Snowflake, *the* Cloud Data Platform

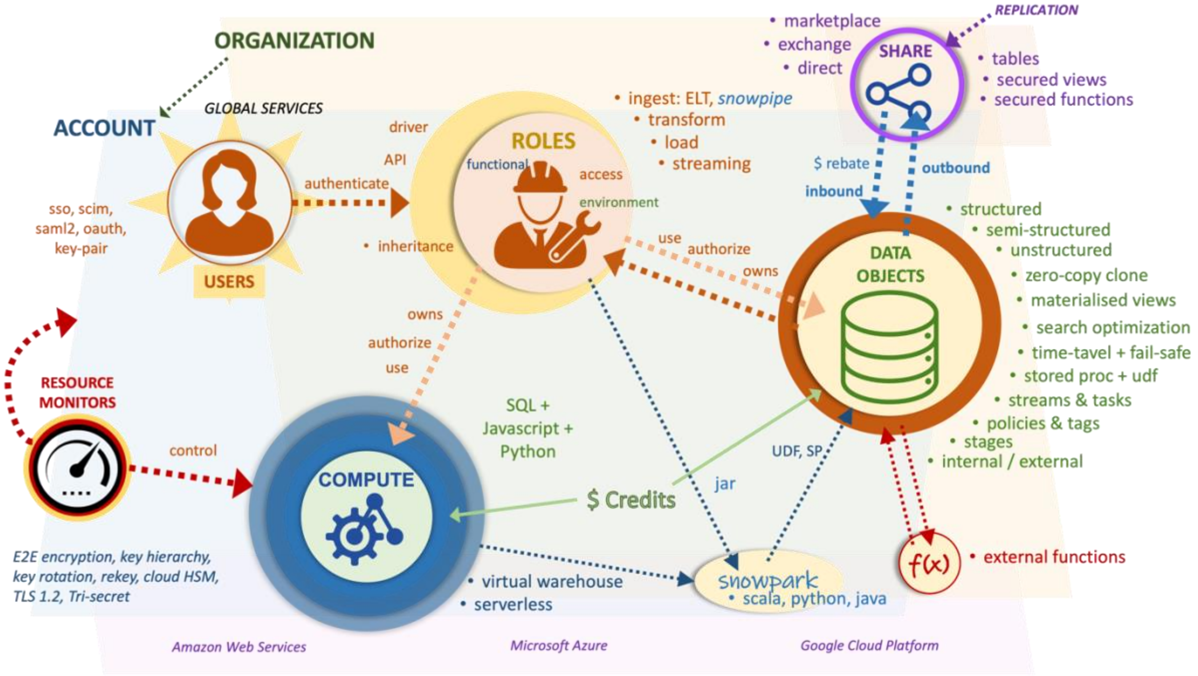

Figure 0-1 Out the Box

If you haven't heard, *the* Cloud Data Platform is **Snowflake**. There is no other platform on the market that offers a secure, native **Data Cloud** architecture with the scalability and flexibility of what you expect from Cloud quite like Snowflake. Andy Jassy of AWS explained back in 2012:

"Invention requires two things: One, the ability to try a lot of experiments, and two, not having to live with the collateral damage of failed experiments."

And this is true for building on Snowflake; Snowflake contains all the traditional logical components of a database (and more) but without the maintenance typically seen on other Cloud Data Platforms. On Snowflake you will not need to do the following:

- **Never** need to enact a table **rebuild** when altering table or column properties
- **No** need to define a **distribution key** or **data replication** to best serve queries depending on data locality on disk
- **No** concurrent **connection limit** and/or **query limit**
- **No** need to disconnect users when **reclustering** of tables occurs
- **No** need to perform **vacuuming** to recover or reclaim storage and **no** need to schedule periods of **downtime** for maintenance

Snowflake's query engine is *relational* enabled and performs all the ANSI compliant operations you expect, like executing recursive *common table expressions* (CTE) without needing to know the depth of the hierarchies being queried. You're able to query structured and semi-structured content **in place**, clone databases, schema, and tables in seconds. Share data between Snowflake accounts thereby eliminating the need to move data between partners and customers alike! You can even load *unstructured* data into Snowflake!

