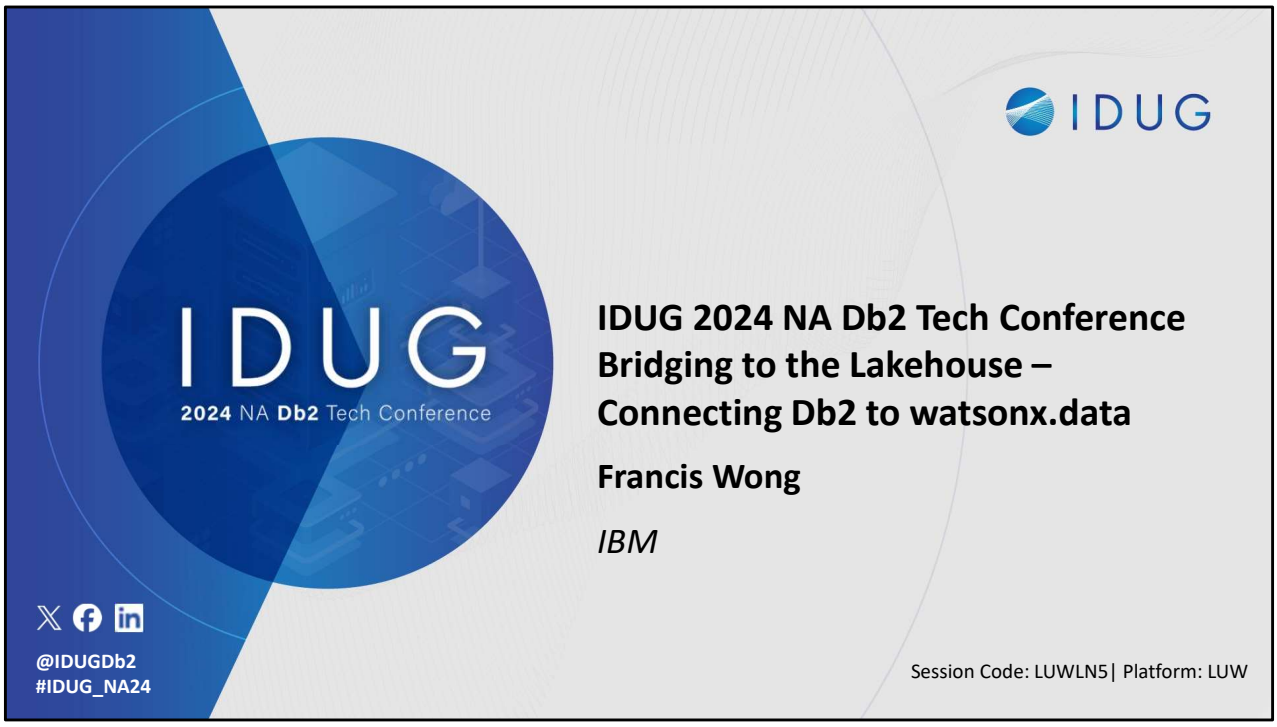


IDUG 2024 NA Db2 Tech Conference Bridging to the Lakehouse – Connecting Db2 to watsonx.data

