Dedicated Semantic Layer Architecture for Effective Data Analytics and Visualization: A Case Study

Ramla Suhra
Department of Digital Data Solutions
H-E-B
Texas, USA

Abstract:- Organizations these days increasingly rely on fast growing data for their critical business decision making. To leverage full potential of data, petabytes of data are being ingested into central data lakes mostly powered by cloud. They also realize that it is not enough to just collect huge amounts of data. To derive value from this data, it must be cleansed, interconnected and translated from its complex technicalities into an easily interpretable and more familiar business terminology. Building a semantic view of the data enriched with business metrics enable users to query, analyze and visualize information as quickly as the business demands. While the semantic layer is perceived as the cornerstone in modern data architecture, there are different perspective towards where or how this should be implemented. Additionally, understanding the evolution of semantic layer over the years can help choose the right architecture when attempting to build one in an organization. With the advent of Artificial Intelligence (AI), it is imperative that we discuss the impact of AI on this topic. This research delves into the evolution of the need and significance of semantic layer, exploring their architecture, benefits. It also analyzes the challenges faced during semantic layer adoption and the outlook.

Keywords:- *Data Analytics; Artificial Intelligenc;, Data Architecture; Semantic Layer.*

I. INTRODUCTION

A semantic layer is a layer of abstraction that separates the physical representation of data from what is to be viewed by business users. It provides a logical view of the data that helps end users access the data using common business terminologies which are easier to understand for them. By providing a more business-friendly representation of data, it acts as a bridge between the raw data and the business users.

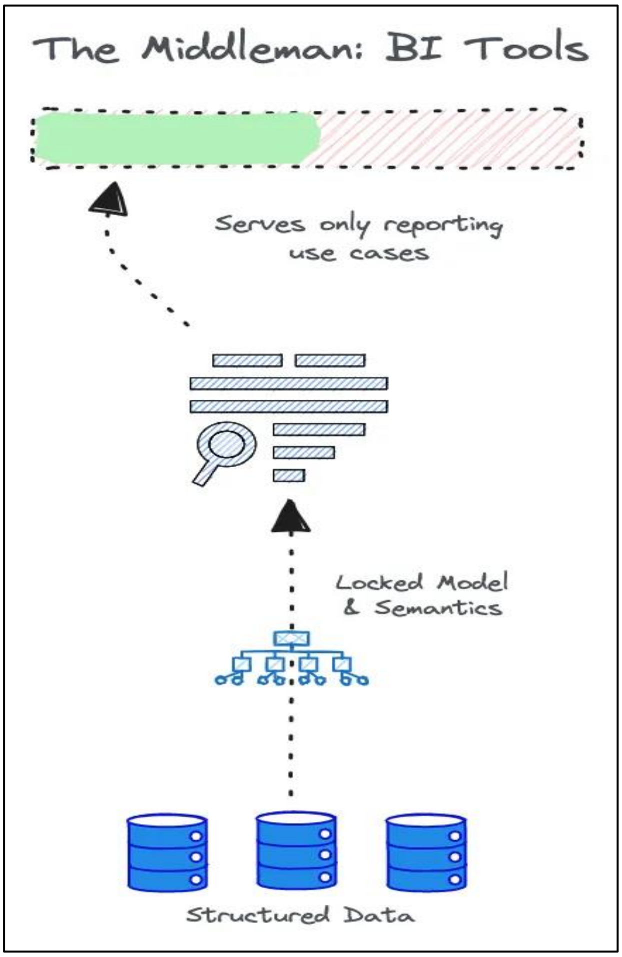
A. Evolution of Approaches

The concept of semantic layer is not entirely new as it is dated back to early 1990s where it was patented by Business Objects and then subsequently challenged by MicroStrategy in 2003[1][2]. While the semantic layer's origins lie in the days of OLAP[3], the concept is even more relevant now in modern data stack.

One of the initial approaches towards building the business-friendly representation was by using Data Cubes [4]. Data was pre-aggregated and stored in multi-dimensional array structure for efficient storage and retrieval. This approach was particularly well-suited for online analytical processing (OLAP) operations, as it enabled rapid slicing and dicing of data along various dimensions. In this approach new reports needed new cubes thus making it less flexible and time consuming to build.

