



GBASE[®]

关系模型与非关系模型的融合：技术实践与展望

武新 博士

CTO / 南大通用数据技术股份有限公司

目录

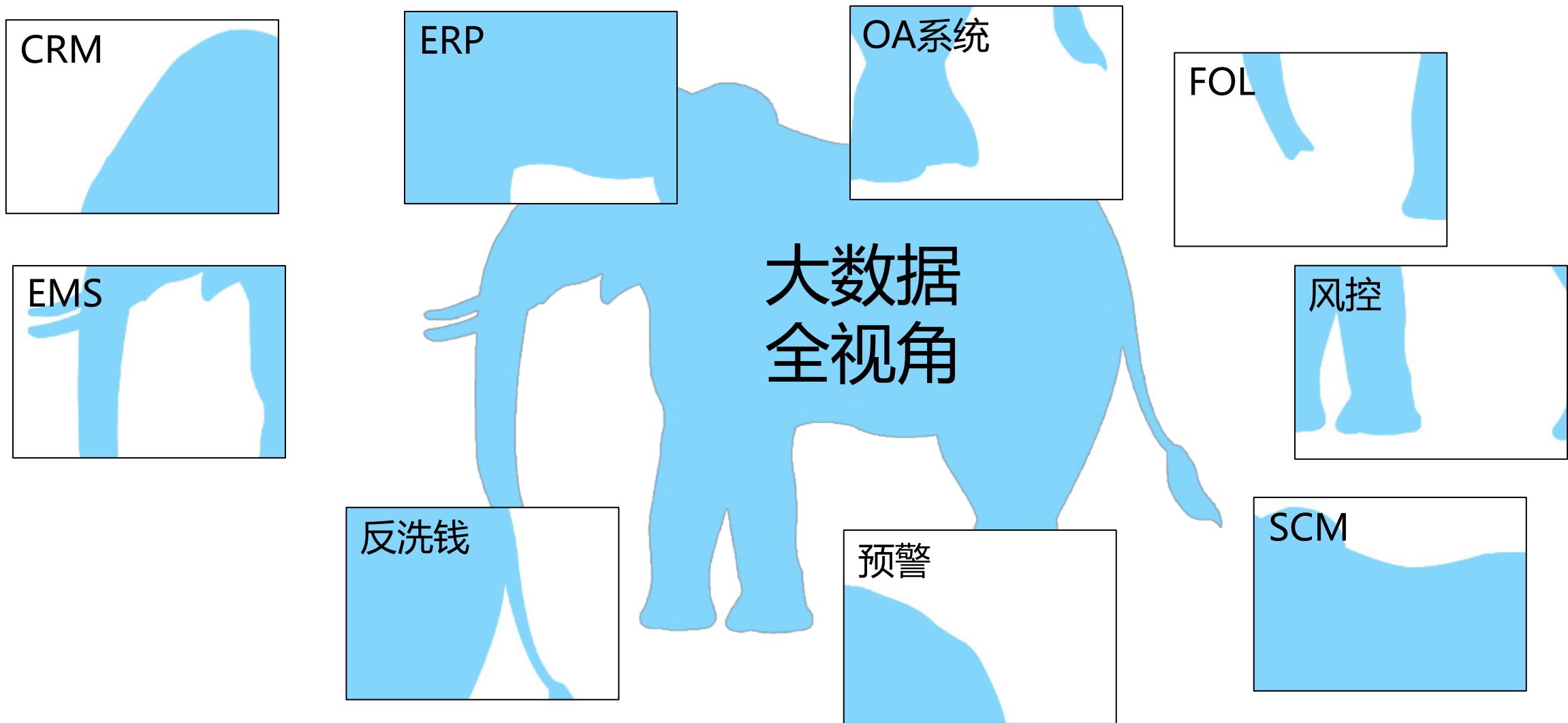
- 大数据需求与IT技术架构演变

- SQL（关系）与Hadoop（非关系）的融合之道

- GBase UP 介绍及应用案例

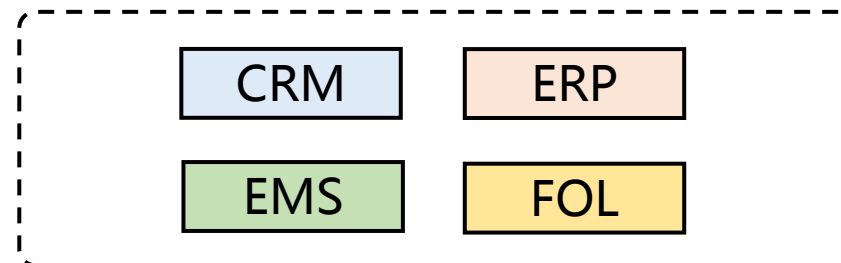
- 总结与展望

从数据孤岛到大数据分析

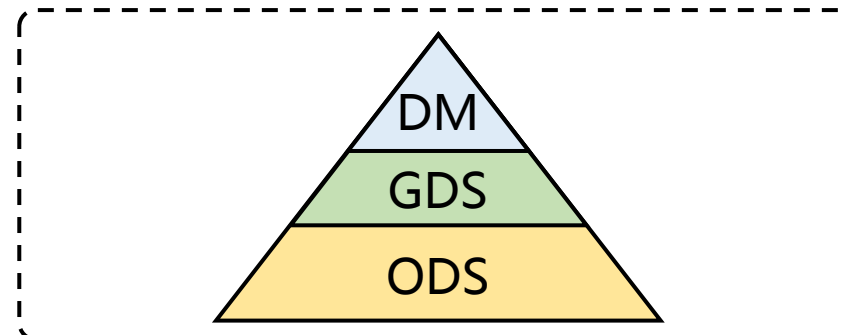


企业数据处理面临的“三座大山”

