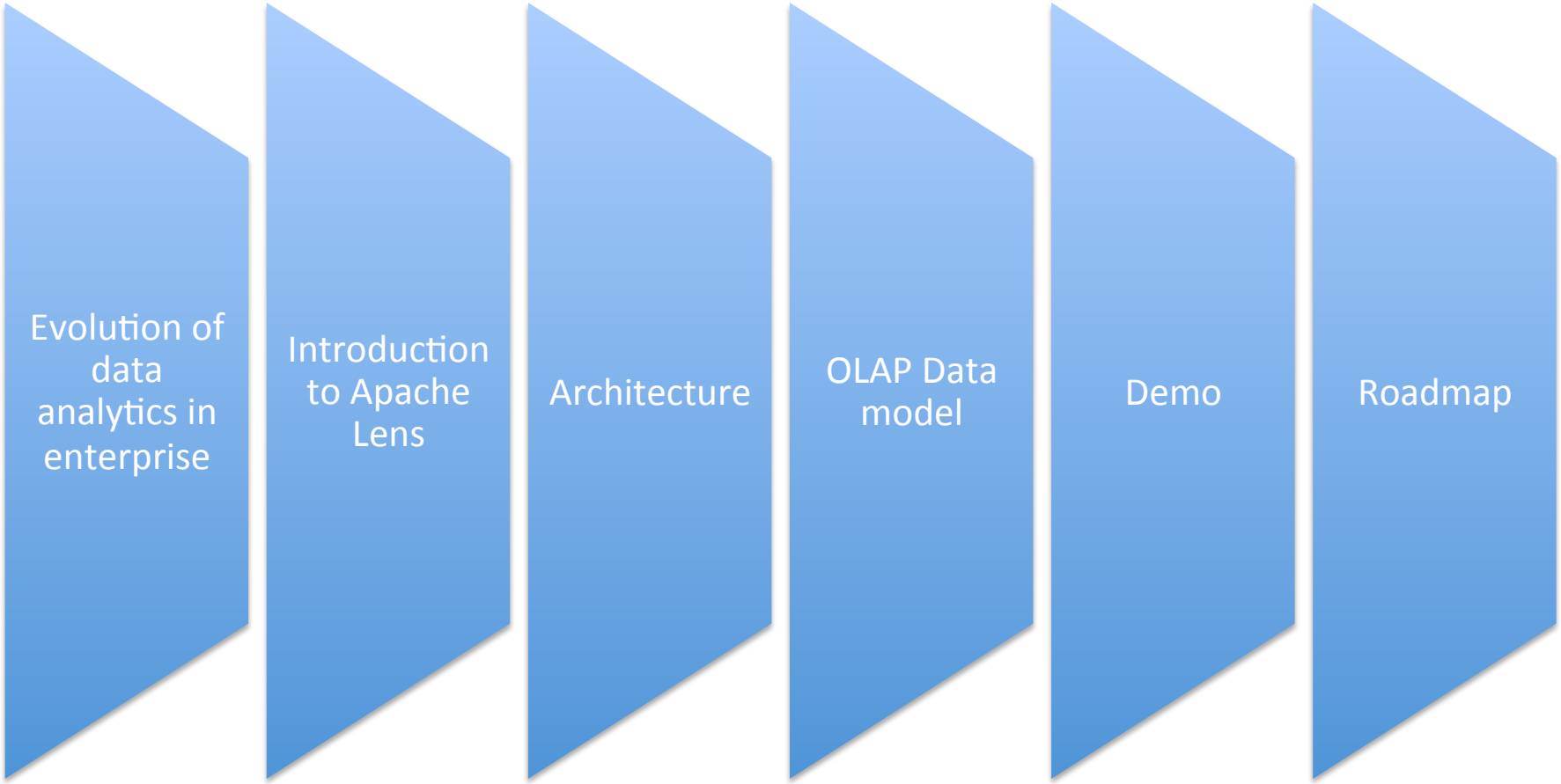
Apache Lens

Cut data analytics silos in your enterprise

Sharad Agarwal & Amareshwari Sriramadasu

Agenda


Evolution of data analytics in enterprise
Introduction to Apache Lens
Architecture
OLAP Data model
Demo
Roadmap

