snowflake®

# 3 TIPS FOR BUILDING MODERN DATA APPLICATIONS

**How to choose a stack to power applications for analytics, machine learning, BI, IoT, and more**

CHAMPION GUIDES

# TABLE OF CONTENTS

# CLOSING THE GAP BETWEEN WHAT YOU HAVE AND WHAT YOU WANT

**By 2026, the global business intelligence and analytics software market is expected to reach $55.48 billion—a 10.4 percent compound annual growth rate over nine years.[1] In addition, at least 70 percent of enterprises will increase spending on real-time customer analytics solutions. Their objective is to use data-driven insights to better attract, serve, and retain customers.[2]**

This increasing demand for data applications is good news for SaaS and other cloud-based application providers. New application providers are disrupting established vendors in markets such as sales and marketing automation, customer relationship management, and across vertical markets such as healthcare, automotive, insurance, and finance.

However, while 83 percent of organizations want data translated into actionable insights at the optimal moment, only 22 percent believe they are achieving this goal.[3] SaaS providers have an opportunity to deliver differentiated and defensible value to their customers who are demanding powerful features and real-time insights so they can run their businesses better, faster, cheaper, and smarter. But SaaS providers must first build their apps on a modern architecture. A huge obstacle in delivering modern applications that delight customers is a continued reliance on legacy architectures that have scalability, concurrency and performance limitations.

Choosing the wrong data stack can be costly and obliterate any chance for success. This eBook addresses important architecture and performance issues, and provides three tips for choosing the right data stack when building data applications.

[1] https://www.prnewswire.com/news-releases/55-billion-business-intelligence-and-analytics-software-market---global-outlook--forecasts-report-2016-2026-300786041.html
[2] https://www.forbes.com/sites/louiscolumbus/2018/07/08/how-to-improve-customer-experiences-with-real-time-analytics/#18ee9a5a6e82
[3] https://www.forbes.com/sites/louiscolumbus/2018/07/08/how-to-improve-customer-experiences-with-real-time-analytics/#5d2e1cae6e82

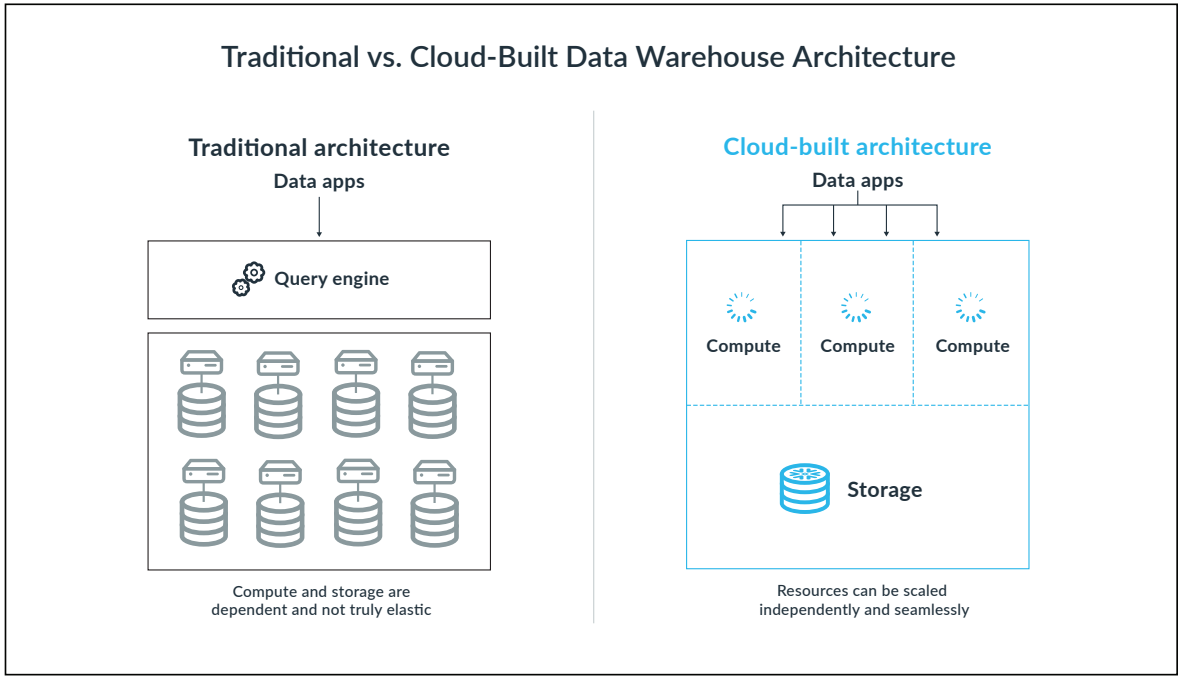
# FIRST, THE BASICS:
# WHY DOES ARCHITECTURE MATTER?