Snowflake provides this abstraction as a relational data store from the native cloud architecture it is deployed on ***as a service***, the database components for you to build your cloud data platform as you wish all through using simple SQL statements. The service is secure and follows the same shared responsibility model of the cloud service provider it is deployed on (see go.aws/315Kf1Q), that is commonly referred to as security **of** the cloud versus security **in** the cloud. Snowflake was also the first to meet the EDM Council's 14 key data cloud controls for protecting sensitive data, an assessment independently conducted by KPMG, see: bit.ly/3DcXO0g

Keeping your data secure is paramount to Snowflake's **value proposition**
Because Snowflake is deployed in a region, it is replicated across availability zones, *see bit.ly/3LIMdJA*, not to mention, state-of-the-art implementation of data security in-transit and at rest, see: bit.ly/3LJ4WVd, and supports integration with the Cloud Data Provider's (CSP) PrivateLink or Private Link or Private Service Connect.

What's in the box?

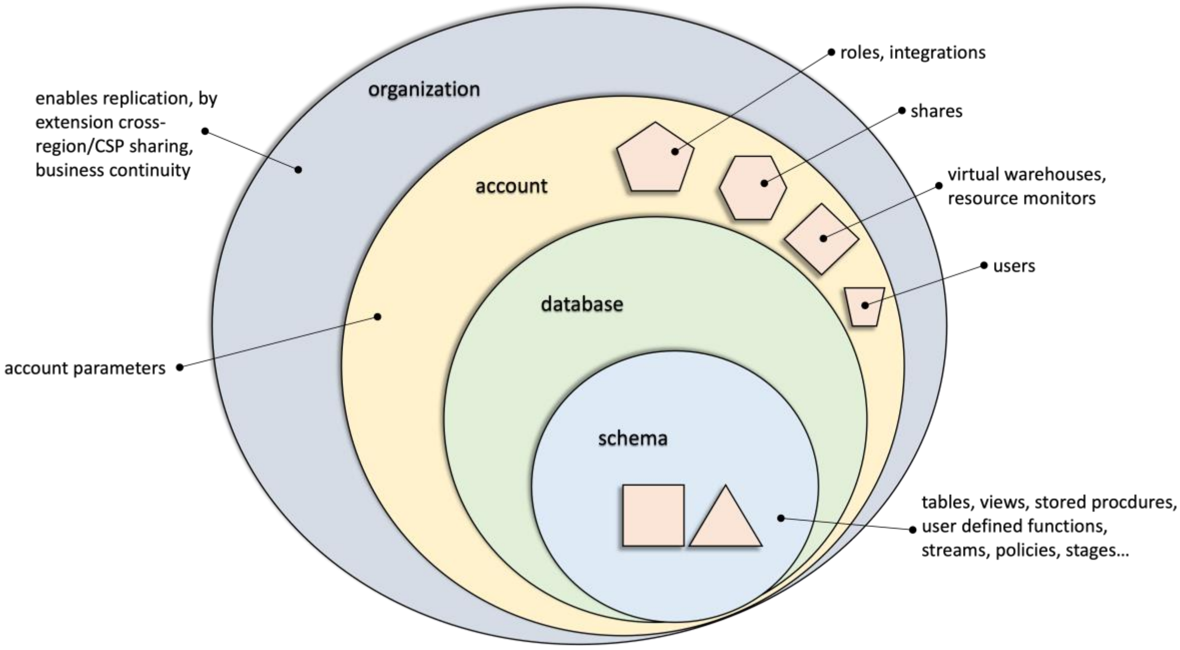
enables replication, by extension cross-region/CSP sharing, business continuity
organization
roles, integrations
shares
account
virtual warehouses, resource monitors
users
database
account parameters
schema
tables, views, stored procdures, user defined functions, streams, policies, stages…

Figure 0-1 Snowflake Object Hierarchy

At the time of writing, *(the list continues to grow)*, these are:

- Role Based Access Control (RBAC): **users own nothing, roles own everything**.
- Virtual warehouses, in essence they are the Snowflake-wrapped AWS EC2 instances (or Azure Virtual Machines, or Google Compute Engine - GCE) that perform *most* of the processing for your Snowflake deployment. These can scale up and out giving you the Massively Parallel Processing (MPP) experience using cloud-native technologies by separating storage and compute.

