brightly

Rewriting Tomorrow

Lakehouse architecture in AWS - AWS User Meetup Oulu

30.11.2022 Pekka Vuorio , +358 40 1257949, pekka.vuorio@brightly.fi

We are Brightly

Tech consultancy & expertise center focusing on development of future digital and data solutions.

Founded by some of the best experts in data-driven digital services and cloud data solutions development. We have created many of the famous & industry leading data-driven solutions for biggest companies in the nordics in consumer services, media, finance, energy, transport, pharmaceutical, housing, pulp, paper, chemical, metal works, and electronics industries.

We are building the most talented tech team to create the next generation data- and AI-driven digital solutions and cloud data solutions that make our customers leaders in their sectors.

Founded in 2020

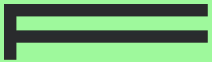
Prior to founding of Brightly we have created many famous & sector leading cloud data and digital solutions.

Team of 50 & Growing

Building growing team of tech experts for development of next generation solutions.

Offices

Helsinki
Tampere
Oulu



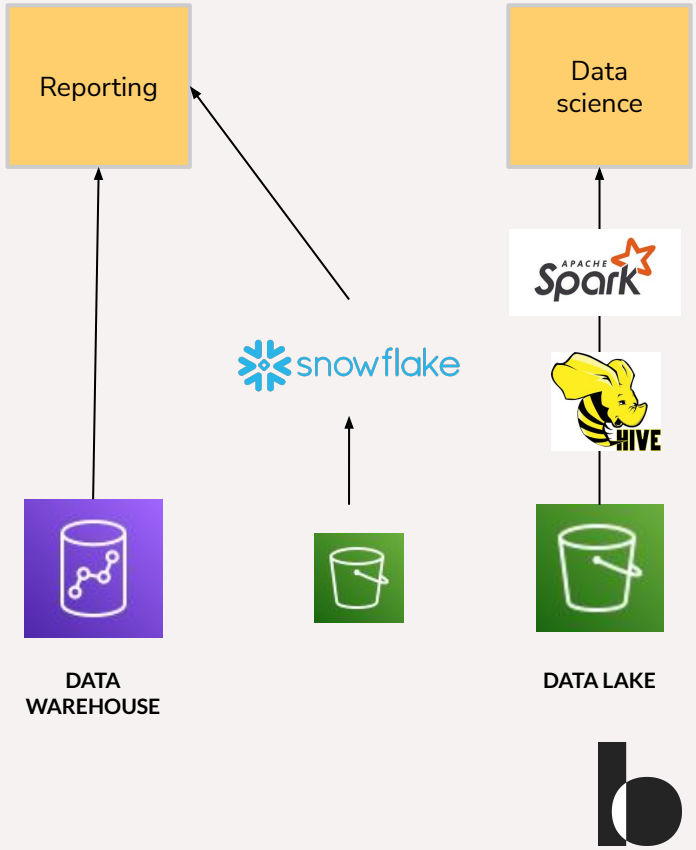
1300+ Experts in our Fusion-ecosystem

Expert companies in design & development of Digital solutions, software, security, devops, testing, maintenance.