Reporting ware house : Generation 1: RDBMS

- **Reporting data in RDBMS**
- **Aggregations/materialized views in DB**
- **~ 1 TB**

Generation 1 :Challenges

- **Data Scale: Loading of data taking ~ 24 hrs**
- **Analysis only upto 3 dimensions**
- **Heavy queries stalling other user queries**
- **Unable to move fast with new reporting requirements**

Reporting warehouse: Generation 2: Columnar DB

- **Small and summarized data in Columnar Database**
- **Rich Dashboards**

Generation 2: Challenges

- **Scalability challenges with data growth**
- **Expensive to grow the capacity on columnar DB**
- **Data modelling and ETL cycles are long**
- **Limited Analytical flexibility**

Generation 3: Columnar DB + Hadoop

- **Small and summarized data in Columnar Database (~10 TB)**
- **Rich Dashboards**
- **Granular data in Hadoop (100s of TB)**
- **Adhoc analysis**

Generation 3: Challenges

- **Maintaining two lines of independent data warehousing systems**
- **Data discrepancies**
- **Schema management**
- **Learning curve for Users**
- **Duplicate datasets**
- **Inefficient Utilization**

Apache Lens (formerly Grill)

- ***Platform to enable multi-dimensional queries in a unified way over datasets stored in multiple warehouses***
  - **OLAP Cube abstraction**
- **Data discovery by providing single metadata layer**
- **Unified access to data by integrating Hive with other traditional warehouses**