Increasing complexity of data and requirements for more flexibility naturally progressed into building a newer approach, i.e. Semantic layers within Business Intelligence (BI) tools ("Fig. 1"). With wider adoption of business intelligence techniques, different departments started using different BI tools as found by the survey by 360Suite which states that 67% of respondents say they can take advantage of multiple Business Intelligence solutions within their organization [5]. This led to data silos, resulting in inconsistent business logic, diverse metrics, and varying interpretations of the same data within the organization. The analytics team could have a fully defined model locked inside their preferred BI tool, which would make it inaccessible or, rather, inoperable for any other team or tool sitting outside this ecosystem.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT1676

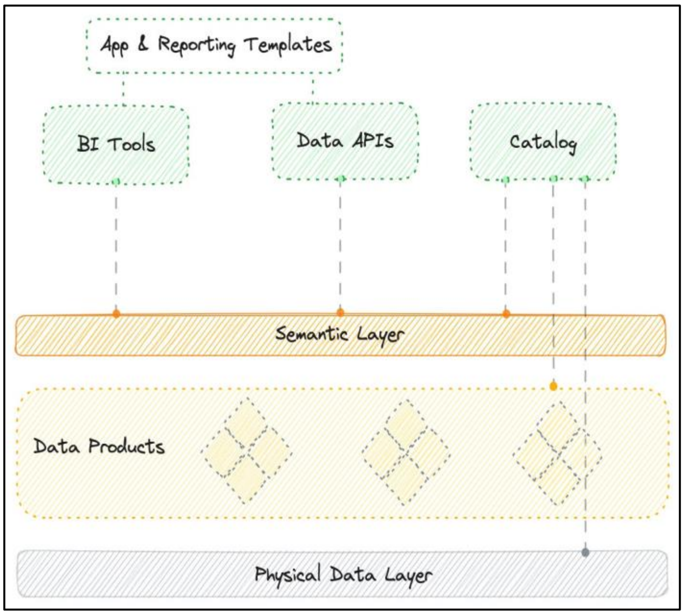


"Fig. 1, BI Tools as Semantic Layer" [6]

The solution for the above stated is a dedicated layer for semantics which is not limited within the boundaries of multiple BI tools but also reachable by a variety of tech stakeholders as per their needs and use cases. In the later years, there has been much increased focus on the "Dedicated Semantic Layer" model. In 2018, Jinja templates and dbt introduced transformation layer into semantic layer. In 2019, Looker and LookML were branded as the first real semantic layer. By 2022, there were more additions like MetriQL, Minerva and dbt.

II. DEDICATED SEMANTIC LAYER (DSL)

DSL is implemented as a dedicated layer between data sources and all the consumers including BI tools. Irrespective of the BI tool users choose, DSL allows them to work with the standard semantics and underlying data layer, ensuring that the data insights and reports are consistent across all the integrations. With clear advantages over the predecessor approaches, DSL has found a critical position within the modern data stack.



"Fig. 2, Semantic Layer" [6]

A. *Advantages*

- Allows business matrix and KPI to be shared with disparate platforms including BI and data science tools.
- Abstracted from the data sources, they act as data virtualization for business users allowing them to access data from multiple sources.
- Complex business logics and calculations can be defined once and reused many times by various sub departments in the organization.
- Decoupling of data lake and business matrix allows for scalability.
- Helps track the lineage of business matrix columns since they are integrated with existing data platform.
- Simplifies governance and compliance processes in the dedicated approach as compared to federated silos.
- Easy onboarding of new integrations on existing KPIs or matrix.
- This can act as application layer through data APIs.
- Enhanced query performance and cost efficiency due to elimination of redundant processes.

B. *Challenges*

Besides a lot of apparent advantages, DSL also brings in some challenges to the picture. Introducing a new layer adds to the complexity of the existing data platform. Another downside is that if newer technologies or solutions are being leveraged to build this layer, it requires operations support leading to increased cost of maintenance. Opening access to the layer using APIs can also introduce performance issues as the data grows.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT1676

III. ARCHITECTURE

The proposed architecture for DSL comprises of various components as described below.

