GOTC

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

OPEN SOURCE , OPEN WORLD

分布式数据库与存储专场

本期议题：打造云原生流式数据库

吴英骏 2021年08月01日

About Me

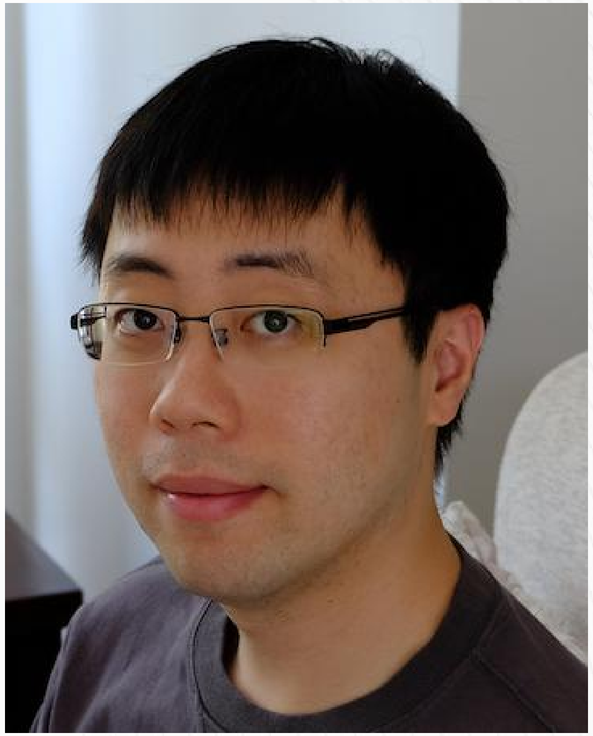
GOTC

Yingjun Wu (吴英骏)

- CEO of Singularity Data Inc.
- Former software engineer at AWS Redshift
- Former researcher at IBM Research - Almaden
- Ph.D., National University of Singapore
- Visiting Ph.D., Carnegie Mellon University

Huanchen Zhang

- Assistant professor at Tsinghua Yao class
- Former postdoc fellow at Snowflake Inc.
- Ph.D., Carnegie Mellon University
- Winner of Jim Gray Dissertation Award (2021)



Andy Pavlo

I am an Associate Professor of Databaseology in the Computer Science Department at Carnegie Mellon University. My research interest is in database management systems, specifically main memory systems, self-driving / autonomous architectures, transaction processing systems, and large-scale data analytics. At CMU, I am a member of the Database Group and the Parallel Data Laboratory. I am the co-founder and CEO of OtterTune.

[BIO] [CV]

Summer 2021 — OtterTune Database / ML Engineer Positions

Singularity Data Inc.

- 10+ full-time engineers with solid background from big techs
- Raised millions of U.S. dollars from top-tier VC firms
- Based in Beijing, Shanghai, and the San Francisco Bay Area

Starting from Redshift/Snowflake...

snowflake

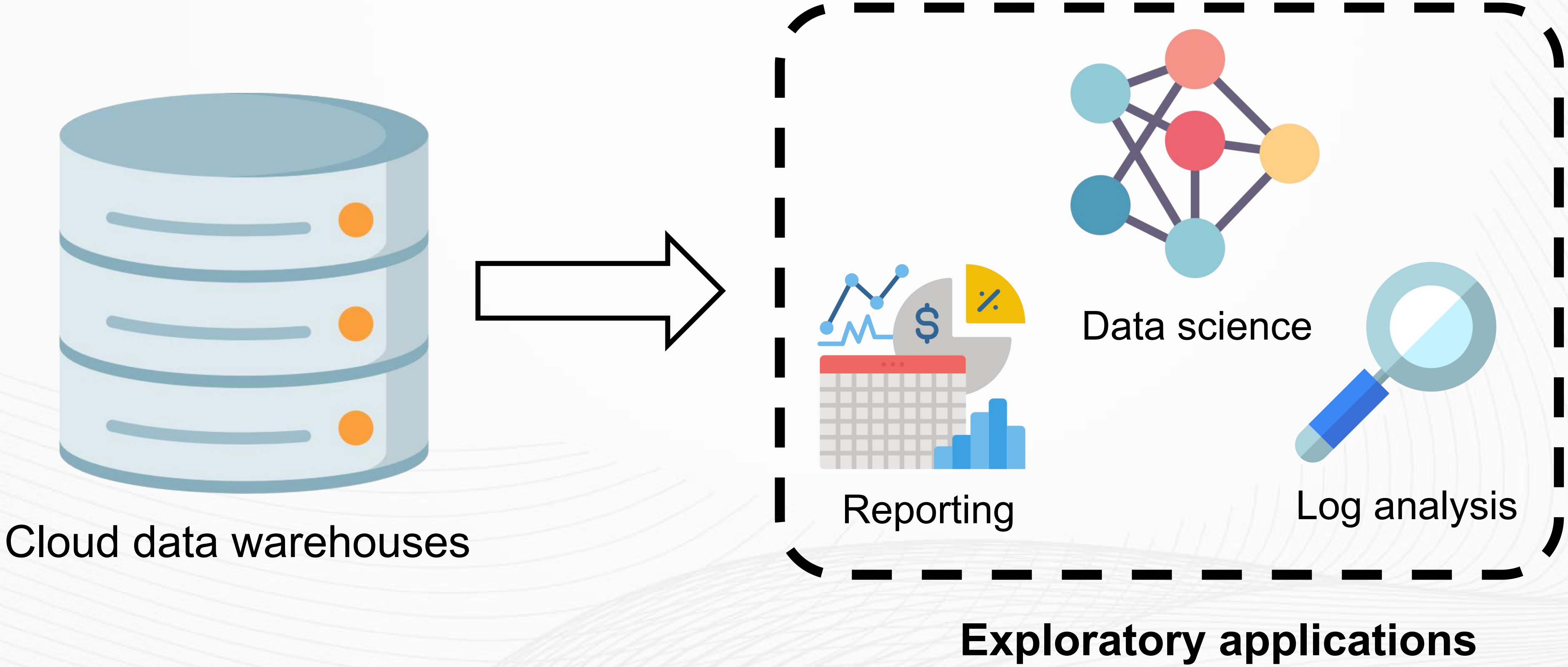
Biggest software IPO in the history
(once reached 80B+ market cap)

