AGILE DATA GOVERNANCE

KOVERSE®
AN SAIC COMPANY

TABLE OF CONTENTS

OVERVIEW

Today's data-driven world and associated collection and management challenges have placed data security squarely in the spotlight. Constantly emerging and evolving threats, new and complex data sources, modifications to operating models, and changing security access requirements necessitate that data governance be both effective and agile.

Effective data governance necessitates a certain amount of control and understanding of the data within an organization.

Specifically, it includes knowing:

- **WHAT** and how much data is available, the structure, types, and meaning of the values, how accurate, complete, and consistent it is, and what applications or decision-making processes it supports
- **WHERE** the data came from, including its provenance and lineage
- **WHEN** the data was last updated and by whom
- **WHO** has stewardship or ownership, who can access it, and what sharing agreements exist
- **HOW** the data is protected against unauthorized use, disclosure and destruction, how the data has been used (auditing), and how it complies with relevant laws, regulations, and industry standards
- **AND PERHAPS MOST IMPORTANTLY**, are we getting value from this data, and how can we put the data we have to better use?

All the above could in theory be achieved without taking timeliness or responsiveness to changes in organizational needs into account. In fact, requiring that the above questions be answered before data is available can create organizational drag. But taking responsiveness and agility into account as part of the governance process achieves a high degree of understanding and control over the data while also maintaining agility.

Agile data governance achieves the above goals while simultaneously giving an organization the ability to put data to use everywhere it is needed, and in time for it to make a difference.

In particular, agile data governance includes the ability to:

- Bring new datasets online quickly and make them available for use
- Discover, in a self-service way, what information exists across all datasets
- Update data with low latency—ensuring it is timely and consistent
- Support interactive applications and live decision making
- Change access based on changes in need-to-know
- Use data in ways that were not previously conceived
- Combine two datasets with no advance notice

Agile data governance ensures that not only is data being used in a responsible and effective way, but also that this governance persists throughout changes to organizational needs over time.

BARRIERS TO AGILITY

Agility in data governance can be inhibited by several factors, including the classic problems presented by data silos, data over-protection, and the incomplete application of governance to new data early in the process. Specifically, how well the data stores that hold and manage the data support governance can have a great impact on effectiveness and agility.

Data silos make everything more difficult since it causes actions to govern data to be applied multiple times, and in potentially different ways. But they also make it difficult, if not impossible, to achieve the ability to use data in new ways or combine data with little notice because those two actions are more likely to require data movement. Joining two siloed datasets requires either moving all of one dataset to the other, which may or may not be allowed or technically feasible (e.g., if one data store cannot handle both structured and unstructured data), or pulling both datasets into a third data store where the join can take place. Even if someone is allowed to do this and both datasets technically can live in one place, the process still may take a prohibitively long time to execute for non-trivial data sizes.

Data over-protection prevents users from accessing data they are authorized to use. Over-protection occurs as a result of access controls being too coarse-grained (i.e., data can only be protected at the directory, table, or file level when some data elements within those differ in sensitivity). For example, an employee may be allowed to see patient diagnoses and outcomes, but if patient identities within the data cannot be protected individually, that employee may not be allowed to use this information at all.

Insufficient tools for handling new data creates barriers to agility because the datasets may be messy, ill-understood, and require significant effort to analyze and clean before they meet the standards of the governance process. Consequently, these datasets may remain offline and unavailable for use while the issues are addressed.

THE GOLDILOCKS ZONE OF DATA GOVERNANCE

How data is organized directly influences its security, availability, usability, and flexibility. In the Goldilocks zone of data governance, the data organization is “just right” because all the data is highly secured, while remaining findable, searchable, and analyzable.

Read more about How to Achieve the Goldilocks Zone of Data Governance on our site.

PARTIAL SOLUTIONS

There are several partial solutions to the challenges of achieving agility, including the establishment of data lakes, adding search systems, and adding a security layer to the data architecture (see Figure 1).

Data lakes attempt to eliminate silos by bringing data together into one centralized system. This is effective in making it possible to combine datasets and simplify access by being able to go to one place for the data. Additionally, data lakes often can handle both structured and unstructured data in one system and can serve as a good place to put new datasets so they can be analyzed, often in bulk, in order to gain a better understanding of the data.

However, most data lake offerings lack some important features, such as the ability to search all the data within the lake to identify and retrieve important information. They often do not provide built-in support for discovering the schema of datasets, and they commonly only offer coarse-grained access controls at the directory and file level, which can cause data over-protection.