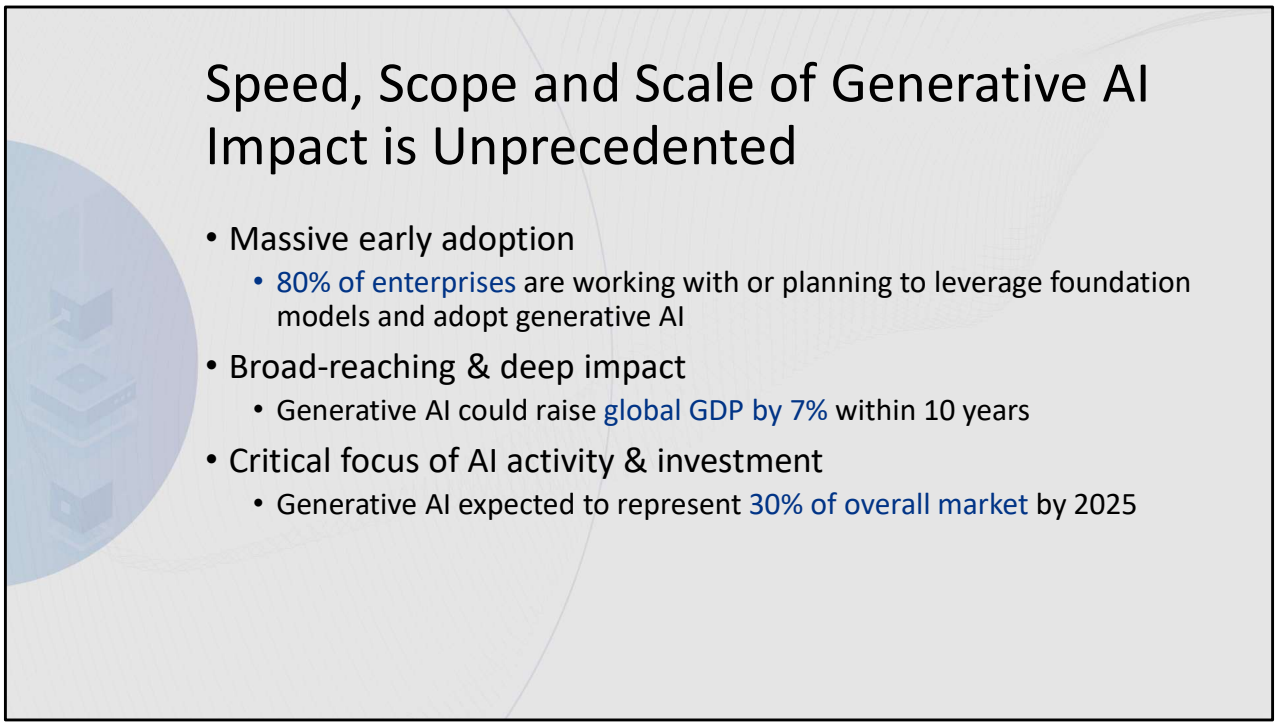
Francis Wong

IBM

Session Code: LUWLN5| Platform: LUW

Agenda

1. Introduction to watsonx.data
2. watsonx.data Use Cases
3. Connecting watsonx.data to Db2



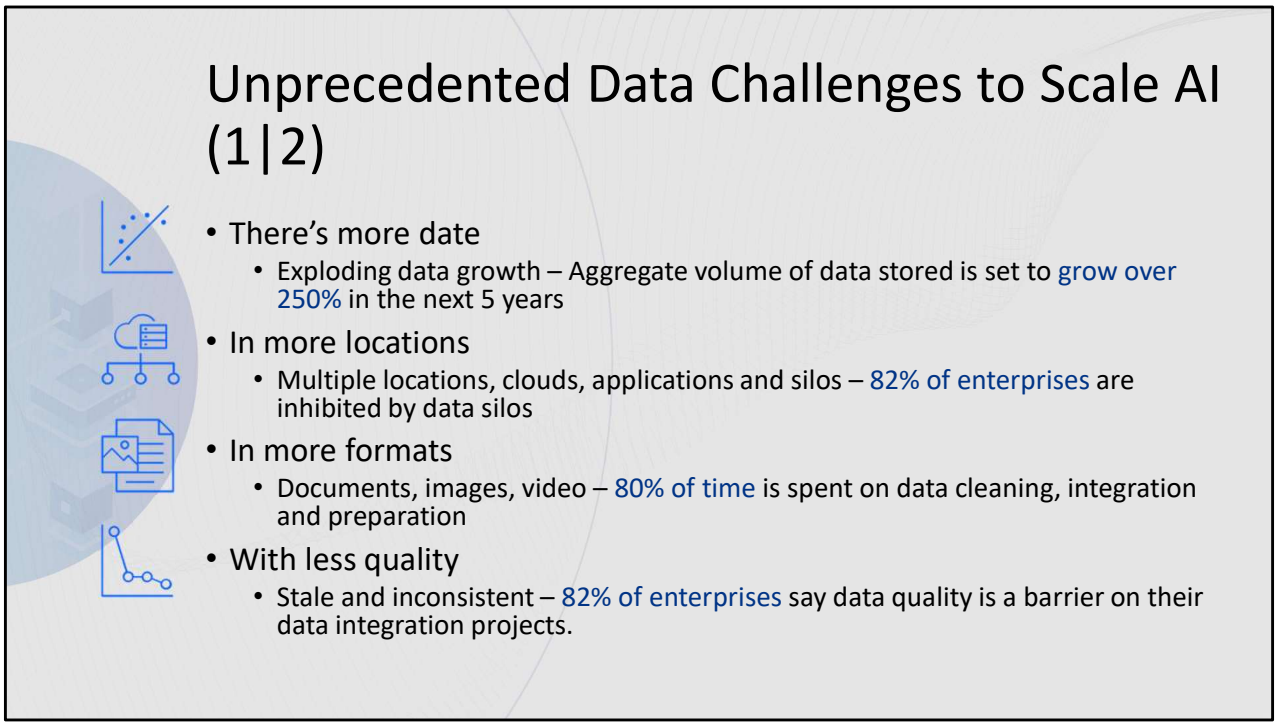
The public release of ChatGPT by OpenAI in November 2022 led to a massive surge of interest (and hype) in artificial intelligence (AI). ChatGPT has also sparked significant interest around large language models (LLMs) and generative AI.

Sources:

- Threads Shoots Past One Million User Mark at Lightning Speed, Statista, July 2023 - https://www.statista.com/chart/29174/time-to-one-million-users/
- ChatGPT sets record for fastest-growing user base - analyst note, Reuters, February 2023 - https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/
- Generative AI could raise global GDP by 7%, Goldman Sachs, April 2023 - https://www.goldmansachs.com/insights/pages/generative-ai-could-raise-global-gdp-by-7-percent.html

- Generative AI, Boston Consulting Group, 2023 – https://www.bcg.com/capabilities/artificial-intelligence/generative-ai

- Gartner Experts Answer the Top Generative AI Questions for Your Enterprise, Gartner, 2023 - https://www.gartner.com/en/topics/generative-ai#:~:text=We%20predict%20that%20by%202025,marketing%20copy%20and%20personalized%20advertising.



Scaling artificial intelligence (AI) requires trusted data, however most organizations still struggle with fundamental data challenges detailed on this self-explanatory slide. Data is stored in multiple locations, applications, and clouds, leaving 82% of organizations inhibited by data silos. To add even more complexity to these problems, the uses of data have become more varied – with data in varying and complex forms, but also with poor quality.

And things are about to get worse. According to International Data Corporation (IDC), stored data is expected to grow 250% by 2025. These growing volumes of data across disparate silos not only pose challenges to data governance and compliance, but also drive up the costs associated with storing and managing data for analytics and AI.

Sources:

- Data volumes growing 250% over next 5 years: IDC, May 2022- https://www.idc.com/getdoc.jsp?containerId=US49018922

- 80% proportion of time on data cleaning, integration, and preparation – Data Integrity Trends: Chief Data Officer Perspectives in 2021, Corinium - https://www.precisely.com/app/uploads/2021/06/Data-Integrity-Trends-2021-Corinium-Intelligence.pdf

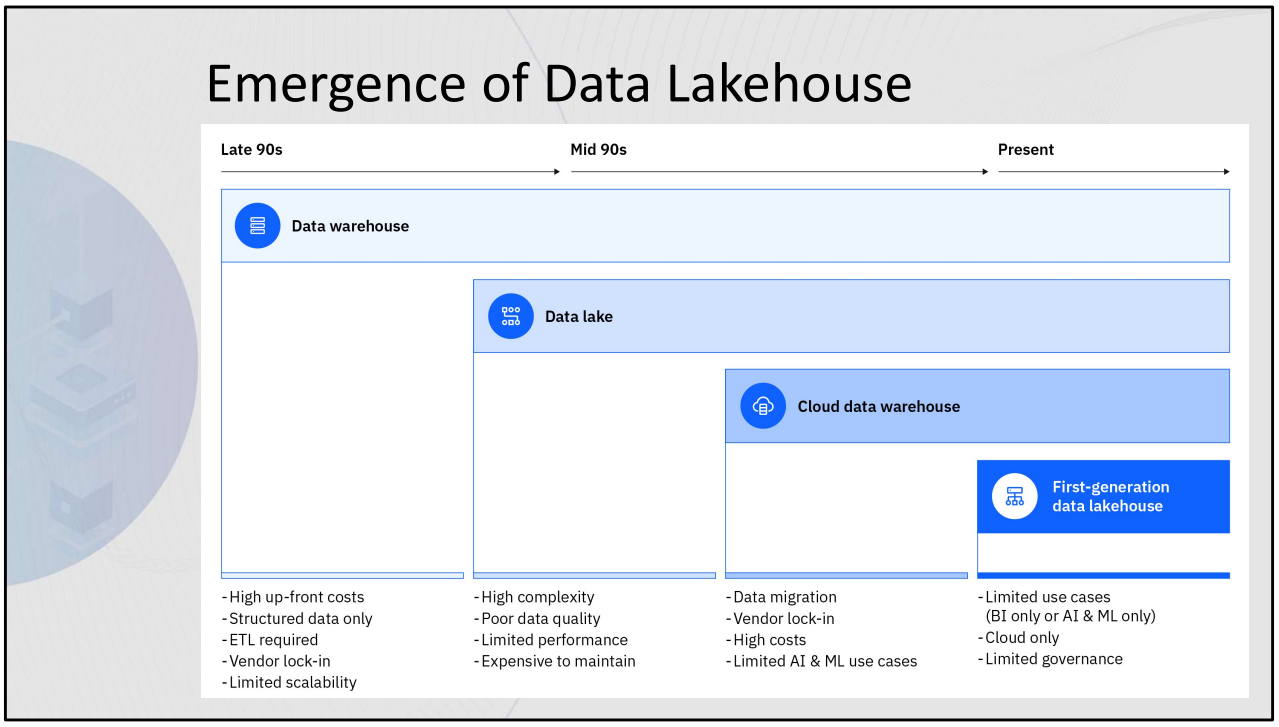
- 82% data quality a barrier to their data integration projects – Data Integrity Trends: Chief Data Officer Perspectives in 2021, Corinium - https://www.precisely.com/app/uploads/2021/06/Data-Integrity-Trends-2021-Corinium-Intelligence.pdf

- 82% of enterprises are inhibited by data silos – Are Data Silos Killing Your Business? Michael Goldberg, Dun & Bradstreet, May 2018 - https://www.dnb.com/perspectives/marketing-sales/data-management-strategies-avoiding-data-management-silos.html

Unprecedented Data Challenges to Scale AI (2|2)

- Traditional approaches to addressing these challenges have created more overall complexity and cost
- Today, most large enterprises manage their data and workloads using a mix of data repositories and data stores in hybrid environments
- The overall cost across all these repositories remains high
- Difficult for leaders to effectively leverage and govern the data across multiple environments and use enterprise data for analytics and AI

Traditional approaches to solving the challenges detailed on the previous slide create more complexity and cost over time. Traditional data warehouses were designed to offer high performance for processing terabytes of structured data for reporting and for business intelligence (BI) workloads. However, these data warehouses have become expensive to scale (for example, they require expensive block storage, performance tuning, and more) to support new workloads.



