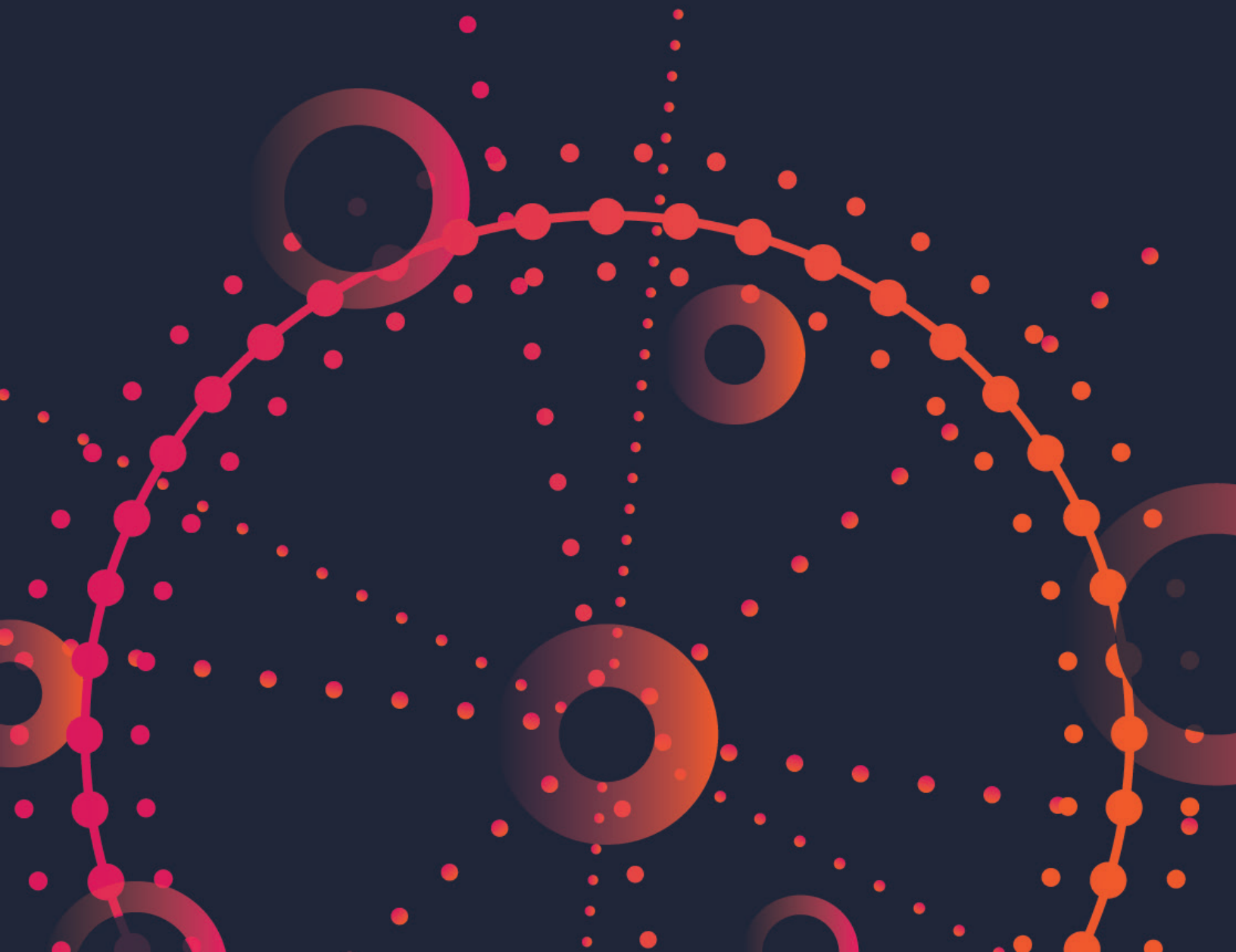
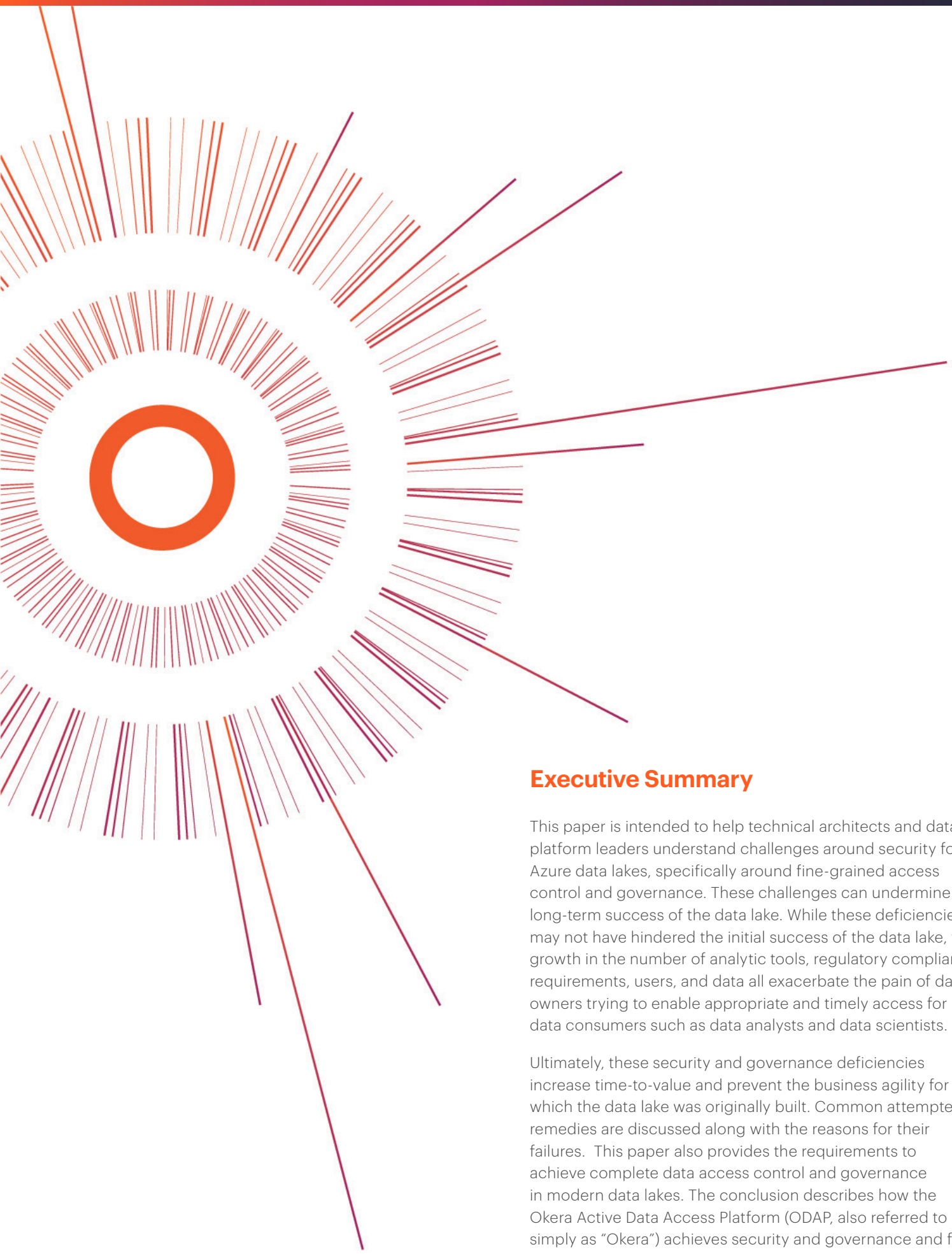
OKĒRA

