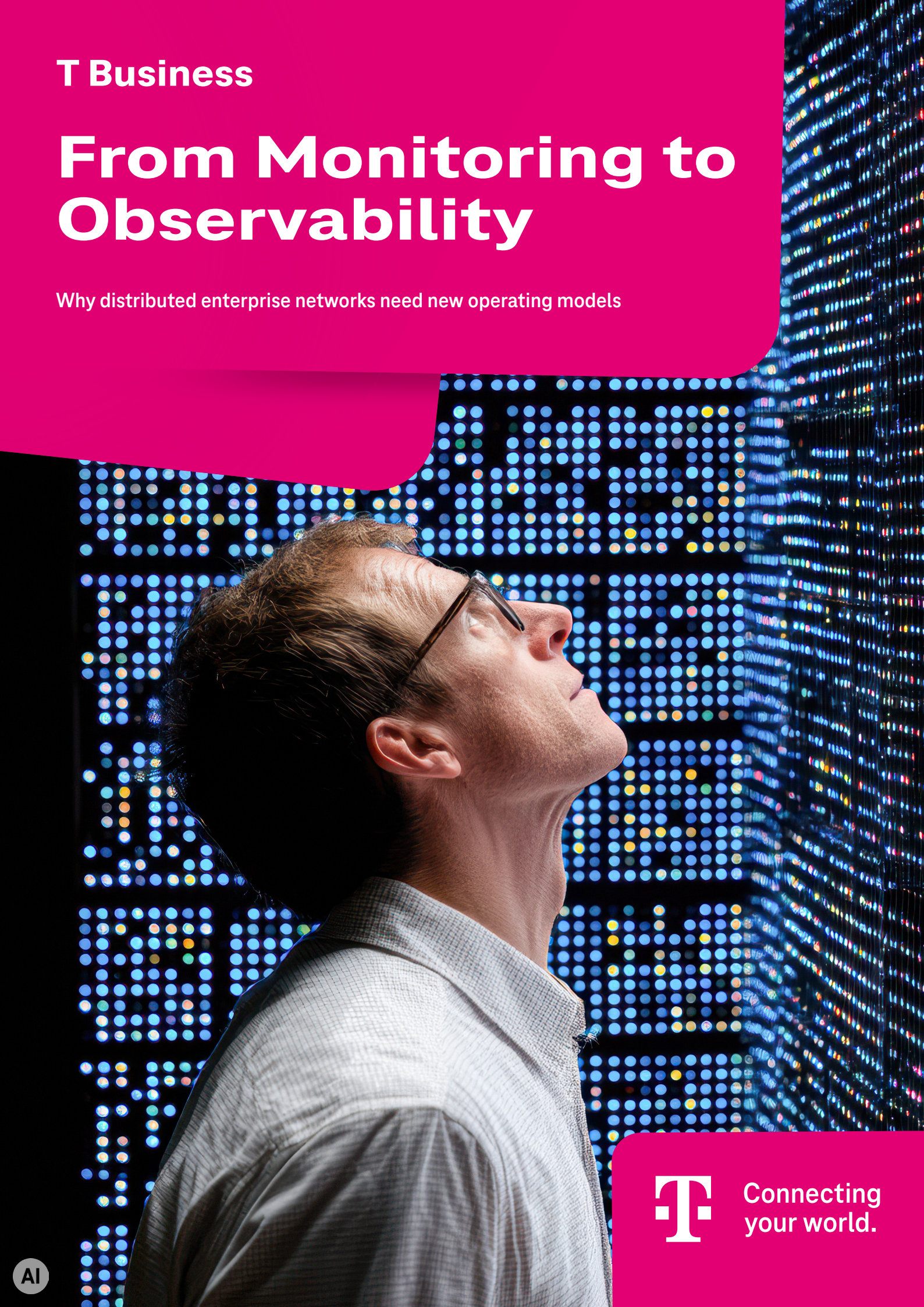
T Business

From Monitoring to Observability

Why distributed enterprise networks need new operating models



Connecting your world.

Introduction

Enterprise networks once served primarily to connect central data centers and clearly defined sites. They have since become a critical part of how businesses operate. Across manufacturing and logistics sites, branch offices, home offices, and mobile workforces, networks provide the essential link to cloud platforms, SaaS applications, and central IT services.

When disruptions occur, particularly when users access applications, a single network component is rarely the sole cause. WLAN infrastructure, wide area network paths, routing tables, DNS, security policies, endpoint configurations, service providers, and external platforms can all affect user experience and performance.

Traditional monitoring shows whether systems are reachable, thresholds have been exceeded, or metrics such as latency, packet loss, and utilization indicate a problem. Understanding why a service is impaired and which dependencies are involved requires additional context. Observability connects relevant technical signals with information about sites, services, users, providers, and changes. This helps teams assess causes and impacts more accurately.

This is particularly important for organizations with many distributed sites. Different connection types, providers, security requirements, and local operating conditions make consistent management difficult. At the same time, IT teams must assess service quality, narrow down the causes of incidents, and explain their impact clearly to business teams and management.

This whitepaper explains how monitoring and observability differ, the challenges that arise in distributed enterprise networks, and the target operating model these challenges call for. The aim is not to collect as much data as possible. What matters is integrating relevant information into operations and processes so that it supports better analysis and informed decisions.

Deutsche Telekom helps organizations work toward this target model by bringing together relevant capabilities, from site and cloud connectivity to software-defined network control, LAN and WLAN operations, integrated security solutions, and managed services. The capabilities required in each case depend on the organization's site structure, cloud usage, security requirements, data quality, and operating model.