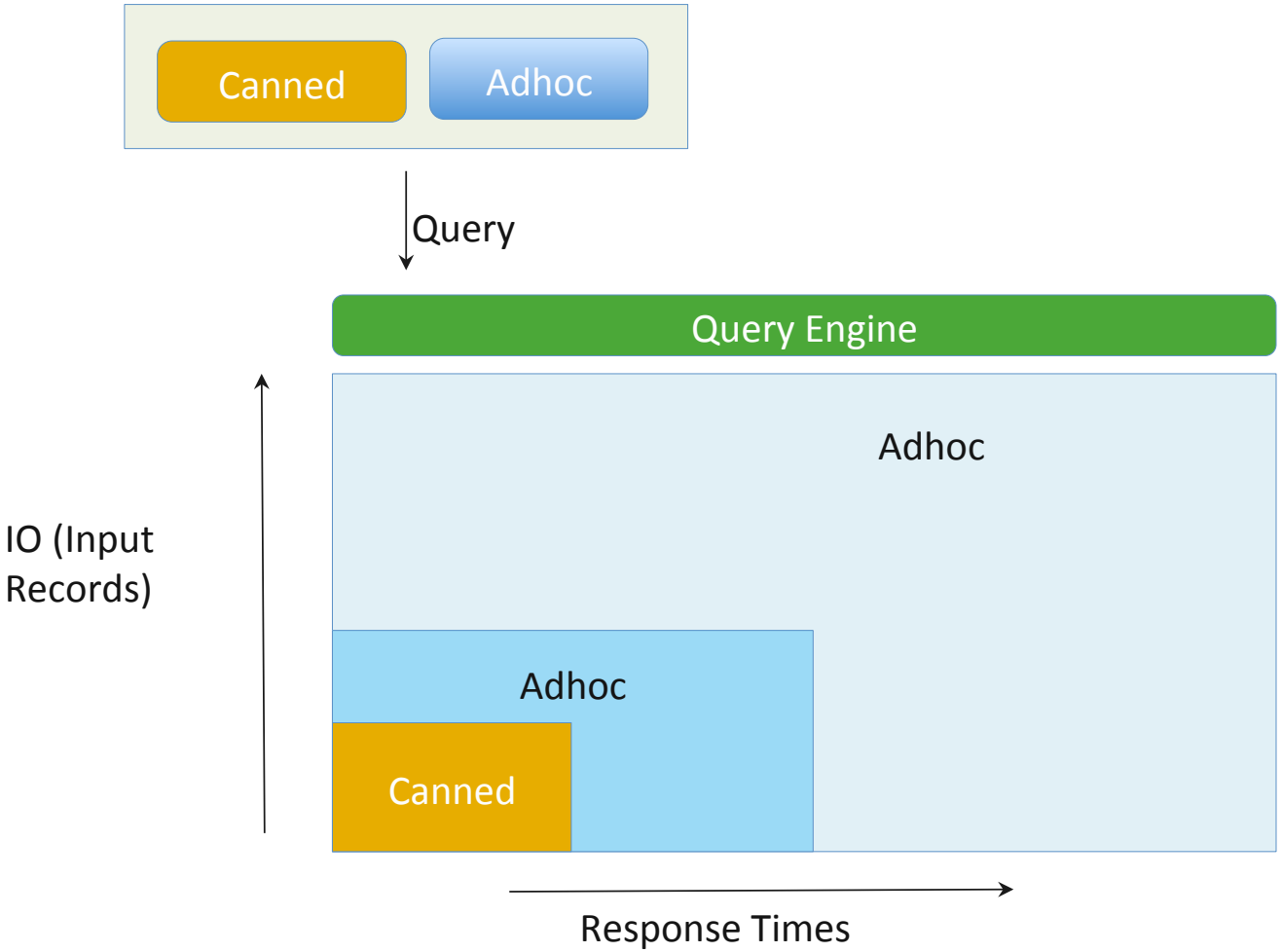
Apache Lens

- **Queries get pushed to where data resides**
- **Central Catalog management: All applications talk same language**
- **Query analytics for optimizing hot datasets**
- **Workload based experimentation with newer systems: AWS Redshift, Apache Spark, Apache Tez**

Analytics Use cases

- Reporting queries
- Adhoc queries
  - Interactive/Batch queries
- Scheduled queries
- Infer insights through ML algorithms

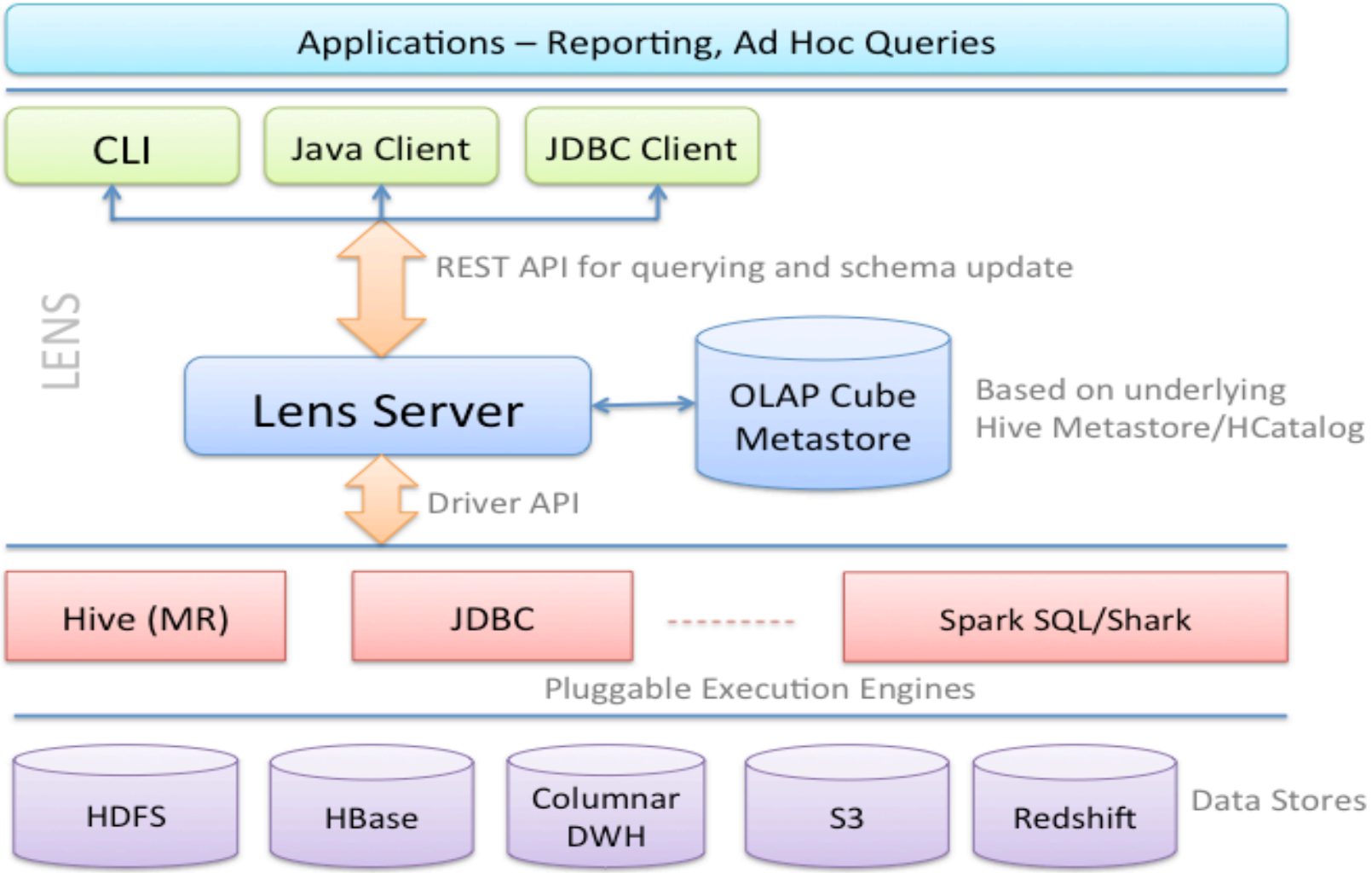
Why both Hadoop and traditional warehouse?