MAY 2019

SECURITY AND GOVERNANCE FOR AZURE DATA LAKES

Scalable security and governance enables business agility





Executive Summary

This paper is intended to help technical architects and data platform leaders understand challenges around security for Azure data lakes, specifically around fine-grained access control and governance. These challenges can undermine the long-term success of the data lake. While these deficiencies may not have hindered the initial success of the data lake, the growth in the number of analytic tools, regulatory compliance requirements, users, and data all exacerbate the pain of data owners trying to enable appropriate and timely access for data consumers such as data analysts and data scientists.

Ultimately, these security and governance deficiencies increase time-to-value and prevent the business agility for which the data lake was originally built. Common attempted remedies are discussed along with the reasons for their failures. This paper also provides the requirements to achieve complete data access control and governance in modern data lakes. The conclusion describes how the Okera Active Data Access Platform (ODAP, also referred to simply as "Okera") achieves security and governance and full interoperability with the Azure data lake ecosystem.

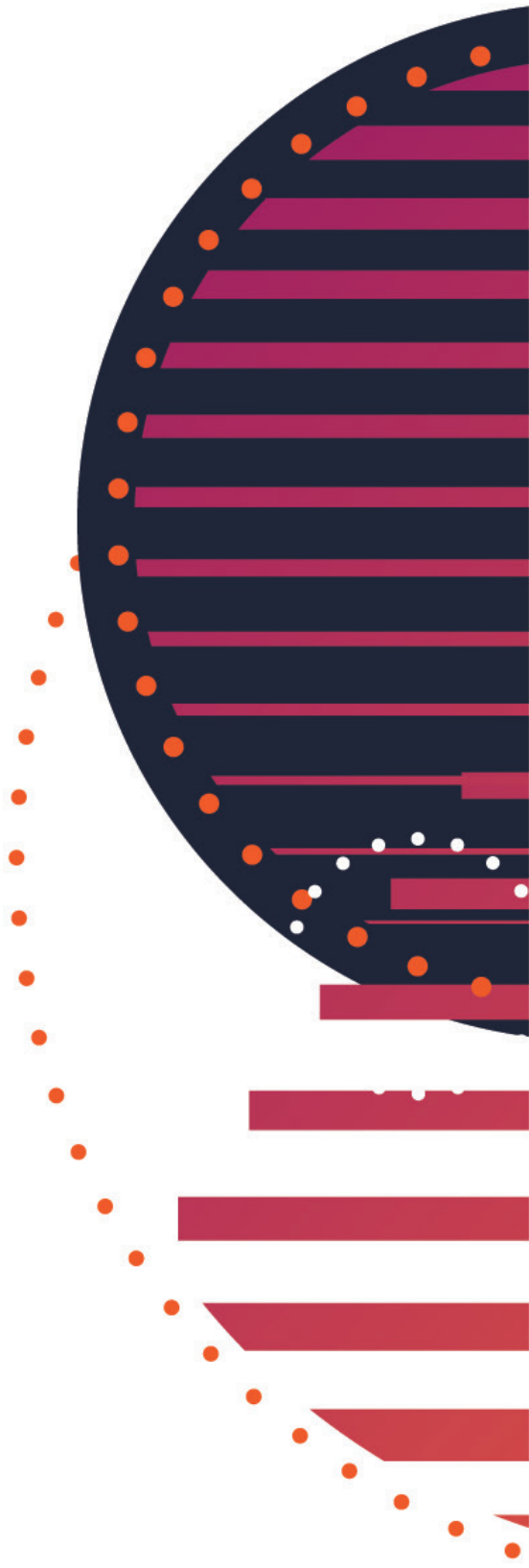
Introduction

The most successful companies recognize data is an asset to serve their customers better and drive business value. Data lakes were built for these purposes. Working with different kinds of data, whether structured or unstructured, and running a variety of workloads on a variety of data give organizations the flexibility to use the data assets more effectively. Making this a self-service experience is commonly top-of-mind for platform teams.The cloud is driving this closer to reality, where numerous lines of businesses and different user personas are able to use best-of-breed tools for the job at hand.

Azure-based data lakes are becoming increasingly popular. In typical architectures, Azure Data Lake Storage (ADLS and ADLS Gen2) is the core foundational building block and is treated as the scalable and flexible storage fabric for data lakes. Multiple compute engines from different providers can interact with ADLS Gen2 to enable the workloads. These include, but are not limited to, frameworks offered by HDInsight, Databricks, Qubole, and several other flavors of both open source and proprietary data processing, analytics, and machine learning frameworks. To truly realize the potential of data lakes, multiple technologies have to work in concert. The diversity of tools and abstractions, however, can get in the way of accomplishing this. The major area where this creates friction is security and governance, and this threatens the long-term viability of the data lake.

Security (access control) and governance (auditability and visibility) are inherently lacking in a data lake architecture and thus typically overlooked in the early stages of a data lake deployment. In the early days of their data lake journey, organizations typically work around these deficiencies by creating plumbing to create copies of data for different users and use cases depending on the entitlements. Even with that, there are significant gaps in the functionality: fine-grained access control across multiple compute tools and different kinds of data assets, with detailed visibility into user activity. People make do in the early days of their data lake journey, but this problem is increasingly exacerbated over time with the addition of more tools, users, compliance regulations (GDPR, CCPA, and so on), and data. These factors drive up inefficiencies, costs, and risk — in addition to increasing time-to-value by slowing analyst productivity, which ultimately prevents organizations from accomplishing the main goal of the data lake: agility for the business.