Adding separate search systems can make it possible to discover important information within datasets at the record level, which helps with understanding new datasets so they can be properly governed. But often search systems are added onto data lakes, which does not guarantee that all data is searchable, and it creates complexity by having to administer a separate system. It can also complicate access control by having two ways that data needs to be protected—at the data lake level and also in the search engine.

Adding a security layer across existing data stores can help with addressing the inconsistent application of security, but it does not address the problems of data being physically siloed, and it cannot always address the coarse granularity of security controls or retrieval mechanisms that underlying data stores support. For example, one might be authorized to access some records within a file in a data lake, but unless the data lake can retrieve only those records from that file, they are inaccessible to that person.

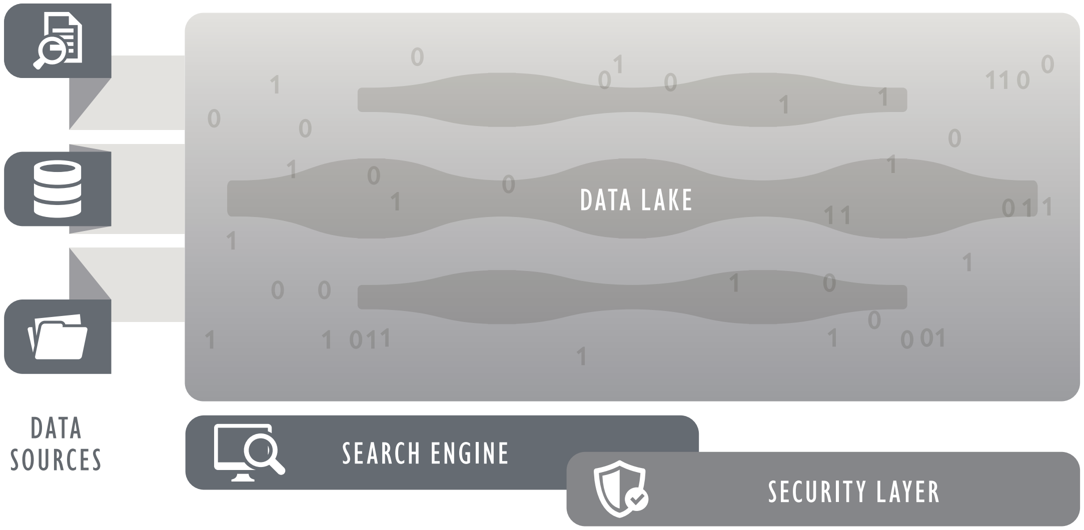


FIGURE 1. Partial, disconnected solutions are limited in their ability to deliver agility.

EFFECTIVE SOLUTIONS

A better approach to achieving agile data governance requires that data stores participate actively in supporting the data governance process.

This means that a data store should combine the features of the data lake, search systems, and security layers so they can be effectively used together to govern data in a responsive manner (see Figure 2).

An agile data layer allows both unstructured and structured data to be physically co-located, and it provides logical isolation of those datasets the way data lakes do. But the agile data layer goes further by supporting indexing of any and all data within the layer and not in a separate system. This requires that the system be scalable both in terms of storing the data and the indexing on that data.

The structure of the data is automatically discovered, rather than requiring users to do this discovery and inform the data layer themselves. This automates one major part of the data governance effort and eliminates incorrect, inconsistent, and out-of-date descriptions of the data.

When fine-grained security controls are built into the data layer, new and messy datasets can be quarantined and available for cleaning, data over-protection can be eliminated, and access control can be managed in a uniform and responsive way. Getting access to new data because of a change in need-to-know no longer requires setting up an account on a new system and potentially moving data to combine and analyze it; it simply becomes a change in authorizations.