过去：业务数据孤岛



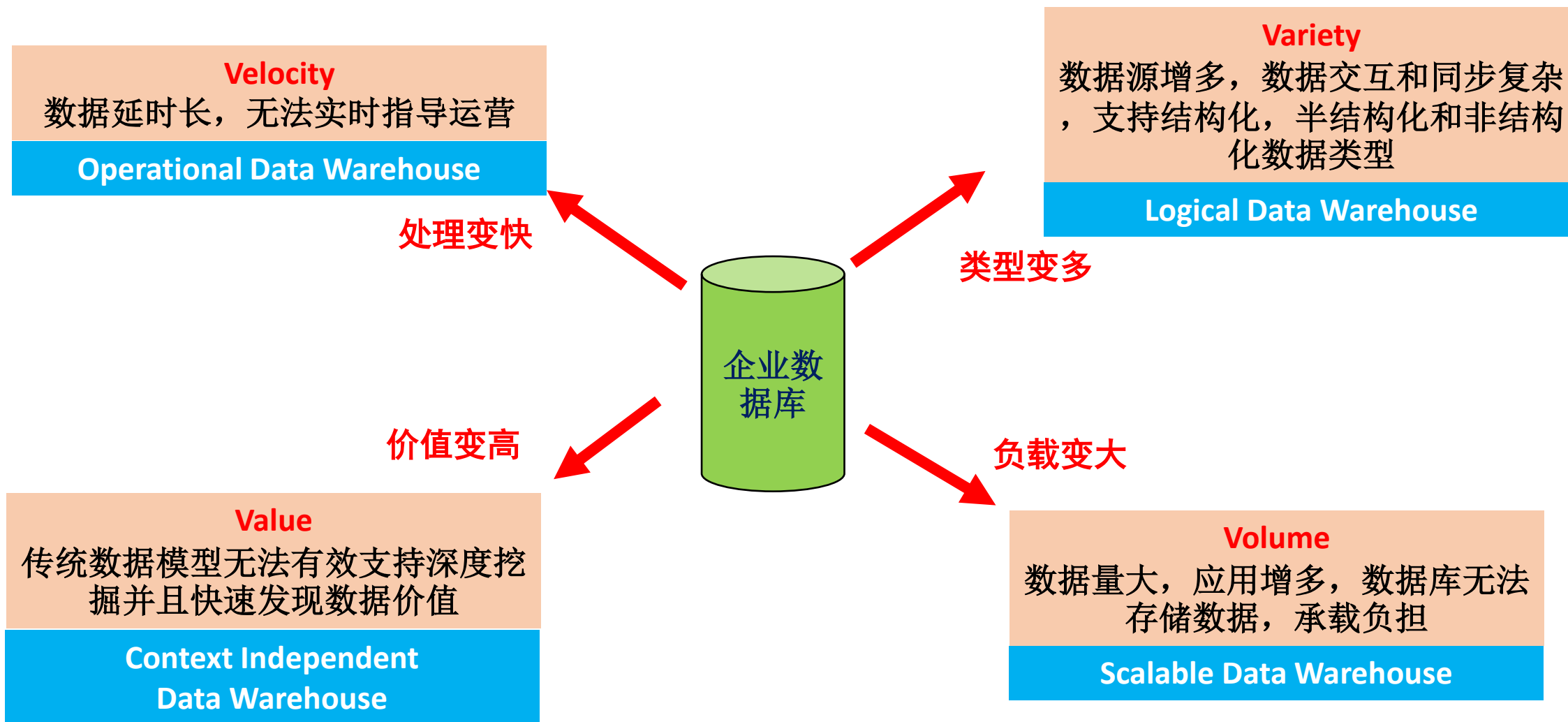
现在：MPP数据库解决了结构化业务数据的分析问题



未来：大数据的挑战，全数据、多模型、异构、新技术...



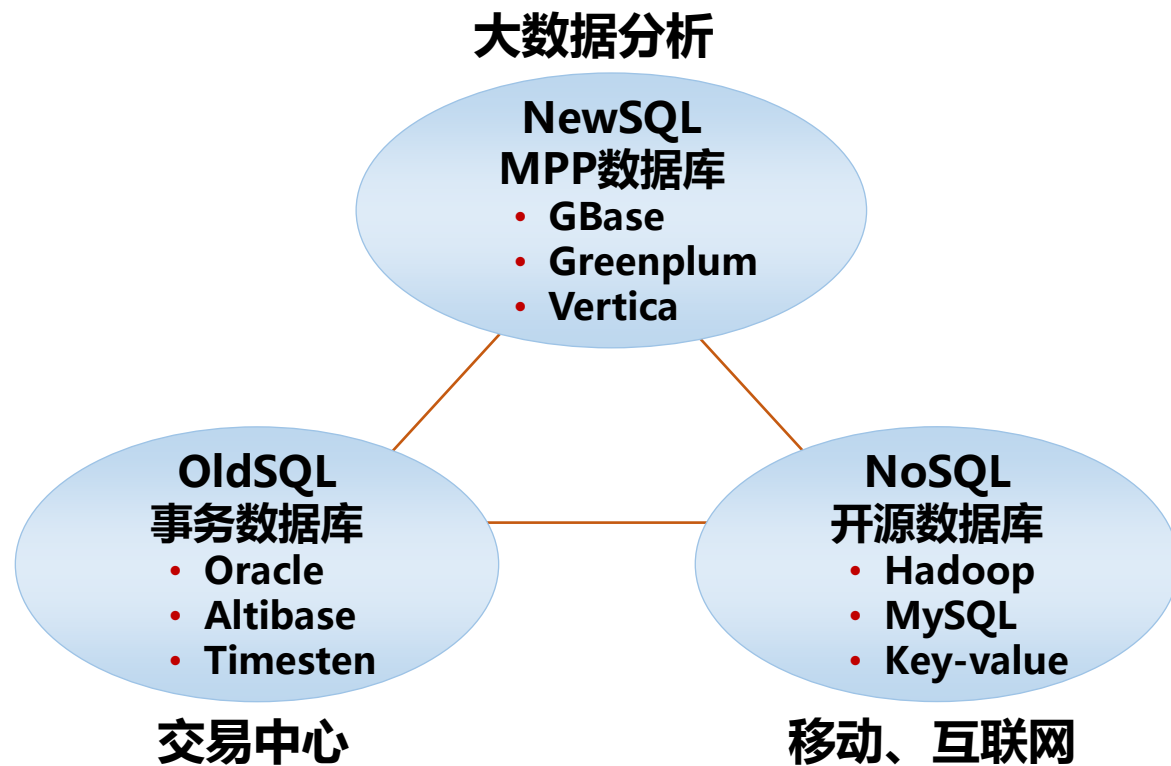
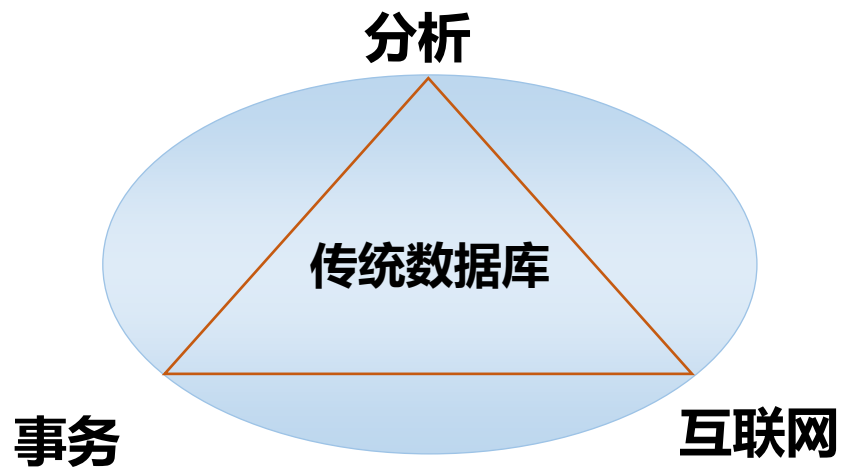
Gartner预测：企业数据库面临的4大挑战



