APPLICATION MAPPING: CONCEPTS & BEST PRACTICES FOR THE ENTERPRISE

Modern enterprise IT environments comprise complex system architecture. IT workloads, apps, and hardware system components interact and depend on each other to deliver the necessary functionality. The infrastructure and apps evolve as more lines of code are added to the codebase that scales IT apps and services to a growing user base, running on software-defined, virtualized, or cloud-based IT environments.

Discovering these dependencies is critical to understanding how the IT environment behaves, which touches many aspects of an IT-driven operational environment. So, in this article, we'll discuss key concepts in application mapping. We'll take a look at:

- Application mapping
- How it works
- Its role in DevOps use cases
- Best practices on mapping application dependencies

What is application mapping?

Application mapping is the process of discovering and identifying the interactions and interdependencies between application components and their underlying hardware infrastructure.

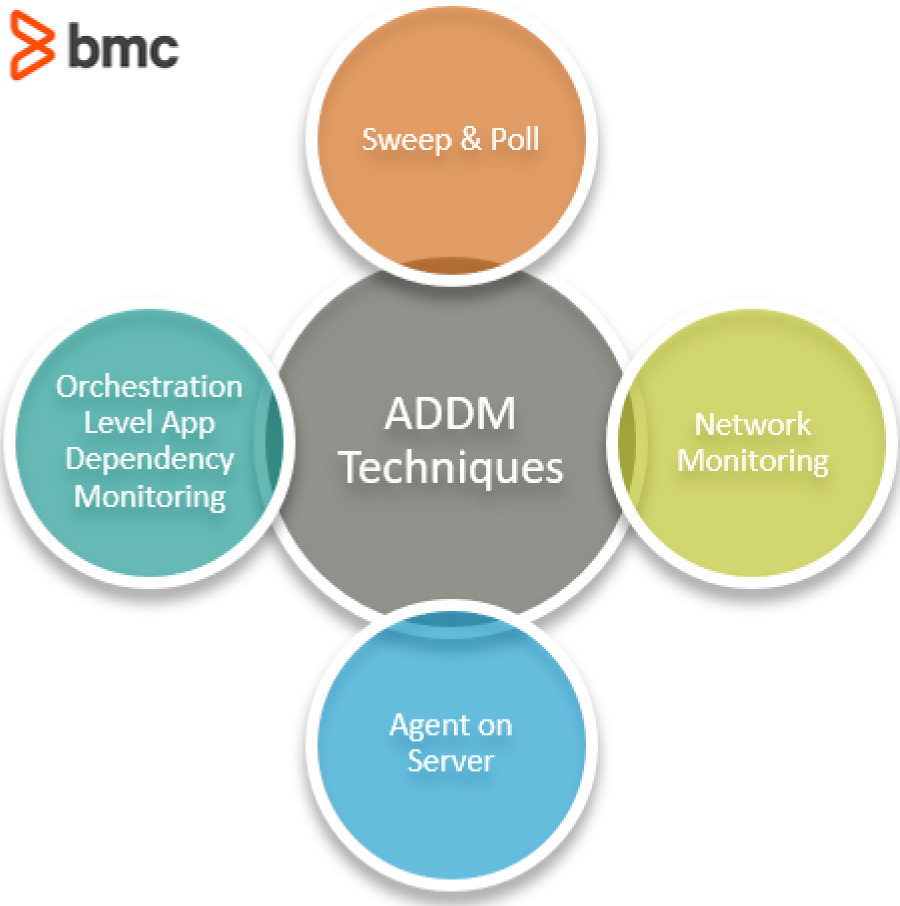
To ensure that apps perform optimally, it's important to discover and map the underlying dependencies. The technology that enables this capability is common called "application mapping", but Application Discovery and Dependency Mapping (ADDM) is another word for it. Application mapping solutions are:

> *A management solution that discovers the relationships of app components and the underlying components and maps them to deliver a comprehensive insight into the resources running in the IT infrastructure and their dependencies.*

(See how IT discovery & service mapping work.)