Why distributed networks need more than monitoring

Organizations with many sites often have connections, LAN and WLAN infrastructure, security solutions, and provider services from different generations. Cloud and SaaS services, mobile access, and new network architectures add to this mix. As a result, an issue may originate in one part of the environment and become apparent elsewhere. A slow application in a branch, for example, may be linked to the local WLAN, the WAN path, a DNS configuration, a security inspection, or an external cloud service.

Monitoring provides important status information. It detects when devices or connections become unreachable, latency or packet loss increases, or defined thresholds are exceeded. It primarily answers the question: **What is happening?** Its limitations become apparent when several possible causes need to be considered. A single alert rarely explains which dependencies are involved or which users and services are affected.

Observability adds context and correlation to monitoring, helping answer the question: **Why is it happening?** It connects relevant metrics, logs, and events with topology and service information. Teams can then investigate whether an anomaly coincides with a change, a security policy, a particular provider, or a condition elsewhere along the service path. This helps them develop hypotheses about the cause, assess the impact, and make more focused decisions about the next steps.

A particular challenge arises when the teams involved use different systems. Network operations may see latency or packet loss, security operations may see policy or inspection events, cloud operations may see platform metrics, and the service desk may see tickets. Providers supply their own status information as well. Without shared context, it remains unclear how these observations relate to one another. If an incident affects only some sites, for example, teams need to investigate whether a particular provider, region, site type, security policy, or routing change is involved. Without these connections, tickets are often passed from team to team instead of being used to narrow down the cause.

In international site networks, different providers, connection types, service levels, operating windows, and escalation paths make consistent management even harder. Central IT must assess service quality and produce comparable reports, even though the operational data available may differ by country or site. This requires site, provider, and service context alongside technical signals.

Network performance and security cannot always be assessed separately, either. As security inspections move closer to users, sites, and cloud access, they can affect user experience without a conventional network outage occurring. Architectures such as SASE combine networking and security functions, making it increasingly important to assess performance and security events together.

The same applies to changes. A new SD-WAN policy, an adjustment to firewall rules, a different cloud path, a new DNS configuration, or a change of provider can affect sites differently. Unless the relevant data is brought together, it is difficult to establish whether an incident stems from the change, local conditions, or an external dependency. Observability can help make these effects easier to trace.

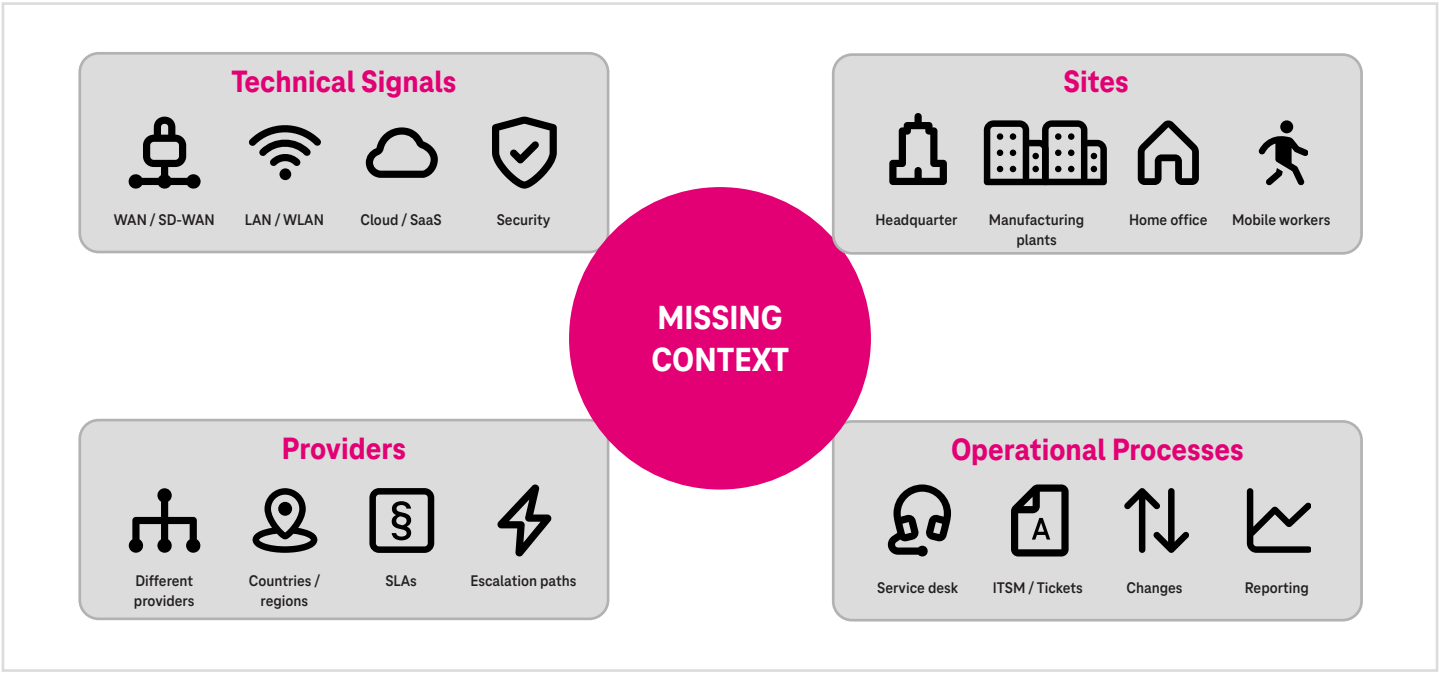


Figure 1: Connecting technical signals, sites, providers, and operational processes creates the context for fast, informed decisions

Business teams and management need an assessment that goes beyond individual technical measurements. Their concern is the stability of business processes, rather than packet loss or routing paths alone. Observability can connect technical signals with services, sites, and operational processes to support a better-informed assessment. It does not replace monitoring, security architecture, or operations. What matters is making the relevant signals available in the right context.