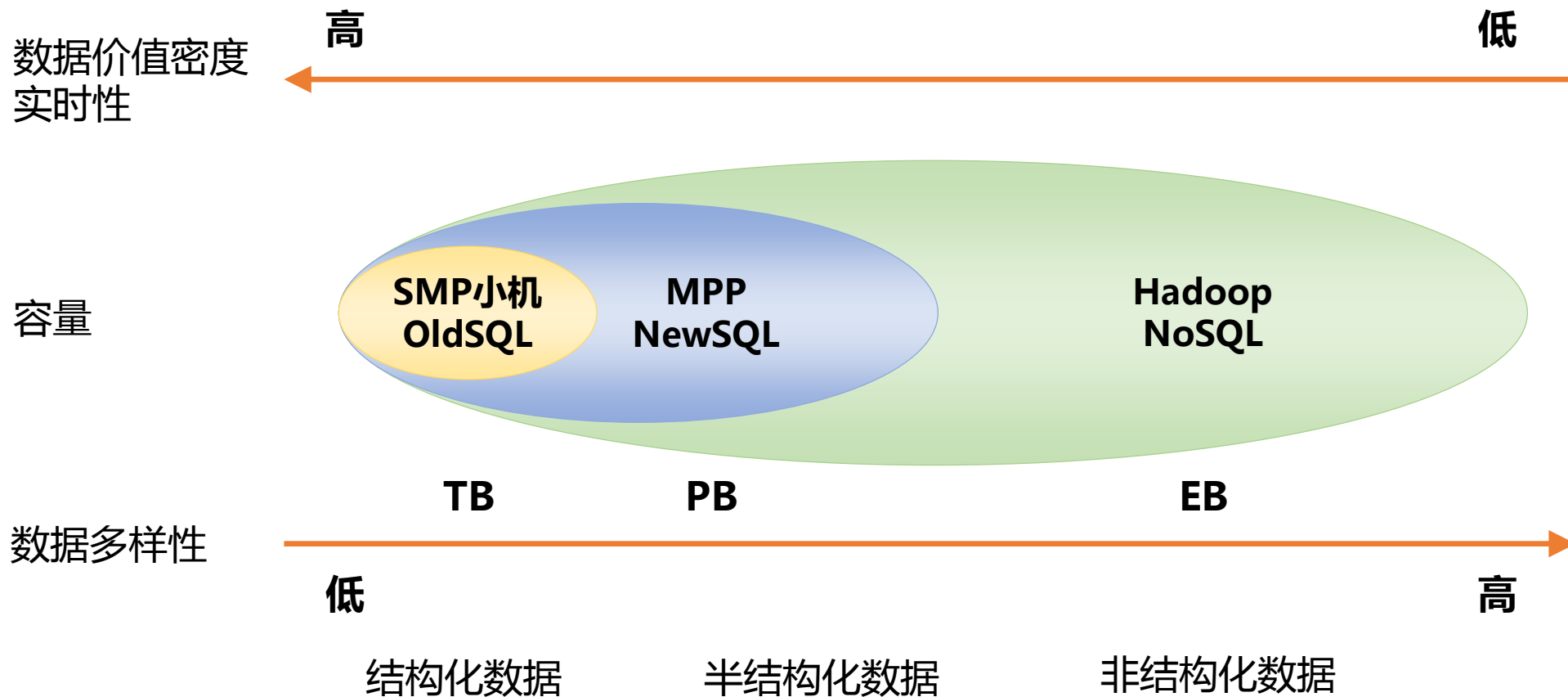
大数据引发的处理架构多元化：M. Stonebraker

一种架构支持多类应用
(One Size Fits All)

多种架构支持多类应用
(Not only one Fits All)



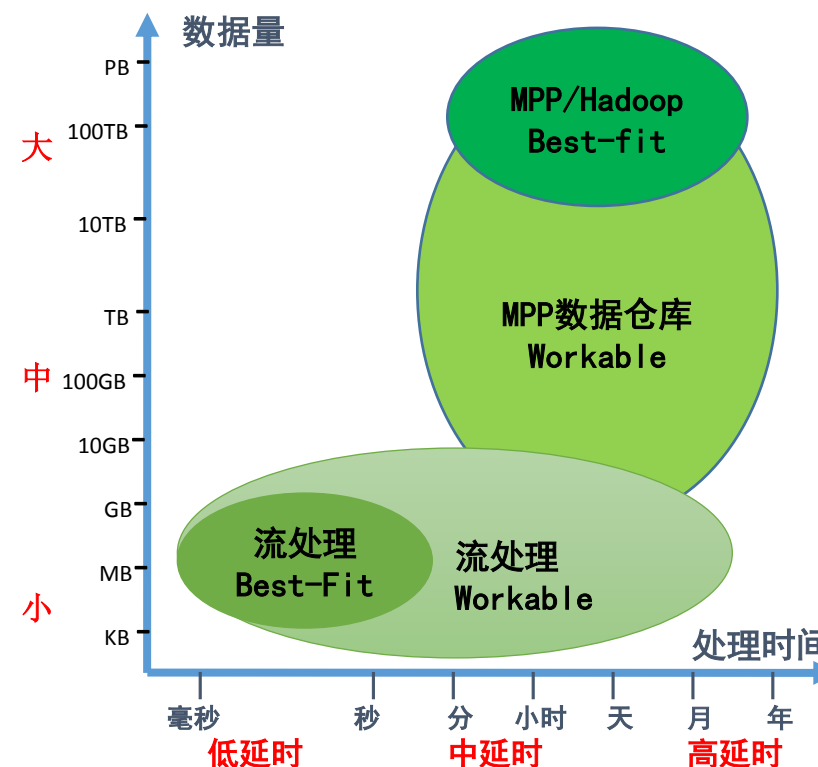
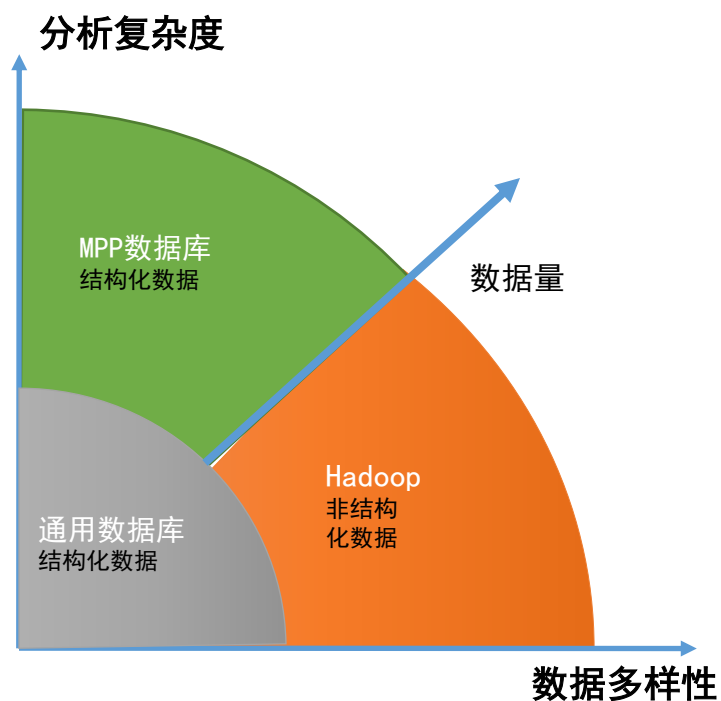
目前数据处理的有效架构：混搭架构，多种技术



SMP 垂直扩展、单一数据处理引擎的时代正在成为过去

目前数据处理的有效架构：混搭架构，多种技术

- 大数据平台类型复杂，既涵盖了结构化数据，又涉及到实时流数据，以及各类非结构化数据，单一技术无法满足需求
- 未来大数据平台的技术选择应以“适才适所”的原则，进行多种平台的数据集成，**集成各个技术的价值**