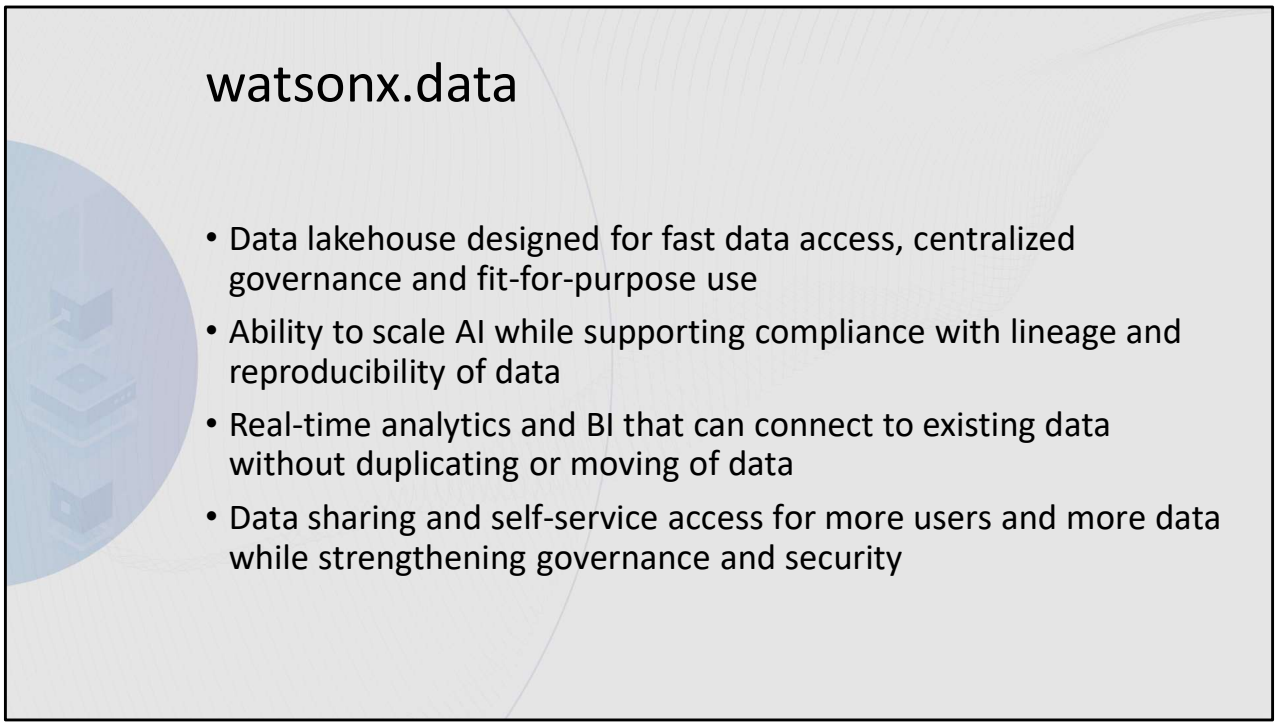
Data lakes were designed with semi-structured and unstructured data in mind, utilizing lower cost storage. However, they are less performant than warehouses and are more complex to maintain and govern because they are really for use by those with programming skills. But rather than having a well-organized data lake, most organizations found themselves with data swamps – a term that refers to badly designed, inadequately documented, or poorly maintained data lakes, usually the result of a lack of processes, standards, and proper governance. These deficiencies compromise the ability to analyze and exploit the data efficiently.

Cloud data warehouses (like Snowflake) promised a way to drive analytics costs down, with pay-as-you-go pricing, but they've led clients to spend more. Without proper financial guardrails to manage and predict costs, full time on cloud can be more expensive than full time on-premises.

Given the prohibitive costs of high-performance on-premises and cloud data warehouses, and performance, governance, and maintenance challenges of

legacy data lakes, neither of these repositories satisfy the need for analytical flexibility and price-performance. The new approach is the data lakehouse architecture.

However, first-generation lakehouse vendors (which include for example, both Databricks and Dremio) have key constraints that limit their ability to address cost and complexity challenges. These types of vendors only support a single query engine, which limits the types of workloads that can be effectively run on it. Also, most are available on cloud only, with no support for multicloud or hybrid cloud deployments, or on-premises for that matter. Another limitation is governance. Most of these first-generation lakehouse vendors have a governance strategy, but they are limited in scope and capabilities. For example, Dremio lacks the ability to track data lineage (the origin, transformation, and movement of data). Finally, their value proposition forces clients to move all data and that introduces risk.



To meet the needs of enterprises, a robust data architecture is essential. Specifically, clients need a data architecture that delivers fast data access, centralized governance, and customized usability.

A robust data architecture is comprised of the following:

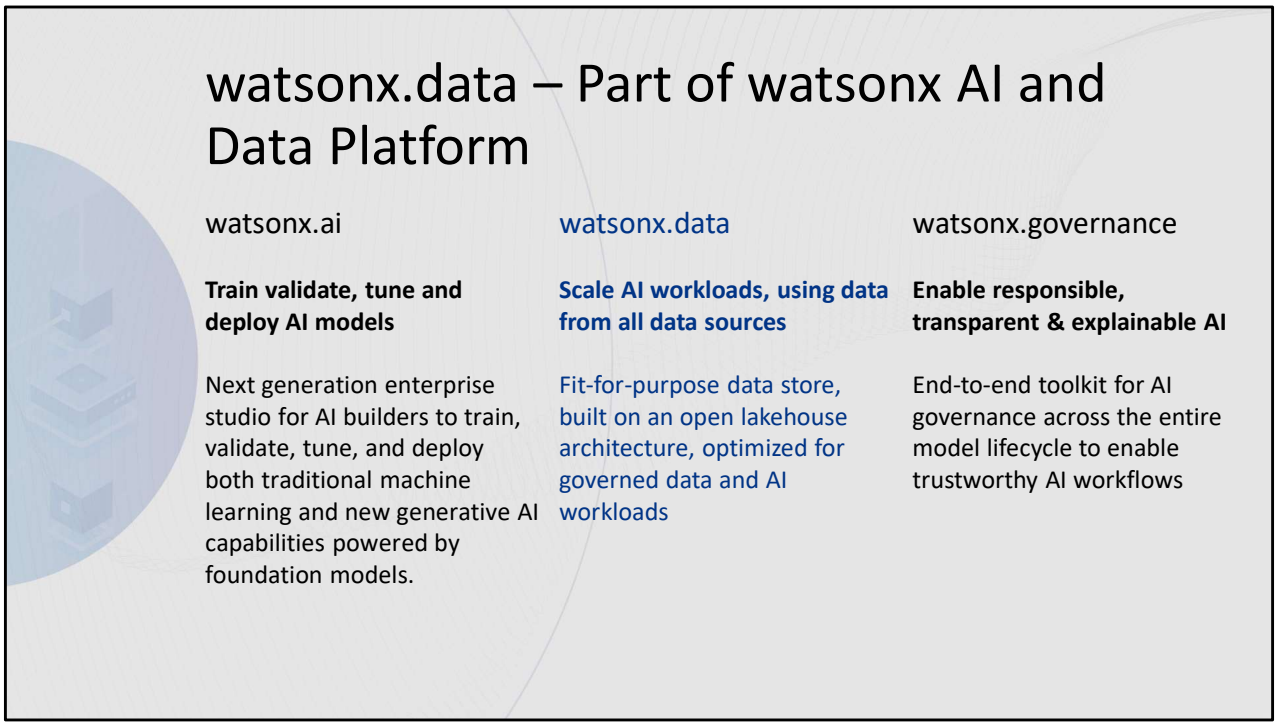
- **Scaling artificial intelligence (AI) with compliance**: The architecture must allow for seamless AI expansion while upholding data lineage, reproducibility, explainability, and other compliance standards.