Defining a target model for enterprise observability

A viable target model for observability starts with the operational capability the organization wants to achieve, rather than with a platform. For organizations with extensive site networks, the aim is rarely to make every technical detail visible at all times. What matters is bringing together the signals relevant to business-critical services so that causes, impacts, and responsibilities become clearer.

At the center is a shared data and context layer. It connects information from WAN, LAN, WLAN, SD-WAN, SASE, cloud connectivity, security components, endpoints, applications, and ITSM systems. ITSM stands for IT service management and covers processes such as incident management, problem management, change management, and service reporting. These processes matter because observability data delivers operational value only when it informs decisions, tickets, escalations, and management reports.

The target model must distinguish between signals, context, and decisions. A signal might be unusual latency, packet loss, a log event, a policy match, or a failed access attempt. Context emerges when that signal is connected to a site, provider, application, user group, topology, change history, or security rule. A decision, in turn, requires that context to be translated into operational action: escalating to a provider, rolling back a change, adjusting a policy, upgrading a connection, investigating an endpoint, or prioritizing a modernization initiative.

For many organizations, this transition is the real challenge. Dashboards are quick to build, but they do not automatically create a shared understanding. A site's network dashboard may show everything in green while users still experience poor application performance. A security console may enforce a policy correctly while a business team experiences a noticeable delay. A provider may meet its service parameters while the end-to-end experience falls

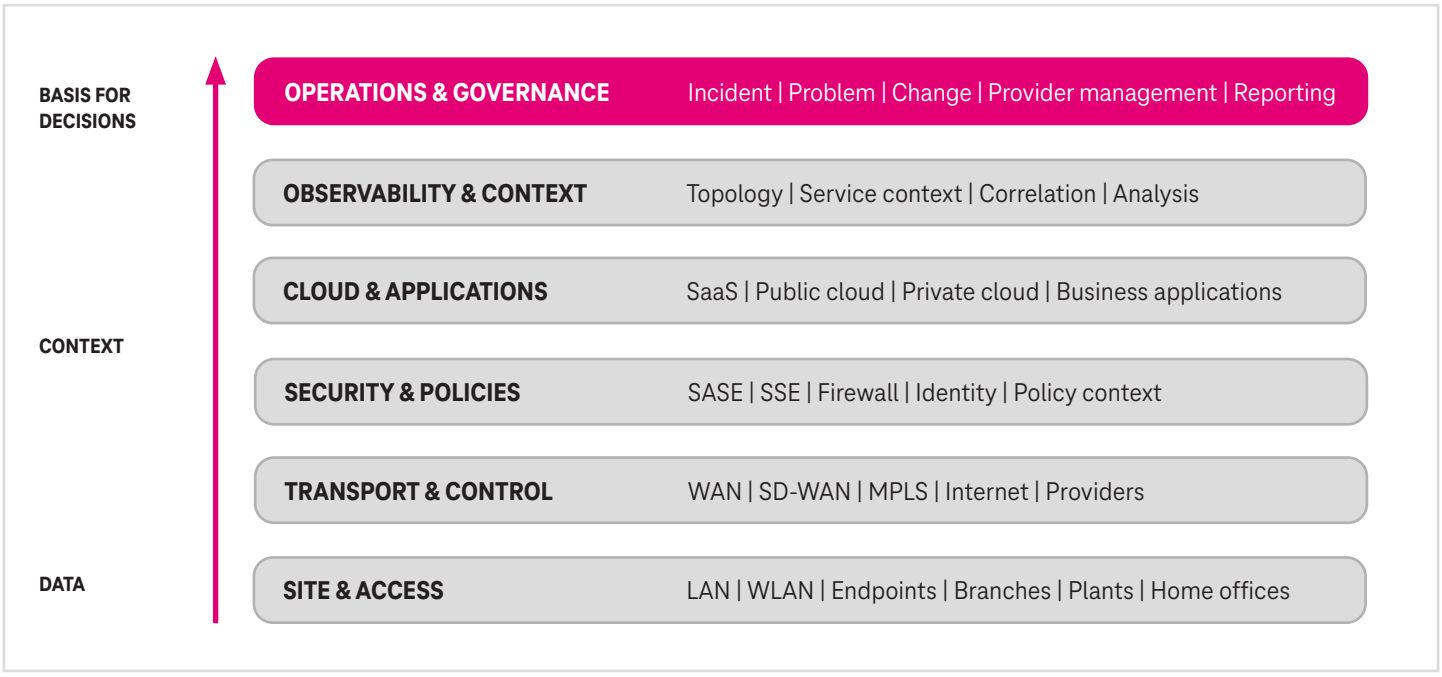


Figure 2: Informed operational decisions based on observability require a continuous chain connecting data to decisions

short of expectations. Observability must bring these perspectives together while preserving clear accountability across teams.

A sound target model therefore needs clear design principles. Data sources must be relevant and reliable. Correlation needs to be based on an understandable model. The most important services and sites must be prioritized. Access rights, data protection, and data residency must be addressed. Roles and responsibilities must be defined so that observability does not become another tool silo. The technical architecture must also remain open enough to accommodate different vendors, providers, cloud platforms, and operating models.