SQL价值：TPC-DS MPP与Hadoop性能对比

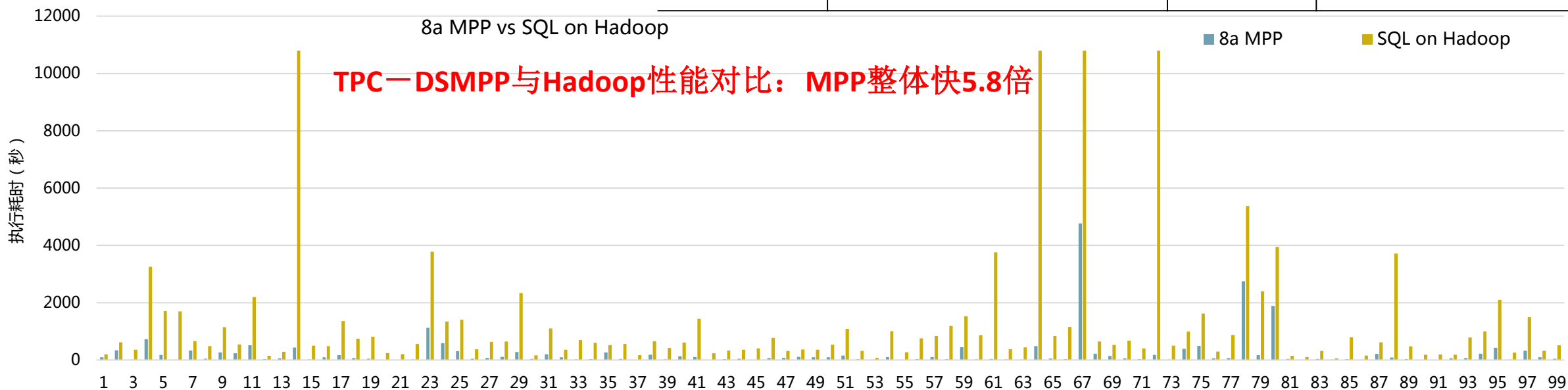
TPC-DS基准测试的特点：

- 共99个测试案例，遵循SQL99和SQL 2003的语法标准，SQL案例比较复杂
- 测试案例包含各种业务模型（如分析报告，迭代式联机分析，数据挖掘等）
- 分析的数据量大，并且测试案例是在回答真实的商业问题
- 几乎所有的测试案例都有很高的IO负载和CPU计算需求

测试产品	
8a MPP	GBase 8a MPP Cluster
SQL on Hadoop	某Hadoop商业发行版 (SQL引擎基于Hive on Spark)

测试环境

服务器台数	4		
操作系统	Redhat 6.5 x86_64	硬盘	SAS 15K rpm (RAID 0)
CPU	Xeon E5-2650 2 * 16 cores	网络	10Gb/S
内存	128GB	TPC-DS	1000 Scale factor



SQL价值：TPC-DS MPP与Hadoop性能对比

TPC-DS MPP与Hadoop性能对比：MPP整体快10倍以上！ GBase MPP整体快7.3倍以上

TPC-DS SQL查询特征	
SQL特征	查询数量
子表达式	31
关联的子查询	15
不相互关联的子查询	76
Group By	78
Order By	64
Rollup	9
Partition	11
Exists	5
Union	17
Intersect	2
Minus	1
Case	24
Having	5

SQL on Hadoop异常语句说明	
Q14、Q64、Q67、Q72	超3小时没有执行完毕，均按3小时计时
8a MPP/SQL on Hadoop性能比	
8a MPP慢于SQL on Hadoop语句个数	0
10倍以上语句个数	47
20倍以上语句个数	24
总耗时（秒）	
SQL on Hadoop	129,025 (35.8小时)
8a MPP	22,320 (6.2小时)
耗时比	5.8

SQL价值：TPC-H MPP与Impala性能对比

GBase MPP整体快7.3倍以上

TPC-H基准测试的特点：

- 同数据、同环境下的性能，Impala与MPP对比，根据数据特点不同，Impala较MPP会有低于7~12.3倍的性能差异
- Impala优化手段极其复杂，需要专业人士才能完成
- 不支持高精度decimal类型, 因此在实际生产环境中基本不可用
- 没有事务能力，加载和执行insert过程中可以看到脏数据，因此在实际生产环境中基本不可用
- 750G左右的数据，MPP加载17分钟，Impala + kudu加载需要23.5小时，因此在实际生产环境中基本不可用
- 查询执行不稳定，执行SQL有时报错，当不执行任何任务时，也发现后台在做大量的IO操作，此时执行任何SQL都会报错（包括执行select count(*)这样简单的SQL），因此在实际生产环境中基本不可用。

序号	impala耗时（秒）	MPP耗时（秒）	结果集
SQL1	6.72	3.45	1
SQL2	155.71	19.1	999
SQL3	47.74	7.26	1
SQL4	171.21	20.17	1907
SQL5	9.3	1.81	1907
SQL6	5.31	2.46	1676
SQL7	11.05	4.17	1907
SQL8	3.94	2.56	1879
SQL9	6.42	2.5	1907
SQL10	3.92	2.78	1907
SQL11	3.8	2.19	1907
SQL12	4.25	2.17	1879
SQL13	4.2	1.15	1907
SQL14	7.9	4.17	1907
SQL15	4.69	0.97	0
SQL16	176.55	22.98	0
SQL17	316.61	29.48	0
SQL18	246.62	33.1	0
SQL19	58.3	5.65	95350
SQL20	5.33	1.53	3814
SQL21	3.62	1.26	1907
总计	1253.19	170.91	--