Whether data apps run in the cloud or on-premise, it's the infrastructure and architecture underneath the apps that are paramount. While the apps themselves have modernized, the unfortunate truth is that the infrastructure powering them is often ported over from the legacy world. For example, consider how legacy platforms such as the data warehouse allocate disk space. SaaS providers must select a certain amount to associate with storage and compute nodes, and the only way to increase availability is to manually add more nodes. To do so, the platform must be shut down or placed in read-only mode while data is manually redistributed. SaaS providers can attempt to plan for future peak capacity and demand by over-provisioning. However, this strategy is expensive and rarely allows for scaling back to match actual usage.

Why do these constraints occur? Because traditional data stacks were created long before the cloud existed. They were not architected to run massive-scale SaaS applications with semi-structured data. Their design assumed small clusters of machines with predictable amounts and types of structured data created largely by internal sources.

Not surprisingly, traditional data stacks struggle with today's volumes of data, which frequently arrive in schema-less and semi-structured formats from external sources such as application logs, web applications, mobile devices, social media, sensor data, and IoT (Internet of things) data.

Unfortunately, many software vendors start their data analytics app journey by adopting generic architectural plumbing and tools that allow for quick development without upfront investment. This shortcut saves money initially, but inherent technical challenges arise that cause lower throughput and a frustrating and disappointing customer experience. Eventually, SaaS providers realize they must invest in re-architecting if they want to deliver on the promise of modern, real-time app experiences.



# REACH THE NEXT FRONTIER:
# HIGH PERFORMANCE, MASSIVE SCALING SAAS

SaaS providers must invest in a modern stack that addresses the shortcomings of legacy architectures, enables real-time analytics in a cost effective manner, and delivers fast and reliable experiences customers actually find useful.

