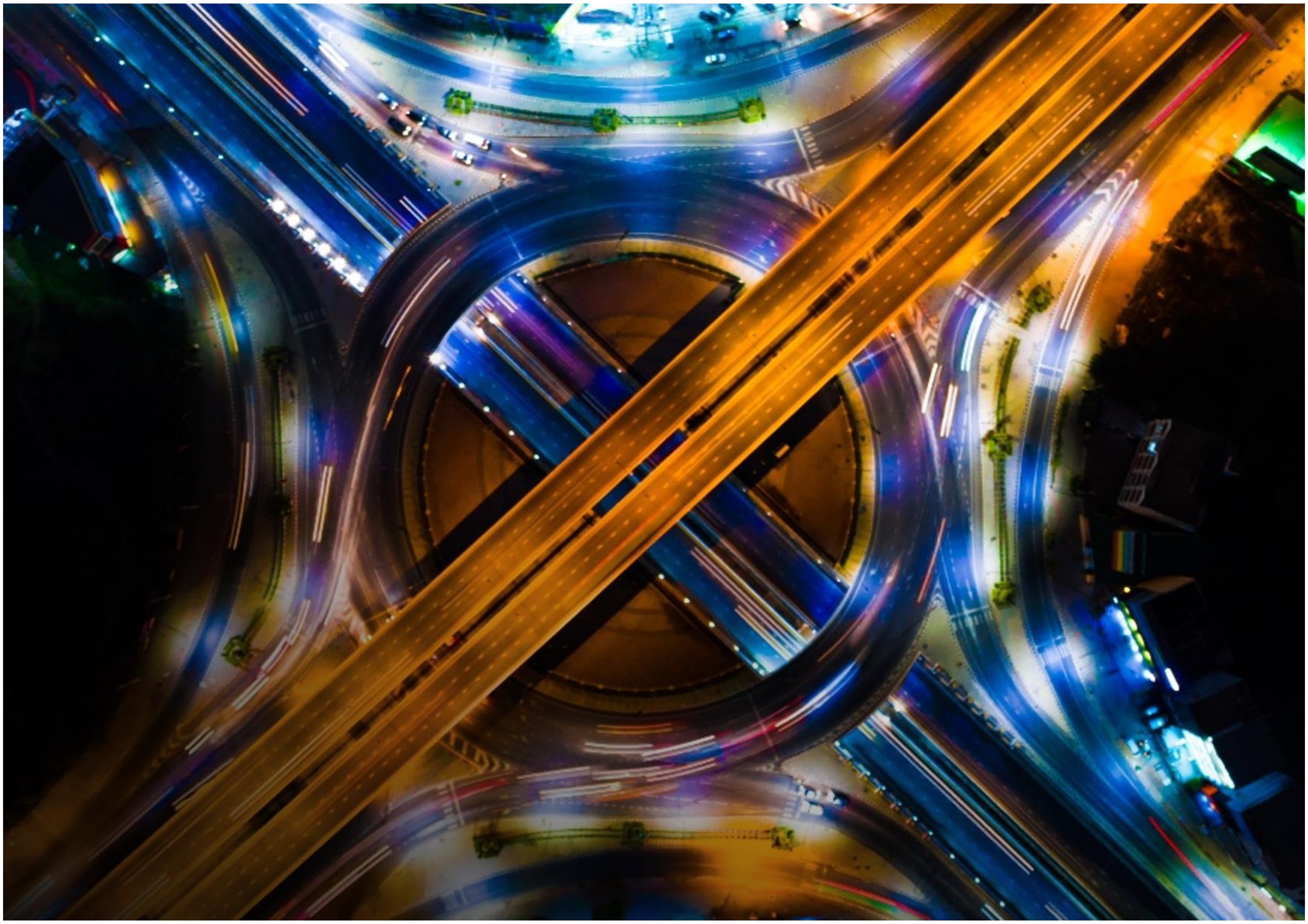
Deloitte.



Observability in GPS: Knowing more, recovering faster, achieving cloud agility

Contents

Introduction

Not knowing in IT can cost you time, money, and your reputation. A delay in root cause analysis during a system outage or failure to digest the complexities and interdependencies of modern applications will lengthen recovery time and drive up the cost of cloud migrations and modernizations. Issues like these can have real world consequences in government especially for critical services such as health care and citizens' benefits. Government mission leaders and agency mission leaders never want to hear "we didn't know about that dependency in our migration" or "we are still searching for a root cause of the outage".

Your IT world has never been more complex, and managing complexity requires centralized monitoring with the tools of observability like automated application service mapping and anomaly detection. Observability removes ambiguity to know what is at fault, unravels complexities to know how systems connect and pinpoints areas for efficiency gains to know that you are maximizing your IT spend which can have far reaching business and mission benefits.

What if you could increase your release quality and velocity, never miss an end user impacting anomaly or event and accelerate your cloud modernization initiatives all while managing costs?

This paper introduces the benefits of this capability, illustrates how applying it helped two GPS clients, and outlines the common outcomes and lessons learned and how to approach applying observability.

Benefits of Observability

All technical and managerial teams can benefit from more data on the behaviors and statuses of their systems. Timely, accurate, and digestible data can be used by these teams to avoid outages, improve migrations, and better plan for future evolving needs. Further, as an organization increases the number of applications, or increases the footprint/complexity of architectures, the Information Technology (IT) operations require better tools and processes to ensure that user access to vital services remains predictable and consistent.