Hadoop价值：非结构化数据 & 深度机器学习

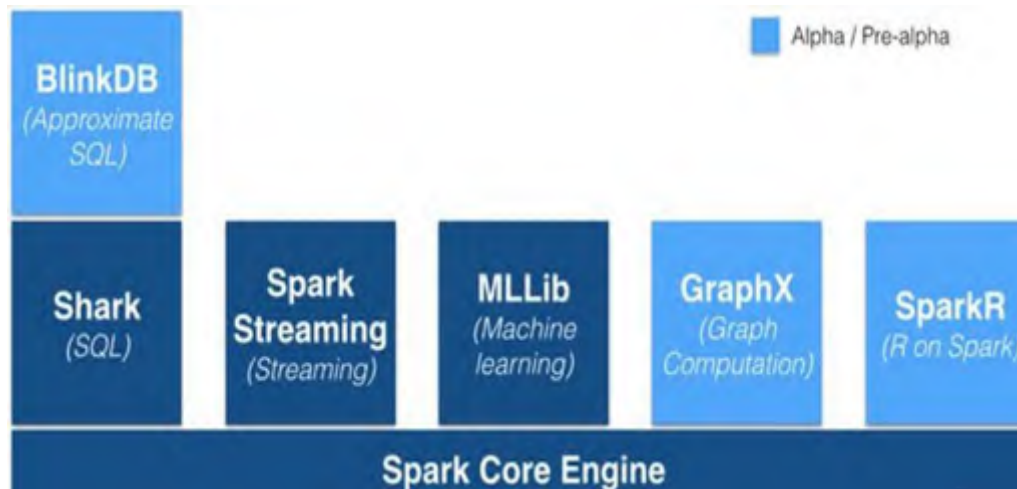
Hadoop

- ❑ 非结构化数据接入和实时分析
- ❑ 历史数据查询分析

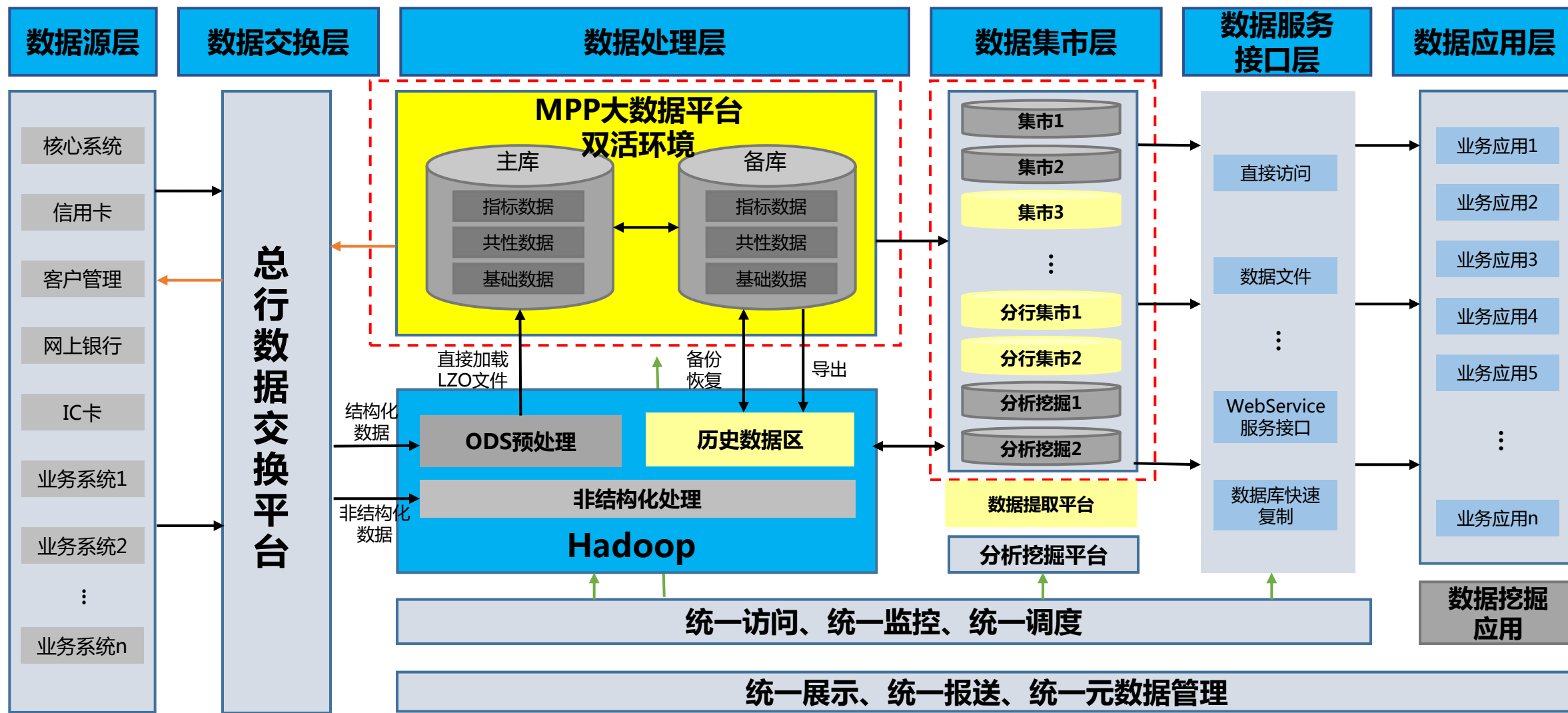


Spark

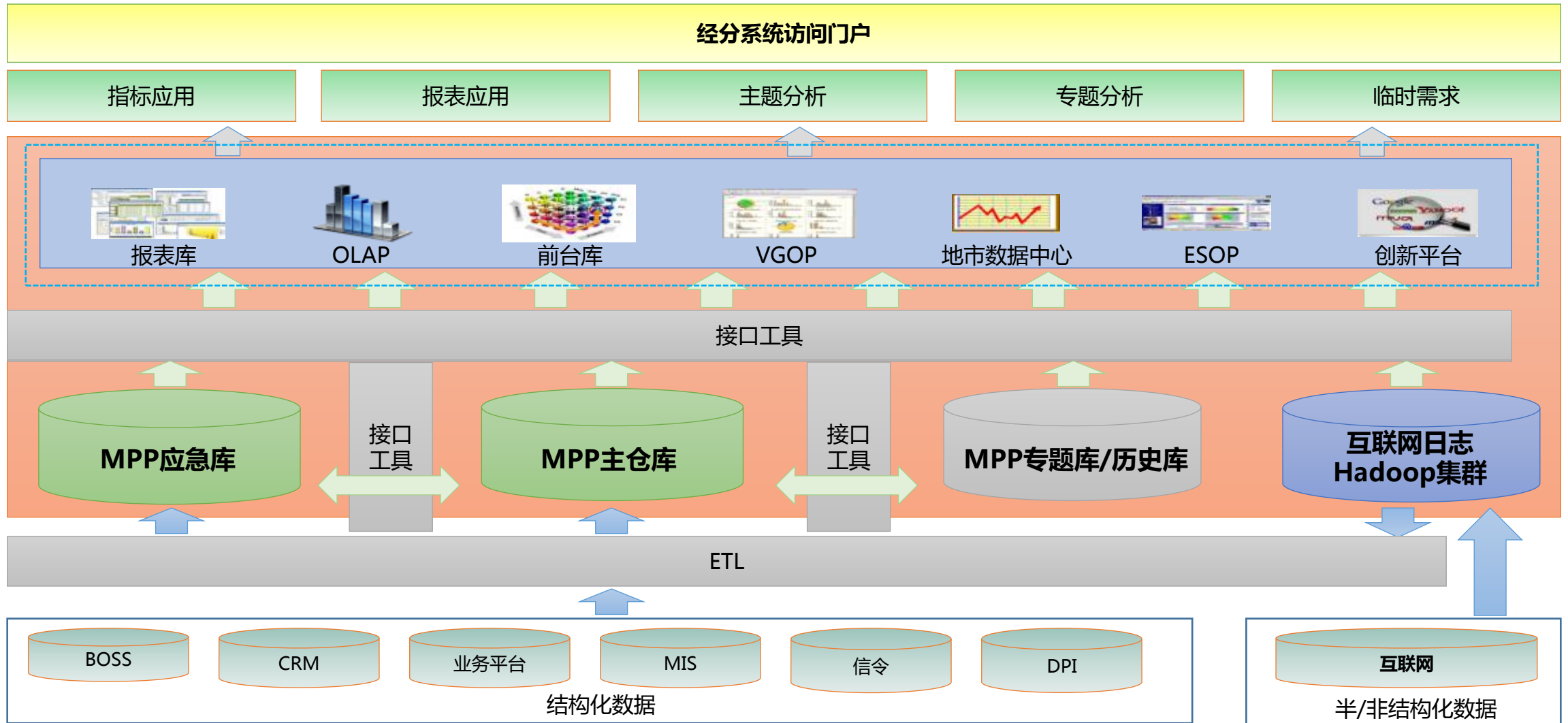
- ❑ 流数据处理
- ❑ 深度机器学习
- ❑ 图算法引擎
- ❑ R语言非结构数据算法分析