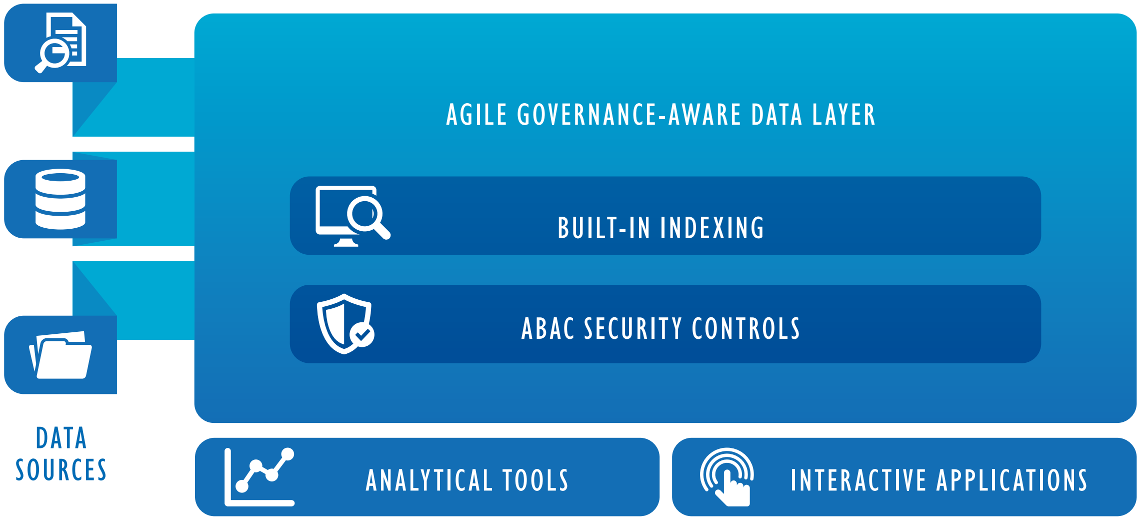


FIGURE 2. Agile solutions combine the features of the data lake, search systems, and security layers.

EFFECTIVE SOLUTIONS

Finally, the data layer should support both interactive, selective access of data via search as a result of indexing the data, and also the bulk processing of data to support analytics and cleanup of messy datasets. Bulk processing of data should be architected in a way that allows computation to be pushed to the data, rather than requiring the movement of potentially large datasets.

This way, the progression of data from unrefined bronze data up to higher-standard gold data can be supported all within one system, and users can benefit from the ability to reuse data that has been partially processed. They can also choose to clean up or normalize data in more ways than one since the original data is always available (see Figure 3).

Analytical results can also be written back to the data layer so they can be indexed and access controlled, making them available to a wide audience of consumers who can use that information in decision-making processes.

In a system like this, users have one place to go to see all the data they are authorized to access. They can do a broad search across multiple datasets and discover in seconds what datasets may be relevant to a particular task. And they can choose to retrieve a small subset of the data or process one or more entire datasets using analytical tools. Additionally, changes in a user's authorizations are immediately effective whether they have lost or gained new authorities.

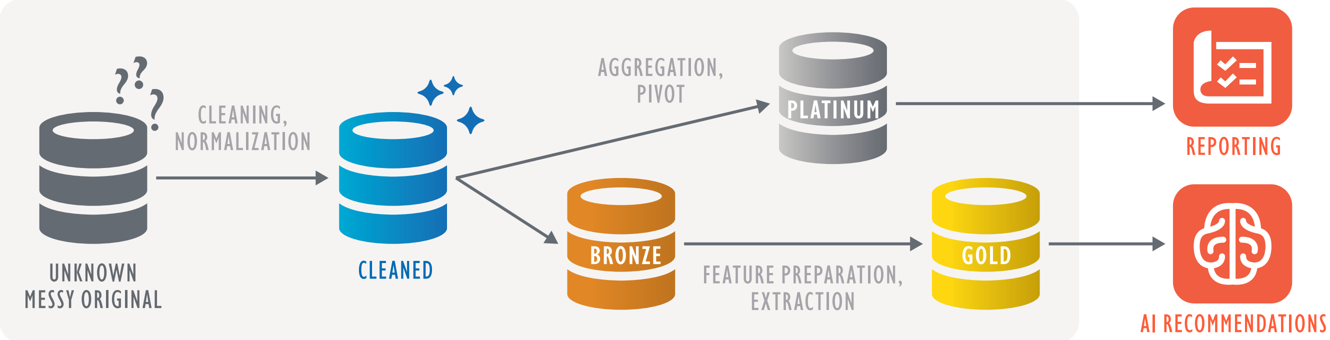


FIGURE 3. Data cleanup, normalization, and progression are supported within one system.

HOW ZERO TRUST SUPPORTS AGILE DATA GOVERNANCE

We have talked about how fine-grained security controls, implemented uniformly within the data layer, can be a game changer for the agility of data governance. Now, we would like to describe what that approach—a Zero Trust Architecture for data—consists of.