For organizations with many sites, standardization must also account for local conditions. Sites differ in criticality, bandwidth, applications, and security requirements. A large manufacturing facility with systems closely connected to operational technology (OT), such as programmable logic controller (PLC) devices or manufacturing execution systems, has different requirements than a small sales office. A retail branch with point-of-sale systems and customer access has different requirements than an office focused on collaboration. The target model should therefore provide different levels of coverage rather than impose a rigid approach. Critical sites and services need deeper observability, while less critical areas should receive proportionate coverage without unnecessary complexity.

Security and compliance also belong to the target model, but expectations must be realistic. Observability improves traceability, provides data for investigations, and reveals relationships between network events and security context. It does not automatically make an organization compliant or prevent attacks. Claims about regulatory compliance, risk reduction, or security outcomes must be substantiated and reviewed by the relevant specialists.

A good target model therefore answers three questions: Which services and sites really matter? What data and context are needed to understand their behavior? And which operational processes will use these insights? Only when these questions have been resolved, observability can develop from a technical data collection exercise into a manageable operating model.



How observability works technically

The technical architecture of observability can be described as a flow of data from the site to the decision. It begins with infrastructure signals: metrics from routers, switches, and access points; flow data from network paths; logs from security components; status information from SD-WAN controllers; data from cloud connections; DNS information; endpoint information; and synthetic tests that reflect the user perspective. These signals are valuable in and of themselves, but they rarely explain the cause of a problem on their own.

The next step is to connect them with topology and service context. A latency measurement becomes more meaningful when it can be associated with a site, provider, application, user group, and specific path. A log entry becomes more useful when the associated policy, service, and change are visible. A ticket becomes more actionable when it is linked to technical events and affected services, rather than simply describing a complaint. Observability depends on this translation of raw data into operational context.

SD-WAN and SASE data play an important role in modern site networks. SD-WAN refers to software-defined control of wide area networks. For observability, the important point is that application paths, policies, and traffic steering decisions should be visible and included in root cause analysis. When a site experiences a problem, knowing that it is online is not enough. Teams need to know which path was used, which application was affected, and which policy was applied.

SASE adds security context to this perspective. As network access and security inspections become more closely integrated, security decisions also influence paths, latency, and user experience. Observability should therefore capture network metrics and show which security components and policies are involved in an access request. This is particularly important when sites, mobile users, IoT devices, and cloud applications are managed through a common security and access model.

Alongside WAN and security, the local site environment remains critical. Many problems originate outside the wide area network: in the LAN or WLAN, in local power or device infrastructure, in incorrectly sized access point deployments, or in local configuration differences. Without visibility into this layer, a central observability model may draw the wrong conclusions. Slow cloud access may appear to be a WAN problem even though the cause lies in the local wireless LAN. Organizations with many sites therefore need to incorporate local signals into the same analytical context as WAN, cloud, and security data.