Let's look at how people have tried to solve the problem of access control and governance in Azure-based data lakes.



Common Approaches and Pitfalls

Assuming that protecting and governing data is a requirement, you have three general approaches with your data lake:

1. **Rely on Azure Identity and Access Management (IAM)**
2. **Restrict technology choices**
3. **Use open source authorization tools**

RELY ON AZURE IDENTITY AND ACCESS MANAGEMENT (IAM)

The most common approach for ADLS Gen2 data lakes is to use IAM policies and permissions (including Azure Active Directory and its variations) to manage access as it is native to Azure. However, it has **SEVERE DRAWBACKS**, including:

Coarse Granularity of Permissions
IAM enables coarse-grained permissions, with the most granularity being files. Anything more fine-grained requires you to create copies of files with the sensitive content stripped out.

Cumbersome Management of Policies
Managing IAM policies requires you to update objects every time you need to grant new access to users. This is error-prone and becomes hard to manage, even in a reasonably small data lake.

No Obfuscation Capabilities (Anonymization, Tokenization, Masking, and Redaction)
Using IAM policies enables you only to give access to a file or not. You can't have anonymized, tokenized, masked, or redacted versions of the data without creating copies of it.

No Auditability or Visibility
IAM defines which roles or users can get access to which files. Auditability and visibility can be accomplished using Azure Diagnostics logs, which aren't very granular and don't provide detailed, easy-to-use reports. For example, it's difficult to get the visibility you need to understand what users are doing with what datasets and which tools they are using. It's also almost impossible to answer questions like: "Who has access to this dataset?" "What is their view of it?" "What does this user have access to?" "How does this user get access to this dataset?"

Difficulty Changing the Underlying Dataset
If new partitions or files are added to a particular dataset, depending on how you define IAM policies, they may have to be updated. If you define them at the finest granularity possible, you'll need to update them. If you define them at the coarsest granularity, you will have far less control than you should need for meeting various regulatory and privacy requirements.

Difficult/Impossible to Map to Other Systems
Down the road, IAM policies to manage data access cannot be mapped to other systems, such as Azure Database, Azure SQL Data Warehouse, Snowflake, or Kafka.

RESTRICT TECHNOLOGY CHOICES

The second approach is to restrict the technology choices and use only specific compute frameworks from specific vendors. For example, you could use only Azure HDInsight or Azure Databricks for all your data processing and analytics needs and not use anything else. This way you

can secure and govern via HDInsight's engines and not open up access to the data to any other kind of engine, e.g., Azure Databricks, Tensorflow, Qubole, or Presto. An extreme version of this approach is to use databases for everything; an approach the market has dismissed (as SQL is not the answer to everything, for example with machine learning, and so on). The drawbacks of this approach include:

Limited Agility
Each technology, including the database, is purpose-built to solve a set of problems, and any mature, data-driven organization will have multiple technologies in their environment at any point in time. However, with this approach, access control and governance is bundled into a subset of the compute tools that severely limits business agility and data professionals are forced to select tools from a limited toolkit.