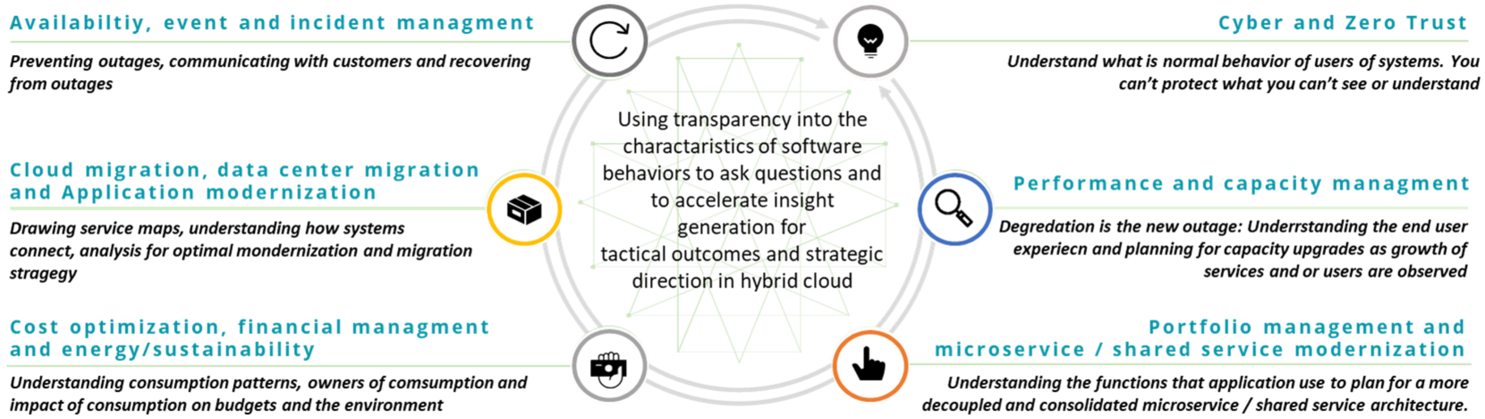
Observability can be a solution to many problems as it enables the collection and analysis of a massive amount of data that has wide applicability. Government agencies have considered using this data to gain insight to help increase the availability of systems, plan for cloud migrations, optimize cloud consumption costs, supplement dedicated Security Information and Event Management (SIEM) systems, help manage performance, and even to understand the energy impact of IT compute and storage. The most common use case among government agencies is using monitoring and observability to increase system availability and improve customer communications around availability.

Whether state or federal, and whether they serve citizens, constituents, internal customers, or interagency customers, the Chief Information Officer (CIO) organization's highest priority is the availability of services. However, maintaining high availability can become costly and out of reach for many organizations as they grapple with modernizations, migrations, and operating at scale. Multiple applications, hosting solutions, and nodes under management increase the complexity and points of administration that prevent engineers and administrators from devoting time to project work associated with modernization initiatives. Further, maintaining regulatory compliance with the Sarbanes-Oxley Act (SOX), Health Insurance Portability and Accountability Act (HIPAA), National Institute of Standards and Technology (NIST) 800.53 related requirements, and general security compliance creates more overhead that distracts from improvements needed to increase availability.

While goals for availability monitoring prefer having a "single pane of glass", preventing failures through observability, reducing mean time to recover, and even "self-healing" are common threads, many GPS CIOs, and IT leaders do not know where to start, and have difficulty justifying the infrastructure and licensing costs for the capability. A key question is how they can reap the benefits of data management and transparency while controlling costs and maturing the capabilities needed for self-healing and single pane of glass dashboards.

Most CIOs and IT leaders understand that while necessary for many dimensions of planning and management, creating human-generated management reports requires overhead from analysts and systems administrators who could otherwise be managing or executing projects. That said, human-initiated collection and analysis will often serve as the basis for the business case and implementation for automated collection and dashboard-based analysis in the future. Developing and refining periodic deliverables to include defining the metrics, meta tags (I.e., cloud tags to associate say cost centers with application workloads), stakeholders, and desired analysis/presentations create a foundation for when in future phases, collection and code are institutionalized through technology investments.

Figure 1: Multiple Use Cases for Observability



Availabiltiy, event and incident managment

Preventing outages, communicating with customers and recovering from outages

Cyber and Zero Trust

Understand what is normal behavior of users of systems. You can't protect what you can't see or understand

Using transparency into the charactaristics of software behaviors to ask questions and to accelerate insight generation for tactical outcomes and strategic direction in hybrid cloud

Cloud migration, data center migration and Application modernization

Drawing service maps, understanding how systems connect, analysis for optimal mondernization and migration stragegy

Performance and capacity managment

Degredation is the new outage: Underrstanding the end user experiecn and planning for capacity upgrades as growth of services and or users are observed

Cost optimization, financial managment and energy/sustainability

Understanding consumption patterns, owners of comsumption and impact of consumption on budgets and the environment

Portfolio management and microservice / shared service modernization

Understanding the functions that application use to plan for a more decoupled and consolidated microservice / shared service architecture.

Centralization of monitoring is a key institutional steppingstone to realizing the benefits of observability. Often, monitoring tools are sold or included with commercial products which create decentralized monitoring and fragmentation. While these tools will provide the benefits of integrating configuration management with monitoring, they often are suboptimal tools for cross-product monitoring and custom software end-to-end observability. The process of centralization and consolidation of monitoring forces tool rationalization and capability analysis where vendor tools and custom tools often show higher costs and lower benefits than modern observability tools.