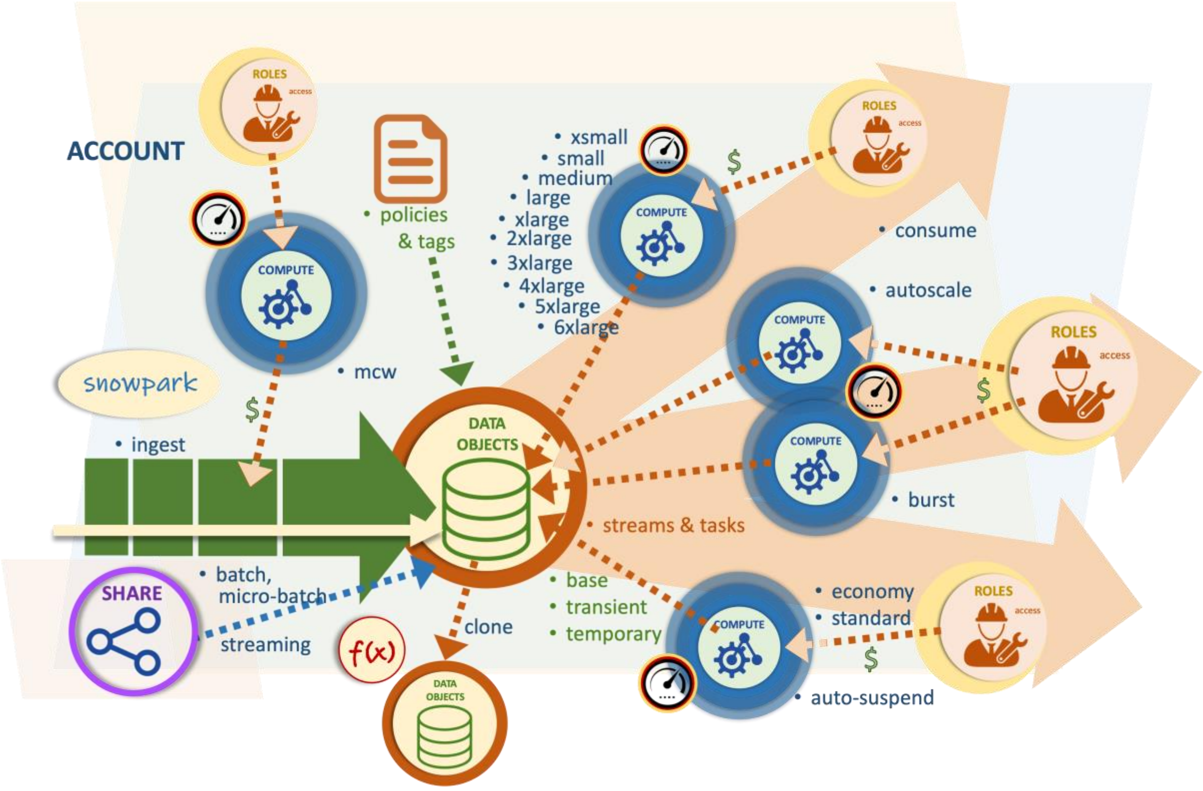

Figure 0-2 Elastic MPP

- As many databases as you want, a database can have zero or many schemas and within a schema schema-level objects are stored. Snowflake has its own proprietary storage format that is both row and column optimised from a few to potentially thousands of micro-partitions (16MB) in size that are **encrypted at rest** and **compressed by default** to achieve a better than blob-store compression ratio than parquet (for example).
- As an extension of the above, ***Time Travel*** is the live backup Snowflake provides for your base tables. Using Snowflake SQL, you can query a Snowflake table as it appeared at a point in time (up to 90 days). What's more is that you can take that snapshot of the table and clone it to a new table, *a very strong use case for DevOps development!*