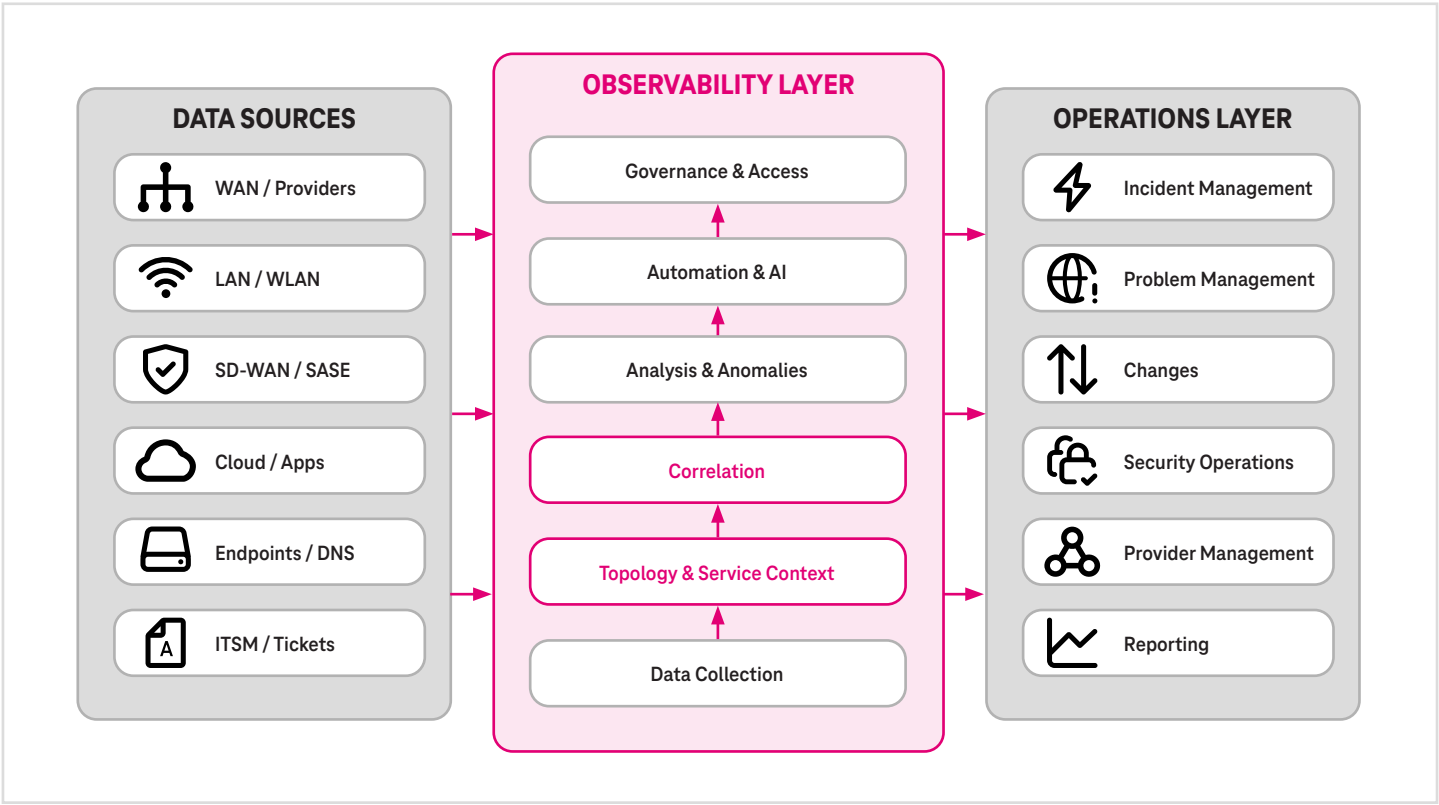


Figure 3: An architecture model for observability

The architecture also needs an integration layer. Observability platforms, network management systems, security tools, cloud monitoring, ITSM systems, and provider portals do not necessarily have to be consolidated into a single system. What matters is connecting relevant information so that teams can work together. In practice, this can be achieved through APIs, event streams, data models, dashboards, automation, or ITSM integrations. From a technical standpoint, it's less about completeness and more about relevance: Which data actually helps explain the most important use cases?

Artificial intelligence and automation can also help. Pattern and anomaly detection, along with automated correlation, facilitate analysis when data quality, the context model, and process integration are sound. Without reliable data and clear responsibilities, however, automation simply accelerates misinterpretation.

Technically, observability consists of several layers: data collection, contextualization, correlation, analysis, process integration, and governance. Value comes from the ways these layers interact, rather than from a single layer. A platform can collect data, but it cannot establish accountability. A dashboard can show patterns, but it cannot replace escalation logic. A telemetry standard can structure data, but it cannot resolve conflicting organizational objectives. The technical architecture must therefore be connected to the operating model.

How operations, governance, security, and compliance work together

Observability does not deliver value through technical data collection alone. It becomes effective when organizations establish who interprets the data, who initiates action, who manages providers, and who explains decisions to business teams or management. This is challenging in organizations with extensive site networks because network operations, security, cloud teams, service desk, application operations, and external service providers view the same incident from different perspectives.

A slow application across several branches illustrates why shared service context is necessary. Network, security, and cloud data, together with tickets and provider information, must be associated with the same service, site, and time period. Only then can separate observations be combined into a shared hypothesis about the cause. Observability must therefore embed these relationships in operational workflows and responsibilities, not just in dashboards.

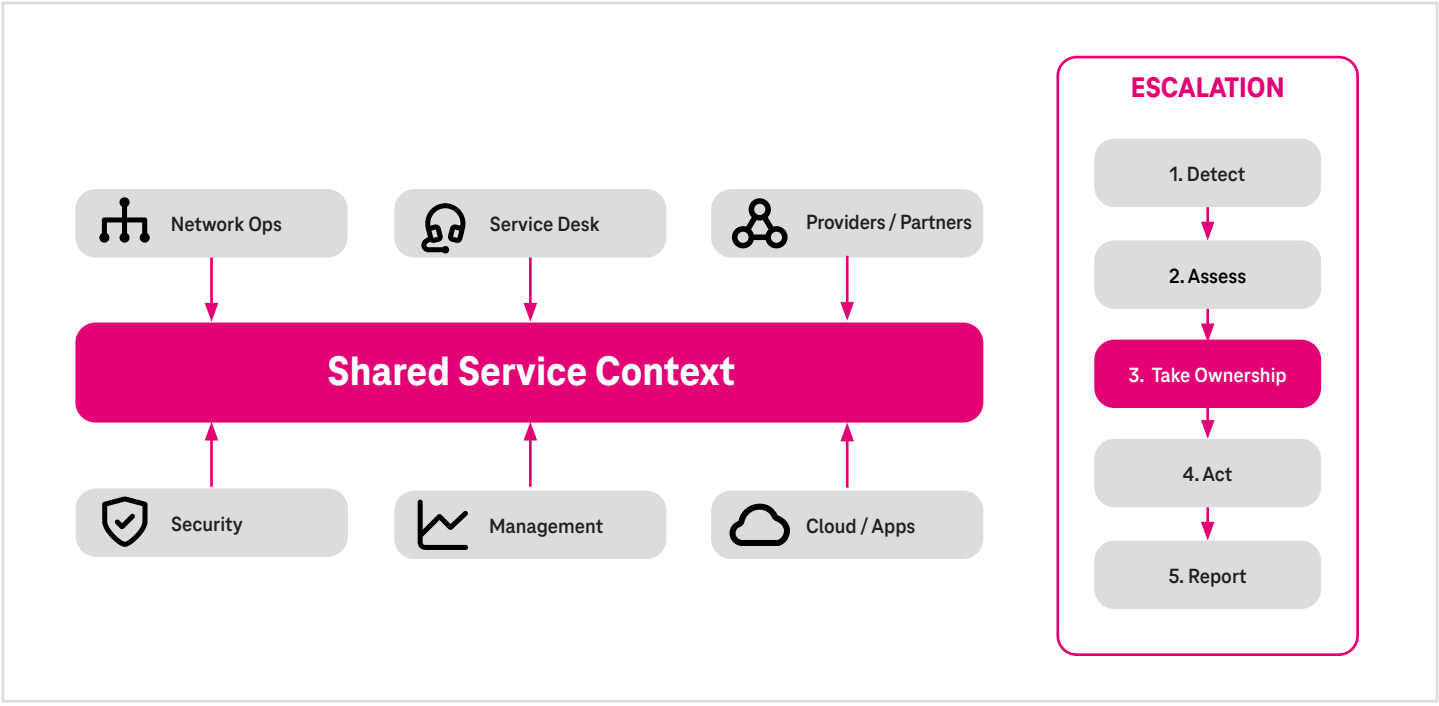


Figure 4: Observability requires shared service context to deliver value

In practice, this means that observability data needs to be integrated into incident management, problem management, change management, and service reporting. Incident management addresses current disruptions, problem management investigates recurring causes, and change management assesses the impact of changes. If observability signals are not connected to these processes, they remain analytical aids for individual teams. Integrating them into tickets, escalation paths, and service reports creates a shared basis for decisions.