At this point, it is worth distinguishing between centralized monitoring and observability. Both capabilities focus on collecting 4 different types of data often referred to as MELT (Figure 2).

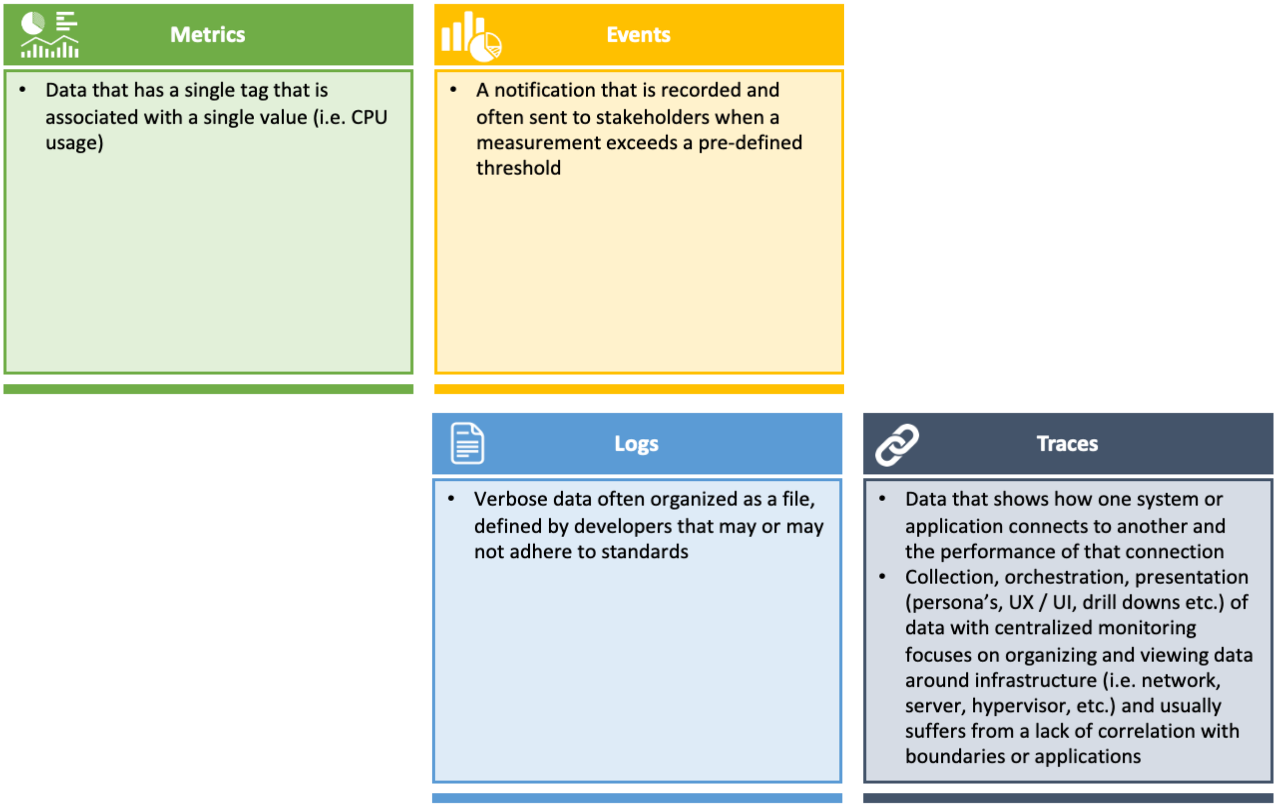
Centralized monitoring is using non-vendor-specific tools to collect data in a central location for the event management team and technical teams to use for notification and troubleshooting of outages.

Observability is when a critical mass of data is collected to inform the characteristics of software that constitute normal behavior or behavior by design compared to anomaly behavior or suboptimal behavior for a given problem set (I.e. energy usage, availability, migration, performance etc.). Observability is the response to a renewed focus on the user experience where degradation poses a similar business problem to an outage. “Degradation is the new outage” represents customer desires for a seamless user experience and in the absence of such a willingness, to abandon a cart, process, or experience frustration. The new frontier that observability seeks measure from MELT data is what google terms the “golden signals” of latency, saturation, and error rates, where tools supporting synthetic transactions, collection of application telemetry data, and service maps created from machine learning and AI (Artificial Intelligence).

These same data on what is “normal” behavior for use of systems can also serve as the foundation for zero-trust systems that depend on detecting anomaly behaviors to prevent exploits.

In the journey to observability, not everything is a one size fits all. In two government success stories: A U.S. Government Agency had DIY tools and preferred to keep licensing and data storage costs low; A U.S. State healthcare eligibility and enrollment system needed a solution to maintain availability while also supporting application modernization.