Lakehouse architecture

Background

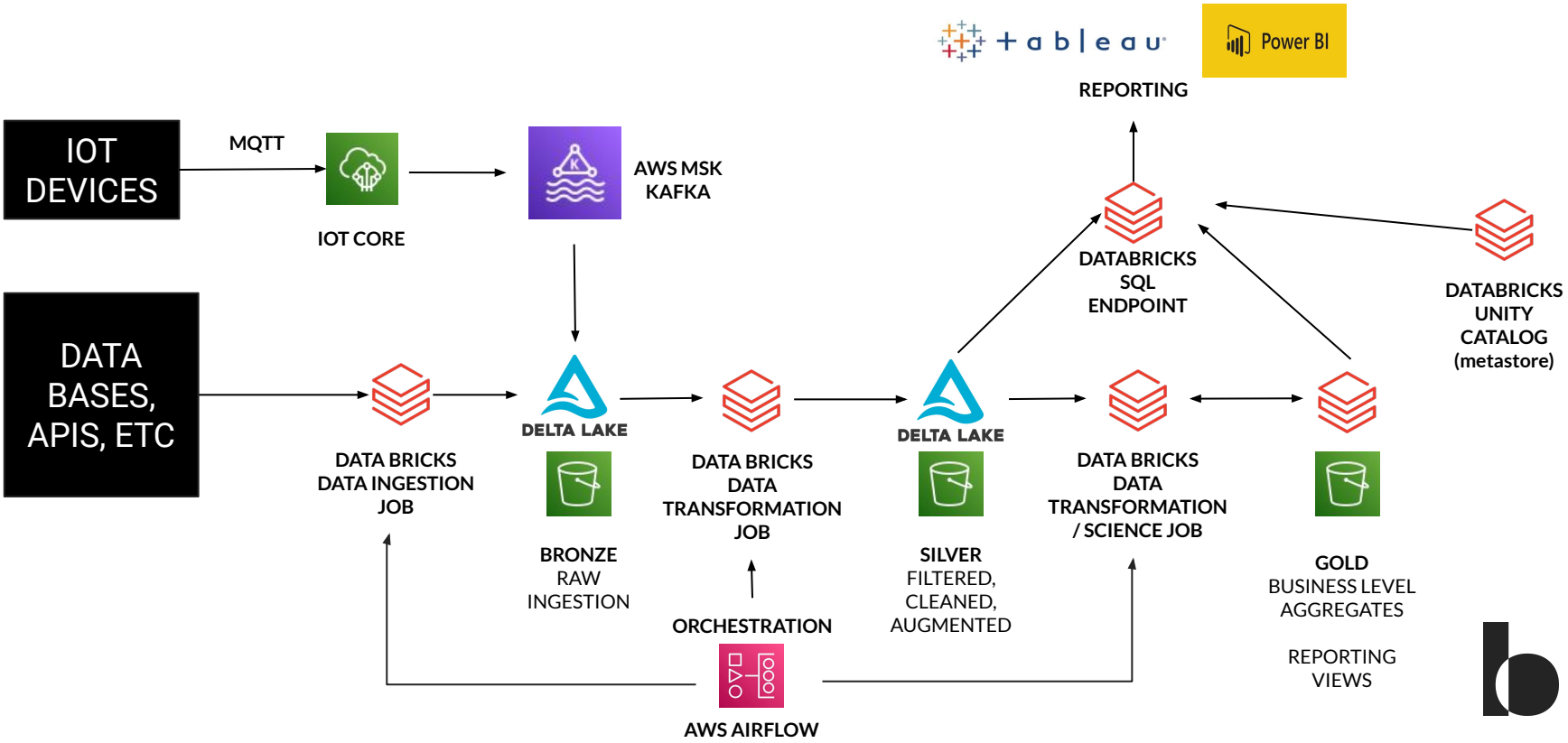
- Reporting use cases built on top of data warehouse or relational database
  - First to on-prem Oracle, Teradata, SQL Server, etc
  - Cloud based solution AWS Redshift, Azure SQL DW, etc.
    - Compute and storage capacity based on the instance type and cluster size → scaling issues → cost issues
  - Snowflake was a first solution separating compute and storage capacity and enabling easy scaling based on the usage
- Data lakes were build for data science and advanced analytics use cases
  - Data in files and partitions
    - Problems with DELETEs and UPDATES
  - Hadoop-clusters and Spark for parallel processing of data
  - HIVE metastore provides ODBC/JDBC access to data in datalake
    - Poor performance compared to DWs
- **Lakehouse combines data lake and DW into one solution**