Governance plays a pivotal role. Organizations should establish which data sources are authoritative, which dashboards are intended for which roles, how data quality is checked, and which escalation rules apply across provider boundaries. Organizations operating in several countries must also account for local providers, different service levels, time windows, and regulatory frameworks. Observability cannot eliminate this complexity, but it can help make technical and organizational dependencies visible in a more structured way.

In integrated networking and security architectures such as SASE, network and security data must be analyzed together. This requires organizations to define which teams interpret the data and who determines the actions to take.

Compliance is relevant to observability particularly where data is collected, stored, correlated, and analyzed. Logs, flow data, user context, site information, and security events may have data protection or regulatory implications. Data residency, access controls, retention periods, role-based permissions, and traceability should therefore be assessed early. Claims such as “meets NIS2 requirements” or “ensures compliance” would be risky without legal and specialist review. A structured observability model can help organizations assess technical and organizational requirements more clearly and prepare supporting evidence more effectively. Whether specific regulatory requirements are met depends on the architecture, industry, country, allocation of responsibilities, and other measures.

The central operational question is therefore whether data, roles, and processes are connected in a way that enables teams to act consistently, rather than whether the organization has enough data. Observability thus becomes part of governance, connecting technical visibility with clear responsibilities, coordinated escalation paths, and traceable decisions.

Introducing observability: key steps and common pitfalls

Common pitfalls at a glance

- **Choosing the tool first:** Selecting a platform before use cases and objectives have been defined
- **Too much data without context:** Adding telemetry creates a flood of data instead of better root cause analysis
- **Unclear roles:** Teams interpret data differently, and responsibilities remain unresolved
- **An overly ambitious pilot:** An overloaded pilot overwhelms the teams involved
- **Data protection and compliance:** Involving the relevant teams too late and omitting reviews can force subsequent changes
- **Overstating the benefits:** Claims lose credibility when the organizational prerequisites are missing

Introducing network observability should not begin with platform selection. A tool can collect, visualize, and correlate data. It cannot decide which business-critical services take priority, which data is reliable, which teams take responsibility, or how escalations should work. Organizations should therefore start with a specific operational problem rather than a complete target architecture.

A sensible starting point is to prioritize a small number of relevant use cases. For organizations with many distributed sites, these might include recurring performance issues in branches, unexplained SaaS delays, cloud latency of uncertain origin, WLAN problems affecting particular site types, the effects of security inspection, or provider issues that require extensive escalation. The use case determines which data sources are needed. It prevents observability from turning into a broad data collection project whose benefits are difficult to explain.

The next step is to assess existing data sources and their quality. Relevant signals may come from WAN, LAN, WLAN, SD-WAN, SASE, firewalls, DNS, cloud connections, endpoints, application monitoring, synthetic tests, and ITSM systems. The objective is not completeness at any cost. The more important question is which data helps teams develop specific hypotheses about the cause of a problem. Large volumes of data without site, service, and provider context often create more analytical work rather than greater clarity.