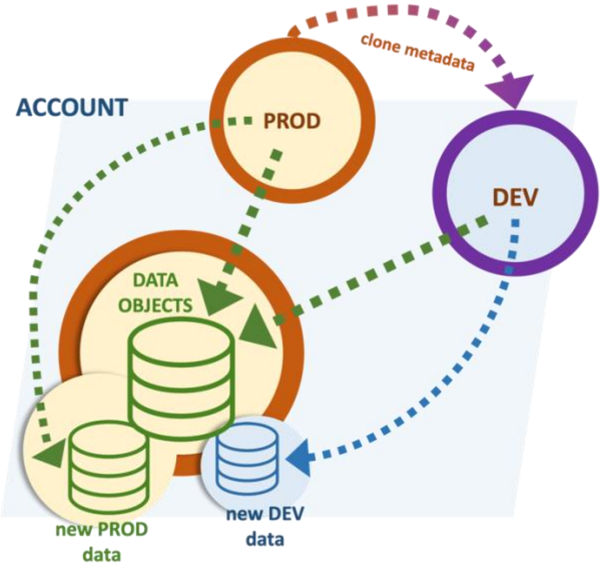

Figure 0-3 Rapid DevOps using clones

- To help orchestrate the movement of the minimal amount of data through the platform Snowflake released Streams and Tasks that can be used as automation of processing only **what is new** to an underlying table, view, share or even external table.
- Snowflake also supports **external tables** of such as Parquet, AVRO, ORC, CSV, and JSON. On top of these external objects, you can deploy streams and materialized views just like you would on top of Snowflake's native storage.
- **Data Governance** (DG) defined as a separate object, **policies and tags** can be deployed as a framework over your data that achieves a variety of data governance postures. The access of which is determined by the roles you design and setup.
  - Column Level Security through Dynamic Data Masking (DDM) Policies and external tokenization, Row Level Security through Row Access Policies.
  - Access history supporting Data Provenance and Object Dependencies supporting Impact and Reverse Impact Analysis
  - Auto-Classification and Tags, the tags in turn can be used to assign DDM
- Snowflake provides the ability to create your own user-defined stored procedures and functions (**UDFs**) in **JavaScript**, **Scala**, Snowflake **Scripting**, **Java** and **Python**. The UDFs in turn can be *scalar* or *tabular* and can even be defined as **external UDFs** to call externally developed code like AWS Lambda Functions.
- Snowflake's commitment to handling all your data in the cloud extends to **Snowpark** (portmanteau of Snowflake and **Spark**) where you can bring your Python, Scala, and Java workloads to Snowflake **without** the tedious task of performance tuning your Spark logic.

Is there a recommended Data Model?