Figure 2: Four Types of Data (MELT)



Metrics

- Data that has a single tag that is associated with a single value (i.e. CPU usage)

Events

- A notification that is recorded and often sent to stakeholders when a measurement exceeds a pre-defined threshold

Logs

- Verbose data often organized as a file, defined by developers that may or may not adhere to standards

Traces

- Data that shows how one system or application connects to another and the performance of that connection
- Collection, orchestration, presentation (persona’s, UX / UI, drill downs etc.) of data with centralized monitoring focuses on organizing and viewing data around infrastructure (i.e. network, server, hypervisor, etc.) and usually suffers from a lack of correlation with boundaries or applications

Two Success Stories with Common Threads:

Increasing availability of health identity services in the U.S. Government

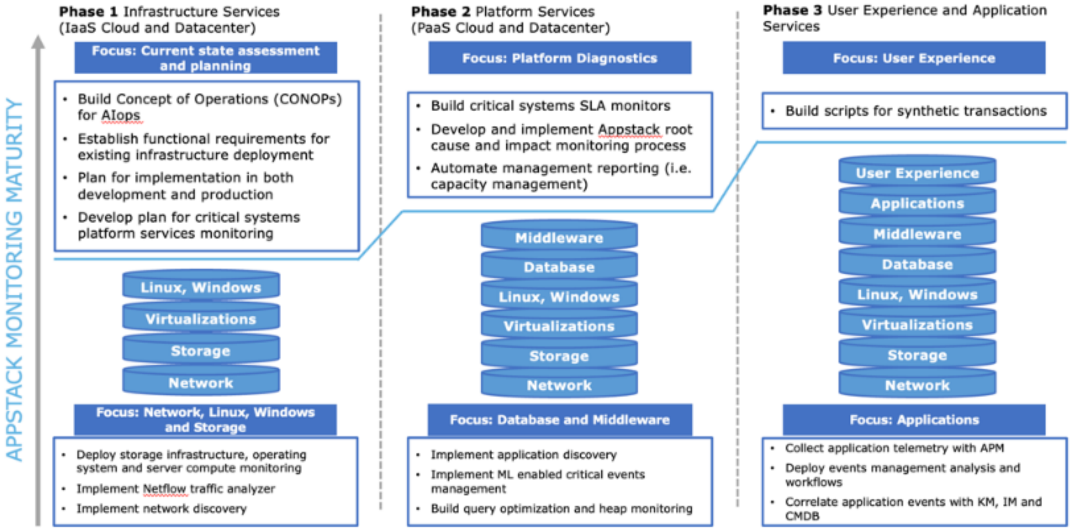
When Deloitte took over the day-to-day operations at an agency that provided identity services for health care, availability was the IT leadership's top priority. Having more than 400 applications, 2500 infrastructure nodes, 3-5k users, and multiple interagency customer teams under management, leadership knew that centralized monitoring was needed to reduce the time to information needed to prevent outages, plan for capacity, and recover from outages quickly.

Due to budget constraints, the client had decided to reduce the use of commercial off-the-shelf monitoring tools and replace them with custom DIY tools, while these tools did not require ongoing licensing costs, they failed to provide little more than a general up or down the status for applications. This pendulum swing of high licensing costs with questionable value to low licensing costs that only provided okay capabilities had left them wanting to build a different future that balanced the need for insight and information from its IT systems with managing the costs of commercial software.

They decided that they needed a new approach with a new contract that included a concept of the Integrated operational dashboard or IOD for short. The IOD requirement included a single pane of glass view of all operations, automatic data collection and reporting, and multiple personas from technician to manager to executives. As the concept of IOD evolved, it came to include concepts around observability, where information is turned into insights that can be used for discovery, service mapping, and predictive capabilities for capacity planning and to avoid outages. As the hosting evolved, the requirements for IOD evolved to include multi-cloud and hybrid-cloud observability, the client recognized that observability data would help them with the financial management of cloud assets, migration of applications to the cloud, and transaction to show the impact issues on business metrics. Further, they recognized that their ambitions for more automation using ServiceNow, their desire for automated builds and patching with DevOps, and leveraging Infrastructure as code solutions; could be tied together using real-time data found in the IOD.

The Deloitte team needed to start with where the client was. Deloitte established an event management team and an incident management team which was coined the Virtual Network Operations Center (vNOC). We needed to give the team what we had as soon as we had it and simplify the means to get it. Most of the non-DIY systems in place were vendor-based systems with the remaining being from Elactic.co for log capture. In the first phase, Deloitte used Application Programming Interface (API) queries from a Business Intelligence (BI) platform to consolidate the distributed data into application groups that could be displayed on dashboards using a BI platform, JavaScript, and D3 libraries (for visualizations). While this presented minimum functional product to the vNOC, and a quick win, phase 2 was about maturing the centralized capability, performing a tools rationalization, and maturity of the capability at the collection and orchestration level. Phase 3 was expanding the elastic product to monitor containerized workloads, Application Performance Management (APM), and building multi-cloud, hybrid monitoring solutions that included Infrastructure as a service (IaaS), Platform as a service (PaaS) APIs, native cloud builds, and cost analysis.

Throughout the journey of building technology, enabling the vNOC, and building a continuous improvement process, Deloitte increased the number of incidents avoided through active monitoring and reducing the time to recover from outages address the client's goals for availability. The people, process, and technology helped isolate the problem faster, bring the right incident response team together sooner and communicate situational awareness broader to increase the availability of services to customers.