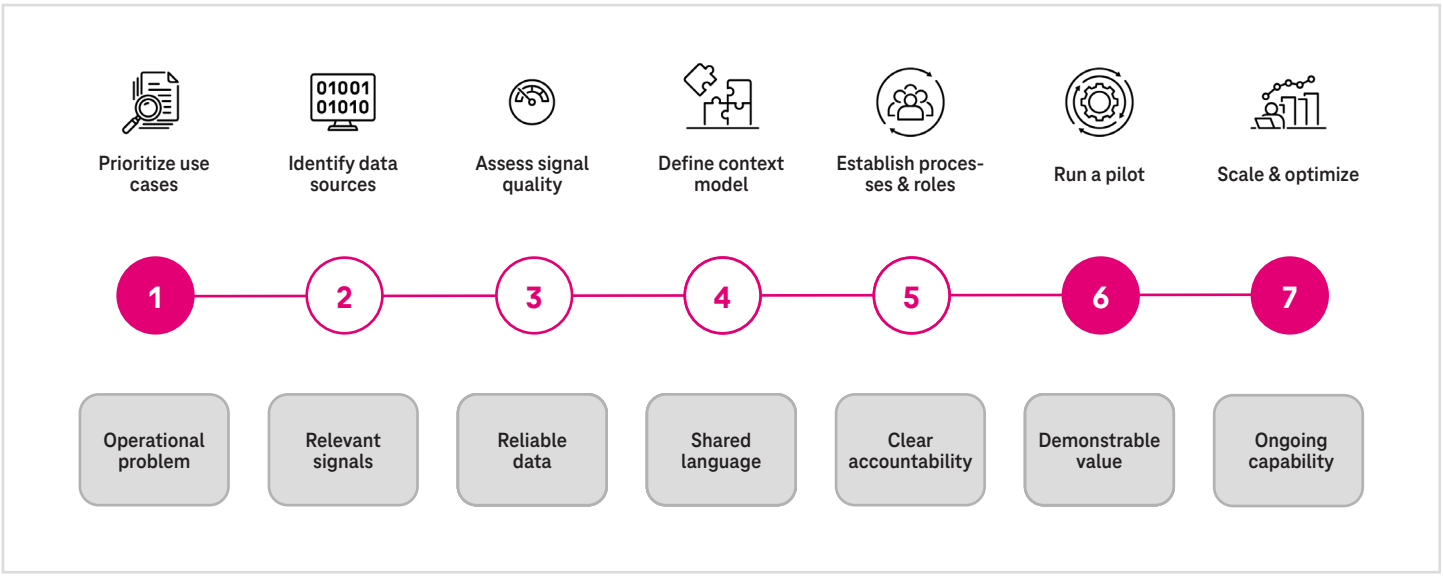


Figure 5: Successful observability implementation starts with defining an operational problem, rather than selecting a platform

Organizations then need a shared context model. They should define how sites, applications, user groups, providers, policies, service classes, and operational processes are connected. For a retail group, the business criticality of individual branches may be decisive. A manufacturing company may focus on plants, networks closely connected to OT, and campus infrastructure. A financial services organization may place greater emphasis on documenting access, roles, and service quality. Observability must accommodate these differences without making the model excessively complex.

Only on this basis should organizations consider platforms, integrations, and automation. An observability platform should ingest and correlate data and make it usable for different roles. It must also fit existing processes across the service desk, incident management, change management, security operations, provider management, and management reporting. Technical integration without process integration often produces new dashboards without improving decisions.

A pilot is the right way to test whether the model is viable. Its scope should be manageable but sufficiently relevant to address real operational questions. Suitable success criteria include clearer classification of causes, less manual effort to find data, more effective provider escalations, more informative reports, and greater traceability of changes. A pilot can thus show whether causes can be narrowed down more quickly and responsibilities assigned more clearly.

Scaling should follow only after this. Organizations can bring in additional sites, applications, providers, and data sources once the data model, roles, dashboards, and processes work. Observability should be treated as an ongoing operational capability. Site networks change, cloud access grows, security policies are adjusted, providers change, and new applications are introduced. Observability must therefore be maintained, reviewed, and developed over time.

Typical pitfalls arise when organizations reverse this sequence. A tool-first approach often leads to rapid data collection but slow clarification of its value. More telemetry without context creates data overload. Unclear roles lead to conflicting interpretations. Overly broad pilots overwhelm teams. Omitting data protection and compliance reviews can force subsequent changes. Overstated benefits undermine a program's credibility when the organizational prerequisites are missing.

Implementation therefore works best as a gradual development of operational capability. Use cases, data sources, a context model, process integration, a pilot, and scaling form a connected sequence. If any element is missing, network observability remains technically interesting but operationally incomplete.

Realistic organizational and business outcomes

The business value of observability does not come from data visibility alone. Visibility in itself does not reduce costs, prevent incidents, or improve service quality. Value emerges when data informs better decisions: faster identification of likely causes, clearer escalations, better prioritization of modernization initiatives, more transparent provider management, and more reliable communication with business teams.

IT decision-makers need to describe the benefits realistically. Observability can help make the effort involved in manual investigation more visible when data sources are connected consistently. It can simplify provider management when technical signals, service levels, and escalation paths are brought together in a traceable way. It can support change processes when changes to the network, security, or cloud connectivity can be linked more clearly to their subsequent effects. These outcomes are plausible, but they depend heavily on the starting point, data quality, architecture, and operating model.

Observability can reduce silos between teams when everyone has access to the same context. Teams can then assess causes together while maintaining clear responsibilities. A good model makes it clear who interprets the data, initiates action, and is accountable for escalations. This provides structure for collaboration.

Observability can also help with management communication. Executives do not need raw packet loss data or flow statistics. They need a reliable assessment of which sites are affected, which business processes are impacted, whether there is a provider problem, whether a change may have caused the issue, and which actions should take priority. Observability can help translate technical analysis into service-oriented language.

Organizations with many sites can also benefit when planning modernization. Visibility into recurring incidents, site types, provider dependencies, and application paths helps organizations prioritize investment. They can more readily determine whether a problem calls for local WLAN modernization, a WAN adjustment, an SD-WAN policy change, a change to cloud connectivity or the security architecture, or a process change. This can provide a basis for better budget decisions, but it does not replace a business case.

The realistic impact therefore lies in three areas: better visibility across distributed infrastructure, greater control over operational and provider processes, and a stronger basis for modernization and governance decisions. This is less dramatic than a blanket promise of savings, but it is far more credible for large enterprise networks.