amazon REDSHIFT

5-10 times larger than Snowflake

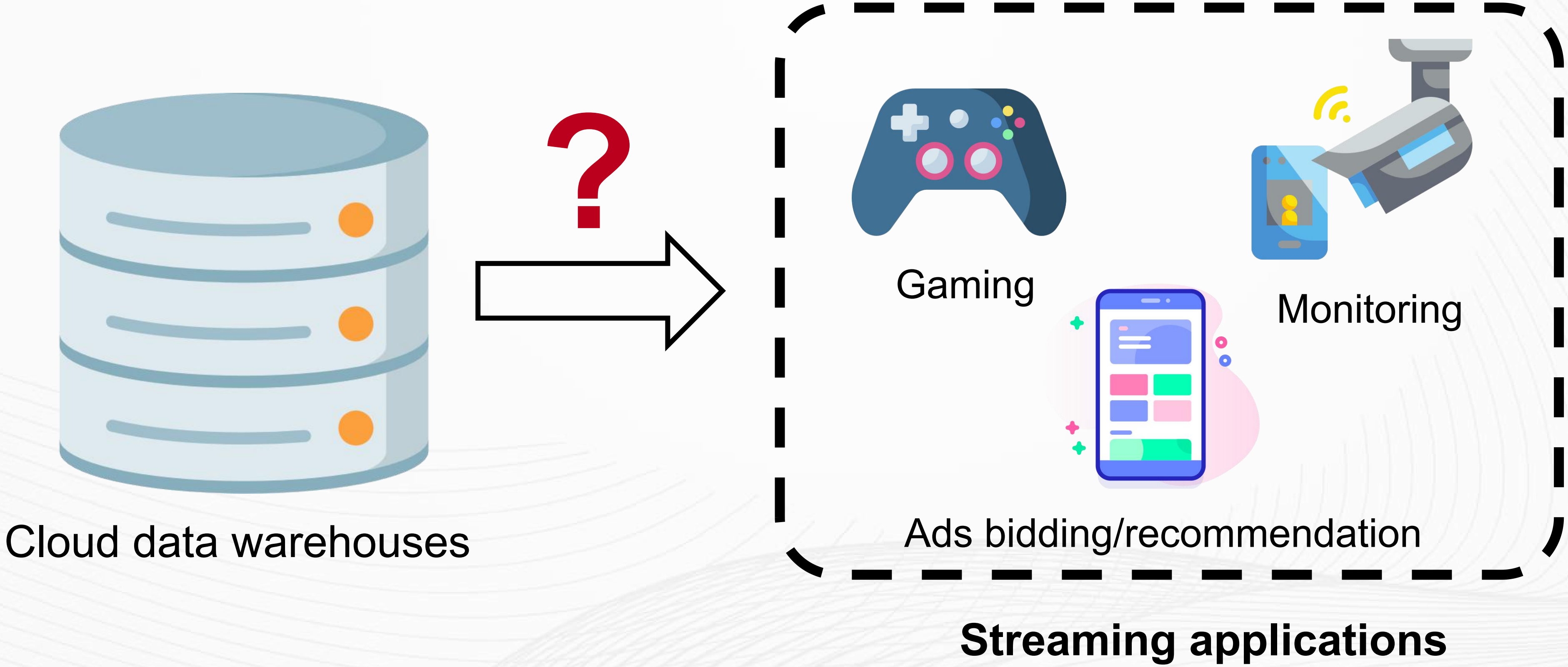
GOTC

Starting from Redshift/Snowflake...