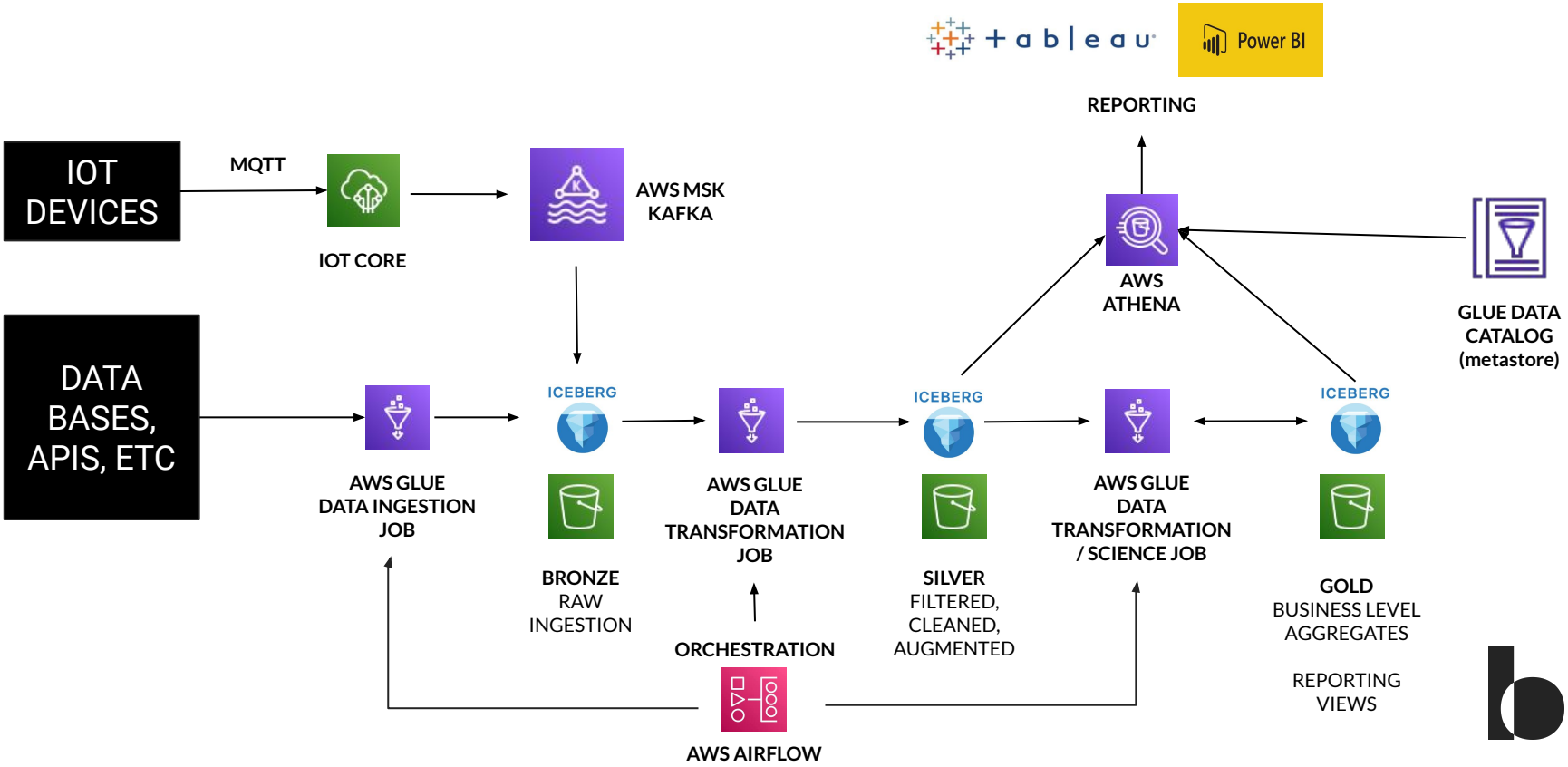
Delta lake and Iceberg

- Both provides basically same features
- File based storage + “transaction log”
- Features from DWs
  - ACID transactions: inserts, deletes, updates
  - Snapshots: Enable writing and reading data on a same time. Reader accesses always complete dataset.
  - Time travel
  - Performance optimisations
    - Z-order indexing
    - Metadata management
- Same storage for streams and batch loads
- Both are open source, no vendor lock in
  - Libraries at least for Python and Java (and spark)
- Metastore in used for definition of databases and tables
  - AWS Glue Data catalog or Databricks Unity Catalog can be used