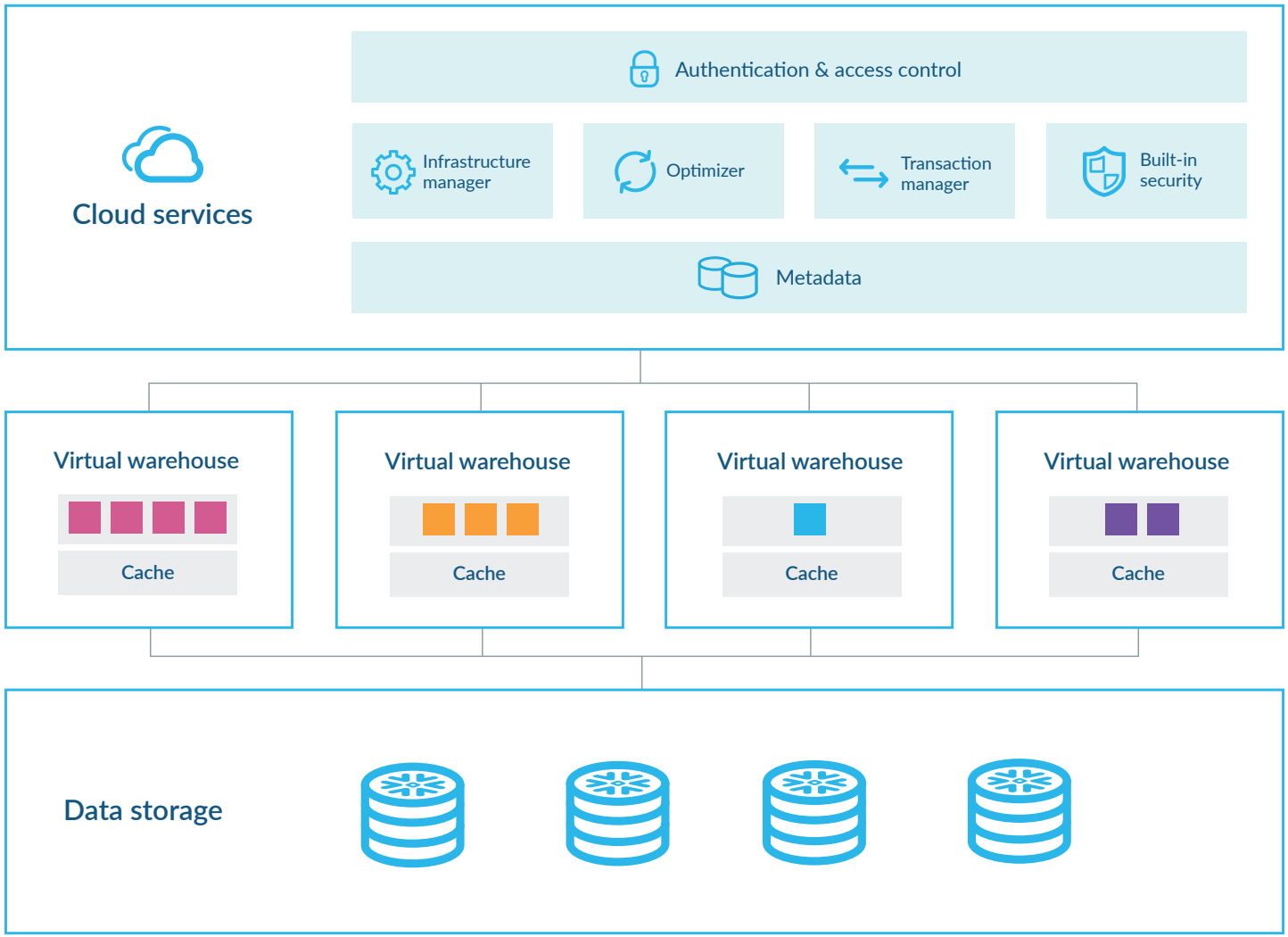
What core capabilities should a modern data platform for analytics apps include?

- **Decoupled resources:** Apps should scale compute resources independently and linearly for each job without manual intervention or disruption. When technology "building blocks" are separated by design, resources can scale instantly as needed. This decoupling enables optimization that's fast and efficient, which means applications never miss SLAs. Decoupling also enables virtually unlimited concurrency so you can run multiple queries against the same data without conflict. As an added bonus, it's easier to rewrite individual components of the app when they're smaller and less intertwined with other parts.
- **Elasticity, self-service, and pay-as-you-go:** The architecture should enable apps to adapt to workload changes by growing or shrinking resources dynamically and automatically. With a modern cloud-built architecture, organizations can start small, increase usage as needed, and benefit from auto-scaling to ensure resources are allocated as needed. In addition, only active clusters should accrue charges so organizations pay solely for what they use, down to the second.
- **Support for breadth of data types:** With modern architectures, apps must support all structured, semi-structured, and unstructured data in order to provide customers with a holistic view of their data. Traditional data stacks will always fall short because of their limited processing capabilities, inadequate memory, high data storage costs, and singular support for structured data.
- **Developer tooling and automation:** Applications must scale with ease and enable developers to "plug in" services rather than re-architect with each addition. While traditional data platforms require a lot of manual heavy lifting, modern stacks leverage APIs to connect with other services in the data ecosystem. This "building blocks" approach to development provides speed and eliminates unnecessary complexity.
- **Security:** Security must be baked into the architectural design in order to enable fast development while protecting against present-day security threats. Modern stacks address this dual need by providing fine-grained, role-based access control for data and actions, always-on encryption of data stored in the cloud, and automatic protection against accidental or malicious loss of data.