APPSTACK MONITORING MATURITY
Phase 1 Infrastructure Services (IaaS Cloud and Datacenter)
Focus: Current state assessment and planning
Linux, Windows
Virtualizations
Storage
Network
Focus: Network, Linux, Windows and Storage
Phase 2 Platform Services (PaaS Cloud and Datacenter)
Focus: Platform Diagnostics
Middleware
Database
Linux, Windows
Virtualizations
Storage
Network
Focus: Database and Middleware
Phase 3 User Experience and Application Services
Focus: User Experience
User Experience
Applications
Middleware
Database
Linux, Windows
Virtualizations
Storage
Network
Focus: Applications

- Build Concept of Operations (CONOPs) for AIops
- Establish functional requirements for existing infrastructure deployment
- Plan for implementation in both development and production
- Develop plan for critical systems platform services monitoring

- Deploy storage infrastructure, operating system and server compute monitoring
- Implement Netflow traffic analyzer
- Implement network discovery

- Build critical systems SLA monitors
- Develop and implement Appstack root cause and impact monitoring process
- Automate management reporting (i.e. capacity management)

- Implement application discovery
- Implement ML enabled critical events management
- Build query optimization and heap monitoring

- Build scripts for synthetic transactions

- Collect application telemetry with APM
- Deploy events management analysis and workflows
- Correlate application events with KM, IM and CMDB

Figure 3: Building Observability in a Hybrid Cloud

Maintaining Availability while Modernizing Applications at a State Health Exchange

As part of the affordable cares act of 2010 state governments were required to establish healthcare insurance exchanges for individuals and small businesses. Deloitte took over the support of a State Government's automated system that supports account creation, consumer application, eligibility rules, and health plan selection for an insurance affordability program that launched in June 2012. Deloitte took over operations and executed a "lift and shift" project where infrastructure was virtualized and transferred to the Amazon Cloud. While this initiative successfully converted the client's capital expenditures (CAPEX) spend to operational expenditures (OPEX), there remained a desire to control costs, increase availability, and increase the flexibility of their applications. The client asked Deloitte to rearchitect their monolithic applications and establish a tech stack that supports containerization, microservices, and cloud-native technologies.

The Deloitte team needed to move quickly from the lift and shift to a more modern architecture. The client was accruing costs in the cloud without really capturing the benefits of the cloud. As part of the technology modernization effort, a monolith app was broken into multiple microservices using the 'Strangler Fig Pattern.' With so many microservices interacting with each other and with external entities, the client needed a solution to understand dependencies, minimize blind spots and process data loads coming from many different directions.

Most of the existing monitoring solutions were highly focused on vendor-specific solutions that showed how their products were performing in the tech stack. For example, Oracle Enterprise Monitor would show how fast Oracle processed a request but did not show the overall end-to-end user experience for the request. These tools are not equipped to monitor across the stack to show the overall performance and the overall end-user experience. Deloitte used a 2-step approach to first, focus on observability for availability by addressing blind spots and issues with monitoring and observability tooling and second, use the data collected from the new tools and the new tools themselves to support the client's modernization strategy.

As a first step in the solution, Deloitte introduced Dynatrace, a leading observability platform, creating observability to establish end to end performance and availability monitoring for distributed applications. Vendor tools and the lack of an observability tool to highlight the customer experience and understand the business impact of degradations and outages left for areas of improvement. Thus, the priority for end-to-end monitoring was to build a collection of data that could show the business impact of IT service quality. Tying application performance to business outcomes was a critical need given the national attention on health exchange services. Deloitte implemented a solution that showed:

- Real-time business KPIs (Key Performance Indicators), to protect against abandonment of enrollment and quickly get to the root cause of customer issues causing these problems
- Real-time visibility into the user experience, app performance, errors, and new features or releases that impact the business KPIs
- A common view of business metrics–including page names and audience segments–through a shared business lens
- An AI engine that alerts leaders to business anomalies and identifies the root cause for the issue.
- The trend in resource utilization, enabling right-sizing workloads and cost optimization

Centralized, automated alerts that are intelligently filtered are essential to monitoring. They allow operations teams to spot problems anywhere in the infrastructure, rapidly identify causes and minimize service degradation and disruption. However, too many alerts and real problems can often get lost in a sea of noisy alarms. Thus, Deloitte implemented intelligent alerting where we identified various levels of alerts and the actions needed in response, centralized the alert generation tool instead of configuring alerts on all tools and designed with future opportunities to automate alert response in preparation for a future self-healing ecosystem.

This preparation included realizing the value from the steppingstones to applying AI and building a self-healing ecosystem. The systems and platforms used to deliver modern services and applications generate a lot of data for the centralized observability platform to collect, even using a constrained set of signals where we targeted a subset of critical applications. Analytics and dashboards are required when starting "small" to make this data useful as a critical mass of data is accumulated to think "big" with AI-based solutions identifying to assess health and performance, as well as analysis to make the data useful. Ultimately, building analytics and thinking big helped Deloitte deliver **Adaptive thresholds for anomaly detection**: Determining what "normal" is and notifying based on deviations from that, **Machine learning**: Constructing a complex model of system behavior, correlating events and data using the model, with reinforcement such that it "learns" over time and **Predictive analytics**: Identifying impending events and proactive responds, such as

provisioning additional resources before exhaustion to reduce downtime.
As a second step to support a strategic transition to cloud modernization, containerization and microservices, Deloitte built observability into the Kubernetes platform that would serve as the foundation of its modernization strategy. OpenShift from Red Hat was chosen for the Kubernetes container orchestration platform. It was especially important to get complete visibility into the health of clusters and containerized workloads. The typical instrumentation process in a non-containerized environment (an agent residing in the user space of VM (Virtual Machine) or host) does not work well for the containers because of its small, independent processes and low dependencies. Further, data such as container run time and Kubernetes metrics such as running pods and node resource utilization can be used for different use cases and runbooks. In this case, multiple tools were used, and the Deloitte team needed to integrate these tools to achieve a centralized repository from which to perform correlation analysis, service mapping and centralized alerting.