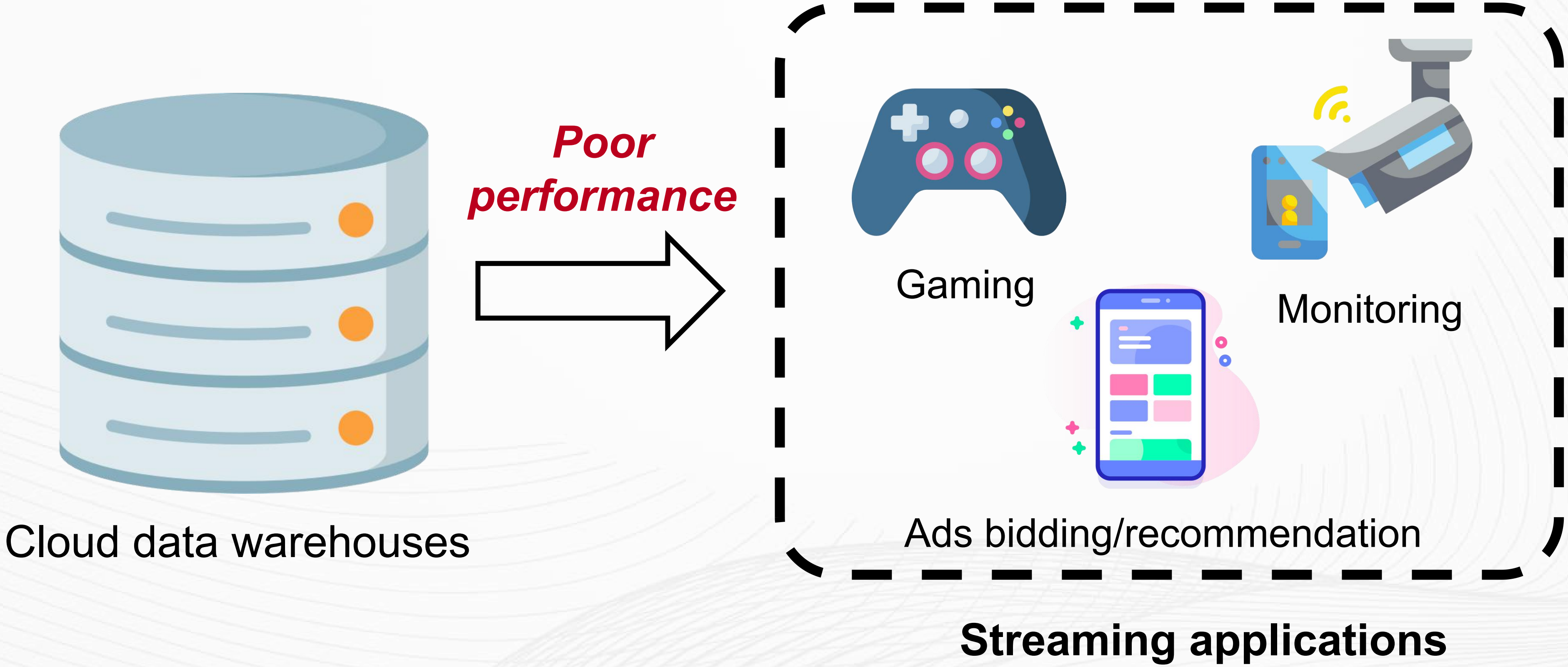
GOTC

But what if supporting streaming applications?



GOTC

But what if supporting streaming applications?