A Zero Trust Architecture means that users are never implicitly trusted, but rather all actions they take must be explicitly authorized. This requires the reduction or elimination of implicit trust zones. One example of an implicit trust zone is a network where users who can access the network are implicitly trusted to access all data and applications on that network, rather than being authorized to access each of those resources individually. Another example is a data table. A user with access to the table may be implicitly trusted to read all the rows and columns within it, rather than being authorized for those specifically.

A Zero Trust Architecture ideally reduces implicit trust zones to zero. Many Zero Trust efforts focus on the network and applications, but a great degree of implicit trust can be eliminated by applying security controls to the data layer.

Another important concept in a Zero Trust Architecture is attribute-based access control, or ABAC. This is a more scalable form of access control than the more commonly used role-based access control (RBAC) because it allows users to be authorized in terms of their unique set of attributes, rather than explicitly having to create groups for every set of users (see Figure 4).

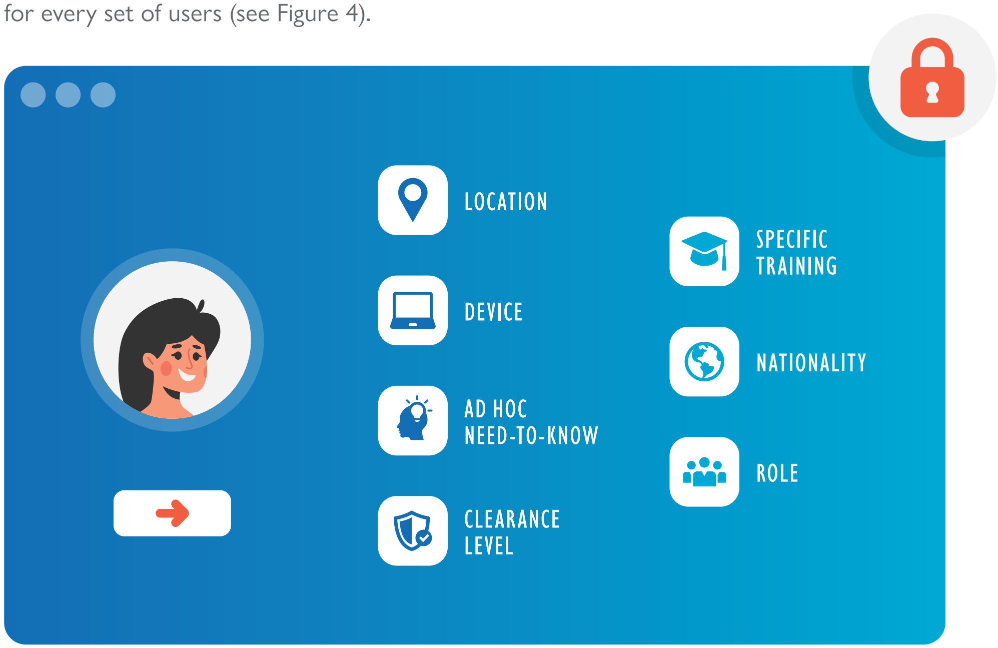


FIGURE 4. ABAC allows individuals to be authorized based on their unique attributes and permissions.

HOW ZERO TRUST SUPPORTS AGILE DATA GOVERNANCE

Adopting ABAC consists of the following steps:

1. Assign attributes to people based on various security dimensions, such as clearance, data handling training, nationality, organization, etc.
2. Label data elements with the combination of attributes required for access. Critically, this should be done at whatever granularity of data is required, whether at the row level, row and column, document, paragraphs within documents, etc.
3. Check attributes on every data access request and authorize each specific data element requested.
4. Audit all access requests.

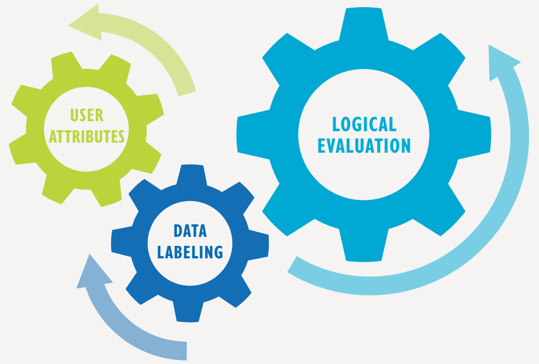
Data labeling is often done by a person who is trained to label data appropriately but can also be done via rule sets or other automated methods, based on the source, metadata, and content of data.