A. Data Modeling

Data modeling is the creation of business-oriented, logical data models that are directly mapped to the physical data structures in the data warehouse. Data is de-normalized and standardized with definitions of hierarchical dimensions that are used in business analysis, for example organization's definition of fiscal year to quarter to months.

This unit is critical as it will take care of definitions of key performance indicators (KPIs) metrics used in analytics. Even though every quantifiable data point captured from the business is a metric, not all metrics are equal in value from a decision-making perspective.

In this layer, we can have metrics serving feature as well in addition to metrics creation and management.

B. Data Transformation

Data transformation based on the models is the integral unit in the DSL which helps with curation of business-ready data. The semantic layer must be able to orchestrate transformations in the data platform. This data transformation workflows needs to be orchestrated using standard orchestration engines used within the organization. The semantic layer dynamically translates incoming queries from consumers (using the metrics layer logical constructs) to platform-specific SQL (rewritten to reflect the logical to physical mapping defined in the semantic model). Managing performance and cost requires the capability to materialize certain views into physical tables. The output of this layer must be stored in a low latency storage for quicker access.

C. Data Storage

While DSL is more about the semantics, the underlying storage is also critical while designing the serving layer for DSL. Data storage can leverage existing storage components within the data platform or add an additional layer of structured low latency storages.

➢ *In-Memory Semantic Layers*

In-memory semantic layers load data into the server's memory, allowing faster retrieval and analysis. They are helpful for organizations that require real-time or near-real-time analysis of data.

In-memory semantic layers leverage high-speed memory for fast data access, enabling quick loading and real-time analysis. This makes them ideal for time-critical applications. As data is stored in memory, there is no need to retrieve it from disk, resulting in low latency.

Nevertheless, in-memory semantic layers require significant memory, which can be expensive. The available RAM limits the amount of data that can be stored in memory. Furthermore, horizontal scaling of in-memory semantic layers might be more challenging compared to other types.

➢ *Relational Database Semantic Layers*

Relational database semantic layers store data in a relational database, such as SQL Server, PostgreSQL, or Oracle. These are ideal for organizations with substantial data volumes and demanding data management requirements. Relational databases offer robust data management features, including data integrity and security. They can scale horizontally by adding more servers to the cluster. Being a well-established technology, relational databases integrate seamlessly with other systems.

However, relational databases can be slower than in-memory semantic layers due to disk-based data retrieval. They can be complex to set up and manage, and maintaining them, especially with high availability and redundancy, can be costly.

➢ *Graph Database Semantic Layers*

Graph database semantic layers utilize graph databases like Neo4j, AWS Neptune for data storage. They are suitable for organizations handling complex and interconnected data. Graph databases are flexible and can handle intricate, interconnected data. Graph databases can outperform relational databases for specific queries. Graph databases can scale horizontally by adding more servers to the cluster.

However, graph databases might not offer the same capabilities as relational databases. Graph databases can be complex to set up and manage. Maintaining graph databases, especially with high availability and redundancy, can be expensive.

➢ *Cloud Storage*

Distributed storage can be cheaper and easy to integrate with the enterprise governance and catalog They need to be integrated with low latency caching or view like abstractions to improve query performance. Setting up the infrastructure will be easier as this is already setup in the platform.

Inbuilt governance capabilities make it more reliable. This can scale horizontally as needed and is cost efficient. Adding a consumption layer on top of this storage will help faster reads based on access patterns.

D. Monitoring and Alerting

All the orchestration and transformation workloads need to be monitored for reliability. Job failures, resource utilization latency are some key areas to be included. Data quality monitoring is very important too and needs to be part of the monitoring component. Data accuracy, completeness and validity can be monitored and if required reported to the stakeholders in case of any anomalies. Performance monitoring is another aspect that needs to be taken care of in this component. Query performance, API response time, error rates are some of the factors to be considered.

Monitoring and alerting techniques involve setting up thresholds for each metric and alerting if they are exceeded. ETL Monitoring tools, DB monitoring tools, Log analytics, data quality tools etc. can be leveraged for this purpose. Proactive monitoring, regular review and automated alerts are

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT1676

some of the best practices that can be followed to help the semantic layer to work seamlessly providing continuous support without interruptions. Following this approach will ensure that the production support process is streamlined.