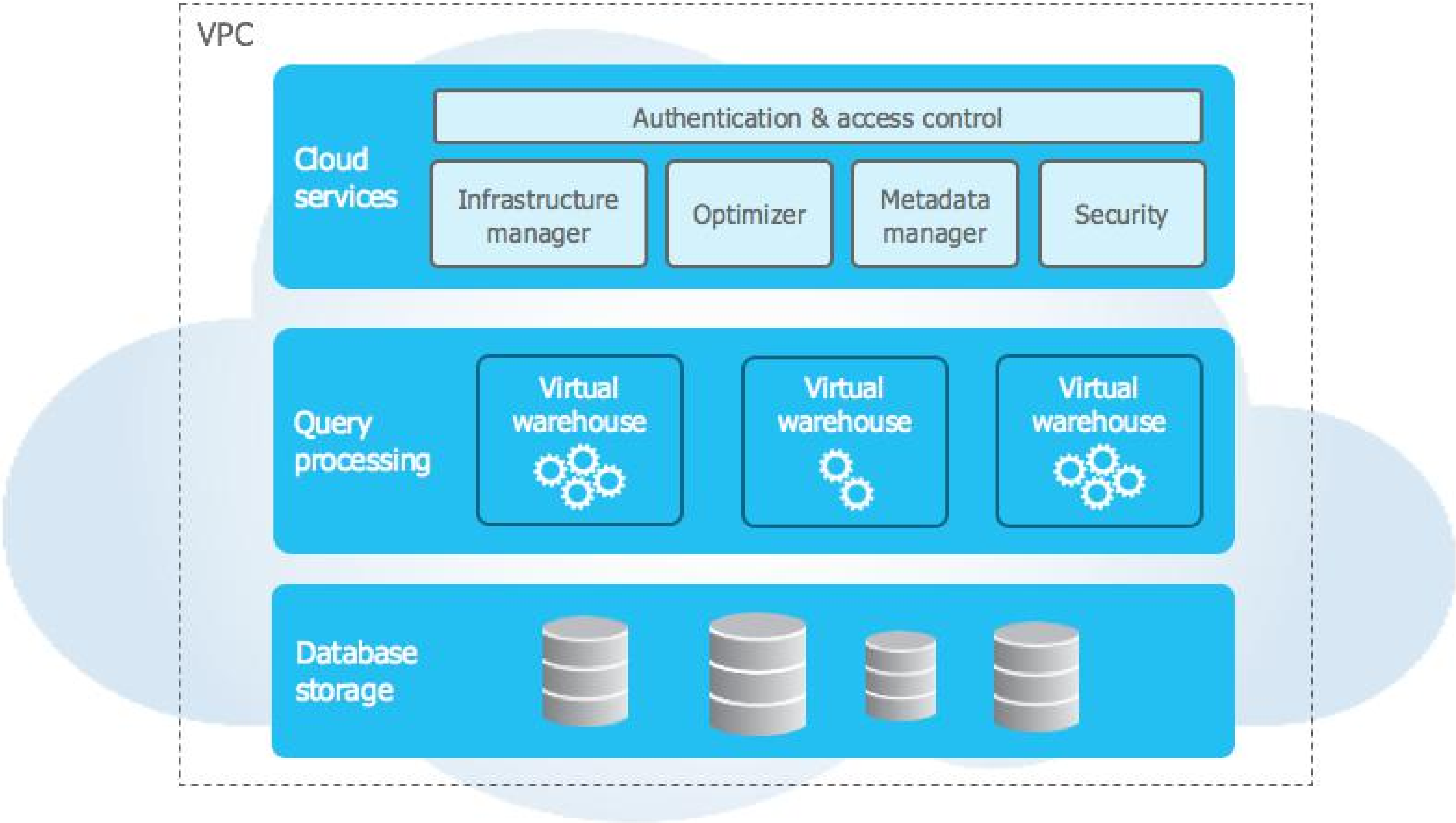
But what if supporting streaming applications?

- Understand the workload characteristics
- Revisit Redshift/Snowflake's architecture

Why streaming applications are different?

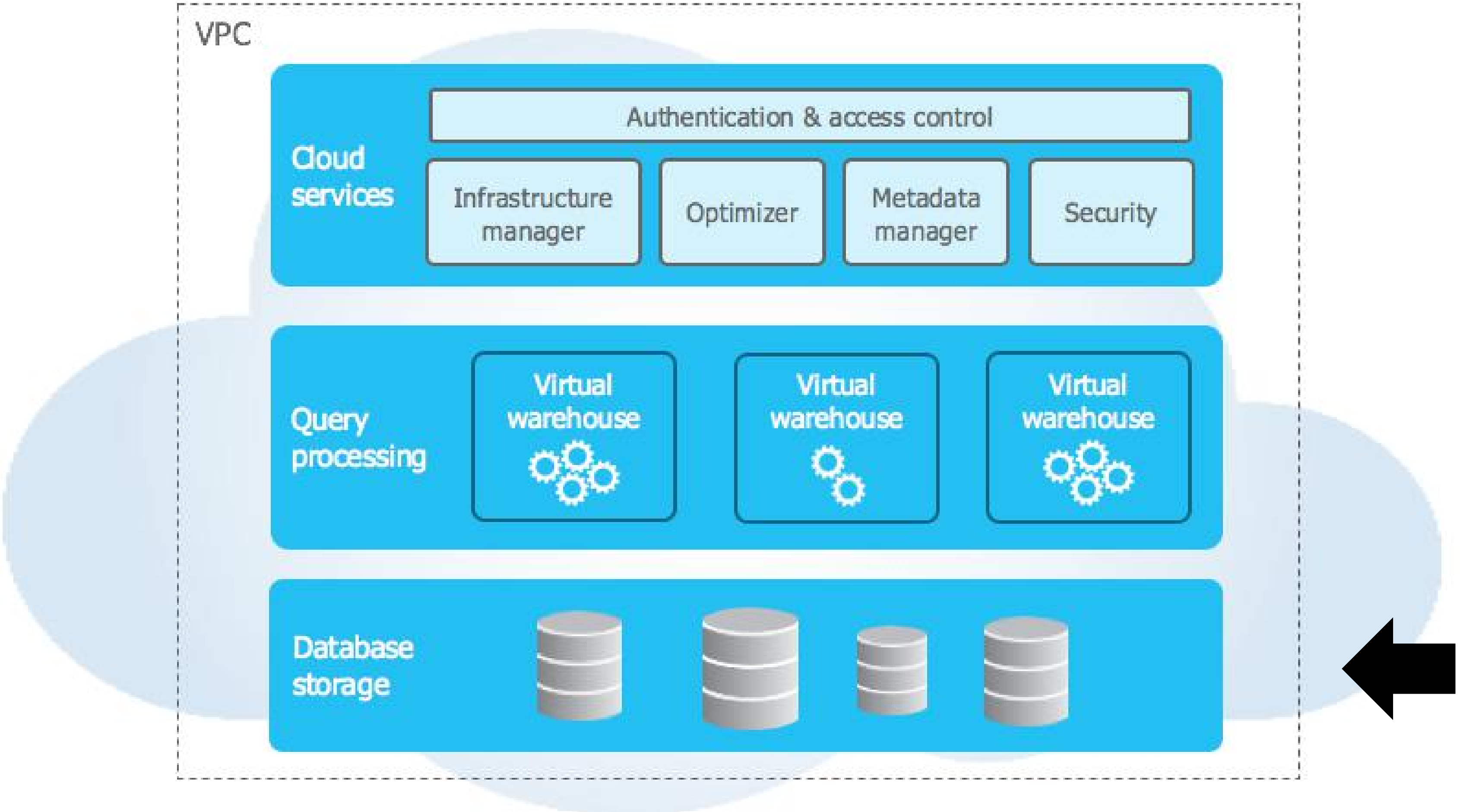
- High ingestion rate
- Low query latency
- Continuous computation