Competitive Disadvantage
By restricting technology choices, users rarely get to try out new analytic tools that could represent a competitive advantage. This reduces the value of the data lake today and limits the ability to quickly evolve as business requirements change.

TRY USING OPEN SOURCE AUTHORIZATION

The third approach is to try to use an open source policy engine like Apache Ranger. The challenge with this approach is that it covers only HDInsight and thus does not handle other compute engines such as Azure Databricks, Qubole, Azure Machine Learning Service and Tensorflow . The challenges with this approach include:

Significant Customization
While Ranger appears to be an effective way to unify access management, it requires a lot of custom plumbing in the form of connectors to each compute type.

Functionality Gaps
Even with a customization effort, this approach leaves significant functionality gaps, especially in the ability to enforce data-centric access policies consistently across different analytic engines, such as Apache Hive and Apache Spark.

Dependence on Vendor Alignment
Using open source frameworks like Ranger that depend on deep integration with the storage and compute engines makes you highly dependent on all vendors aligning and doing the heavy R&D work to build the complete functionality. They must also be willing to maintain their solutions for the long term

For more details see this blog on "Using Apache Ranger for Access Control and Data Governance in the Cloud."

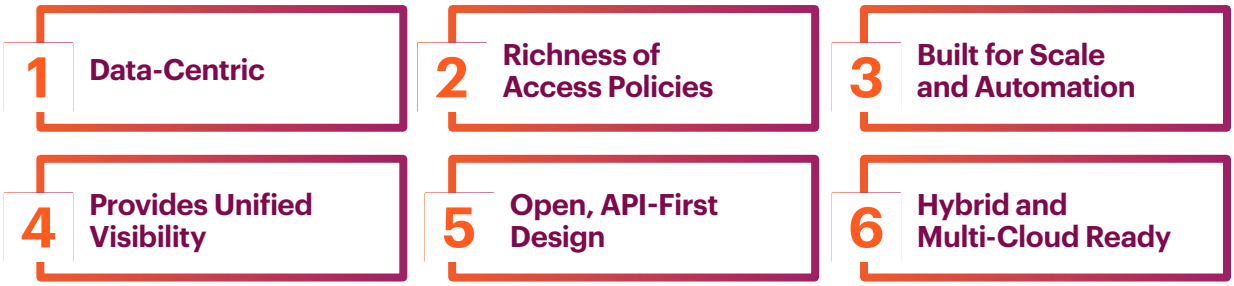
Common Approaches Fall Short

As you can see, if you need to protect the business, the existing approaches leave much to be desired. At the very minimum, the first and third approaches increase the amount of data engineering and plumbing costs and slow down the entire experience for the data lake users. They create a lot of friction for the data consumers and create dependencies and long lead times that hamper their work. Despite that, there are functionality gaps. The second approach completely undermines the purpose of a data lake and the flexibility of analytics and data types; it's like betting everything on the SQL database again.

Let's take a step back and define the ideal capabilities of a data lake access management and governance solution that will enable the data lake to deliver on its promise

Six Tenets of Data Lake Access Control and Governance

Achieving complete access control and governance for a data lake requires following six tenets, which together yield flexibility, scalability, openness, and agility. Your access control and governance solution has to align with these tenets in order to achieve complete business agility for the enterprise. The six tenets are:



DATA-CENTRIC

Data access policies and governance must be data-centric and not based on the storage system or the analytics engine being used. The solution needs to have the ability to enforce policies consistently, regardless of the consuming analytics engine. Even if you use only a handful of tools today, your data lake needs to be capable of supporting future requirements including pure SQL, structured but not SQL (data frames, Spark), hybrid, Machine Learning (ML) and Business Intelligence (BI) at scale. The access control and governance solution has to contribute to future-proofing your architecture, not limiting it.

RICHNESS OF ACCESS POLICIES

The solution needs to provide access control for both structured and unstructured data and at various granularities. For unstructured data, the granularity should range from several folders to individual files. For structured data, the granularity should range from a set of data sets, to individual datasets, to columns, rows, and even cells. Along with that, the solution needs to support anonymization, tokenization, masking, and redaction of data, generally referred to as obfuscation, for different users and use cases including Differential Privacy, i.e., inability to determine a person's identity from given data. It should handle role-based access control (RBAC) as well as attribute-based access control (ABAC) to easily tag sensitive data and assign policy to it at scale. The policies need to be able to support regulations such as GDPR and CCPA by enabling the core requirements of consent management and right to erasure.

BUILT FOR SCALE AND AUTOMATION

Your data lake is built to handle a scale that was previously impossible — without also having to scale people and integration costs. Your access control and governance solution needs to be able to handle definition, enforcement and ongoing management of access policies at scale in the same way:

Definition
The simplest way of defining access policies is based on roles and dataset definitions. As policies get more sophisticated, the solution needs to support richer and more scalable constructs, such as context-based dynamic views and attribute-based policies. These will make defining policies at scale much easier.