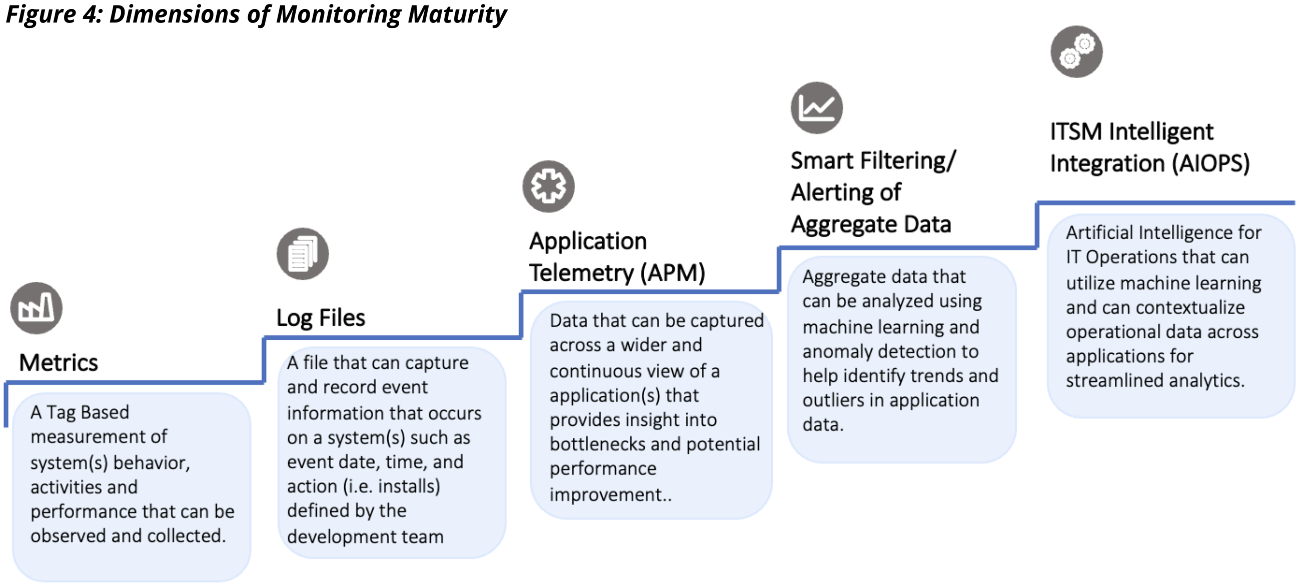
A combination of market-leading tools was used with open-source tools bundled with the container platform for complete visibility of the containerized platform. This combination was needed since breaking down monolith applications into containers and microservices without the benefit of end-to-end tracing of user requests across services and automatic services mapping is a daunting task. Further, after the move to microservices traditional methods alone cannot connect performance and dependencies across the now-distributed architecture.

In this phase two Deloitte focused on:

- Understanding the most frequently used endpoints as a function of time. This allowed teams to see if anything noticeable has changed in the usage of services, whether it be due to a design change or a user change.
- Finding bottlenecks in service delivery and digging deeper to identify any associated problems, or at the very least, pointing to areas that need the most optimization in your system.
- Distributed Tracing: This is establishing the ability to trace service calls through the entire system. While typically used by developers, this type of profiling helps understand the overall user experience while breaking information down into infrastructure and application-based views of your environment.

Implementing these deep monitoring tools allows operations teams to perform end-to-end tracing for calls happening across multiple components and the ability to inspect them for the next phase of their modernization planning and design, which was a move to hyperscale management of the Kubernetes platform and exploring serverless implementations.

Figure 4: Dimensions of Monitoring Maturity



A Tag Based measurement of system(s) behavior, activities and performance that can be observed and collected.

A file that can capture and record event information that occurs on a system(s) such as event date, time, and action (i.e. installs) defined by the development team

Data that can be captured across a wider and continuous view of a application(s) that provides insight into bottlenecks and potential performance improvement..

Aggregate data that can be analyzed using machine learning and anomaly detection to help identify trends and outliers in application data.

Artificial Intelligence for IT Operations that can utilize machine learning and can contextualize operational data across applications for streamlined analytics.

Key Outcomes & Lessons Learned

Building observability and unlocking its value is a transformative journey that can yield new insights about the strengths and weaknesses of applications, infrastructures, and architectures as well as the potential for improvements. Deloitte's experience with hybrid cloud operations, native cloud modernizations as well as data and AI helped these public sector clients realize measurable improvements to availability and agility in their modernization projects.

In both cases, Deloitte took steps to discover, plan and execute the transformation of the people, process, and technology needed to create operational efficiencies through observability. The results: increased availability, more manageable migrations, higher quality releases, and more optimal architecture that included:

- A single pane of glass monitoring solution that enables developers, operations, security, and business teams to collaborate on shared data using the same view to enable early detection of problems pointing to the problem areas and supports deep analysis of telemetry data.
- Real-time views into container workloads with actionable alerts using machine learning "noise" filters that map to business services rather than just the technology equipment in the enterprise for impact analysis and communications
- Automated service maps of applications that supported modernization efforts and break down monolithic services and large migrations into microservices and manageable migrations
- Nonoverlapping, optimized solutions where multiple monitoring tools are rationalized and reduced in favor of the right tool for the specific use case and a bias towards centralization to reduce costs overheads and support end-to-end monitoring
- Self-Healing actions where tickets are created automatically from event and pre-tested healing routines can act on unhealthy Kubernetes containers and traditional processes.