E. Data Archival and Retention

This involves systematic storage and cleanup of data based on the usage and data retention policies. Data lifecycle management is critical to ensure that data is only stored for the time it is required. If the data is ready for archival it can be first moved into a less frequently accessed storage for some period after which it is permanently deleted. Data can be hence categorized into frequently accessed, less frequently accessed, unused. The unused data qualified for data purging which will handled by this component. Determine the necessary retention period for different data types based on their business value.

Data Archival strategies can be carefully planned based on the storage requirements. This unit also need to have data restoration mechanism in place to cater to restoration needs as it comes. There should be inbuilt audit and compliance validation mechanism to ensure that the data access and compliance is intact.

F. API Access

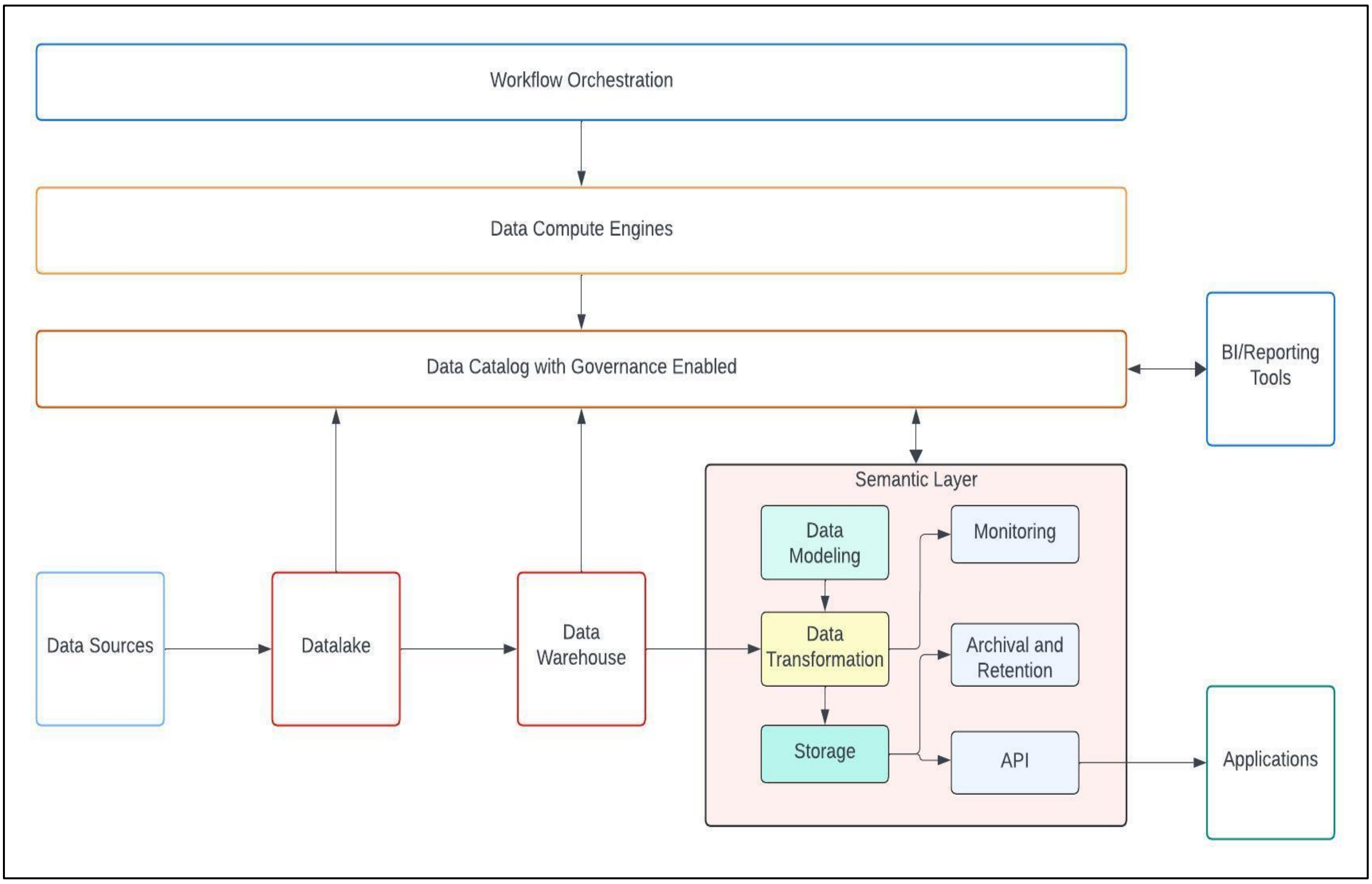
API access allows DSL to serve as an application layer by letting disparate systems or applications connect to this layer. API enables users to query metrics and dimensions using the JDBC protocol, while also providing standard metadata functionality. Popular API approaches are Rest/Graphql.