混搭架构的成功案例1——中国农业银行大数据平台案例

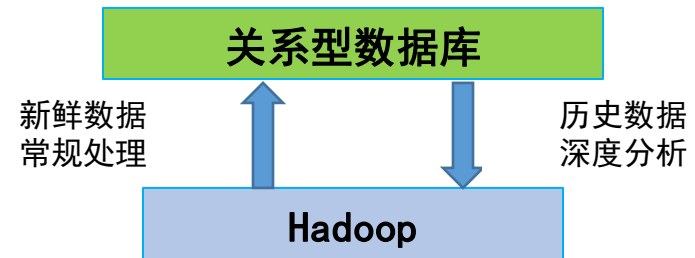
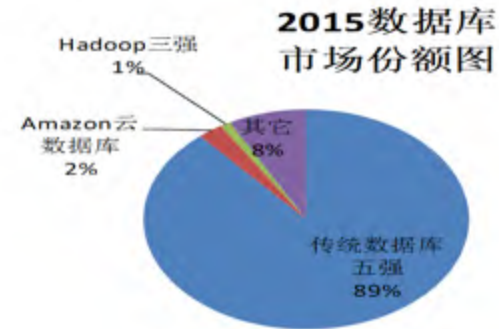


混搭架构的成功案例2——浙江移动大数据平台案例



Gartner预测：RDBMS仍然主导价值，但数据融合更能产生价值

- 当前数字商业的场景越来越复杂，数据容量越来越大，数据也越来越分布。数据的集成，分享和组织管理成为重要目标，统一管理是大势所趋
- 市场表现方面，传统数据库5强依然占据88.7%市场份额。表明传统SQL数据库依然有着很大的用户市场
- 尽管传统SQL数据库和Hadoop有各自独立的工作场景，它们之间的数据融合和双向数据流通越来越成为趋势



Next : 混搭架构2.0 = 融合架构

企业用户的传统数据库应用和大数据应用

企业用户的传统数据库应用和大数据应用

融合架构 (GBase UP 统一层) :