- **Real-time analytics and business intelligence (BI) integration**: Enterprises require analytics and BI capabilities that can swiftly connect to existing data sources, eliminating costly data duplication or movement.

- **Data sharing and self-service access**: The architecture should facilitate secure data sharing and self-service access, accommodating more users and larger datasets while maintaining governance and security measures.

In summary, enterprises demand a data architecture that provides quick access

to data, centralized governance, and fits their specific requirements. This includes scaling AI with compliance, real-time analytics and BI integration, and enables data sharing and self-service access, with robust governance and security.



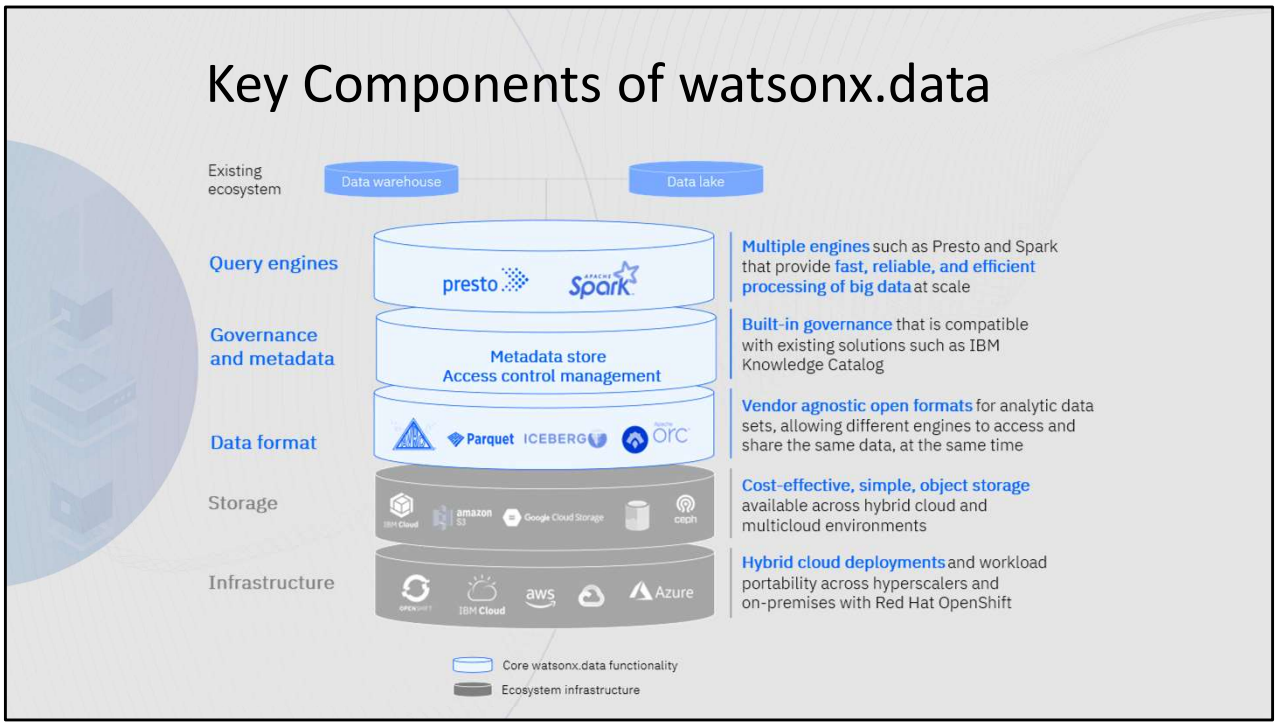
Watsonx is a new artificial intelligence (AI) and data platform from IBM that is designed with the three critical elements of an AI strategy in mind. It empowers enterprises to train, tune, and deploy AI across the business, leveraging critical, trusted data wherever it resides. This platform has three components:

Watsonx.ai is a studio that clients can use to train, validate, tune, and deploy both machine learning (ML) AI models as well as foundation models for generative AI. These models combine best-of-breed architectures with a rigorous focus on data acquisition, provenance, and quality, to serve enterprise needs.

Watsonx.data makes it possible for enterprises to scale AI workloads using all their data with a fit-for-purpose data lakehouse service optimized for governed data and AI workloads, supported by querying, governance, and open data formats to access and share data. This is based on open-source technologies, including Presto and Iceberg.

Watsonx.governance helps companies put AI into production by providing an

end-to-end solution that encompasses both data and AI governance to enable responsible, transparent, and explainable AI workflows. AI governance helps business analysts understand the trustworthiness of their AI solutions.



This slide provides a high-level summary of the key components that make up IBM watsonx.data.

Starting from the bottom, there is the base infrastructure on which watsonx.data runs. Watsonx.data can be deployed across any cloud that a client desires, or on-premises, because it's built on Red Hat OpenShift. However, watsonx.data is especially well-tuned for IBM Cloud and Amazon Web Services (AWS).

Moving up the stack is the storage of the data in watsonx.data. Watsonx.data supports the use of S3-compatible object storage to store data at a fraction of the cost compared to the traditional block storage found in high-performance data warehouses. These files can contain data of different types, stored in different formats.

How these files and the data within them are organized is very important, and that's where the watsonx.data open data file and open table formats (the next layer up the stack) come in.

In a traditional data warehouse, each warehouse has its own proprietary (typically) data file and table format for organizing and managing its data on disk. Db2 has its own native data and table format, NZ has its own native data and table format, Snowflake has its own native data and table format, and so on. Other query engines and data consumers don't understand these formats, which makes it difficult to share data between them.

However, with open table and data formats, apps can access data from any engine that understands them. Watsonx.data embraces common open-source data file formats, such as Apache Parquet, Apache ORC (Optimized Row Columnar), and Apache Avro, as well as open table formats like Apache Iceberg (which supports ACID transactions). With watsonx.data support for Iceberg, data can be accessed by popular query engines, such as Presto and Spark, as well as any other engine that supports Iceberg, including Snowflake. Additionally, and very importantly, starting in 2023, Db2 Warehouse and Db2 Warehouse on Cloud has the ability to read from and write to the Iceberg format. **This means that Db2 and watsonx.data can all share the exact same data (and metadata for that matter) in object storage.**

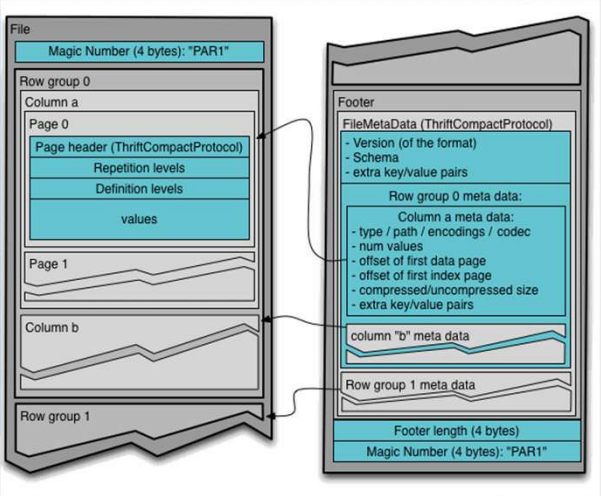
Moving further up the stack, the shared metadata store is a critical component of the watsonx.data lakehouse architecture, because it enables all engines to have a consistent view of the data being managed in the lakehouse storage. It also offers built-in access control for the environment and includes an enhanced integration with IBM Knowledge Catalog (IKC). This ensures that only the people that need to access data can do so. With IKC, clients can define policies, capture lineage, and setup advanced governance. And any policies clients define in IKC can automatically be enforced by the lakehouse metadata store.