The API layer allows seamless integration with machine learning frameworks. Machine learning models can access data from the semantic layer for training and prediction. Developers can build custom applications that leverage the data and insights provided by the semantic layer. Data can be integrated with other systems, such as ERP, CRM, and marketing automation tools.

G. Integration with Enterprise Data Catalog and Governance.

DSL can be integrated with data catalog and governance for enhanced data security and compliance. Users can interact with data catalog to read all the sematic layer tables by leveraging existing compute engines in the platform. The semantic layer can directly access metadata from the data catalog and create metadata in the catalog for users of DSL.

This is a powerful combination as the semantic layer will be fully secure and enriched with all relevant features due to this integration. The data catalog serves as a central repository for metadata, including data definitions, relationships, and usage guidelines including storage and tagging guidelines. Integrating DSL with catalog automatically inherits all these principles without needing to reinvent the wheel for the DSL. Another advantage of this approach is that both DSL and data catalog can enforce consistent data definitions and usage standards.



“Fig. 4, Proposed Architecture for DSL”

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT1676

IV. IMPLEMENTATION APPROACHES

The architecture outlines the components and details of each of them. This section talks about the implementation approaches that can followed for building DSL.

A. Open-Source Tools

The most cost-efficient ones which when integrated with the existing data platform will be good for DSL. For many tools, there are active communities providing support and documentation to help troubleshoot issues.

B. Software As Service

With dedicated support SaaS products help reduce the operation and maintenance overhead.

C. Build Your Own

With the advancement of AI and ML technologies, major data platform products are also trying to get all the required features including Sematic layer capabilities packaged within it. Thus, building a layer will be very easy and cost efficient rather than buying a new product.

V. USE CASES OF DSL

The use cases of DSL are like the use cases of BI and reporting except that introduction of dedicated layer can ensure that the data set can be reused by multiple business units.

- ***Sample use case #1:*** A retail company wants to provide its executives and business analysts with a unified view of sales performance across regions, product categories, and customer segments.

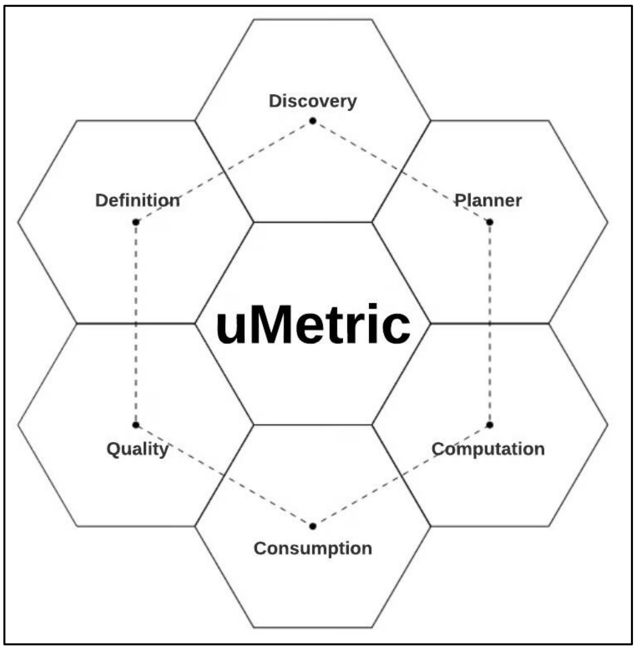
DSL can be used in this case to abstract underlying sales data from multiple databases and systems. It can be used to define the standardized dimensions, and hierarchies, allowing both executives and BI users to explore the same data set from two different views like a portal application vs BI tool.