Cloud Data Warehouse Architecture



Cloud Data Warehouse Architecture

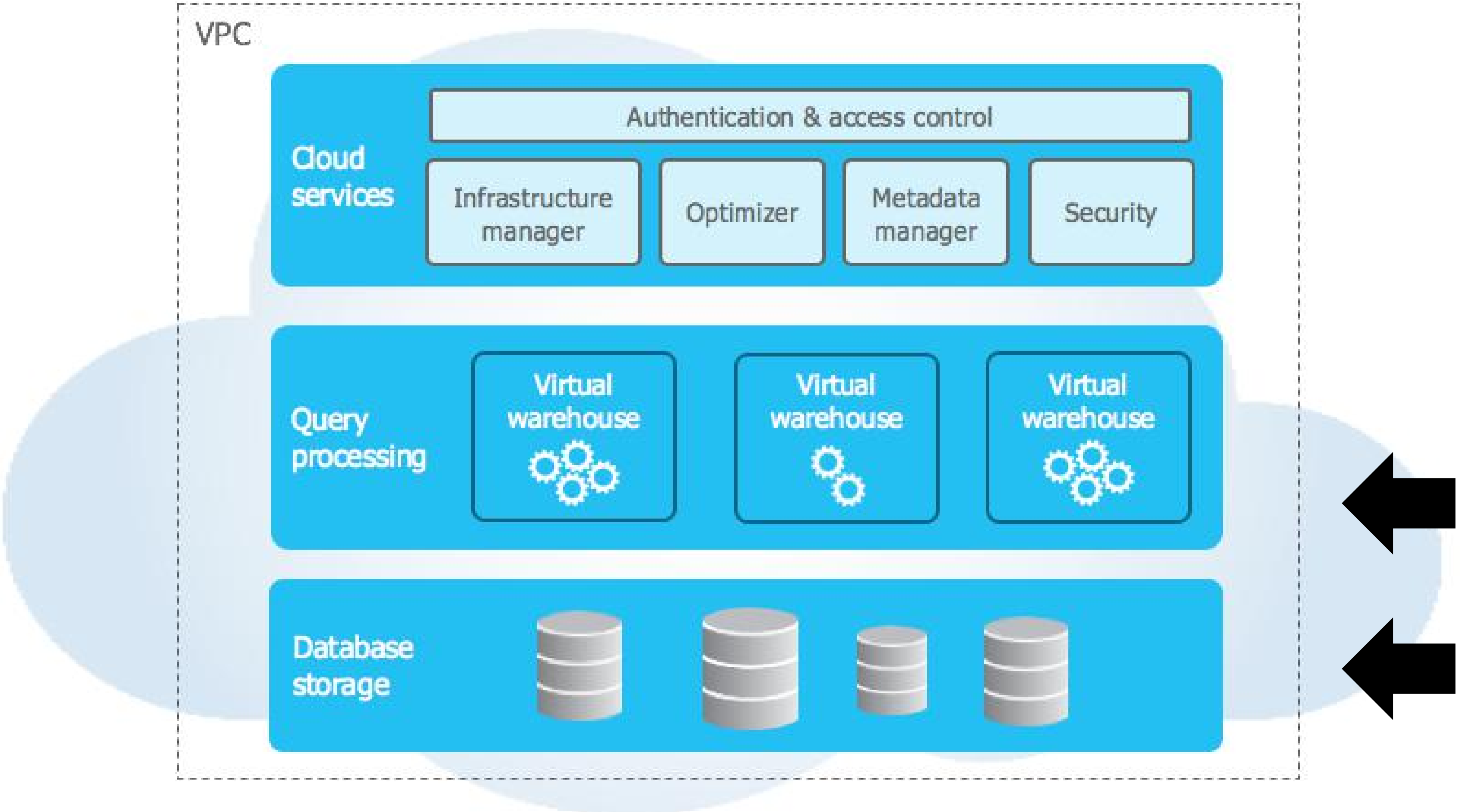
GOTC



High S3 commit latency results in low ingestion rate

Cloud Data Warehouse Architecture

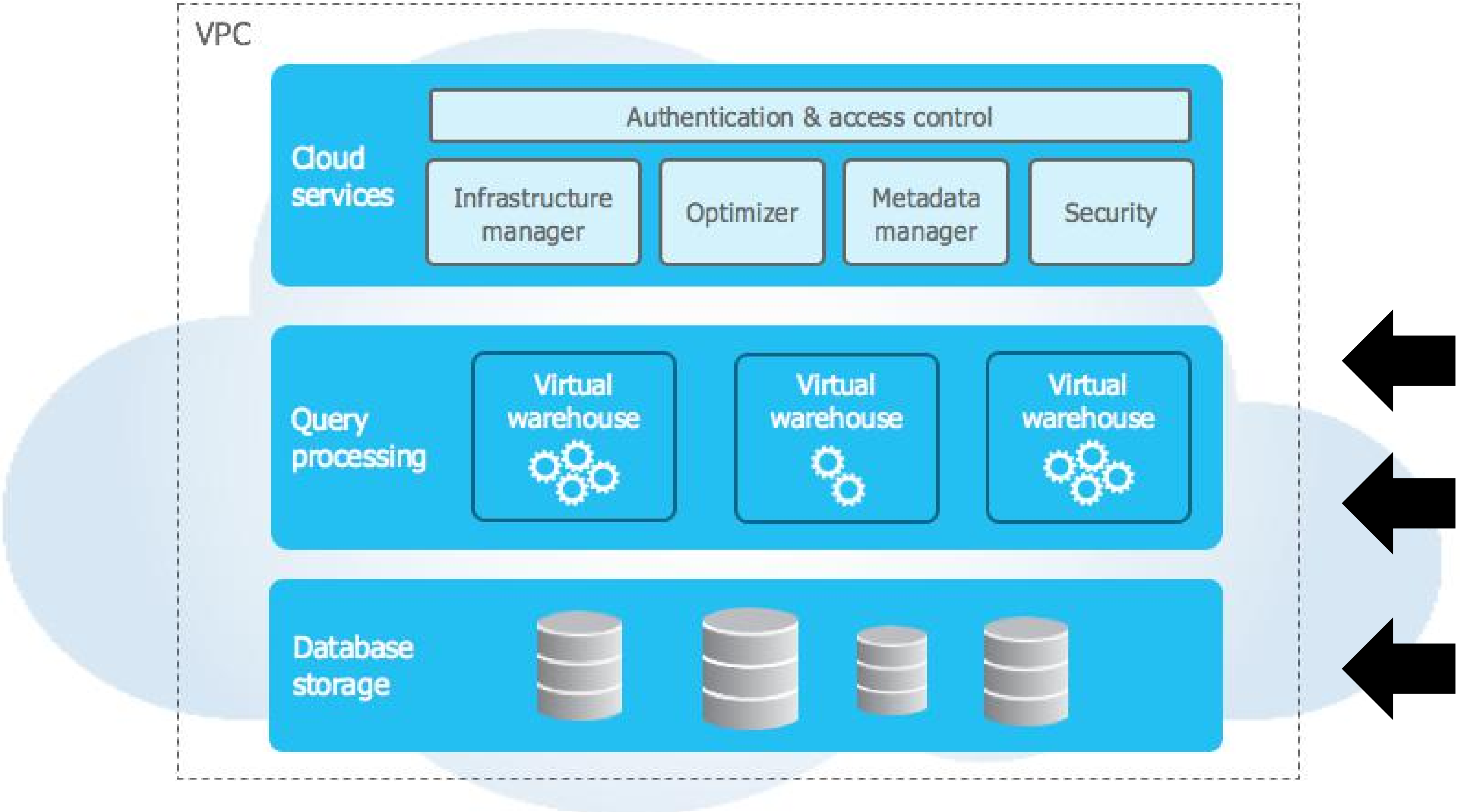
GOTC