Fully managed service

Dedicated compute clusters and independent from storage

Single source of truth separate from compute



Here are three tips to ensure data apps live up to your customer expectations.

## TIP 1: BE OPTIMISTIC: DESIGN FOR GROWTH AND INCLUDE FLEXIBILITY

Customers now demand real-time analytics about their business. In response, some vendors are adding embedded analytics to current SaaS applications as a way to increase their value proposition. Others recognize the opportunity to create an additional revenue stream by using their vast stores of data to serve up standalone analytics.

Whether you're building an analytics app from scratch or augmenting an existing app with new capabilities, it's critical to align your technology decisions with long-term product needs. It doesn't matter if you're releasing version 1.0 or version 10 of your app: Technology decisions should never reflect present-day requirements only.

- **Account for scalability from the get-go.** The fastest way to disappoint customers and waste development effort is to use components that don't scale. Because workloads aren't predictable and downtime is practically forbidden, you need the ability to instantly and infinitely scale compute up, down, and out to any and all workloads without impacting performance. Unfortunately, open-source solutions don't scale automatically, which puts the burden solely on your development team.
- **Be ready to support evolving requirements.** Assume your app has longevity and know with certainty that your customer segments will change, as will their needs. The underlying stack must be able to support these future pivots and shifts in product requirements. And it's much less costly to keep long-term technology needs in mind at the beginning rather than pay the piper later to re-architect completely. Otherwise, you'll be forced to build apps twice, which is expensive and eats up valuable development time that could be used to build new features.
- **Avoid getting locked-in.** Many organizations are embracing a multi-cloud strategy (i.e. using multiple cloud providers such as AWS, Azure, and Google) because it allows them to decrease risks of downtime. However, the only way to make multi-cloud work is to have a cloud-built architecture running underneath your apps that provides a single code base across cloud providers. This is no small feat, given that cloud providers are incompatible by nature. If you want to be multi-cloud as you grow, you must select an architecture that supports that strategy.

In short, decisions about selecting a data stack can't be made with a short-term lens. Instead, it's important to focus on flexibility and always keep in mind evolving customer needs.

## TIP 2: NOTHING COMES FREE: "FREE" SOFTWARE IS NO EXCEPTION

New data apps tend to be built almost entirely on public cloud infrastructure and use APIs to bring together core features. However, not all cloud-based solutions are created equally.

Many software vendors start their app journey by adopting generic low-cost tools that allow for quick development without upfront investment. For example, open-source tools such as PostgreSQL, Elasticsearch, and NoSQL databases are tempting because they are an easy starting point to get up and running. This shortcut allows a new analytics app to be sent to market without the upfront cost of procuring a database, which alleviates a potential friction point between development and finance teams.

Sound too good to be true? It is. Four common development challenges arise when you don't consider thoroughly what's needed from a data stack to deliver powerful data analytics apps.

1. **The problem many apps face** is the increasing data storage and compute strains on the system that come from more and larger customers. Open-source tools require manual and disruptive scaling that impacts the customer experience and requires intensive time and effort from an organization's most valued engineering resources.
2. **Native support for semi-structured data.** Using data types such as JSON, XML, and Avro is a huge challenge if the open-source solution doesn't natively support the vast troves of semi-structured data available today. In these scenarios, data engineering teams are forced to build and maintain complex data pipelines, making real-time insights difficult to achieve and deliver consistently.
3. **Maintenance and security.** While development teams should spend their time writing and developing analytics applications, open-source solutions require overhead in the form of frequent maintenance and upgrades. As a result, developers end up dealing with system maintenance instead of coding. Open-source solutions also leave companies open to security vulnerabilities if teams aren't on top of critical fixes.
4. **Expertise.** The use of open-source tools requires specific skills that may not exist within an organization. As a result, companies need to hire more resources, which can be challenging to find and expensive to acquire. Also, open-source solutions are not specifically designed with business outcomes in mind, which puts the burden on a team of engineers to configure a stack that can match customer requirements.