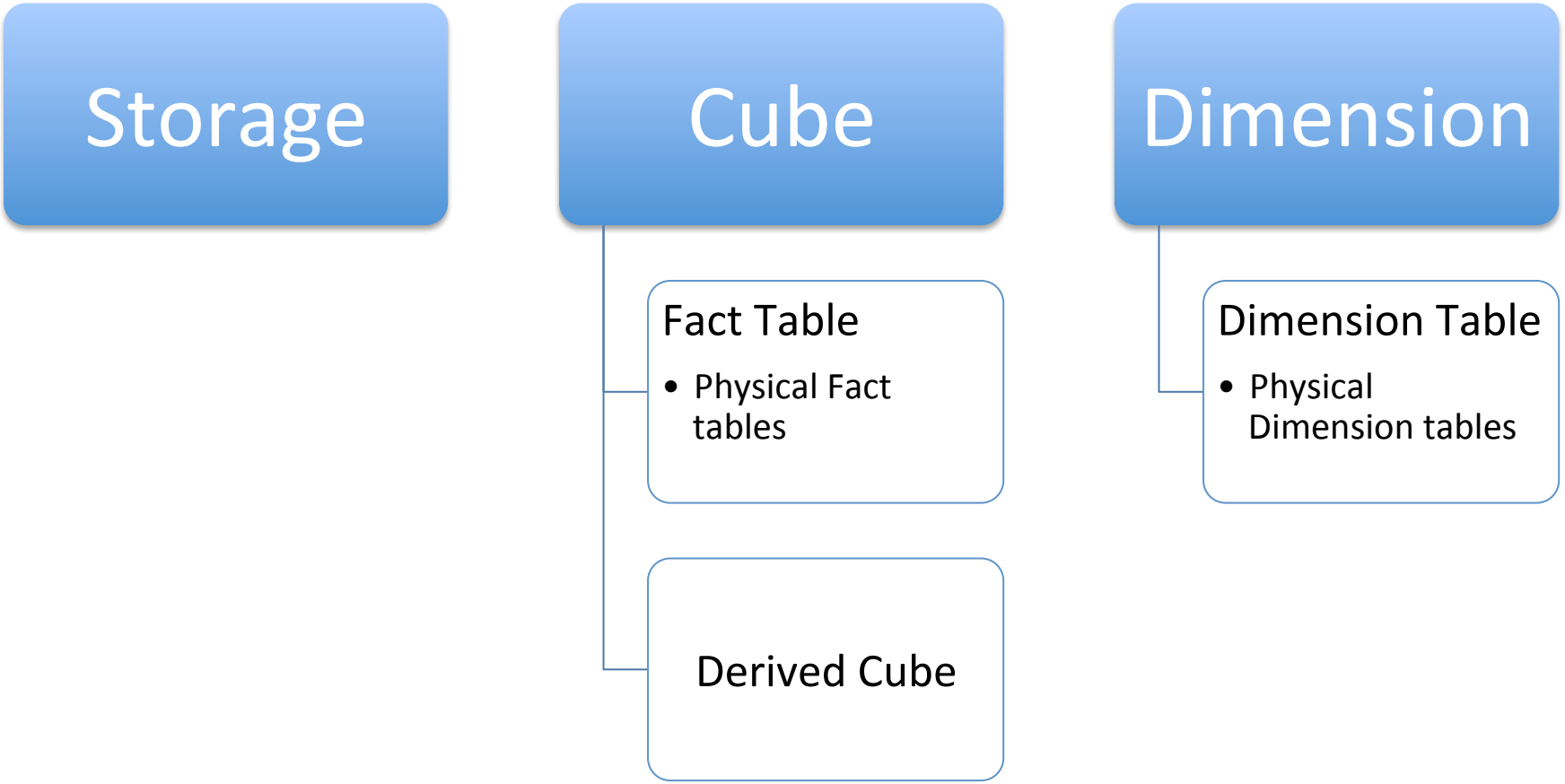
Canned queries are mostly Interactive

Adhoc queries can be Interactive or batch depending on the data volumes and query complexity