Zero Trust as applied at the data layer (see Figure 5) makes agile data governance possible in two ways. First, it allows any type of data to be stored in the data layer, regardless of sensitivity and regardless of how well understood or governed it is in other respects. New datasets are quarantined by default and always access controlled once added to the system, so there is no time during which data is not controlled, and ill-understood data can be managed using all the resources and capabilities of the system.

The second is when needs change within an organization and it becomes important for some set of users to access new data, or for two or more datasets to be combined. That can be accomplished merely with changes to access control settings as opposed to involving data movement, re-housing data, or worrying about the capabilities of the storage layer.

If data access authorization is handled at the data layer, then applications do not all have to be modified to be ABAC aware. Instead, they can inherit the security applied at the data layer. This is important for migrating legacy applications into a Zero Trust Architecture.

Because every access request is authorized, any change in attributes is reflected immediately, which reduces implicit trust over time. Just because a user had access a minute or an hour ago, it does not imply that they have access right now.



USERS ARE ASSIGNED ATTRIBUTES according to their clearance level, training, need-to-know, location, device, etc.
E.g.: User A →HIGH, MED, US

LABEL INDIVIDUAL DATA ELEMENTS on ingest based on sensitivity — paragraphs, rows, sub-rows, etc.
E.g.:
Row 001 → HIGH & (US | UK)
Row 002 → MED & UK

DATA LAYER EVALUATES user attributes against every data element for every data access request.
E.g.:
Row 001 → HIGH & (US | UK) = Y
Row 002 → MED & UK = N

FIGURE 5. A Zero Trust Architecture for data with ABAC enables agile data governance.

AGILE DATA GOVERNANCE IN PRACTICE

So, what does a day in the life look like for data stewards/owners, data workers, and decision makers who are benefiting from agile data governance?

DATA STEWARDS have much more confidence and control in terms of knowing the data they are responsible for is protected (see Figure 6). They are empowered to add new datasets to the data layer on a self-service basis, knowing that any data they add is immediately protected and visible only to them.

They can immediately see what structure the data has, how much there is, and what types of data values are present. They have the ability to explore the data via search or in bulk using analytical tools to discover what information exists, and what potential problems are in the data.

Data stewards can then coordinate efforts with data workers to clean and normalize the data according to how they understand the enterprise wants information to be represented. At this stage, the data stewards themselves may apply data cleaning tools, or it may be done in concert with data workers.

At any time, data stewards can access audit logs relevant to the datasets they manage and determine quickly who is using what data, what the data access controls are, and whether the datasets are up to date.