Enforcement
Policies need to be applied to datasets and workloads at all kinds of scale for all kinds of tools, ranging from single digit gigabytes to multiple petabytes. All this has to be done without incurring any performance overhead.

Management

Managing the access policies needs to be based on an API-first design specifically around fine-grain access control and allow automation. The number of datasets and people engaging with your data lake will be high, and manual methods of managing policies will inhibit your ability to scale your data lake over time

Additionally, it must enable tooling and automation as well as background processing to make the data lake more intelligent, such as by organizing files for performance or auto-detecting data content..

PROVIDES UNIFIED VISIBILITY

A critical part of governance is visibility. This can be broken down into two aspects: Historical and Current State. Historical visibility entails giving you a view into user activity and access patterns that have already taken place. This is provided by an audit trail that the system generates. In addition, the system needs to provide Current State visibility. This means the ability to answer the auditability and visibility questions noted above, including "Who has access to a given dataset, and what is their view?" "What can a specific user access?" "How can access to a dataset be obtained?"

Your access control and governance solution needs to address both aspects of visibility. Moreover, the quality and richness of the content in the audit trail cannot vary depending on the consuming application or the source system. While potentially sufficient for auditing in some cases, this inconsistency makes it impossible to build the next set of capabilities for your data lake, such as usage analytics, charge-backs, resource management, and throttling. The solution needs to provide rich visibility such that you can accomplish these capabilities easily.

OPEN, API-FIRST DESIGN

DData lakes are all about flexibility and extensibility over time. As the world of analytics and machine learning continues to evolve, new tools, frameworks and vendors will join the ecosystem. All of them will need to respect and participate in access control and governance. The solution you lay down today needs to make this possible. This means utilizing a simple, service-oriented architecture that is API-first in design.

Insisting on an API-first design will mean easy integrations with current and future enterprise tools, such as AD or SSO systems for identities, log management frameworks for diagnostics and anomaly detection, and catalogs for business metadata.

Your access control and governance solution also needs to be agnostic to the storage and analytics platforms and vendors you choose over time. A solution that is limited to a given platform (storage or analytics) will likely create a challenge for your data lake — because of both a lack of openness and a misalignment between the incentives of the vendor and your business.

HYBRID AND MULTI-CLOUD READY

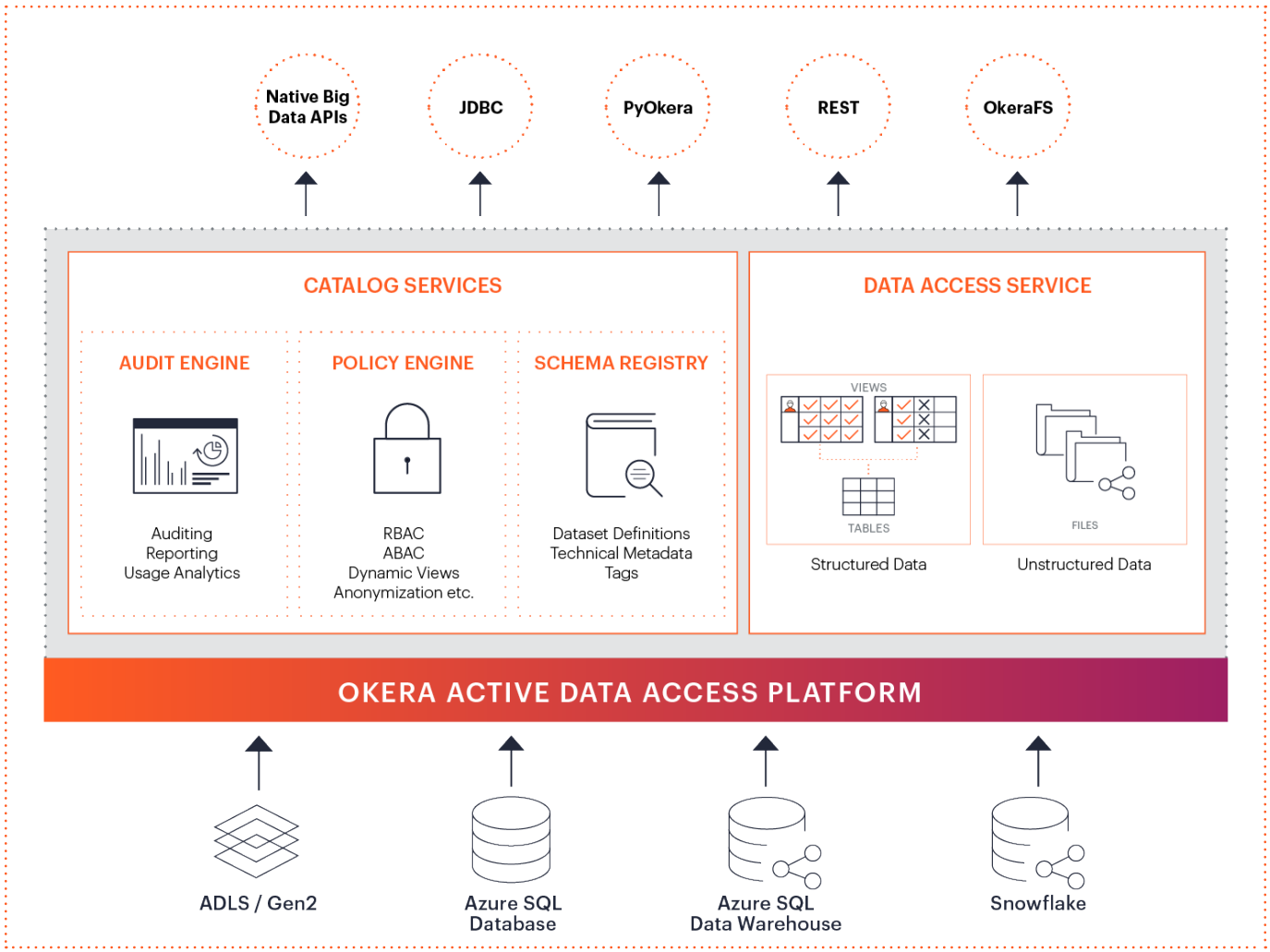
For a modern organization that aspires to be agile and use best-of-breed technologies to create business value faster, it's important to make technology decisions that will not impede the broader goals. This means that in addition to the API-first design noted above, the data lake access control and governance solution should be agnostically cloud-native in nature and support hybrid infrastructure. This combination will ensure that you are picking a solution that future-proofs your architecture, and you won't need to revisit it any time soon.

