The value of embedded analytics is clear. Application teams that embed dashboards and reports drive revenue, reduce customer churn, and differentiate their software from the competition[1]. Given the benefits, it's no wonder the big question for embedded analytics has shifted from "Whether to?" to "How to?"

Application teams can choose between two development approaches to embed analytics: building what they need with the help of components libraries, or buying a solution from an analytics vendor. When does each make sense?

When to build analytics with components libraries	VS	When to use an embedded analytics platform
If your application has only simple analytics requirements, basic security, and no future plans to modernize or enhance the analytics capabilities, building with the help of components libraries may be right for you.		Applications that need embedded analytics to deliver end-user value, drive revenue, and differentiate their products from the competition will be better off using an embedded analytics platform.

"Organizations with internal developers are always tempted to build their own analytics," writes Wayne Eckerson in *The Ultimate Guide to Embedded Analytics*. "But unless the analytics are simple and users won't request change, it's always smart to outsource analytics to a commercial vendor. Commercial analytics products deliver best-of-breed functionality that would take in-house developers years to develop, distracting them from the host application."

Consider these 9 factors when comparing building with components libraries to embedded analytics platforms.

Effect on Core Application Development

Time spent building analytics is time taken away from your core application. Choosing to build analytics yourself takes time, and you end up replicating capabilities you could have easily bought from a vendor. This distracts your product team from focusing on core features and mission-critical functionality.

Time to Market

"Many enterprise application providers try to build analytics in-house, which can impact time to market," writes Gartner in *5 Best Practices for Choosing an Embedded Analytics Platform Provider*. Unless your requirements are limited to standard charts and reports, an analytics development platform will be faster to build, deploy, maintain, enhance, and upgrade than building analytics on your own. A platform supports everything from basic features to advanced capabilities, including data tables, visualizations, reports,

[1] Logi Analytics, *2018 State of Embedded Analytics Report*

dashboards, self-service analytics, workflow, write-back, and predictive analytics. And because a platform leverages your existing investments in data, security, and servers, you can develop, test, and deploy faster than building.

Analytics Maturity

As your application becomes more successful, your users will want more capabilities. An analytics development platform helps future-proof your analytics and makes it easy to incorporate modern features such as self-service analysis, predictive analytics, write-back and workflow capabilities, and bi-directional interactivity with secure parameters passing between the host application and embedded analytics. If you build, you'll be forced to research and develop each new capability one at a time, which delays every product update.

Security

Building analytics means implementing and maintaining security for every component you use. Managing security in multiple places takes significant effort and introduces unnecessary security risks. Unlike components, an embedded analytics platform globally reuses your existing security policies to ensure your security is always up to date and consistently applied for every tenant throughout the entire application.

Branding

Since every component you use will support theming on an individual level, white-labeling your embedded analytics means branding each component one at a time. Any modification to the brand means individually applying the new themes to every component—which puts you at risk of breaking one or missing a key component altogether. A platform makes white-labeling easy by globally theming the embedded analytics to match your application's look and feel and offering granular control over any attribute. After all, you've spent countless hours designing your application's user interface and user experience (UI/UX), making it core to your intellectual property (IP). Your embedded analytics should protect that work and enhance your brand.

Stability and Enhancements

Relying on third-party components libraries introduces risk when it comes to integration and upgrades. You have no guarantee all the components libraries you choose will work together forever or have consistent versioning, and a single upgrade may lead to regression issues. An analytics development platform ensures your application is always on solid footing. Every capability is thoroughly tested and works seamlessly together, now and with future versions.

Licensing

Many components come with complicated pricing structures that can hinder your application growth. If one component prices on a per-user basis, costs can quickly skyrocket as your business grows. Also keep in mind that using multiple libraries means managing multiple different licensing policies. On the other hand, platforms have predictable and easy-to-understand licensing structures that will scale with your business needs.

Support

Support and documentation will vary widely from component to component. Even with online communities, if something goes wrong, you'll need to solve the problem yourself (if you can solve it at all). Platforms offer reliable documentation and support to ensure your application is always running.

Long-Term Viability

Some of your developers will inevitably leave the company. What happens when they have all the institutional knowledge of your analytics solution? It will have a serious impact on your ability to maintain or upgrade your embedded analytics. Unlike components, platforms are thoroughly documented, so anyone can build, maintain, and enhance the embedded analytics with minimal disruption.