At the top of the stack are the query engines, including Presto and Spark. These engines allow analytics and AI workloads to run against the data in watsonx.data. Presto is an open-source, distributed SQL query engine, designed for analytic queries against data of any size, that's fast, reliable, and efficient at scale. Apache Spark is an open-source, unified analytics engine for large-scale data processing. With Spark practitioners can run data engineering, data science, and machine learning workloads with data parallelism and fault tolerance. As mentioned earlier, these query engines (and any other engines that support the Iceberg table format) can access the same shared data in object storage.

Watsonx.data doesn’t require the “lifting and shifting” of a client’s warehouse or existing data lake to the watsonx.data platform. Watsonx.data integrates with

these existing data stores, allowing data to be joined amongst them, as well as supporting the movement of workloads to the engines that are best suited for them.

Benefits of Open Data Format – Parquet



- Open
  - Open Source. Reference implementation / format specifications publicly available
  - Support available for multiple tools and multiple programming languages. No vendor lock in.
- Optimized
  - Column organized for analytics use case fast reads & compression optimization
  - Self describing with file footer & pages carrying statistics enabling data skipping / predicate pushdown

Parquet, ORC, and Avro are open data file formats. What is a data file format? Consider the common comma separated values (CSV) data file format. CSV files contain data records, where each row in the file corresponds to a single data record (one row, one record). Each of these data records consists of one or more fields, delimited (separated) by commas. The fact that the CSV data format is simple and text-based makes it a common storage and interchange format. However, other file formats have been devised over time (many within the open-source community), to better support the operational and performance characteristics of data and analytics workloads.

Parquet (initially developed by Cloudera and Twitter) is an open-source, columnar storage file format designed for efficient data storage and fast retrieval. Parquet provides efficient data compression and encoding schemes with enhanced performance to handle complex data in bulk. Similar to Parquet, ORC (initially developed by Hortonworks and Facebook) is an open-source columnar storage file format. Avro, on the other hand, is a row-oriented data format and data serialization framework. All of these formats have their beginnings in Hadoop and are commonly used today.

Advantages of Parquet:
- Wide tooling support. Guarantee they will be readable for the forseeable future.
- Guarantee of interoperability across multiple engine & technology. No vendor lock in.
- Columnar format suited for analytics workload.
- Additional structure & embedded statistics allowing for predicate pushdown / data skipping

New Class of Open Data Formats – Iceberg

- Full open-source, Open Data Table format, quickly becoming an industry standard
- Relies on Open Data File formats for storage, but provides an additional layer of metadata for enhanced capabilities

Iceberg is a high-performance, open table format. Think of it as sitting one level up from the data files. Iceberg is responsible for the organization of table data and metadata, regardless of the data file format used to store the table's data on disk (which could be Parquet, ORC, or Avro). There are many benefits to using Iceberg (which is optional in watsonx.data), including support for ACID transactions (more on this below), as well as time-travel capabilities (allows users to examine changes over time and be able to quickly rollback tables to a previous state). While use of Iceberg within watsonx.data is not mandatory, without it, clients lose the benefits it provides. Most major vendors have started embracing Iceberg, although some lakehouse competitors are focusing on their own table formats. For instance, Databricks uses their own Delta Lake storage format. While they've released it to the open-source community, it's effectively proprietary due to other vendors not supporting it and they are the only committers on the project – which means data lock-in for clients.

With watsonx.data support for Iceberg, data can be accessed by popular query engines, such as Presto and Spark, as well as any other engine that supports Iceberg, including Db2, Netezza and Snowflake.

Regarding ACID transaction support in Iceberg … ACID refers to four properties of transactions that guarantee data validity even in the face of errors and system failures: atomicity, consistency, isolation, and durability. ACID transaction support is a common characteristic of transaction-based databases that need to be accurate; almost all proprietary and many open-source databases support this capability (certainly most SQL databases do, but many NoSQL databases do not). Businesses depend on this behavior to ensure that multiple users can concurrently query and update data without fear of corrupting it or seeing it in an intermediate state (think of a banking transaction or booking an airline ticket).

Specifically, ACID provides the following:

- **Atomicity:** All changes to data are performed as if they are a single operation. That is, all the changes are performed, or none of them are. For example, in a bank application that transfers funds from one account to another, the atomicity property ensures that if a debit is made successfully from one account, the corresponding credit is made to the other account, or all activity is rolled back – think all or nothing.

- **Consistency:** Data is in a consistent state when a transaction starts and when it ends. For example, in a bank application that transfers funds from one account to another, the consistency property ensures that the total value of funds in both the accounts is the same at the start and end of each transaction.

- **Isolation:** The intermediate state of a transaction is invisible to other transactions. As a result, transactions that run concurrently appear to be serialized. For example, in a bank application that transfers funds from one account to another, the isolation property ensures that another transaction sees the transferred funds in one account or the other, but not in both, nor in neither.

- **Durability:** After a transaction successfully completes, changes to data persist and are not undone, even in the event of a system failure. For example, in a bank application that transfers funds from one account to another, the durability property ensures that the changes made to each account will not be reversed after the transaction has committed – even if the database goes down at this point.

Agenda

1. Introduction to watsonx.data
2. watsonx.data Use Cases
3. Connecting watsonx.data to Db2

Use Cases

Share data through an open format

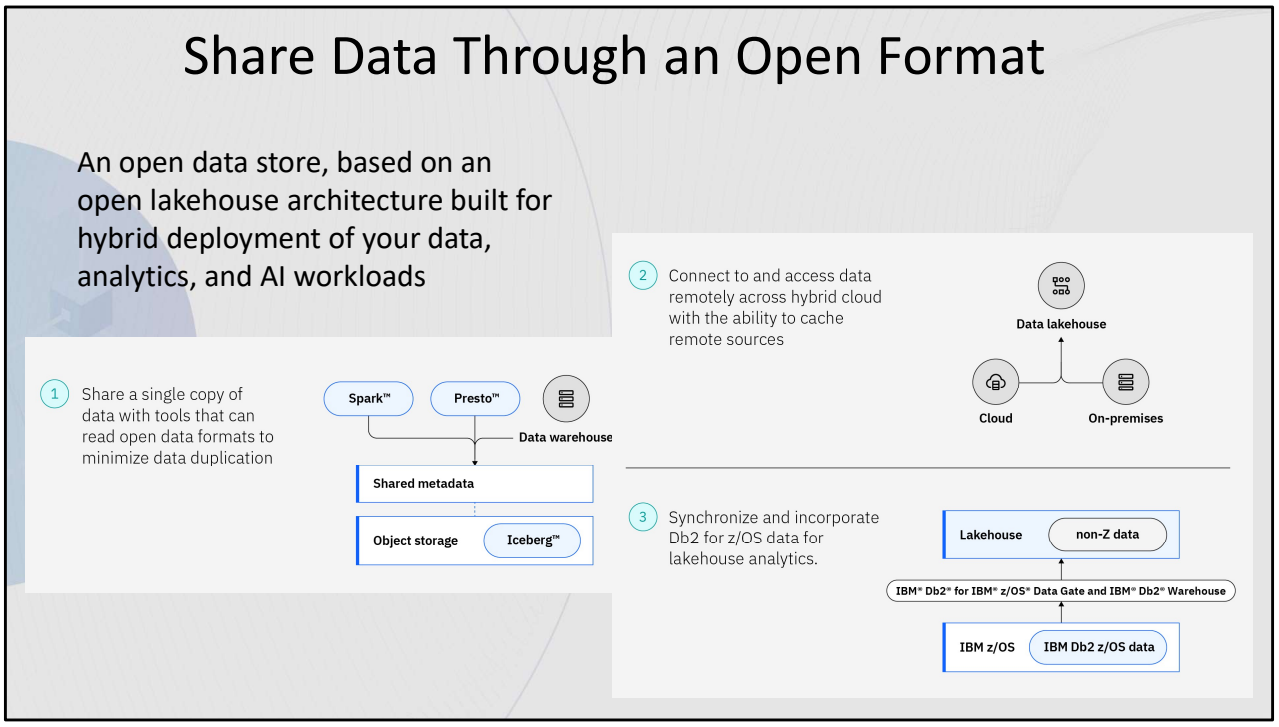
Eliminate data silos by sharing Db2 tables with data lakes and lakehouse engines.