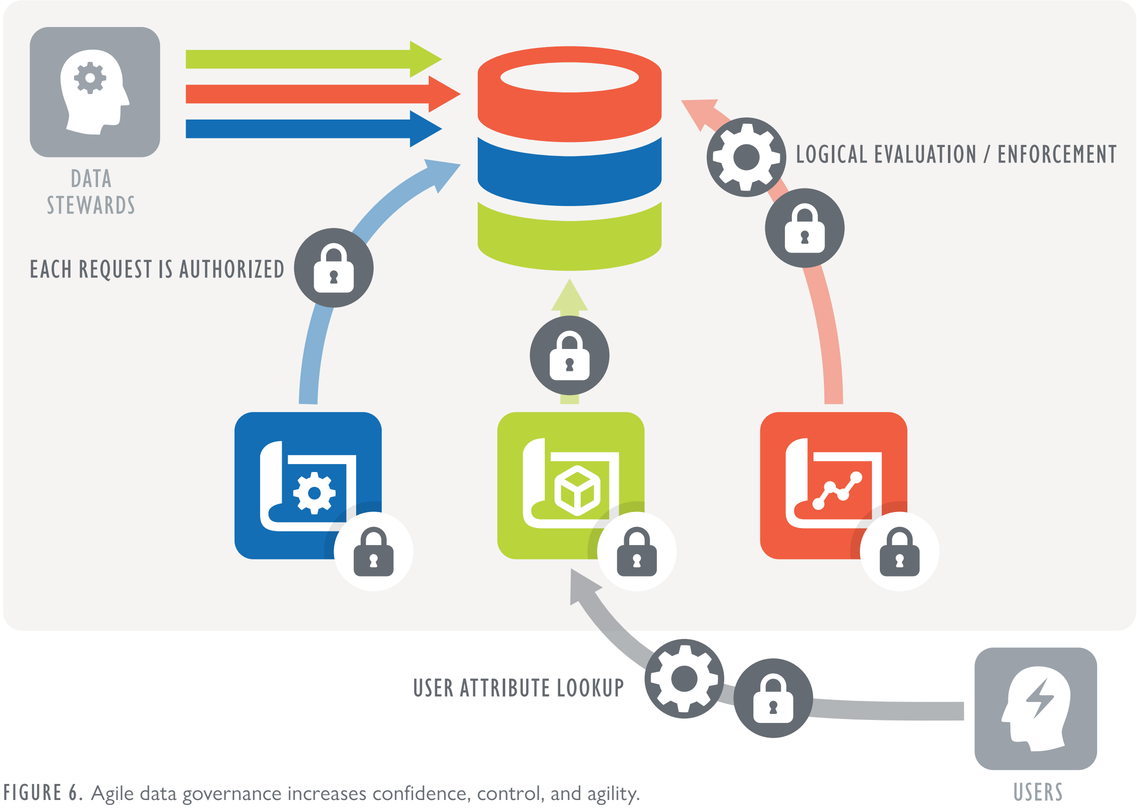


FIGURE 6. Agile data governance increases confidence, control, and agility.

AGILE DATA GOVERNANCE IN PRACTICE

DATA WORKERS such as Data Engineers, Machine Learning (ML) Engineers, Data Scientists, and Data Analysts have a straightforward experience in discovering what datasets are available and what structure each has, and in accessing those datasets via search or loading them into their tool of choice. Their work is streamlined because the data layer handles the converting of datasets into the format that their data processing tools require, such as Pandas DataFrames, Spark DataFrames, tables for SQL processing, etc.

Data processing times are greatly reduced by the ability of data processing tools to push computation down to the data layer, which has already distributed data across several machines.

Data workers and data stewards do not have to worry about whether any sensitive elements that workers should not be allowed to see are present, because the data layer automatically filters out any data elements a particular worker is unauthorized to see.

When analysis is complete and if new derivative data has been created, data workers have the option to write that data back to the data layer so it can be access controlled and indexed. They can then make these analytical results available to a broader audience of decision makers because the data layer makes them available for interactive applications through a RESTful web API, or via search directly. This makes getting useful information into existing decision-making processes possible.

DECISION MAKERS can access any and all data that is relevant for their job by searching the data layer directly. They may also work with data workers to generate new datasets that yield specific insights into business decisions. Decision makers can use the tools they are familiar with because data workers have a way to deliver useful data to their existing applications and tools via the data layer.

Ultimately, agile data governance with a Zero Trust Architecture turns your data into an asset. By making your data rapidly and securely available to only those with need-to-know, it enables faster and more effective analysis, decision making, and actions. It is an imperative for surviving and thriving amidst modern business challenges.

To learn more about how to achieve agile data governance with Koverse, please visit koverse.com or contact us at info@koverse.com.

ABOUT THE AUTHOR

Aaron Cordova

CO-FOUNDER AND CHIEF TECHNICAL OFFICER

Aaron has built multiple, large-scale, big data systems that are used by the intelligence, defense, finance, and healthcare industries. He has a passion for delivering products that are transformative, based on close customer collaboration and iterative development.

Aaron spent five years as a researcher for the National Security Agency (NSA) where he developed and deployed into operations dozens of advanced analytical techniques. He is also the founder of Apache Accumulo, a scalable and secure data store on top of Apache Hadoop and the author of the O'Reilly book, Accumulo: Application Development, Table Design, and Best Practices. Aaron holds a Bachelor of Science degree in Computer Science from the University of Maryland College Park.

in https://www.linkedin.com/in/aaroncordova/

ABOUT KOVERSE

Koverse, Inc., An SAIC Company, empowers customers to use data to gain understanding and drive mission-impacting decisions and actions. Our technology is trusted by government agencies and highly regulated commercial industries such as healthcare, financial services, and more.

Founded by former NSA data architects, Koverse offers a security-first platform with unprecedented scale, performance, and flexibility.

Koverse provides Zero Trust for data by enforcing attribute-based access controls (ABAC), allowing customers to safely work with their complex and sensitive data to power the most demanding analytics, data science, and AI use cases.

Koverse is headquartered in Seattle with hubs in Denver, San Diego,
and the Washington, D.C. metro area.

For more information, please visit www.koverse.com

koverse.com

KOVERSE®
AN SAIC COMPANY