The results are reduced mean time to recovery, fewer incidents and decreased risk for the business and customer experience degradation.

Lessons Learned:

Data enabled and data centric operations using observability is a journey that requires the right set of people on the core implementation team, changing the processes that other teams use, and keeping a balanced perspective on the technology to avoid excessive costs without correlated benefits.

People:
The core implementation team needs to have a set of certain skills and abilities:

- Data skills – Data that is ingested needs to be indexed for user groups and interests. The ease with which this is done varies depending on the vendor.
- Infrastructure skills – Observability data can be massive and unyielding, and memory-starved data shards or indexers can make for a very bad day. The infrastructure needs to be built and/or estimated properly whether it a Software as a Service (SaaS), IaaS, or virtualized deployment, to avoid performance issues and cost overruns.
- UX (User Experience) / UI (User Interface) skills – If dashboards cannot be digested and understood quickly, they will not be used. If they cannot be accessed easily, they will not even be viewed.
- Implementation skills – While many vendors ease the deployment of agents, some do not and sometimes agents are not desirable. The core team needs to understand the intricacies of deploying agents, configuring Simple Network Management Protocol (SNMP), allowing for IP (Internet Protocol) scanning and polling on the network and plugging into other data sources.
- Product and Customer Relationship Management skills – Given that the value of the data is derived from the use of the data, product roadmaps and customer feedback is needed to prioritize use cases, visualizations, data exchanges, etc.

While not core to the observability team management reporting skills, old fashion paper reports and deliverables with human analysis are useful. Adjacent use case noted in the opening may not be satisfied with dashboards or automation alone. Contract and executive reporting always include a "so what" human analysis for a given context that can be enhanced by observability and data but cannot be replaced by it. Domain knowledge is needed to look at the data and understand what analysis might be needed by a given user group, especially if they are busy executives.

Process:
Changing the way other teams operate is needed to realize the benefits of observability. Shifting operational data to the "left" to enable observability-driven development yields the greatest impact for process changes. When we all think of observability for availability, we of course think of improving the information that our event managers , code optimization efforts use, and planning efforts use. But what if we could provide our development teams with greater operational awareness on how their code would react in various operational situations? Tighter integration of observability data with the development process leads to Observability-driven development (ODD). ODD uses data and tooling to observe the state and behavior of a

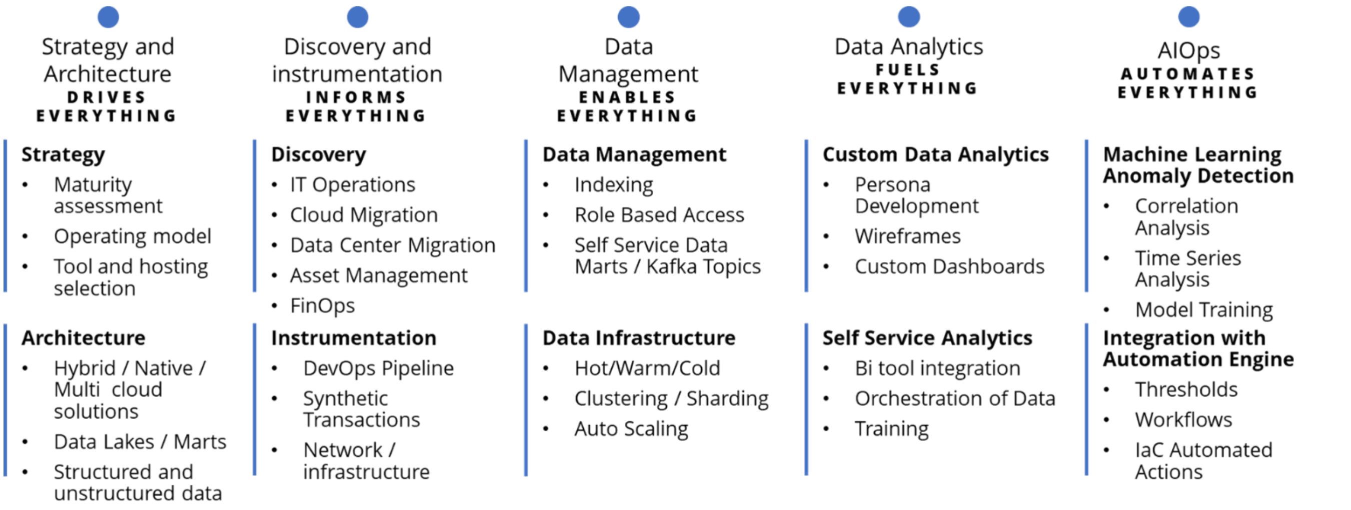
system before, during, and after development to learn more about its patterns of weakness. Having this information sooner in the development cycle allows for developers to tailor their code to the production environment they will be deployed in. For example, if a new production environment for an application includes an unreliable link, developers can introduce a time-out routine or if an environment is saturated during the day, a developer can create an overnight batch process for data that does not need to be real-time. In ODD, the development team and operations team are working with a single concept of understanding the performance of the application. Developers who understand the environment where their code will live develop better working code. When this happens, right sizing your Kubernetes platform becomes easier.