Optimize Workloads

Use the most appropriate tool for the task at hand without having to move or copy the data

Warehouse Augmentation

Gain new insights from your warehouse data by combining Db2 Warehouse and data lakes platform data through open formats engine.

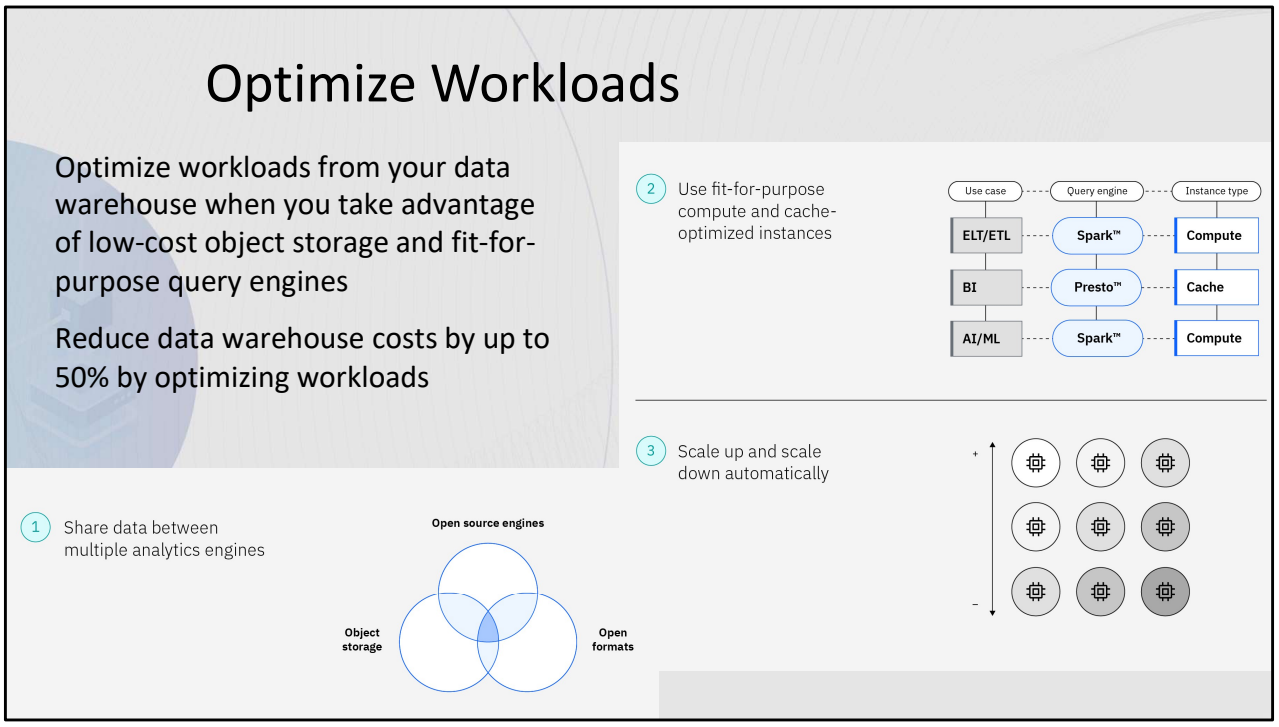


No matter where a client's data lives across the hybrid cloud, watsonx.data can connect and access that data remotely. By reducing data pipelines and simplifying data transformation, clients can seamlessly combine data from existing sources with new data in the lakehouse to unlock new insights, faster.

Watsonx.data leverages the Apache Iceberg open table format, shared metadata, and cost-effective object storage to share a single copy of data across multiple query engines. This facilitates collaboration, minimizes duplication, and can help to reduce security and governance risks by reducing the number of copies necessary to support different users and tools.

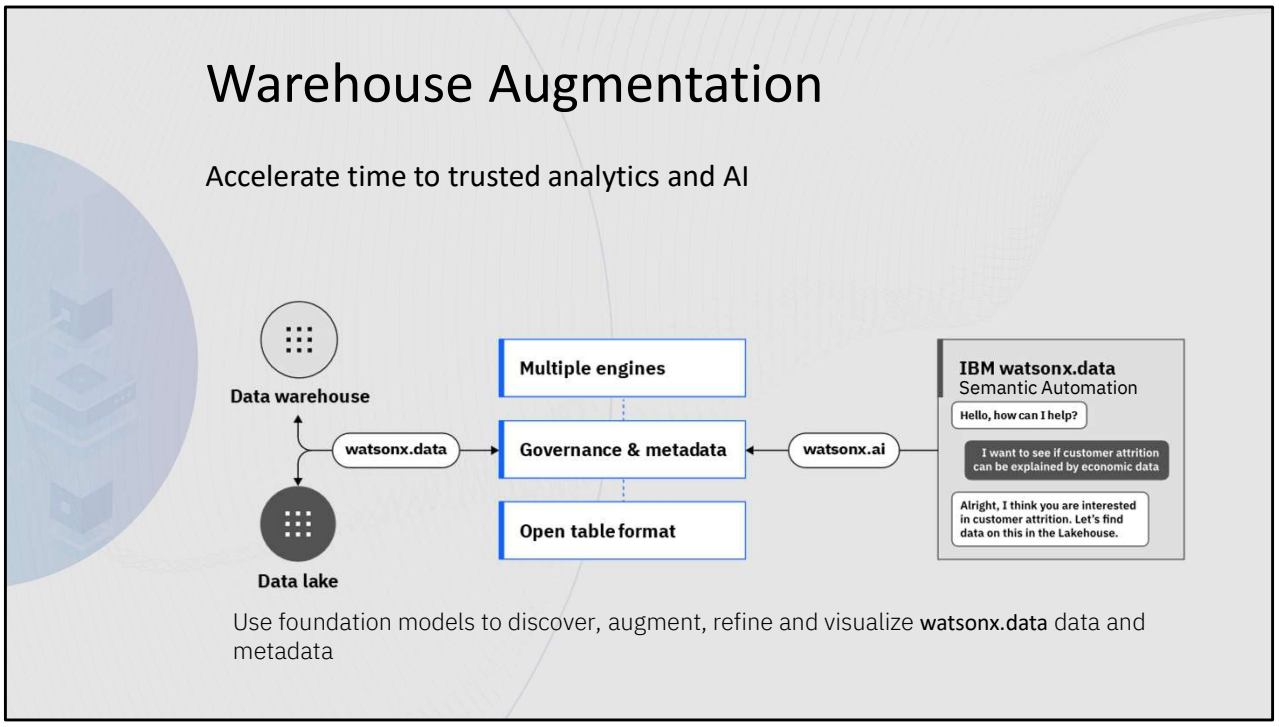
A key component of watsonx.data is the use of multiple query engines such as Presto and Spark. Presto is an open-source, distributed SQL query engine, designed for analytic queries against data of any size, that's fast, reliable, and efficient at scale. Apache Spark is an open-source, unified analytics engine for large-scale data processing and machine learning pipelines. With Spark, clients can run data engineering, data wrangling, data science, and machine learning workloads with data parallelism and fault tolerance.