These challenges are very similar to the problems that development teams encounter when they use a traditional data platform. For example, shared-disk and shared-nothing architectures put limitations on queries because of concurrency issues, and performance is slow because of competing workloads. Traditional data platforms such as data warehouses also present challenges with integrating semi-structured data and require constant management and in-house expertise.

Adopting an open-source data stack typically results in a complete re-architecture down the line to account for these limitations. In the meantime, customers have been left frustrated by latency and incomplete data analysis, and large amounts of development time have been wasted on running and maintaining "free" solutions.

## TIP 3: IT'S SIMPLE, REALLY: MODERN APPS REQUIRE A MODERN INFRASTRUCTURE

There are two straightforward requirements for data applications such as analytics, BI, IoT, and machine learning:

1. **Large volumes of data must be ingested, and**
2. **All data must be analyzed quickly and easily**

In other words, a central repository is needed that provides workload isolation, instant and near-infinite elasticity, unlimited concurrency, and the ability to natively ingest semi-structured data.

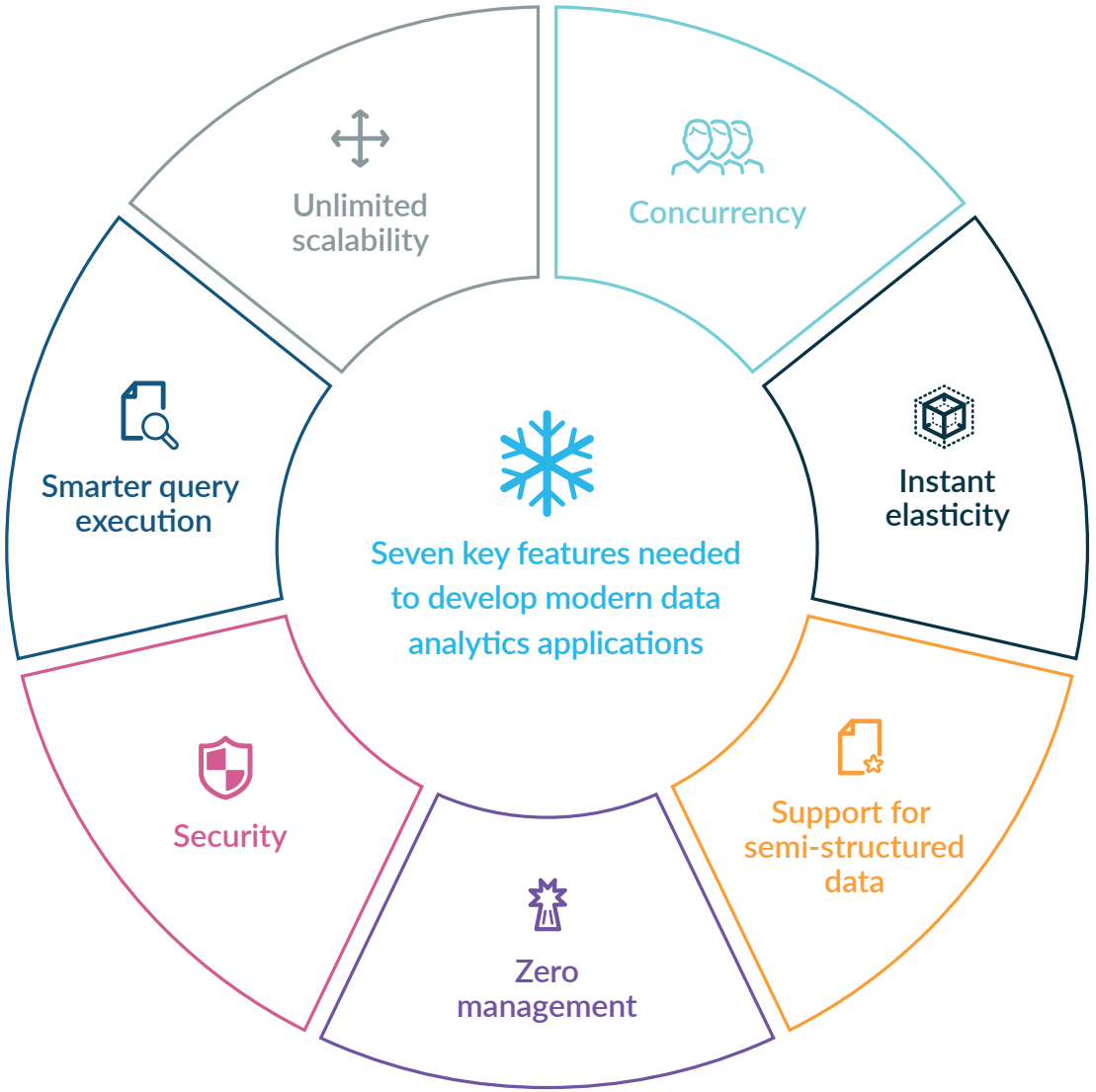
Would it surprise you to learn that a **cloud-built data warehouse** provides the exact platform you need to run modern analytics applications and deliver value to end customers? The last 15 years have seen a shift from traditional data warehouses to modern cloud solutions that provide scalability, elasticity, flexibility, and performance, all with minimal maintenance.