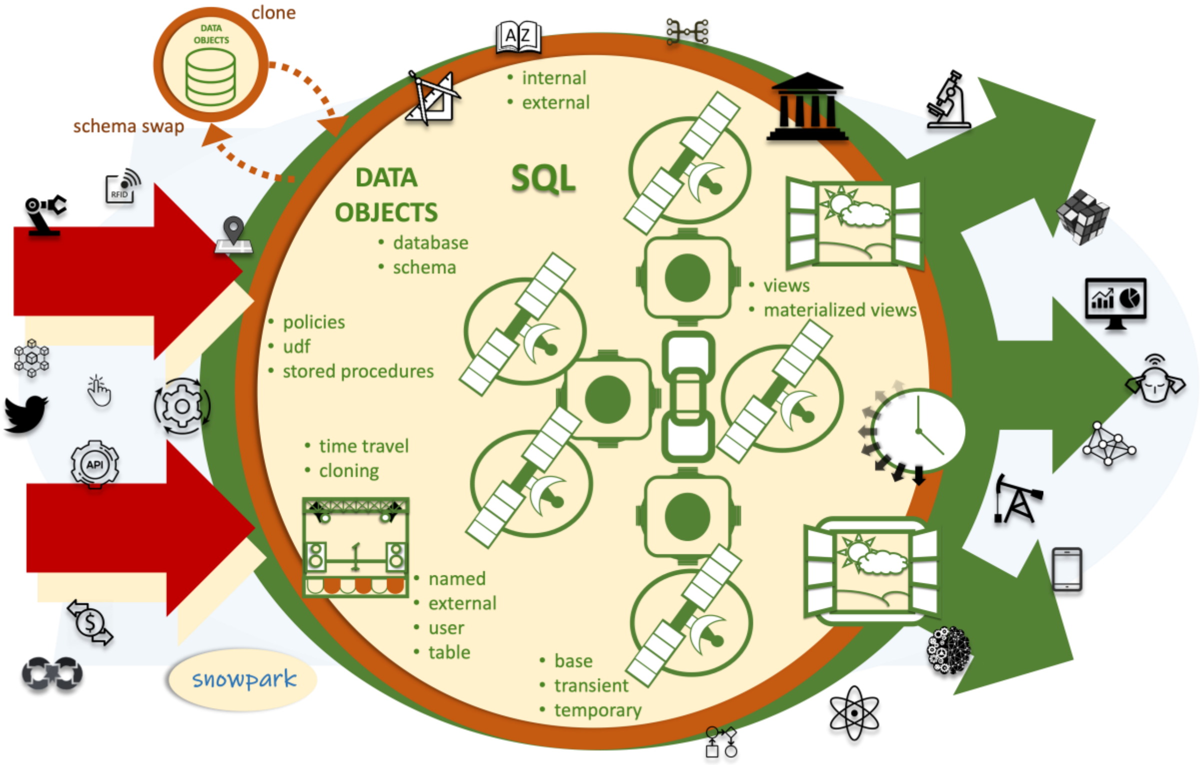

Figure 0-1 Multi-modal

Snowflake is **data model agnostic**, although the platform promises infinite scalability it does not matter whether that is Inmon-style **3rd normal form**, Kimball Style **Dimensional Modelling**, Linstedt's **Data Vault** or building and deploying flat wide tables or views. Snowflake *does* suite Data Vault 2.0 very well because of how immutable storage works in Big Data, and hence Snowflake's micro-partitions. INSERT-ONLY data modelling paradigms will perform better on Snowflake, *which Data Vault 2.0 is*. Learn more about Data Vault on Snowflake by visiting these stories below

- **Data Vault 2.0 on Snowflake…To hash or not to hash… that is the question,** bit.ly/3dn83n8
- **Why EQUIJOINS Matter!** bit.ly/3dBxOQK
- **Data Vault Test Automation,** bit.ly/3dUHPIS
- **Data Vault Dashboard Monitoring,** bit.ly/3BjSg1F
- **Data Vault PIT Flow Manifold,** bit.ly/3iEkBJC
- **Data Vault's XTS pattern on Snowflake,** bit.ly/3aCCRhQ
- **Data Vault Agility on Snowflake,** bit.ly/337Jhp3
- **Kappa Vault,** bit.ly/3JbRf05