Shared computation resource results in performance interference

High S3 commit latency results in low ingestion rate

Cloud Data Warehouse Architecture

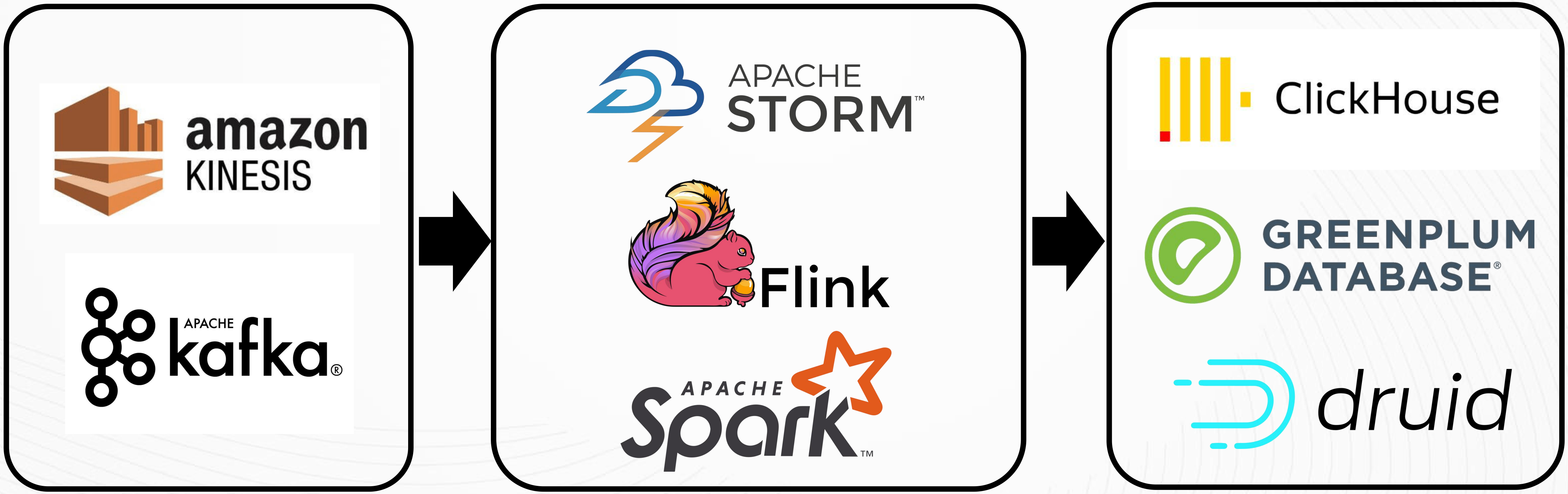


Full query computation results in high query latency

Shared computation resource results in performance interference