With these six core tenets in mind, we built and brought to market the Okera Active Data Access Platform. The following section provides an overview of the product, its various components, and what they do.

Okera Active Data Access Platform Overview

The Okera Active Data Access Platform (ODAP, also referred to simply as "Okera") eliminates the data security and governance challenges of the Azure data lake and guarantees agility, no matter the increase in data, users, compliance regulations, and tools. Okera removes the friction between access and governance by providing an active management layer that makes data lakes accessible to multiple access tools and workload patterns while enforcing fine-grained access policies. Okera eliminates the complexity of underlying storage systems and files and provides a higher-level abstraction of logical datasets, which is much more intuitive for the purpose of analytics and machine learning workloads. This level of abstraction lets you provide data owners and stewards with granular control over and visibility into data usage while providing data analysts and data scientists with fast, self-service access to data from the data lake at scale. Structured data is made easily accessible via familiar tables and views. For unstructured data, Okera provides a file system API, Okera FS, that enables access to unstructured data with easy-to-manage, fine-grained access policies and detailed visibility into usage patterns.

The Okera platform was designed with **high performance, scalable, fine-grained access control and auditability** in mind. Users can employ any analytics tool they want using industry-standard APIs.



The Okera Active Data Access Platform (ODAP)

The Okera Active Data Access Platform consists of two main components:

- **Catalog Services:** contains the Schema Registry, Policy Engine and Audit Engine. Together they manage the fine-grained security policies, audit trail, and metadata.
- **Data Access Service:** enforces the security policies by dynamically providing restricted views to end users. The same user, accessing the same dataset, using any tool will always have the exact same policies applied, even when the dataset is retrieved as a file using Okera FS.

Okera's easy-to-use abstraction of logical datasets is presented as tables. This not only removes the complexity of different APIs, file formats and coarse-grained access methods, but also allows Okera to bring sophisticated data management capabilities that mature relational databases are known for. In addition, Okera is modular in nature and exposes common APIs to make it easy for platform teams to integrate it within their environment. It also gives them the choice to use other services, such as enterprise data catalogs, as they require.

OKERA CATALOG SERVICES

Okera Catalog Services are a unified, common set of technical metadata services that provide vital details directly to users, in conjunction with business metadata coming from other Catalog services, and to the Okera platform itself. The system stores dataset definitions, access policies, and any other metadata that you may choose to include, and this information can be shared across different storage systems and analytics tools. The services include: **Schema Registry, Audit Engine, and Policy Engine.**

These services provide the following primary functions:

- Dataset **registration and publishing** for both structured and unstructured files or data stored in traditional relationship systems (RDBMSs)
- Management of **fine-grained access policies** down to the individual cell level, including user-defined **anonymization, pseudonymization (tokenization), masking, and redaction**, as well as other security related functions. Granularity is down to the cell-level for structured data and down to the file level for unstructured data
- Dataset **search and access** for analytics
- Comprehensive **auditing and reporting** for every metadata operation and any access that is processed by the platform

Typically, a single instance of Okera Catalog Services is deployed and shared across the enterprise. The services expose standard APIs of the Hive Metastore (for schemas) and REST APIs to interact with the metadata. This makes it the long-running, common metastore for different Hadoop components, regardless of the infrastructure they run in (on-premises or cloud) or which analytics tool they use (Azure HDInsight, Cloudera, MapR, Databricks, and so on). REST APIs can be used to integrate with any other systems and workflows that may already be in place.

OKERA DATA ACCESS SERVICE

The Okera Data Access Service is a scalable, fault-tolerant distributed service that lets businesses use multiple analytics tools on their data lake, while ensuring access policies and auditing are always enforced. Okera's Data Access Service handles the heavy I/O while providing data to the analytics tools after applying schema, fine-grained security policies, and other transformations (for instance, UDFs, tokenization, masking) with high performance. Data provisioned in this form is easily consumable and delivered as a familiar abstraction of tables, or in the form of files in a format preferred by the user. Different analytics tools, such as Spark, Python, SQL engines, BI tools, and spreadsheets, can all interact with this service. With Okera, every tool works with the same view of the data based on the individual user security policies that have been applied.