How application mapping works

Application mapping can be implemented in several ways, ranging from brainstorming and manual element polling to automated discovery of the entire IT ecosystem. The techniques can be agent-based or agentless monitoring, as described below:



Sweep and poll

The agentless monitoring technique is the traditional way of discovering IT assets: by pinging IP addresses and identifying the responding devices. The technique identifies app components, devices, and server systems based on information such as discovery ping rate and device group information, established based on known device data.

The sweep and poll method is lightweight and allows users to sweep the entire network from a single connected node location.

The process may be slow for large-scale data centers, which is a drawback considering that dependencies can change during the process and leave critical assets undiscovered. Furthermore, discovering app components, particularly in dynamic virtualized and cloud-based IT environments with limited visibility and control, may be particularly challenging.

Network monitoring

Network monitoring looks at real-time packet information and captures accurate data on application dependencies. The Netflow protocol contains IP traffic information such as:

- Volume
- Path
- Source and destination nodes
- Other IP flow attributes

Using protocols such as Netflow for traffic monitoring has its disadvantages.

Netflow implementation can impact the performance of devices given its large bandwidth requirements. To address this issue, the data is sampled at intervals, which reduces the demand for bandwidth consumption but also collects fewer packet information since the unsampled data is not monitored.

Additionally, Netflow containing IP address and TCP port data cannot differentiate application-level dependencies. As an alternative, data packets can be captured but still provide only limited information collected at the time of probing.

Agent on server

With agent-based monitoring, a software component is installed on the client server to collect data.

A monitoring station at the server side requests the data at periodic intervals based on a predefined policy. The agent monitors incoming and outgoing traffic in real-time and can identify the topology changes in application dependency as they happen.

This capability is particularly suitable for the dynamic virtualized infrastructure environments. Additionally, they can differentiate between apps running on the same server instances and the overall cost of running agents is less than using individual hardware devices to collect packet data.

However, agents must be installed on every server to ensure complete visibility. This can ultimately cause the monitoring agents to consume too much of the computing resources, impacting the server infrastructure's:

- Overall cost
- Operating performance

Orchestration-level application mapping

Automation and orchestration platforms are becoming increasingly popular to manage IT environments. These enable IT to provision resources for specific IT workloads efficiently using advanced software solutions. These solutions combine multiple automated tasks and configurations across app components that must utilize the necessary IT resources. In doing so, the orchestration technologies also keep track of the app components and the underlying server resources.

Together with Application Performance Monitoring (APM), AIOps, and other agentless or agent-based technologies, a hybrid discovery and dependency mapping can both:

- Enable accurate reporting
- Maintain optimal cost and performance of the monitoring systems

Even with the right application monitoring information available, organizations must be prepared to proactively manage risks and prevent performance bottlenecks. Three key aspects should be considered in devising a playbook guideline to achieve these goals:

- **Apps and systems.** Obtain comprehensive information about the apps and systems that can be potentially impacted. Continuously update existing dependency mapping documentation as new features, app components, and infrastructure resources are deployed.
- **Risks and mitigation.** Evaluate and prioritize the risks involved. Analyze vulnerabilities from a security, performance, and cost perspective. Provide specific guidelines and identify roles and responsibilities among team members as part of an effective risk mitigation plan.
- **Feedback and iterate.** The IT environment is constantly evolving as new software components are installed and new, scalable infrastructure resources are provisioned. As organizations shift workloads from legacy on-premise systems to advanced cloud-based infrastructure, the IT environment becomes more complex and dynamic. It is therefore important to follow up with team members, improve the knowledge base, and invest in the right technology resources and practices for app discovery and dependency mapping.

DevOps infrastructure challenges

With this understanding of application mapping, let’s take a look at it inside a DevOps environment. There, application mapping is a crucial activity.

Let’s review common infrastructure challenges in DevOps environments.

Server consolidation & virtualization

DevOps teams require rapid provisioning and management of IT resources in a complex mix of hybrid and multi-cloud environments. In order to consolidate server resources and manage dynamic workloads, IT needs a comprehensive view of all application dependencies.

According to a BMC-Forrester survey, 56% of the responding IT managers cited inadequate view into application dependencies as a key challenge to server consolidation.