Lakehouse Architecture, Delta/Databricks based



Lakehouse Architecture, AWS native solution



DW based lakehouse

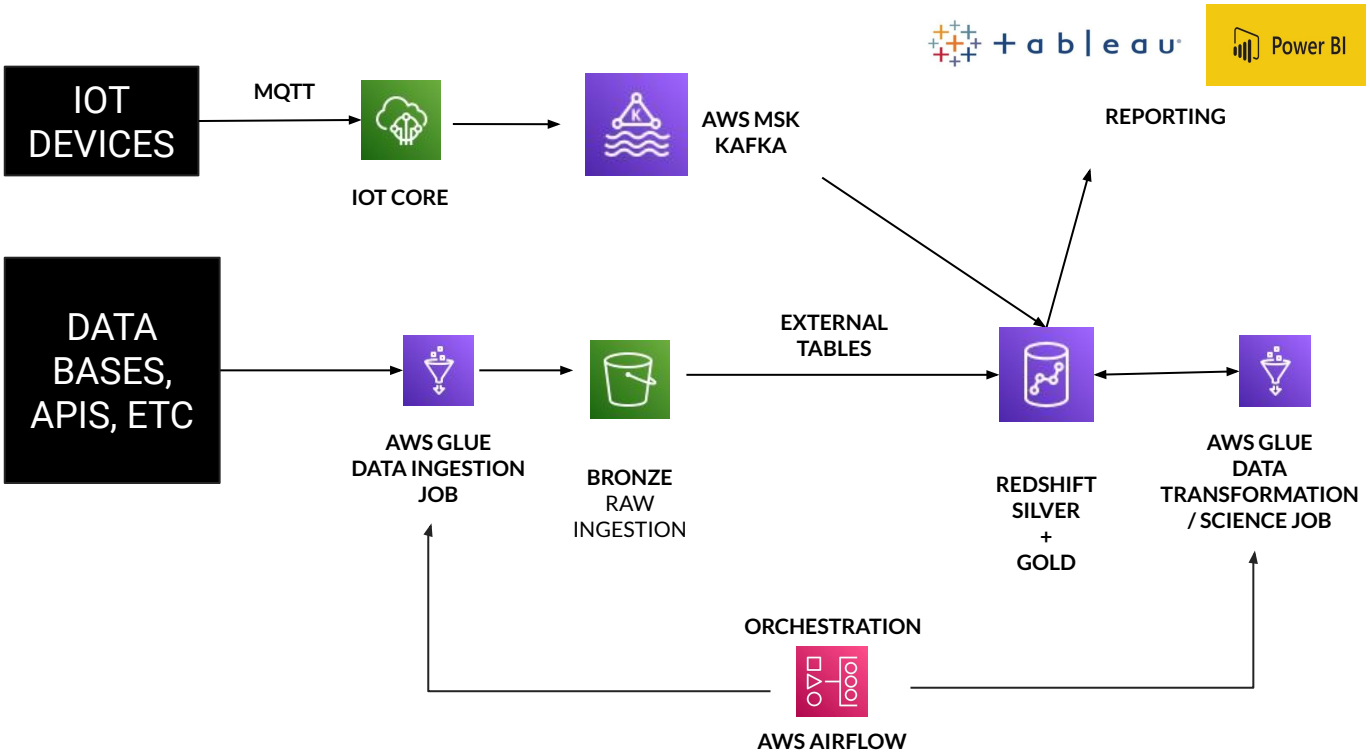
AWS announces Amazon Redshift integration for Apache Spark

Posted On: Nov 29, 2022

Amazon Redshift integration for Apache Spark helps developers seamlessly build and run Apache Spark applications on Amazon Redshift data. If you are using AWS analytics and machine learning (ML) services—such as Amazon EMR, AWS Glue, and Amazon SageMaker—you can now build Apache Spark applications that read from and write to your Amazon Redshift data warehouse without compromising on the performance of your applications or transactional consistency of your data. Amazon Redshift integration for Apache Spark builds on an existing open source connector project and enhances it for performance and security, helping customers gain up to 10x faster application performance. We thank the original contributors on the project who collaborated with us to make this happen. As we make further enhancements we will continue to contribute back into the open source project.

At its core, Snowpark is all about extensibility. It was designed to let data engineers, data scientists, and other developers work with data more efficiently and effectively in their programming languages and tools of choice, including Scala, Python (in private preview), and Java, using familiar programming constructs such as DataFrames. And it was built to move that work right to where the data lives: in Snowflake's scalable, secure compute engine.

Lakehouse Architecture, AWS native solution



Thank you