Users with different use cases and different choices of tools are able to interact with the Okera platform. Regardless of what tool or access API they use, they will get the same view of the data.

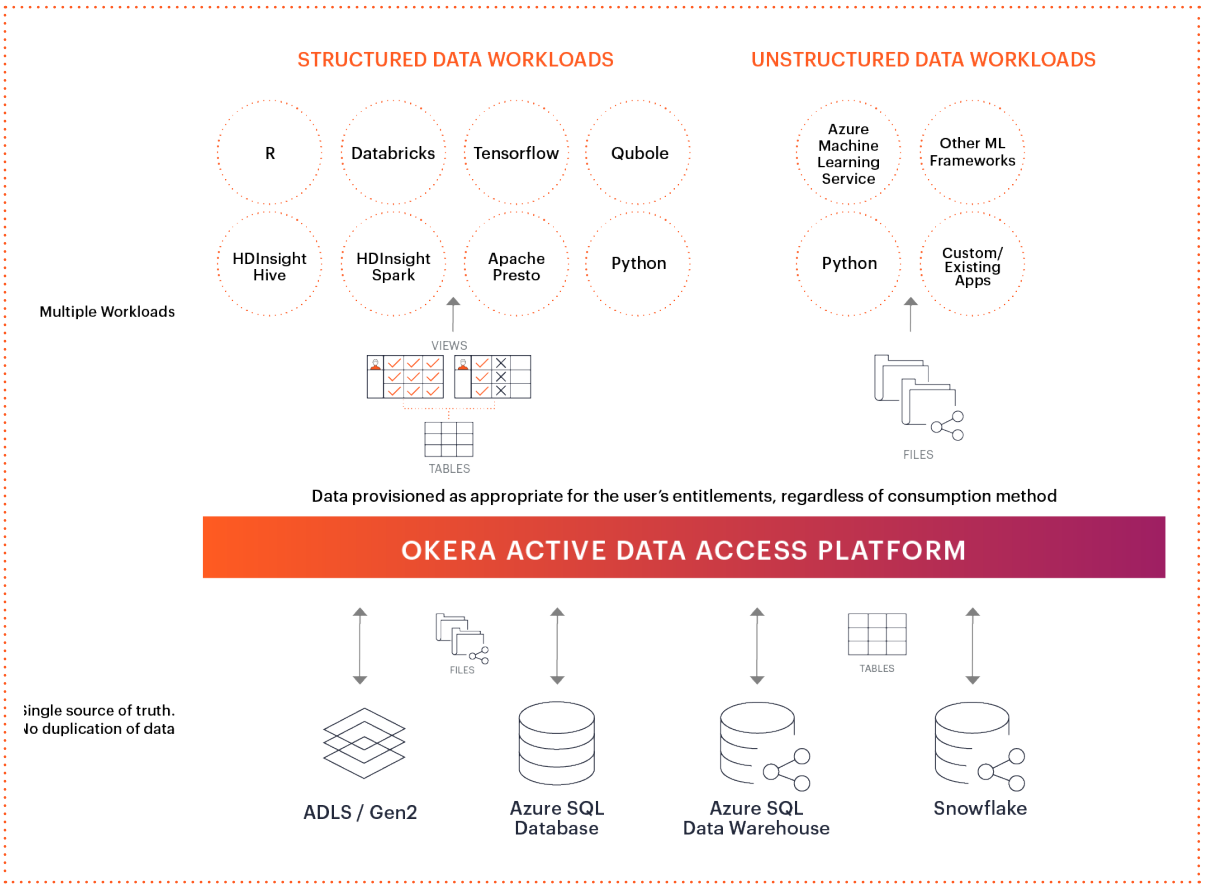
Typically, users will want to run multiple instances of Okera's access service (sharing the same catalog and therefore metadata) running inside their environment. These instances may be ephemeral, while others may be persistent, aligned with different business units or use cases. In other cases, instances may act as independent services, whereas others may be co-located with an analytics framework. You can choose between flexible deployment models based on your performance and isolation requirements.

Now that you know what the Okera product is and what it can do, let's look at how it solves the problem of enabling a secure and governed Azure data lake.

Okera in action: Enabling a Secure Azure Data Lake

Okera deploys inside your Azure account using containers. These containers can be deployed on plain old Azure virtual machine instances or on higher-level abstractions like Azure Kubernetes Services (AKS). Okera uses a relational database to store its state (metadata and configurations). You can use Azure Databases or a self-managed database for this.

A typical Okera deployment on Azurew is shown in the figure below.



Okera easily plugs into your Azure ecosystem

Okera exposes the following APIs in a typical deployment:

METADATA OPERATIONS

Okera exposes APIs for metadata operations such as registration of datasets, defining access policies (including anonymization, tokenization, masking, and redaction functions). These APIs can be used to integrate with end user-focused data catalogs. They are exposed either as a SQL interface or REST APIs.

DATA CONSUMPTION

The Data Access Service exposes several different APIs for data consumption. Okera enforces fine-grained access policies consistently across all of them.

- Native integration with big data frameworks. This includes Spark, Hive, Presto, Impala, and MapReduce. These could be self-managed open source distributions or commercial offerings such as those from Azure HDInsight, Databricks, and Qubole.
- Native JDBC. A fully featured JDBC endpoint can be used to enable data consumption and analysis using any tool that works with that API. These include, but are not limited to, BI tools such as Microsoft Excel, Tableau, Micrsoft Power BI, data prep

tools, data science workbench tools.