In addition to these core engines of Presto and Spark, external engines that support the Iceberg open table format can also work directly with data in the watsonx.data object storage. For example, since Db2 can read from and write to the Iceberg format, it can participate in the lakehouse ecosystem as well, accessing the lakehouse data in object storage directly (just as Presto and Spark can). Extending upon this is the “digital twinning” of data from transactional systems. For example, IBM Data Gate can replicate transactional data from Db2 for z/OS on IBM Z to watsonx.data – where it can be joined with existing data in the lakehouse’s object storage as needed to support analytics and AI workloads.



By sharing data on low-cost object storage between multiple analytics engines, watsonx.data can use fit-for-purpose engines for different workloads. For example, many artificial intelligence (AI) and machine learning (ML) focused workloads are best suited to run on a Spark compute-optimized engine, whereas it's likely the case that a business intelligence (BI) focused use case is better suited for the cache-optimized SQL Presto engine. For high performance analytical workloads, Db2 may be used as an engine.

With watsonx.data, clients can scale up or down automatically, optimizing costs and ensuring the right workload is running on the best engine for the job.



Protect data, manage compliance, and maintain trust with built-in governance, access controls, and enterprise security. Integrate with IBM's centralized governance capabilities for automatic policy enforcement, and enable responsible, transparent, and explainable data, and artificial intelligence (AI) workflows across the enterprise. Use governed data in watsonx.data to train, validate, tune, and deploy AI.

In order to truly realize maximum business value from AI, it is crucial for organizations to provide their analysts, business users, and data scientists with self-service access to high-quality, trustworthy, governed data.

It's clear how watsonx.data makes it possible for enterprises to scale analytics and AI, by providing a fit-for-purpose data store optimized for governed data and AI workloads. But not only is watsonx.data built for AI it's built *with* AI. One such shining example of this is a capability called *Semantic Automation*, which lets business users interact with their data simply by using natural language statements and questions. With Semantic Automation, users can easily

discover, augment, and refine data using self-service, conversational access without the need for coding or data engineering expertise.

Watsonx.ai pre-trained, fine-tuned foundation models semantically enrich watsonx.data, so users can simply search for data using business terms, and conduct other tasks such as importing, joining, and enriching the data via conversational interactions. For example, a user can type the following text into watsonx.data: *"I want to see if customer attrition can be explained by economic data."* Natural language processing (NLP) and natural language understanding (NLU) identifies the intent of this user's request and watsonx.data can subsequently generate SQL to retrieve a list of tables that are relevant to the request. And not only are the table names displayed, but AI-generated content (including tags and descriptions), is also provided. In addition to helping find data, Semantic Automation can also assist in joining data. For instance, a user can type the following request: *"I'd like to add residence data to this table."* Not only does Semantic Automation's AI find candidate tables that can be used to satisfy this request, but it will also look for join keys between the tables that have the right containment (the identifiers used for the join key match the data values between the tables)

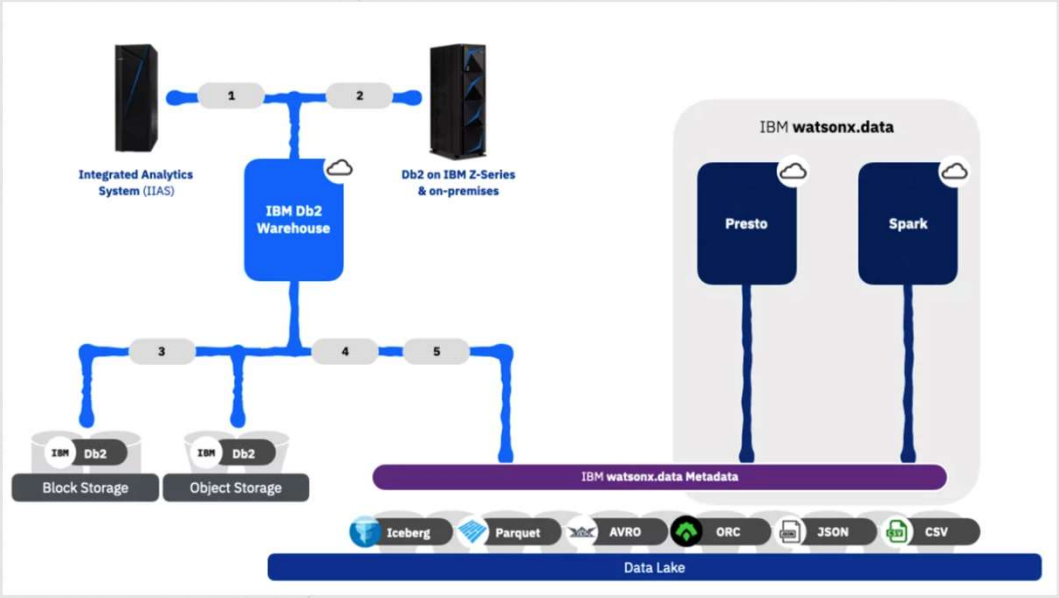
Agenda

1. Introduction to watsonx.data
2. watsonx.data Use Cases
3. Connecting watsonx.data to Db2

watsonx.data and Db2

Sharing data & tables across the 2 systems.

Using the **best tool** for the workload at hand.



Connecting Db2 with watsonx.data

1. Set up a STORAGE ACCESS ALIAS to connect to the Object Storage service

```
CALL SYSIBMADM.STORAGE_ACCESS_ALIAS.CATALOG(‘myalias’, S3',
‘s3.eu-south-2.amazonaws.com’, ‘****’, ‘****’, ‘mybucket’,
‘some/path', ‘R', ‘datalake-user-role')
```

2. Register the Watsonx.data metastore

```
CALL REGISTER_EXT_METASTORE('watsonxdata',
'type=watsonx.data,uri=thrift://hmsauth1.fyre.ibm.com:9083', ?, ?)

CALL SET_EXT_METASTORE_PROPERTY('watsonxdata', 'use.SSL', 'true', ?, ?)
```

3. You can now share tables between Db2 & watsonx.data
(See next slides)

Importing a Table from watsonx.data

```
CALL EXTERNAL_CATALOG_SYNC('metastore-name', 'schema-name',
'table-name', 'exist-action', 'error-action', ‘options')
```

- Brings the table definition into the Db2 catalog. The **data is shared** between the 2 systems. Need to re-synch if the schema of the table changes.
- Multiple tables & schemas can be specified using regular expression.
- The *metastore-name* is the name used to register the metastore when setting up the connection.
- If a table is REPLACEd, it is dropped and re-created.
  - Working on improving that.

Exporting a Table to watsonx.data

Regular Tables

```
CREATE DATALAKE TABLE hiveschema.db2exported(id int, name varchar(32))
STORED AS PARQUET LOCATION 'DB2REMOTE://hive-
bucket//hiveschema/db2exported' TBLPROPERTIES('bigsql.external.catalog' =
'watsonxdata')
```

Iceberg Tables

```
CREATE DATALAKE TABLE iceberg.db2exported(id INT, name VARCHAR(32))
STORED AS PARQUET STORED BY ICEBERG LOCATION 'DB2REMOTE://iceberg-
bucket//iceberg/db2exported' TBLPROPERTIES('iceberg.catalog' = 'watsonxdata')
```

- The table is created in both the Db2 & watsonx.data catalog and **data is shared**
- The value of the property is the name used to register the metastore when setting up the connection.

Those tables are not Db2 tables!

- The storage is not owned by Db2. In fact, the data is not owned by Db2. Db2 is one of multiple engines that can interact with those tables.
- This provides flexibility but comes with limitations as well – e.g. you cannot create indexes on those tables.
- What is provided here is an integration with watsonx.data, not an evolution of existing Db2 tables.

