

EBOOK

Enabling Zero Disruption

Closing the Gap Between
DEX and Application Observability

riverbed®

Table of Contents

- 3 Executive Summary
- 4 What DEX Does Well, and Where It Stops
- 5 The Structural Gap Between DEX and Application Observability
- 6 Why Extending DEX Visibility is Not Sufficient
- 7 Application Observability as a Complement to DEX
- 8 Preserving Context from Detection to Diagnosis
- 9 Operational Implications across Teams
- 10 Conclusion: Closing the Gap is an Architectural Decision

Executive Summary

Devices serve a single purpose: to run business applications. As a result, digital employee experience (DEX) is fundamentally shaped by application performance.

DEX platforms have made meaningful progress by shifting visibility away from infrastructure health alone and toward how applications perform for users. By observing applications from the employee's perspective, DEX enables IT to understand whether critical tools are usable, responsive, and supporting productivity in real time.

In this role, DEX is effective. It can identify degraded application experiences, quantify user impact, and establish when an issue is actively affecting work.

What DEX typically cannot do is explain why that degradation is occurring inside the application. Once experience impact is detected, teams must still determine whether the cause lies in application code, services, dependencies, configuration, or downstream systems. That diagnostic step usually happens outside the DEX platform.

This separation creates a structural gap between experience identification and application diagnosis. Context is often lost as teams move between tools, reconstruct timelines, and revalidate impact. The result is slower resolution and increased operational friction, even when there are strong application observability tools in the environment.

Closing this gap enables a different operational outcome. When teams connect experience detection directly to application diagnosis, they reduce investigative overhead, shorten resolution cycles, and prevent issues from cascading across users. In practice, this is how organizations move toward Zero Disruption operations, where employees remain productive because problems are resolved quickly and with minimal impact.

What DEX Does Well, and Where It Stops

DEX shifted ITOps attention away from infrastructure metrics toward user outcomes. That shift matters. It aligns IT priorities with productivity rather than availability and surfaces issues that traditional monitoring may miss.

Most DEX platforms are effective at:

- Identifying affected users and devices
- Highlighting degraded workflows
- Establishing that an issue is real and active

Where DEX typically stops is at application troubleshooting or what's happening at the code level to cause the issue.

While some platforms provide limited application indicators, most rely on inference rather than direct observation. As a result, when DEX identifies a problem, help desk teams can confirm impact but still lack visibility into:

- Which services or components are involved
- Whether the issue originates in code, configuration, dependencies, or load
- How to move directly from detection to remediation

**DEX answers the question of impact.
It does not answer the question of root cause.**

The Structural Gap Between DEX and Application Observability

DEX and application observability evolved with different objectives and users in mind. DEX focuses on experience outcomes. Application observability focuses on application behavior.

In many environments, these capabilities exist side by side but operate independently. When an issue is detected via DEX, teams must pivot to separate observability tools to begin diagnosis. That transition typically involves redefining scope, reconstructing timelines, and validating impact again.

Several structural factors reinforce this separation:

- Application observability instrumentation is often limited to a subset of critical applications
- Cost and operational complexity restrict broader deployment
- Sampling reduces visibility into intermittent or edge case issues
- Experience context does not carry forward into application tools

The result is a fragmented workflow. Detection and diagnosis occur in different systems, at different levels of fidelity, and often with different assumptions.



Why Extending DEX Visibility is Not Sufficient

DEX tools focus on applications because devices exist to run them. As a result, DEX platforms capture meaningful indicators of application performance from the user's perspective.

Specifically, DEX measures how applications respond as employees use them. It tracks launch times, screen rendering, transaction completion, and user visible errors. In addition, it correlates these signals with users, devices, locations, and time. Therefore, IT can clearly see which applications degrade and who experiences the impact.

This capability plays a critical role. It allows help desk and digital workplace teams to confirm that a problem exists, assess its scope, and prioritize response based on real user impact rather than assumptions.

However, this same focus also defines the limit of DEX.

DEX observes application performance as an outcome. It shows that an application slows or fails, but it does not show how internal services behave when that happens. Consequently, DEX can reliably identify experience degradation, but it cannot explain the underlying cause within the application stack.