Remarkably, many teams don't immediately realize they need a data warehouse as part of their data stack. It's only when technical issues start to arise that the architecture is reconsidered, which can be disastrous: Total cost of ownership (TCO) is much lower when the right technology selections are made from the get-go.

With a cloud-built modern data warehouse as your stack, all of the features needed to develop and scale modern data analytics apps are built into the architecture from the ground up:

- **Unlimited scalability.** Provides independent scalability of compute and storage by separating the two. Scalability is automatic and without limits. That means organizations have the ability to instantly and infinitely scale compute up, down, and out to any and all workloads without impacting performance.
- **Concurrency.** All users share one copy of the data inside of an architecture built to separate compute from storage. Data can be used by multiple users at the same time through multiple compute clusters. Contention is eliminated between workloads since each user has dedicated and independent compute, so users don't experience slowdowns or disruption to queries.
- **Instant elasticity.** Provisioning is immediate for on-demand performance. This elasticity allows users to dynamically match their resources to usage demands, independent of data volume.

- **Support for semi-structured data.** An integrated approach to handling semi-structured data (JSON, Avro, Parquet, and XML) is a must. This means it's loaded and stored natively with structured data, and all data is used for queries and analysis without any additional work or wasted effort.
- **Zero management.** With a data stack offered as-a-service, there's no infrastructure to manage or optimize, and there is no downtime for software updates.
- **Security.** A modern data stack should provide always-on secure data environments with the option of operating in virtual private clouds (VPC) for added protection. No customer configuration or ongoing management is required.
- **Smarter query execution.** Your users should get better price/performance by optimizing how queries run. After all, there's little value in executing a query over 1,000 nodes if another system can do so in the same amount of time using 10 nodes.[4] Plus, these types of warehouses allow organizations to pay only for what is used, down to the second.