High S3 commit latency results in low ingestion rate

Existing Solution: Lambda Architecture



Is lambda architecture perfect?

- High deployment cost
- High maintenance cost
- High development cost
- High performance cost

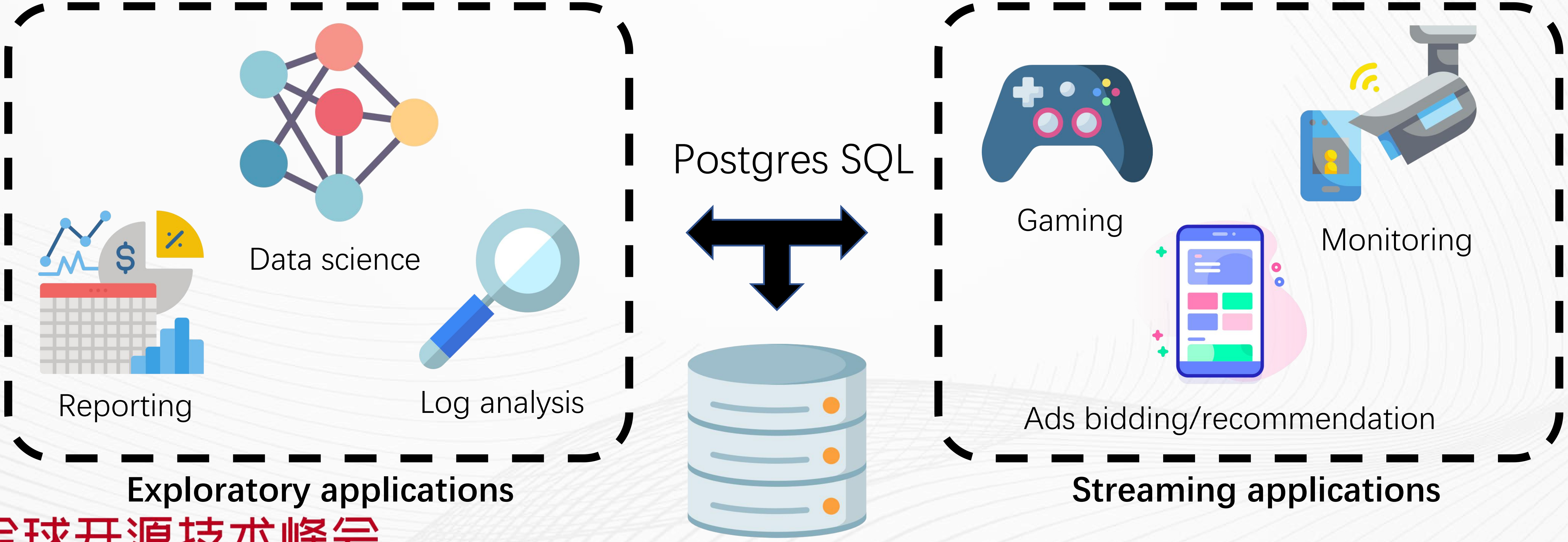
Let’s reinvent!