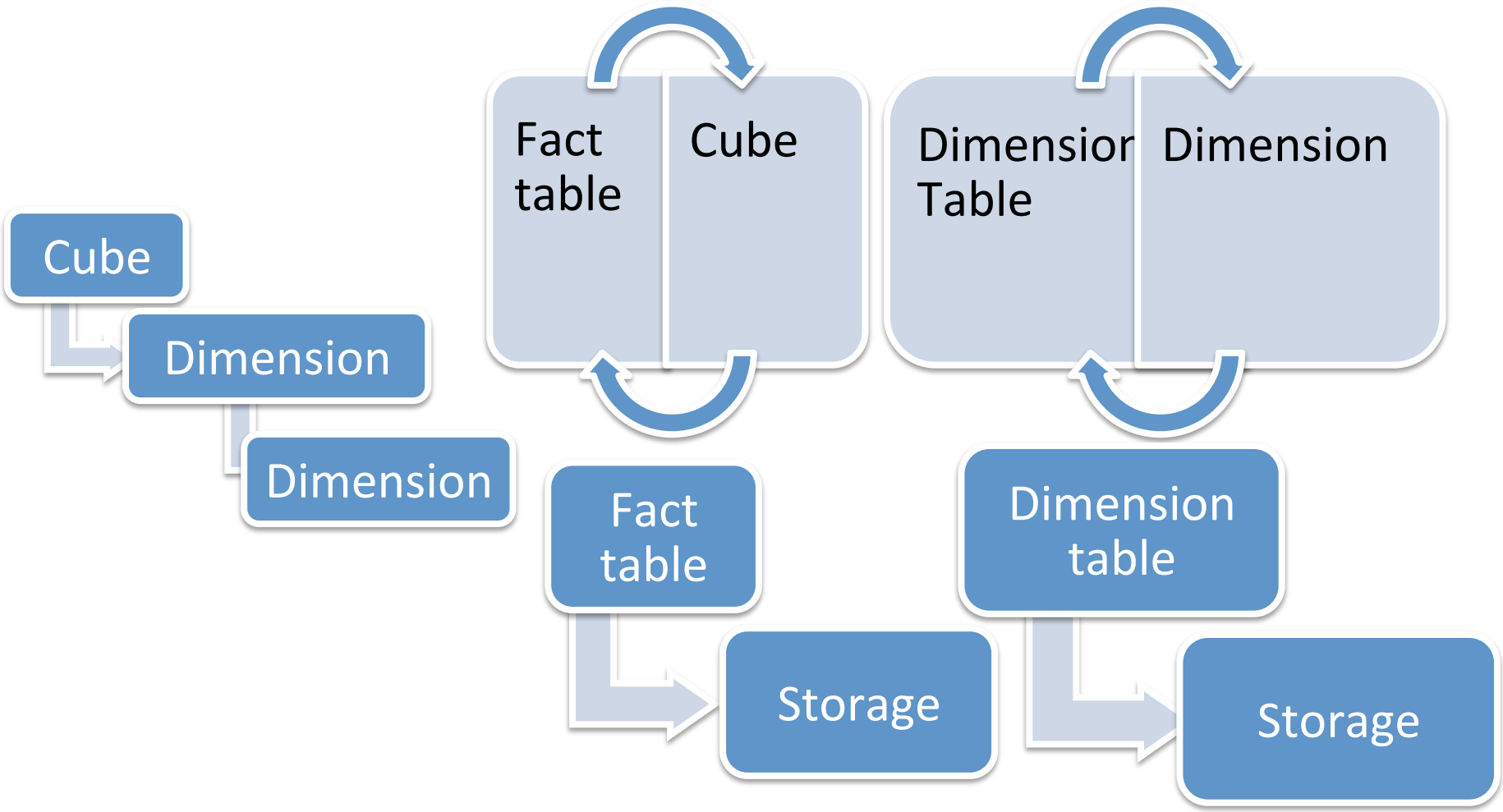
Lens Architecture



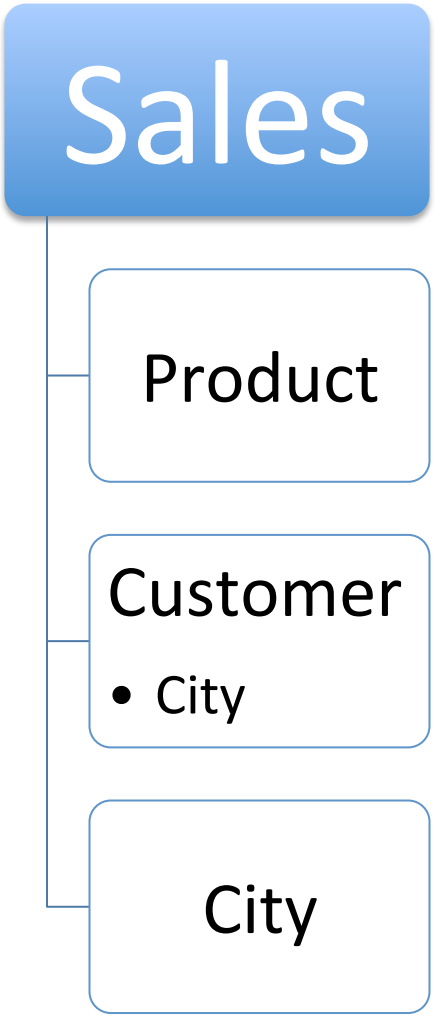
OLAP Data model



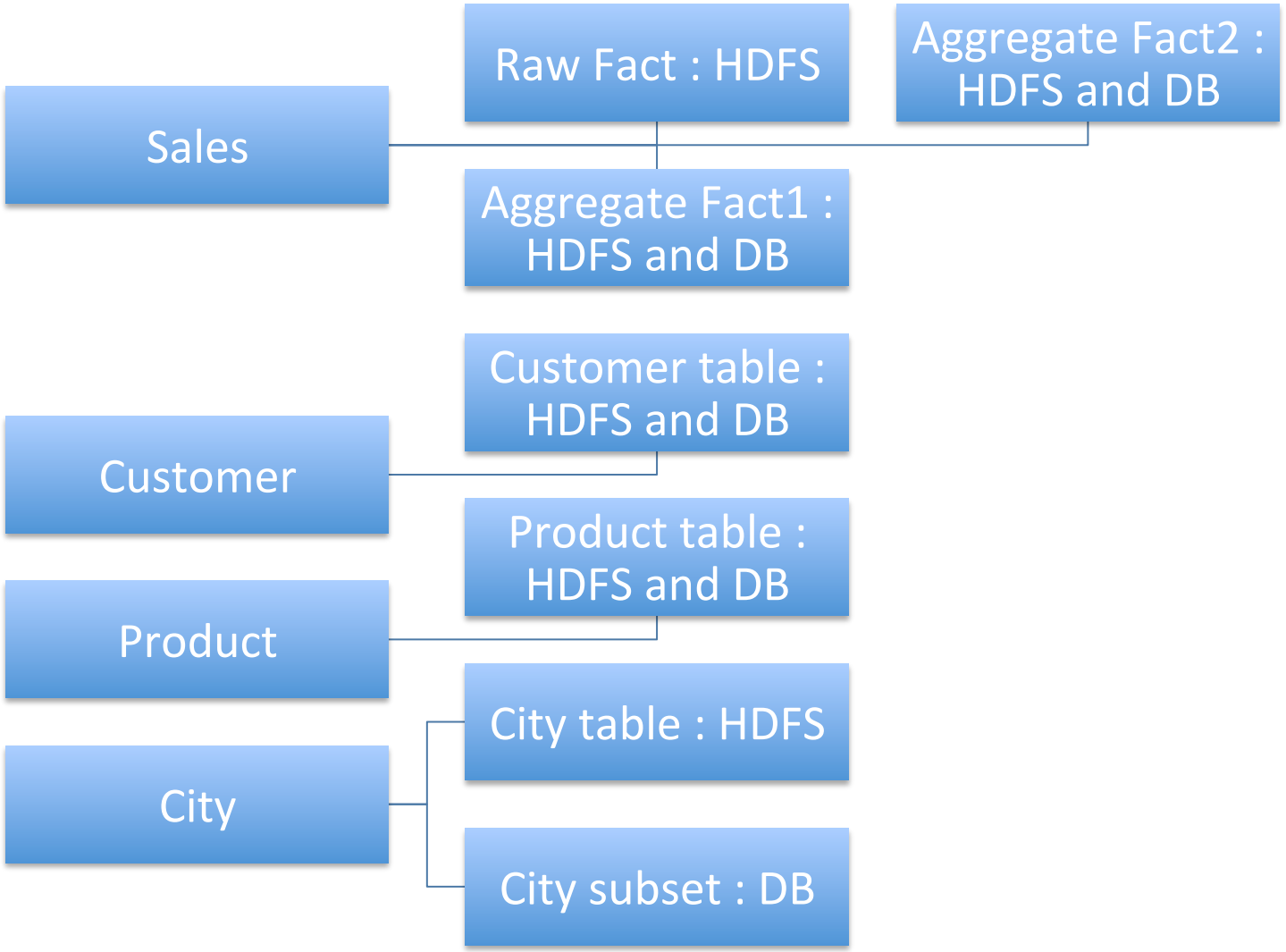
Data model - Relationships



Demo : Example data model

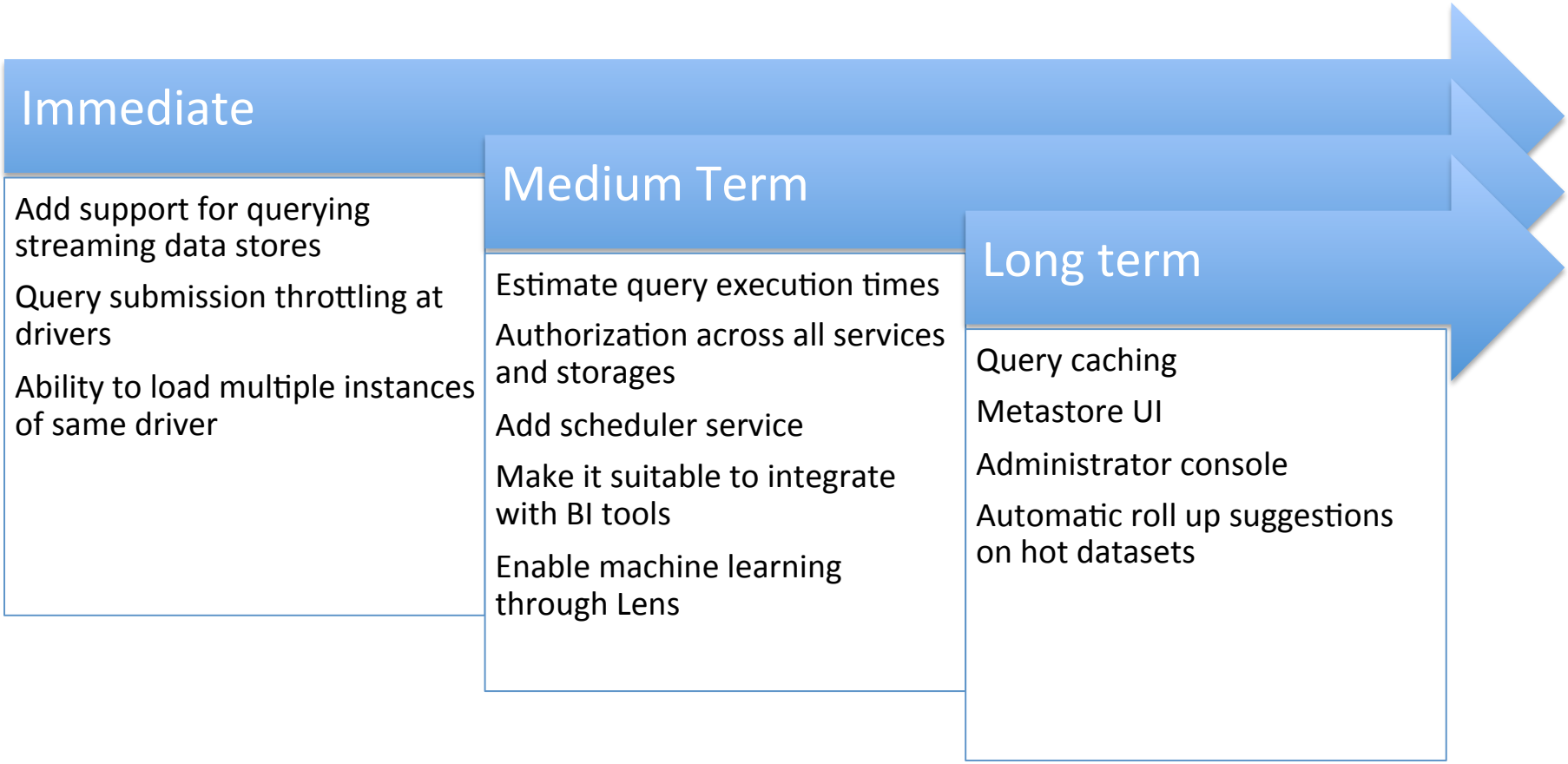


Demo : Example physical data model



Demo

Roadmap


Immediate
Add support for querying streaming data stores
Query submission throttling at drivers
Ability to load multiple instances of same driver
Medium Term
Estimate query execution times
Authorization across all services and storages
Add scheduler service
Make it suitable to integrate with BI tools
Enable machine learning through Lens
Long term
Query caching
Metastore UI
Administrator console
Automatic roll up suggestions on hot datasets

Explorations

- Enable HiveDriver on Tez/Spark
- Explore Zeppelin for web front end
- Newer drivers : Elastic search driver, Druid driver

Stay Involved

Web site

- http://lens.incubator.apache.org/

Source repo

- https://git-wip-us.apache.org/repos/asf/incubator-lens.git

Source repo for Hive

- https://github.com/InMobi/hive

Mailing lists

- dev@lens.incubator.apache.org
- user@lens.incubator.apache.org

Thank You!

- Questions?