- ✓ 数据统一管理, 统一访问
- ✓ 数据易流通
- ✓ 数据易集成

解决了好用的问题, 提升混搭价值

JDBC、ODBC、ADO.NET、CAPI、RESTful API

GBase UP

GBase 8t

GBase 8a
MPP

Hadoop生态

Spark 栈

安全管理、运营管理

目录

- 大数据需求与IT技术架构演变

- SQL (关系) 与Hadoop (非关系) 的融合之道

- GBase UP 介绍及应用案例

- 总结与展望

融合需求：多层次全方位融合

语言：SQL，NoSQL，Not Only SQL

- Not Only SQL 是趋势

架构：SMP，MPP

- 所有架构都朝MPP发展

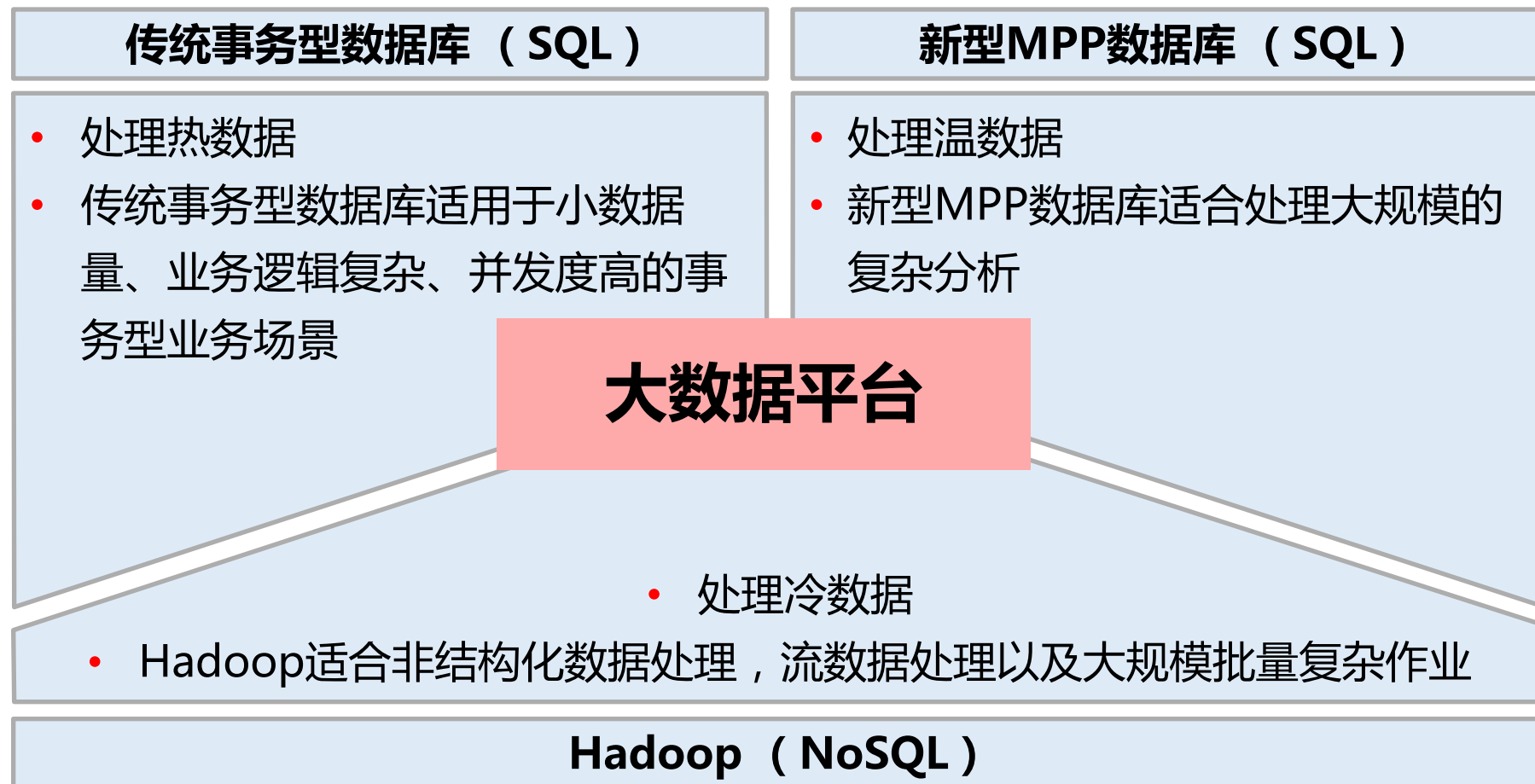
算法：SQL，ML，其他复杂算法

- 这些都需要，最好一起使用

数据类型：结构化、半结构、非结构数据

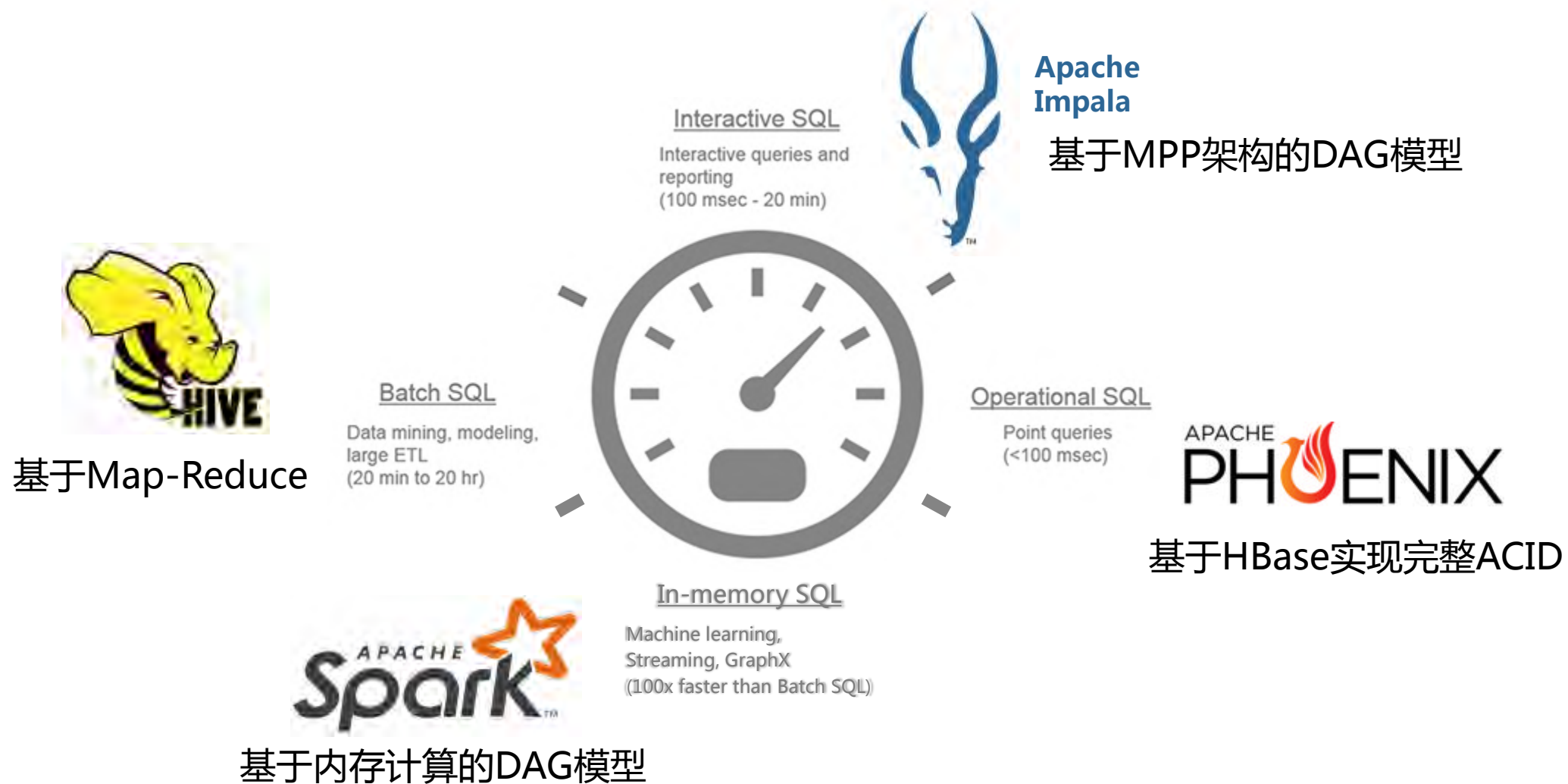
- 需要统一处理

融合需求：大数据融合平台架构需求范畴



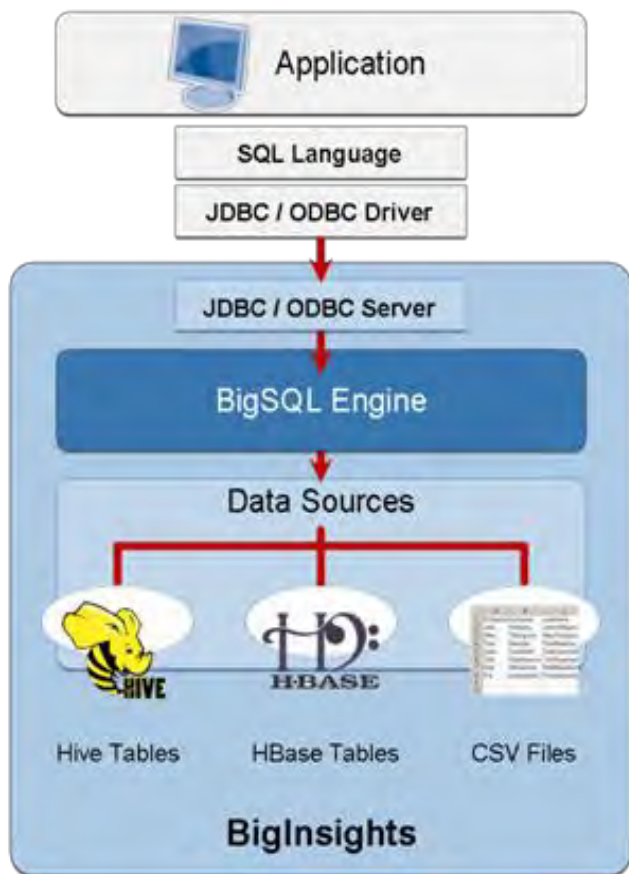
Not Only SQL, Not Only OLAP, Not Only Hadoop