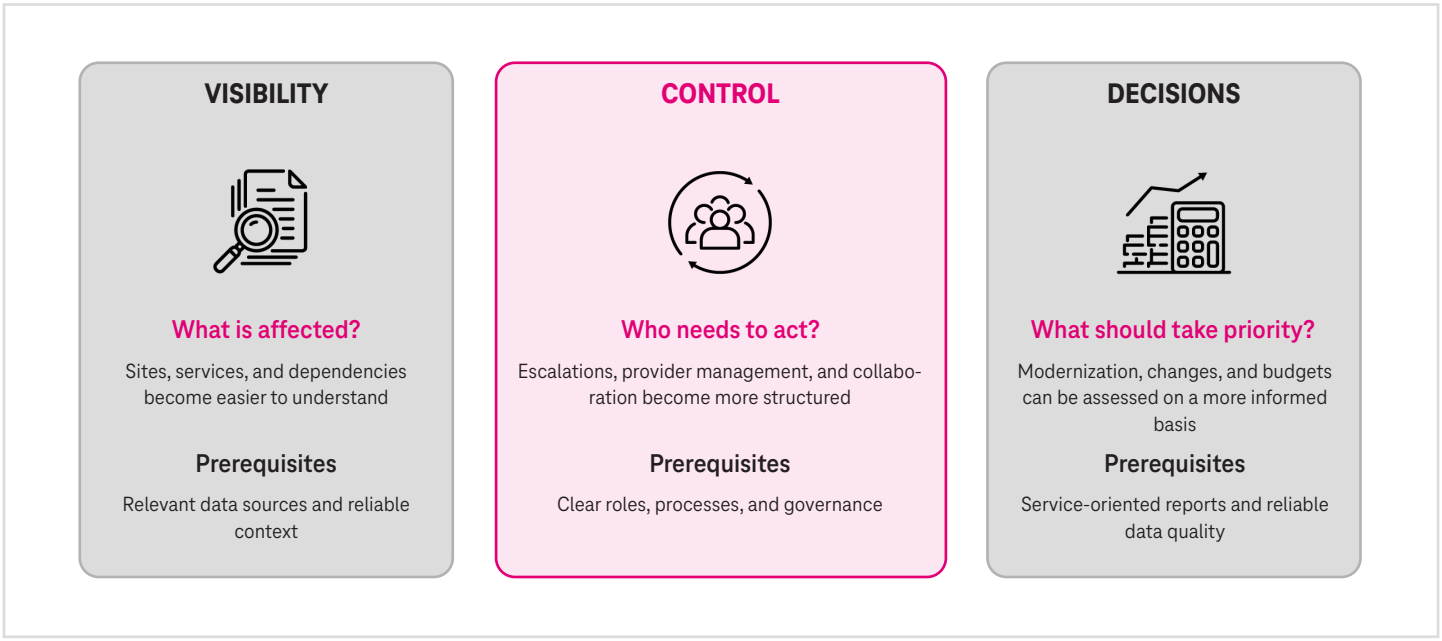


Figure 6: Observability offers better visibility, control, and decisions rather than blanket promises of savings

An observability checklist for IT decision-makers

Before committing to observability, organizations should look beyond platform features. The key question is whether the target model fits their site environment, operational organization, and provider structure.

Strategic relevance and scope

- Which sites, applications, and business processes are particularly critical?
- Which recurring incidents currently require the most analysis or escalation effort?
- Is the primary focus network operations, user experience, provider management, security context, or cloud access?
- Which sites are in other countries, and what additional provider, service, or governance questions does this raise?

Data sources and architecture

- Which data sources are available today: WAN, LAN, WLAN, SD-WAN, SASE, cloud, DNS, endpoints, applications, and ITSM?
- What data is missing to understand causes across site, provider, and service boundaries?
- Can technical signals be linked to a site, application, user group, provider, policy, and change history?
- What integrations with existing networking, security, cloud, and ITSM systems are required?

Security, data protection, and compliance

- What logging, data residency, access control, and retention requirements apply?
- Which security data may be processed and correlated, and in what context?
- Which regulatory claims need to be reviewed by legal, compliance, or the relevant business teams?

Operations and governance

- Who owns data quality, dashboards, alert rules, and correlation logic?
- Which teams interpret observability data, and who decides what action to take? How are provider escalations documented, managed, and tracked?
- How are observability insights incorporated into incident management, problem management, change management, and reporting?

Pilot and scaling

- Which use case is suitable for a manageable but relevant pilot?
- Which success criteria can realistically be measured without making unsupported efficiency claims?
- How will the model be extended to additional sites, providers, applications, and security domains after the pilot?
- What role should a managed services partner play: consulting, integration, operations, escalation, or continuous optimization?



Next steps for organizations to consider

Observability is not a shortcut that automatically simplifies complex enterprise networks. It is a structured way to understand complexity better and make operational decisions more transparent. For organizations with many distributed sites, this capability is becoming more important as networking, cloud, security, providers, and user experience become more closely connected.

The next sensible step is to analyze sites and use cases. Organizations should identify the services, sites, and incident patterns that matter most. They should then assess existing data sources: Which signals are available, which are missing, which are reliable, and which can be connected? Only then can they determine the appropriate platform, integrations, and operating model.

Responsibilities should be clarified in parallel. Observability involves network operations, security, cloud teams, the service desk, application operations, provider management, and management reporting. If these roles are not involved early, the result is a technical project without sufficient integration into operations. A pilot should therefore test escalation, reporting, and decision logic as well as collect technical data.

Deutsche Telekom can support this phase as an architecture, integration, and operations partner. It brings together a broad portfolio of solutions and years of experience in connectivity, SD-WAN, SASE, LAN and WLAN, cloud connectivity, security, and managed services. The right combination should be determined by the prioritized use cases and the target model. This keeps observability focused on the customer's operating model rather than reducing it to a single product decision.

The conclusion is that organizations should approach observability as a modernization of network operations rather than another monitoring project. What matters is the progression from data to context, from context to accountability, and from accountability to better decisions.

Image and graphic credits:
- Adobe Stock (Pages 1, 2, 5, 11)
- Deutsche Telekom (Pages 3, 4, 6, 7, 9, 10)

Deutsche Telekom Geschäftskunden GmbH | 10/2026 | TText created with AI assistance | CF

Get in touch with us for a personal consultation

Contact
telekom.de/business

Publisher
Deutsche Telekom
Geschäftskunden GmbH
Landgrabenweg 149
53227 Bonn
Germany

Connecting your world.