What will databases look like in the next 10 years?

- Single system
- SQL-oriented
- High performance

Cloud-native streaming database

- Elegantly/efficiently support exploratory and streaming applications



It is NOT a …

- OLTP database like Spanner, Aurora, or TiDB
  - Not purposely designed for handling transactional workloads
- data warehouse like Redshift or Snowflake
  - Not just supporting exploratory queries
- streaming processing engine like Flink, Storm, or Spark Streaming
  - Not just an engine that processes streaming data

Is Streaming Database a New Concept?

No!

- First introduced in year 2000s
- From *Human-active, DBMS-passive* to *DBMS-active, Human-passive*

Is Streaming D

GOTC

No!

- First introdu
- From *Huma* ... *uman-passive*

Monitoring Streams – A New Class of Data Management Applications

Don Carney
Brown University
dpc@cs.brown.edu

Uğur Çetintemel
Brown University
ugur@cs.brown.edu

Mitch Cherniack
Brandeis University
mfc@cs.brandeis.edu

Christian Convey
Brown University
cjc@cs.brown.edu

Sangdon Lee
Brown University
sdlee@cs.brown.edu

Greg Seidman
Brown University
gss@cs.brown.edu

Michael Stonebraker
M.I.T.
stonebraker@lcs.mit.edu

Nesime Tatbul
Brown University
tatbul@cs.brown.edu

Stan Zdonik
Brown University
sbz@cs.brown.edu

Abstract

This paper introduces monitoring applications, which we will show differ substantially from conventional business data processing. The fact that a software system must process and react to continual inputs from many sources (e.g., sensors) rather than from human operators requires one to rethink the fundamental architecture of a DBMS for this application area. In this paper, we present Aurora, a new DBMS that is currently under construction at Brandeis University, Brown University, and M.I.T. We describe the basic system architecture, a stream-oriented set of operators, optimization tactics, and support for real-time operation.

1 Introduction

Traditional DBMSs have been oriented toward business data processing, and consequently are designed to address the needs of these applications. First, they have assumed that the DBMS is a passive repository storing a large collection of data elements and that humans initiate queries and transactions on this repository. We call this a *Human-Active, DBMS-Passive* (*HADP*) model. Second, they have assumed that the current state of the data is the only thing that is important. Hence, current values of data elements are easy to obtain, while previous values can only be found torturously by decoding the DBMS log. The third assumption is that triggers and alerters are second-class

require no real-time services.

There is a substantial class of applications where all five assumptions are problematic. Monitoring applications are applications that monitor continuous streams of data. This class of applications includes military applications that monitor readings from sensors worn by soldiers (e.g., blood pressure, heart rate, position), financial analysis applications that monitor streams of stock data reported from various stock exchanges, and tracking applications that monitor the locations of large numbers of objects for which they are responsible (e.g., audio-visual departments that must monitor the location of borrowed equipment). Because of the high volume of monitored data and the query requirements for these applications, monitoring applications would benefit from DBMS support. Existing DBMS systems, however, are ill suited for such applications since they target business applications.

First, monitoring applications get their data from external sources (e.g., sensors) rather than from humans issuing transactions. The role of the DBMS in this context is to alert humans when abnormal activity is detected. This is a *DBMS-Active, Human-Passive* (*DAHP*) model.

Second, monitoring applications require data management that extends over some history of values reported in a stream, and not just over the most recently reported values. Consider a monitoring application that tracks the location of items of interest, such as overhead transparency projectors and laptop computers, using electronic property stickers attached to the objects. Ceiling-mounted sensors inside a building and the GPS system in the open air generate large volumes of location data. If a reserved overhead projector is not in its proper location,

Is Streaming Database a New Concept?

GOTC

Research prototypes developed in early 2000s

- Aurora/Borealis/STREAM

Commercial products released in late 2000s

- Microsoft StreamInsight

History Repeats Itself!

	Traditional era	Big data era	Cloud era
OLTP Databases	ORACLE	cassandra	CockroachDB
OLAP Databases			
Streaming Databases			

History Repeats Itself!

	Traditional era	Big data era	Cloud era
OLTP Databases	ORACLE®	cassandra	CockroachDB
OLAP Databases	TERADATA	hadoop	snowflake
Streaming Databases			

History Repeats Itself!

GOTC

	Traditional era	Big data era	Cloud era
OLTP Databases	ORACLE®	cassandra	CockroachDB
OLAP Databases	TERADATA	hadoop	snowflake
Streaming Databases	Microsoft® SQL Server® StreamInsight	Flink	SINGULARITY

Summary

Cloud-native streaming database is the service we bet that can support both ***streaming*** and ***exploratory*** applications elegantly and efficiently.

We Are Hiring! Let’s connect and chat!

THANKS