Exporting a Table to watsonx.data

Regular Tables

```
CREATE DATALAKE TABLE hiveschema.db2exported(id int, name varchar(32))
STORED AS PARQUET LOCATION 'DB2REMOTE://hive-
bucket//hiveschema/db2exported' TBLPROPERTIES('bigsql.external.catalog' =
'watsonxdata')
```

Iceberg Tables

```
CREATE DATALAKE TABLE iceberg.db2exported(id INT, name VARCHAR(32))
STORED AS PARQUET STORED BY ICEBERG LOCATION 'DB2REMOTE://iceberg-
bucket//iceberg/db2exported' TBLPROPERTIES('iceberg.catalog' = 'watsonxdata')
```

- The table is created in both the Db2 & watsonx.data catalog and **data is shared**
- The value of the property is the name used to register the metastore when setting up the connection.

A Few Gotchas

1. Db2 has a 20 mins (by default) **data cache** for DATALAKE tables.

Force its refresh with the **HCAT_CACHE_SYNC** stored procedure when you insert data into a shared from watsonx.data.

2. Some INSERT statement may implicitly create new partitions. For shared tables, they will **not be registered in the other system metastore**.

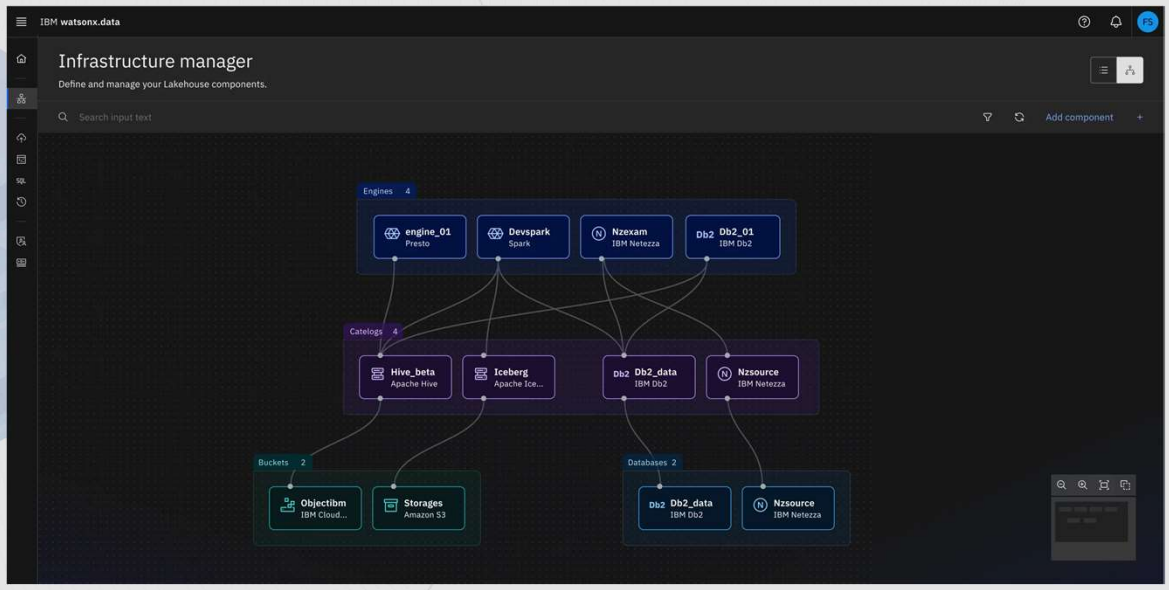
In Db2, run **MSCK REPAIR TABLE** on the table.
In watsonx.data run **system.sync_partition_metadata** on the table.

3. Schema evolution for shared table is disabled in Db2 and must be done from the watsonx.data side.

Performance Best Practices

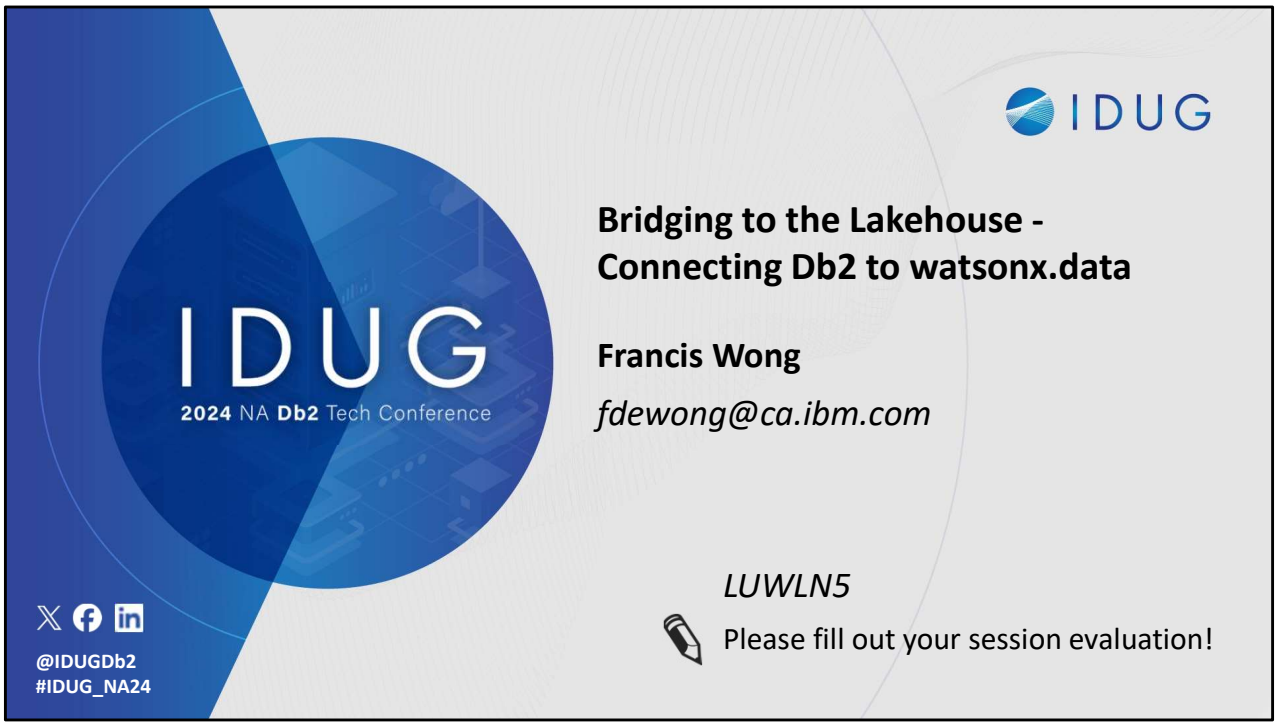
Partitioning	Partition your data with PARTITION BY
No Small Files	Use staging tables to make bulk INSERTS
Stats!	Use ANALYZE TABLE statement to generate statistics for DATALAKE tables
Watch Your Strings	STRING type has no length and causes performance issues. Use VARCHAR(N) ALTER to VARCHAR(N)
SQL Constraints	Help the compiler generate an optimal plan. Constraints are informational only for DATALAKE tables.
MQTs Caching	Native Db2 MQT can be created over DATALAKE table to provide massive speedup.

watsonx.data Console



A Few Links

- Introducing the next generation of Db2 Warehouse
  - ibm.com
- Better together: IBM watsonx.data and IBM Db2
  - ibm.com
- Accessing watsonx.data
  - IBM Db2 Warehouse Docs
- Accelerating your Datalake tables with a Cache of Db2 Warehouse MQTs
  - idug.org


IDUG
2024 NA Db2 Tech Conference
@IDUGDb2
#IDUG_NA24
IDUG
Bridging to the Lakehouse - Connecting Db2 to watsonx.data
Francis Wong
fdewong@ca.ibm.com
LUWLN5
Please fill out your session evaluation!