Technology:
A balanced approach is needed when selecting and building your observability platform. Many clients either focus on the “juice,” the data, or on the “squeeze,” the single pane of glass dashboards. For those who focus on the juice, data is collected for the sake of coverage and often using a premium platform resulting in runaway licensing, computing, and storage costs. For those who focus on the squeeze, they focus on wireframes and dashboard designs for the single pane of glass and find that the underlying data collected is not able to power the designs. A balanced approach to technology is needed. When working with your observability platform it is important to keep in mind the following:

1. Not all data needs to be real time data. Decide which systems are the core, complex systems that have critical mission impact and those that can have lower polling intervals due to simplicity or lower priority
2. Filter before ingesting if possible. This is more important for high-volume data like packet capture where systems can be purchased to filter traffic before sending it to the analysis engine in your observability platform and can even be enabled/disabled for a point-in-time solution.
3. Try to get the most out of your telemetry and trace data. Custom application log files, often created prior to APM (application performance monitoring) solutions either being implemented or existing, can usually be replaced by telemetry logs. Further, trace logs will also show bottlenecks in other parts of the stack that have traditionally been in the domain of SNMP provided metrics.

In all cases support for the platform is key. This includes vendor support and a team that is trained to effectively use the platform. For critical monitoring solutions, it is important to have a premium support agreement with the vendor. The monitoring stack needs to be highly available and any updates in the underlying tools need to be managed and planned carefully so that there is no disruption in the monitoring. If you are using an open-source monitoring tool for less critical workloads, make sure you have the deep expertise needed to tackle any production situation and consider commercial support that is available for most open-source solutions. For an implementation team, it is better to choose a tool that your team can use effectively than say a tool that has greater possibilities (i.e. lower costs or greater capability). Some integrations or exception scenarios will always be needed for the many use cases for observability data and your team needs to be able to navigate the platform with ease.

Figure 5: A Balanced Approach



Strategy and Architecture **DRIVES EVERYTHING**

Strategy
- Maturity assessment
- Operating model
- Tool and hosting selection

Architecture
- Hybrid / Native / Multi cloud solutions
- Data Lakes / Marts
- Structured and unstructured data

Discovery and instrumentation **INFORMS EVERYTHING**

Discovery
- IT Operations
- Cloud Migration
- Data Center Migration
- Asset Management
- FinOps

Instrumentation
- DevOps Pipeline
- Synthetic Transactions
- Network / infrastructure

Data Management **ENABLES EVERYTHING**

Data Management
- Indexing
- Role Based Access
- Self Service Data Marts / Kafka Topics

Data Infrastructure
- Hot/Warm/Cold
- Clustering / Sharding
- Auto Scaling

Data Analytics **FUELS EVERYTHING**

Custom Data Analytics
- Persona Development
- Wireframes
- Custom Dashboards

Self Service Analytics
- Bi tool integration
- Orchestration of Data
- Training

AIOps **AUTOMATES EVERYTHING**

Machine Learning Anomaly Detection
- Correlation Analysis
- Time Series Analysis
- Model Training

Integration with Automation Engine
- Thresholds
- Workflows
- IaC Automated Actions

Recommendations

Deloitte's extensive work with observability platforms and broad capabilities in cloud enablement helped these government clients realize increase their availability and accelerate their cloud migrations. Along the journey, Deloitte helped evolve their IT data management capabilities in presentation, automation, centralized monitoring, observability on the journey to AIOPS.

Along the journey, Deloitte helped these government clients deliver value early and often by focusing on the consumer of the data and the use cases they care about. For example, the availability management teams wanted fewer notifications and fewer screens meaning intelligent filtering and well-designed data experience, development teams want to be able to search all relevant data and design their own experience for their own use cases, and system owners want rapid straightforward communications for end users and other leaders.

Deloitte achieved this by:

- Building the right team
- Selecting the right tool or tools
- Starting with centralized monitoring
- Using periodic reports to build awareness and interest in the platform
- Grooming the observability backlog and prioritizing the right use cases
- Continually improvement the existing processes
- Building with a future AIOps solution and infrastructure as code solutions in mind where we connect observation to analysis and decision to action, all with limited human intervention

Deloitte's cloud services enable us to help clients achieve their cloud ambitions. Deloitte cloud offers end-to-end services to our clients and brings differentiated advice, skills, and assets. We bring deep capabilities in advising, implementing, and maintaining cloud architectures and technologies and delivering services across flexible economic models.

Contact Us

Bryan Pagliano
Senior Manager
Deloitte Consulting LLP
Email: bpagliano@deloitte.com
Tel: 1.571.858.0796

Rajesh Parab
Specialist Master
Deloitte Consulting LLP
Email: raparab@deloitte.com
Tel: 1.916.288.31

Deloitte.

Deloitte refers to one or more of Deloitte Touche Tohmatsu Limited (DTTL), its global network of member firms, and their related entities (collectively, the “Deloitte organization”). DTTL (also referred to as “Deloitte Global”) and each of its member firms and related entities are legally separate and independent entities, which cannot obligate or bind each other in respect of third parties. DTTL and each DTTL member firm and related entity is liable only for its own acts and omissions, and not those of each other. DTTL does not provide services to clients. Please see www.deloitte.com/about to learn more.

This publication contains general information only and Deloitte is not, by means of this publication, rendering accounting, business, financial, investment, legal, tax, or other professional advice or services. This publication is not a substitute for such professional advice or services, nor should it be used as a basis for any decision or action that may affect your business. Before making any decision or taking any action that may affect your business, you should consult a qualified professional advisor.