On top of this some of this data when integrated with ERP data can be used to help KPI metrics for supply chain managers to help them identify the demand forecast and be prepared for the demand. This shows that the same layer can be reused across multiple business domains and areas without compromising data silos, processing similar datasets multiple times and thereby introduce data quality issues.

- ***Sample use case #2:*** At Uber, business metrics are crucial for understanding their performance, evaluating new products, and making informed decisions. These metrics are used for various purposes, ranging from troubleshooting specific issues (like a fare problem) to powering complex machine learning models that optimize pricing strategies on a global scale.

Uber realized that standardizing metrics was essential when democratizing access to data and insights. Without standardization, multiple versions of the same metric could exist, leading to inaccurate or misleading information and ultimately poor decision-making.

To this problem, Uber built a solution called uMetric which is a unified data sematic layer [9]



"Fig. 4, uMetric at Uber" [9]

VI. AI AND FUTURE OF SEMANTIC LAYER

Leading organizations incorporate data science and enterprise AI into everyday decision-making in the form of augmented analytics. A semantic layer can be helpful in successfully implementing augmented analytics.

An AI-ready universal semantic layer can also power customer-facing applications that enable organizations to make the most of their data and their customer interactions. This can be achieved through the integration of DSL with customer facing application via APIs in contrast to BI tools which has been the primary use case so far.

A universal semantic layer that is AI-ready is needed to connect and work with diverse data platforms, protocols, and consumption tools. This decouples the data from consumption, thereby enabling the democratization of data analytics and AI in the enterprise [7].

Semantic layers can be used to publish AI/ML-generated insights to business users using the same analytics tools they use to analyze historical data. Thus, we see that in future there are chances for convergence of analytical tools while maintaining DSL as its meaningful business ready metadata

Semantic layers will expand into more new industries, and they offer potential for all industries that rely on data, including logistics, energy and agriculture. Their implementation could bring significant benefits by extracting insights from different data sources [8].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT1676

VII. SUMMARY

By bridging the gap between technical data and business understanding, semantic layers empower business stakeholders to derive quality KPI and metrics. Thus, it can be summarized that DSL is a vital component of modern data architecture. By providing a simplified, business-focused view of data, it empowers organizations to extract actionable insights and make informed decisions, driving innovation and growth.

REFERENCES

[1]. J.-M. Cambot, B. Liautaud, and S. F. Sa, "US5555403A - Relational database access system using semantically dynamic objects - Google Patents." https://patents.google.com/patent/US5555403

[2]. "Microstrategy, Inc. v. Business Objects, S.A., 661 F. Supp. 2d 548 | Casetext Search + Citator." https://casetext.com/case/microstrategy-inc-v-business-objects-3

[3]. Wikipedia contributors, "Online analytical processing," *Wikipedia*, Oct. 08, 2024. https://en.wikipedia.org/wiki/Online_analytical_processing

[4]. M. 2023 7 M. Read, "What is a data cube?," *365 Data Science*, May 02, 2023. https://365datascience.com/trending/data-cube/

[5]. Emilien, "Business Intelligence Trends 2020," *Wiiisdom | Analytics Governance Solutions*, Feb. 09, 2023. https://wiiisdom.com/ebook/business-intelligence-trends-2020/

[6]. A. Kumar, "The semantic layer movement: the rise & current state," *Modern Data 101*, May 02, 2024. https://moderndata101.substack.com/p/the-semantic-movement-the-story-of

[7]. A. Keydunov, "Real-time AI experiences can't advance without a universal semantic layer," *RTInsights*, Mar. 03, 2024. https://www.rtinsights.com/real-time-ai-experiences-cant-advance-without-a-universal-semantic-layer/

[8]. A. Schwanke, "Semantic layer — one layer to serve them all - Axel Schwanke - medium," *Medium*, Aug. 29, 2024. [Online]. Available: https://medium.com/@axel.schwanke/semantic-layer-one-layer-to-serve-them-all-d0ef7eff1ffa

[9]. "The Journey towards metric Standardization | Uber Blog," *Uber Blog*, Jan. 12, 2021. https://www.uber.com/blog/umetric/