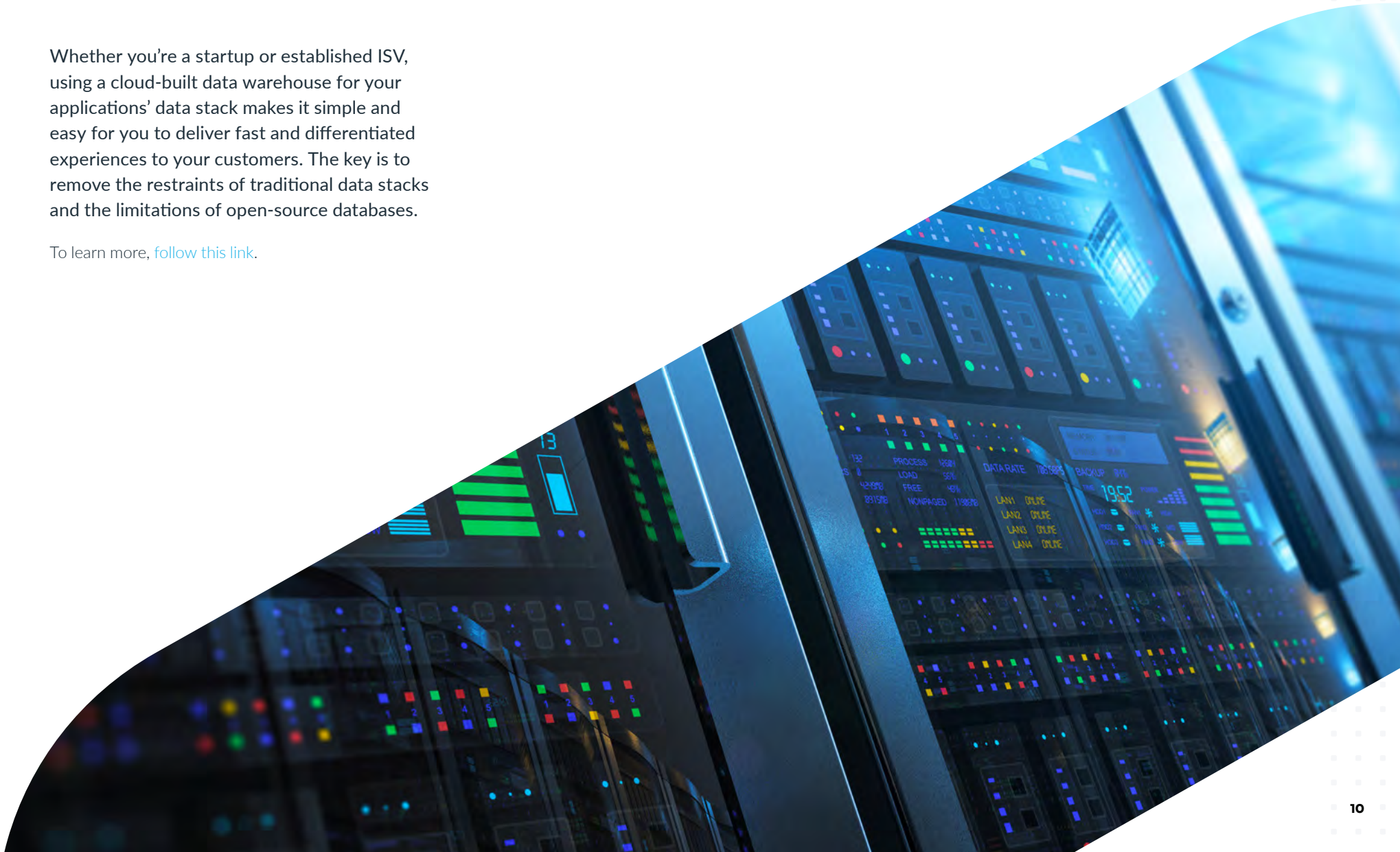



[4] http://pages.cs.wisc.edu/~remzi/Classes/739/Spring2004/Papers/p215-dageville-snowflake.pdf

# DON'T LET THE DATA STACK GET BETWEEN YOU AND YOUR CUSTOMERS

Whether you're a startup or established ISV, using a cloud-built data warehouse for your applications' data stack makes it simple and easy for you to deliver fast and differentiated experiences to your customers. The key is to remove the restraints of traditional data stacks and the limitations of open-source databases.

To learn more, follow this link.



# ABOUT SNOWFLAKE

Snowflake is the only data warehouse built for the cloud, enabling the data-driven enterprise with instant elasticity, secure data sharing, and per-second pricing across multiple clouds. Snowflake combines the power of data warehousing, the flexibility of big data platforms, and the elasticity of the cloud at a fraction of the cost of traditional solutions. Snowflake: Your data, no limits. Find out more at **snowflake.com.**