- Native Python integration. Okera provides a native integration with Python that enables a fast and easy way to build applications using popular python frameworks like Pandas, NumPy, Scikit, Tensorflow, and boto.
- REST API. To build custom applications that aren't covered by the above mentioned integrations, you can use the REST APIs that Okera exposes. REST APIs can be used for both metadata and data operations.

DATA SOURCES

While this paper focuses on Azure data lakes, another benefit of using Okera is that it supports multiple other source systems. In addition to ADLS Gen2, you could have HDFS or a relational database, such as Azure Databases or data warehouses such as Azure SQL Data Warehouse and Snowflake, as data sources. This enables unification of access management across multiple source and consumption systems. You don't have to enable all the sources in one go. The intent of the design is to future-proof the architecture and enable the addition of sources beyond Azure as your needs evolve.

Benefits of using Okera to power an Azure data lake

The underlying key benefit of using Okera's Active Data Access Platform to provide access control and governance on your Azure-based data lake is to enable you to scale the usage of your data lake and extract all the value it has to offer in an easy, agile, and responsible manner. The platform enables this by solving the technical challenges that get in the way of that intent.

SCALABLE AND POWERFUL DATA PROTECTION

Okera provides a single place to store and enforce access policies, delivering fine-grained data access control with dynamic, on-the-fly anonymization, tokenization, masking and redaction across both structured and unstructured data. The platform allows access control down to the level of a single cell and scales to support governance of petabytes of data.

All user activity is consolidated to simplify governance and reporting, and the platform includes built-in analytics and integrates with other enterprise tools to deliver powerful access and governance capabilities

COMPLETE FLEXIBILITY

Designed with heterogeneity and pluggability as core principles, Okera provides powerful and easy-to-use abstractions and APIs that make data consumers more productive with their tools of choice. This is inline with the core tenet of a data lake: running multiple kinds of workloads on multiple kinds of data assets. The platform supports multiple workloads at the same time, and workloads may be isolated as required to ensure no team's workloads impact on another's.

Okera's abstraction layer enables portability workloads between different infrastructures and platforms, eliminating vendor lock-in – something that can't be prevented if any combination of data stores (storing raw bytes), metadata system (storing schemas and access policies), and compute frameworks is provided by a single vendor

UNMATCHED VISIBILITY

Okera provides detailed visibility and insight into user and data access activity. In addition, you can quickly determine what data any particular person can access and who has access to a particular data asset. These different perspectives provide visibility into data access and activity on your Azure lake, allowing you to confidently scale the usage of your data lake, knowing that proper governance will always be in place.

All these benefits in a single platform provide the underpinnings of a governed data lake that serves the goal of creating an agile, data-driven enterprise.

To tie this back to the six tenets of data lake access control and governance, the following table shows how Okera's capabilities compare to other approaches.

Achieving Six Tenets of Data Lake Access Control and Governance

TENETS	APPROACHES			
	Azure IAM Roles	Restricted Tech.	OSS Auth. Tools	OKĒRA
Data-Centric, Not compute-centric	Azure centric	Limited to analytics or storage platform	Limited to platforms you can implement for	✓
Richness Of Access Policies	Coarse-grained No anonymization / tokenization / masking / redaction	Limited to the platform's capabilities	Limited	✓
Built For Scale And Automation	Azure centric	Limited. In most cases, no.	No	✓
Provides Unified Visibility	No	No	No	✓
Open-API First Design	Azure only	No	✓	✓
Hybrid & Multi-Cloud Ready	No	Limited	✓	✓

How traditional approaches compare with Okera in the six tenets of data lake access control and governance

Conclusion

When architected properly, data lakes can offer remarkable business agility. However, in the early days of the data lake journey security and governance are often overlooked or worked around. Eventually, as data, users, regulations, and tools increase, friction accelerates for all the constituents of the data lake — the platform teams, data engineers, and data consumers, producers, and stewards. The friction undermines the core intent of business agility and reduces the power and value of the data lake. By easily solving the hard security and governance challenges with Okera, you can unlock the power of your cloud-based data lake and also be prepared for the ongoing pace of change in the data ecosystem — in terms of both the different kinds of workloads you want to onboard and the rapidly evolving space of data regulations and security requirements. Okera enables our customers to protect and govern the data assets in their data lakes with high-performance and scale, so that they can extract maximum value out of their data in a safe and responsible way. With this, customers are assured fast time-to-value and maximum business agility, no matter how the complexity and usage of the environment increases.

To learn more about what Okera has to offer, contact us today at **info@okera.com**

ABOUT OKERA

Okera enables the management of data access and governance at scale for today's modern cloud data lakes. Built on the belief that companies can do more with their data, Okera's Active Data Access Platform (ODAP) enables scalable fine-grained data protection and visibility on data lakes for both structured and unstructured data. This allows agility and governance to co-exist and gives data consumers, owners and stewards the confidence to unlock the power of their data for innovation and growth. Enterprise organizations receive immediate value from Okera which can be implemented and deployed in less than a day. Okera is headquartered in San Francisco and is backed by Bessemer Venture Partners, Felicis Ventures, and Capital One Growth Ventures.

Learn more at www.okera.com or contact us at info@okera.com