In DevOps environments, this means that the administrative burden, cost, and time spent on server consolidation efforts remains far from optimal and inadequate to guarantee effective CI/CD practices, a fundamental DevOps activity.

Incident management challenge

High availability is a critical DevOps goal—one that requires comprehensive planning and careful system design.

An important characteristic of a highly available system design is the end-to-end mapping of the incident management practice. Application discovery and mapping corresponds to this practice, aiming to find dependencies between software components exposed to risks attributed to various incidents.

When you know these dependencies, you and your DevOps team can proactively isolate and contain damages in the event of an incident. Resource capacity and redundancy can be planned based on the risk factor.

Another aspect of managing IT incidents within a DevOps organization is to maintain full control over the infrastructure and empower DevOps teams to move multi-tier IT environments without having to discover, map, and rediscover IT environments. However, automation of such processes is only possible when you already have full visibility into application dependencies.

Rework & waste processes

DevOps is all about cutting down on waste processes. According to research, more than 75% of the responding organizations cite rework rates of 11% or higher. This trend suggests a huge disconnect between operations and infrastructure processes and teams.

DevOps teams can take advantage of application mapping to guide decisions based on the value created in the process of mapping application components. The process requires teams to understand the implications of interactions taking place at various levels of the infrastructure.

For instance, operating a hybrid mix of private and public cloud and data center deployments makes sense if your goal is to minimize cost. However, adding too many high availability nodes may prevent standardization in node configurations. Managing change controls in such environments could be more expensive and add to the administrative burden—exactly what you were trying to avoid in the first place.

With application mapping, you're better suited to handle this challenge. By automating dependency discovery, application mapping helps you understand how various infrastructure components interact with the software running on top of them.

Regulatory compliance

Proactive and accurate identification of compliance issues is only possible as long as the interactions with application components are both:

- Discoverable
- In compliance with regulatory policy frameworks adopted for sensitive IT workloads

Compliance is an ongoing effort that also requires organizations to identify gaps, prioritize risk, and track compliance progress with every change. A change control management practice would traditionally require IT to track dependency changes and apply the necessary changes. A correct standard operating procedure would involve automated triggering of configuration changes in adherence to a compliance policy that remains consistent regardless of the application changes.

According to research, however, more than a third of organizations cannot track assets or resort to manual asset tracking capabilities—risking compliance failure.

Application mapping best practices: reduce dependencies

The following best practices can help DevOps teams manage challenges associated with application mapping by—yes—avoiding dependencies in the first place:

- **Account for various forms of dependencies.** Dependencies aren't limited to technology. Take

various forms of dependencies into account, including hardware, software, versions, and especially in DevOps teams, soft dependencies such as business cost and culture.
- **Develop agnostic, defensible code.** Avoid interfaces that are expected to deprecate in the near future versions. Encourage and develop with code that's agnostic to operating systems.
- **Skip proprietary dependencies when possible.** Avoid dependencies on libraries and functions limited to specific versions of proprietary technologies.
- **Ensure that all dependencies are testable.** Integrations with internal and external software should be visible and controllable.
- **Ensure the infrastructure can be tested at all levels.** Third-party cloud vendors often offer limited visibility into resources running their SaaS applications.
- **Enhance observability and information flow** across the entire IT infrastructure. This should include both the testing and production environments.
- **Use advanced application mapping technologies.** Common mapping tools include BMC Helix Discovery, SolarWinds, Dynatrace and AppDynamics, among others. Most cloud vendors including AWS, Azure and Google Cloud also offer technology capabilities that enable application dependency mapping.

BMC supports application mapping

BMC supports application mapping in enterprise environments. BMC Helix Discovery is a SaaS-based, cloud-native discovery and dependency modeling system that provides instant visibility into hardware, software, and service dependencies across multi-cloud, hybrid, and on-premises environments.

On average, organizations that use BMC Helix Discovery reported:

- 549% ROI over five years
- Payback period of only five months
- $4.18 million in average annual business benefits

Learn more about BMC Helix Discovery.

Related reading

- BMC Service Management Blog
- BMC AIOps Blog
- The State of ITSM Today
- Get Started with ITAM: IT Asset Management Explained
- Application Performance Management in DevOps