As applications become more distributed, this limitation becomes more pronounced. Performance issues often originate in application code, dependencies changes, or resource issues that remain invisible at the endpoint. DEX captures the symptom accurately. Nevertheless, it lacks direct visibility into the execution path that produces it.

For this reason, extending DEX alone will never close the diagnostic gap. Endpoint based insight remains indirect and relies on inference. In contrast, application observability explains why performance changes occur by observing internal behavior directly.

DEX answers whether applications work for users. Application observability explains how applications behave internally. Therefore, organizations need both. Closing the gap requires connecting experience insight directly to application level evidence rather than treating one as a substitute for the other.

Application Observability as a Complement to DEX

Closing the gap between experience detection and diagnosis requires application observability that organizations can deploy broadly and sustain over time. However, many application observability tools limit coverage because instrumentation is costly or complex to deploy. As a result, DevOps teams focus visibility on a small set of Tier 1 applications while leaving much of the application estate unobserved.

At the same time, DEX continues to surface experience degradation across all applications. Consequently, organizations detect user impact in places where they lack sufficient application level insight.

Riverbed APM+ expands application observability without replacing existing tools. Built on open, standards-based instrumentation, it avoids proprietary lock-in and supports data portability across the environment. Its cost model is designed for broad deployment; not selective Tier 1 coverage—delivering application visibility at 70% lower cost than traditional tools, so IT can extend visibility across all your applications.

In this complementary model, Riverbed Aternity establishes impact and priority. Riverbed APM+ provides internal application evidence. Together, this alignment removes the structural gap between experience detection and application diagnosis.



Preserving Context from Detection to Diagnosis

The most consequential change is not additional telemetry. It is continuity of context.

When DEX and application observability are integrated, an experience alert becomes an entry point rather than a dead end. From a DEX alert, IT can move directly through a single, contextual action—into application level visibility scoped to:



The affected application



The impacted users or transactions



The precise time window of degradation

This workflow reduces the likelihood that investigation itself becomes a second disruption, which is a core requirement of Zero Disruption operations.

The transition does not require rebuilding timelines or redefining scope.

- IT Operations resolves issues when root cause is immediately clear
- IT Operations escalates issues to DevOps with a defined starting point for deeper investigation

In the latter case, DevOps teams receive context rather than just symptoms. They begin with known affected transactions and services instead of generic performance complaints.

**This changes the nature of escalation.
It reduces time spent validating impact and
increases time spent addressing cause.**

Operational Implications across Teams

For IT Operations, tighter integration reduces friction. Fewer incidents stall due to uncertainty, and fewer require escalation solely to determine where to look.

For DevOps, incoming issues are better defined. Investigation begins closer to the source rather than at the perimeter.

For the organization, the effect is cumulative. Incidents resolve faster not because teams work harder, but because less effort is spent reconstructing context.

The operational model shifts from sequential handoffs to a shared investigative baseline.



Conclusion: Closing the Gap is an Architectural Decision

The gap between DEX and application observability is both a tooling and an architectural problem. Many applications remain uninstrumented, and existing tools often fail to connect experience signals to application level insight at scale.

Experience data without application context limits action.
Application data without experience context limits relevance.

Learn more about **Riverbed APM+** and how it extends application observability across the full application estate while connecting directly to experience insights from **Aternity**.

Bringing the two together creates a continuous path from impact detection to root cause. That continuity reduces investigative delay and prevents disruption from compounding during resolution.

The organizations that close this gap are not adding more monitoring. They are aligning detection and diagnosis into a single operational flow.

This continuity enables a move from reactive troubleshooting to Zero Disruption operations, where issues are resolved before they interrupt work and employees remain focused on outcomes rather than incidents.



About Riverbed

Riverbed, the leader in AI observability, helps organizations optimize their users' experiences by leveraging AI automation for the prevention, identification, and resolution of IT issues. With over 20 years of experience in data collection and AI and machine learning, Riverbed's open and AI-powered observability platform and solutions optimize digital experiences and greatly improve IT efficiency. Riverbed also offers industry-leading Acceleration solutions that provide fast, agile, secure acceleration of any app, over any network, to users anywhere. Together with our thousands of market leading customers globally – including 95% of the FORTUNE 100 – we are empowering next-generation digital experiences. Learn more at riverbed.com.