Data Sharing

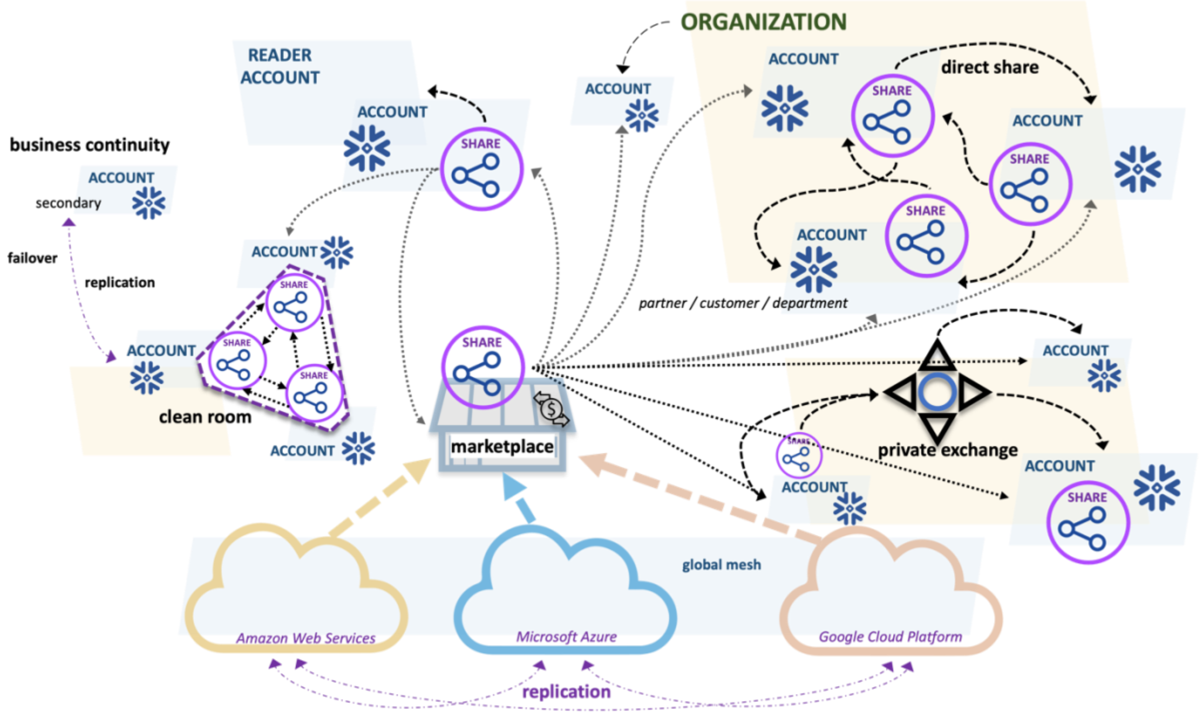

Figure 0-1 Possible through Snowflake's Global Mesh

Another game changer by Snowflake is its pioneering of **Secure Data Sharing** in the cloud that allows for **live real-time** access to your partner, customer, or cross-department data within the same CSP and region (and extended by using replication). You only share what you designate to share and your Snowflake experience is the same regardless of the underlying CSP architecture.

Snowflake provides three main offerings for Data Sharing, these are:

- **Direct Shares** – setting up a share between two snowflake accounts is as simple as executing a few SQL statements. Whether that other Snowflake account is within the same organization or is with a supplier or customer, you have the full power to set that up. This pattern has given rise to **Data Clean Rooms** for performing advertising-lead needs such as advertiser-publisher *overlap* or consumer *data enrichment* with the protection of not revealing plain text data between Snowflake accounts.
- **Private Data Exchange** – a "leader" Snowflake account is nominated to administer a private data exchange with your providers and consumers.
- **Data Marketplace** -to each Snowflake account, is available the Data Marketplace. This is the public exchange of data providers which another Snowflake account can consume from.

When a new Snowflake account is initiated, Snowflake provides a share to every account to monitor your own account's usage statistics, in addition to the information_schema available for each database. This share content is your audit of credit usage, clients used to access your Snowflake account, failed/successful attempts to log in, access history, data lineage, metering usage, impact analysis and so much more!

Finally, Snowflake provides the ability to perform **data replication** from one Snowflake account to asynchronously replicate to another Snowflake account and be able to support business continuity across regions *globally.*

Where can you sign up?

Snowflake has several editions and different ways of signing up. On any one of Azure, AWS or GCP you can sign up for Standard, Enterprise, **Business Critical** or **Virtual Private Cloud** (VPS) for either **on-demand** or contact Snowflake for a **capacity** plan and even provision a Snowflake account through the AWS marketplace to use your AWS credit on Snowflake (same concept is available on Azure or Google Cloud Platform). You do have the option to **try before your buy**, click through to Snowflake and sign up with a free account for a $400 credit to play with the account. You'll get a confirmation email within five minutes and start ingesting data and deploying databases, schema, tables and learn through the online tutorials or on-demand learning through Snow school.

What's more, Snowflake has a partner ecosystem where you can click through to a partner's Snowflake offering and once you have selected and executed the partner's setup routine you will have a "try before you buy" installation of that partner's tool with Snowflake, for the full list see, bit.ly/3DLY6eR, the list of partners covers:

- **Data integration**, ex. dbt Labs, Fivetran, Matillion, Kafka, Informatica, Qlik, Tableau, Wherescape, etc.
- **Business Intelligence**, ex. AtScale, Looker, Power BI, Microstrategy, Tableau, Thoughtspot, etc.
- **Machine Learning & Data Science**, ex. Alteryx, DataIku, DataRobot, SAS, Spark, etc.
- **Security & Governance**, ex. Alation, Collibra, Datadog, Immuta, Hunters, etc.
- **SQL Development & Management**, ex. SnowSQL, DataOps, DBeaver, ErWin, SqlDBM etc.
- **Native Programmatic Interfaces**, ex. JDBC, .Net, ODBC, Python, SQLAlchemy, etc.

Snowflake also provides a powerful web experience called Snowsight; complete with text auto-complete, dashboards and monitoring tools for your account, expect to see more and more features being added to this interface.

For a full comparison of Snowflake **Editions and Features** *see: bit.ly/3LLNdfO*
Snowflake security in the cloud, *see bit.ly/3KkBKDJ*

Figure 0-1 See you at Summit! https://www.snowflake.com/summit/

The views expressed in this article are that of my own, you should test implementation performance before committing to this implementation. The author provides no guarantees in this regard.