融合方式：开源SQL on Hadoop 项目

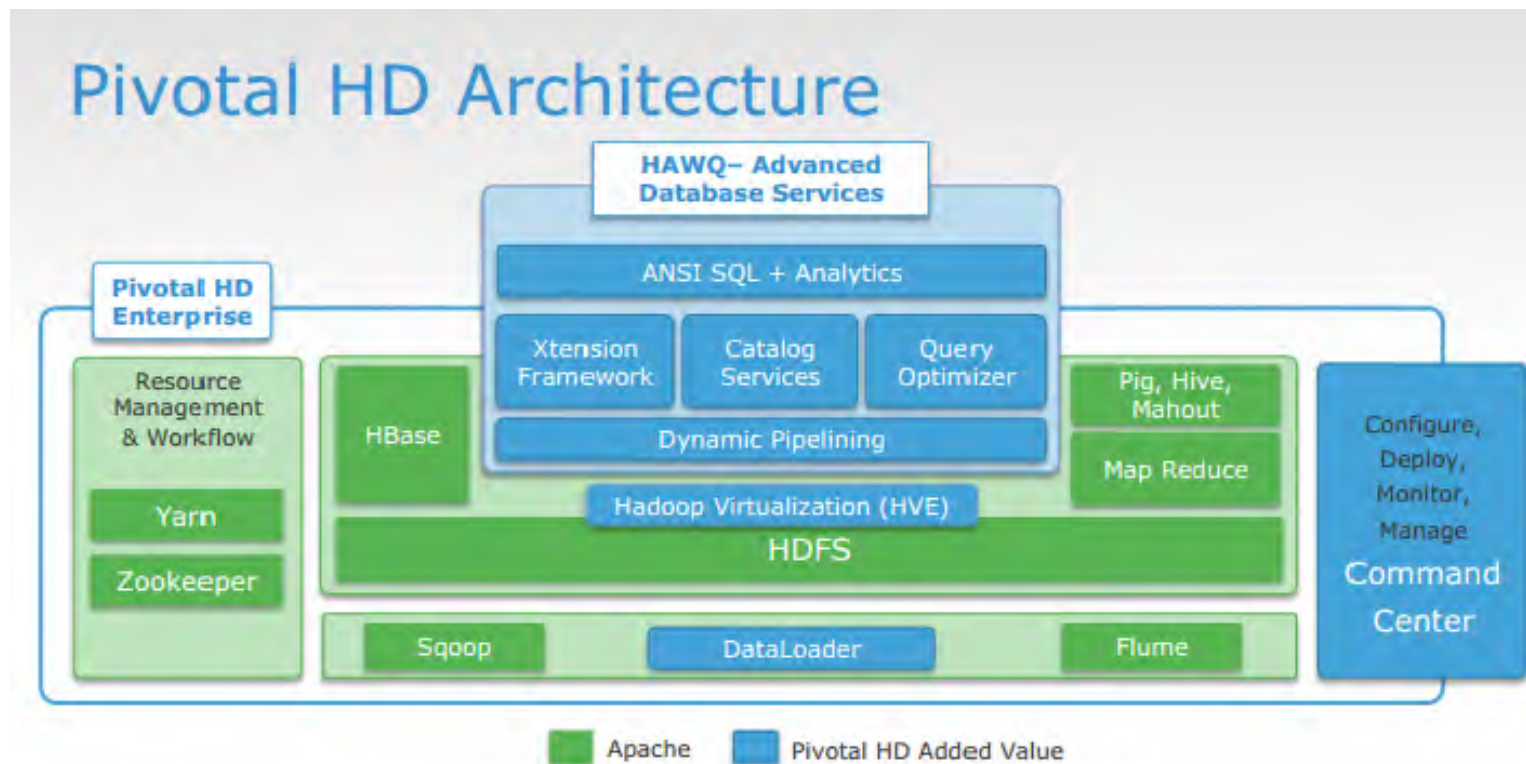


融合方式：SQL on Hadoop 商业产品

IBM , Big SQL

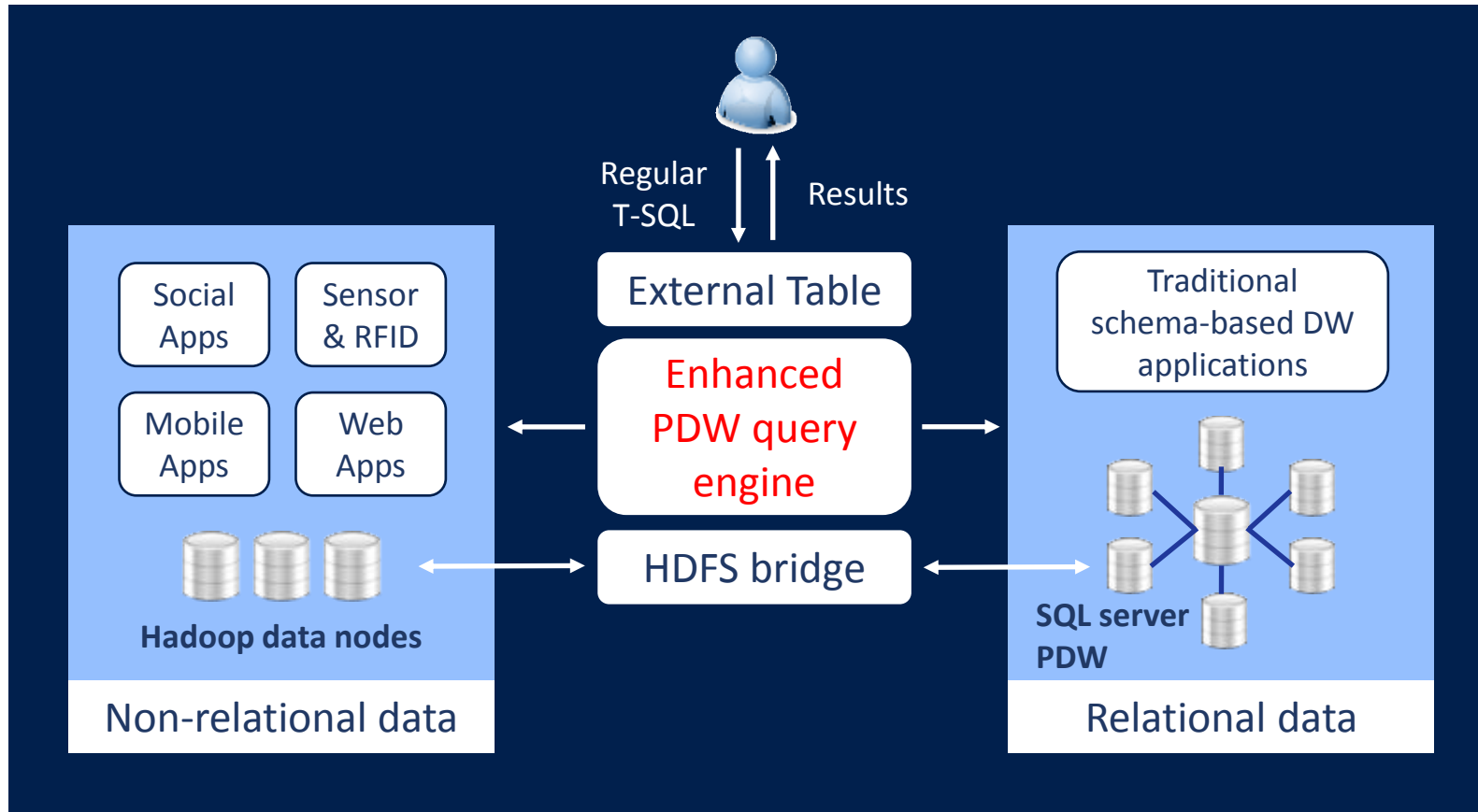


Pivotal HAWQ



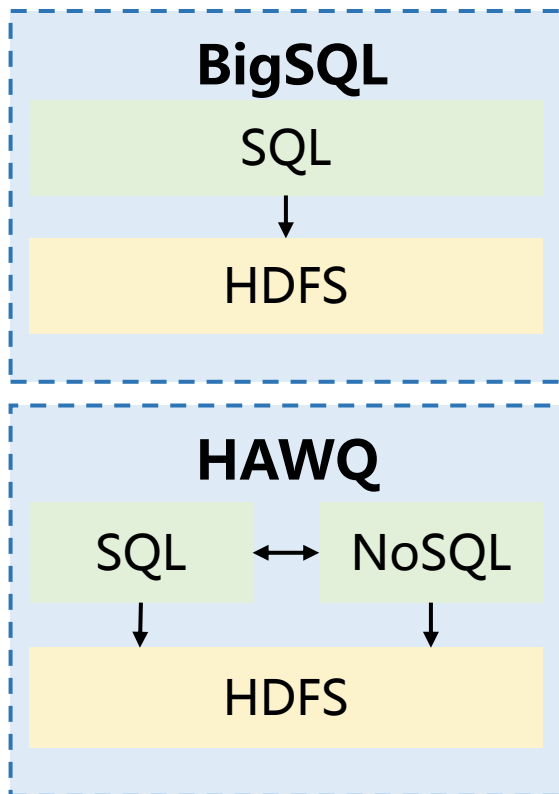
融合方式：SQL and Hadoop 商业产品

Microsoft , Polybase 一体机产品



SQL on Hadoop **OR** SQL and Hadoop

SQL on Hadoop ?

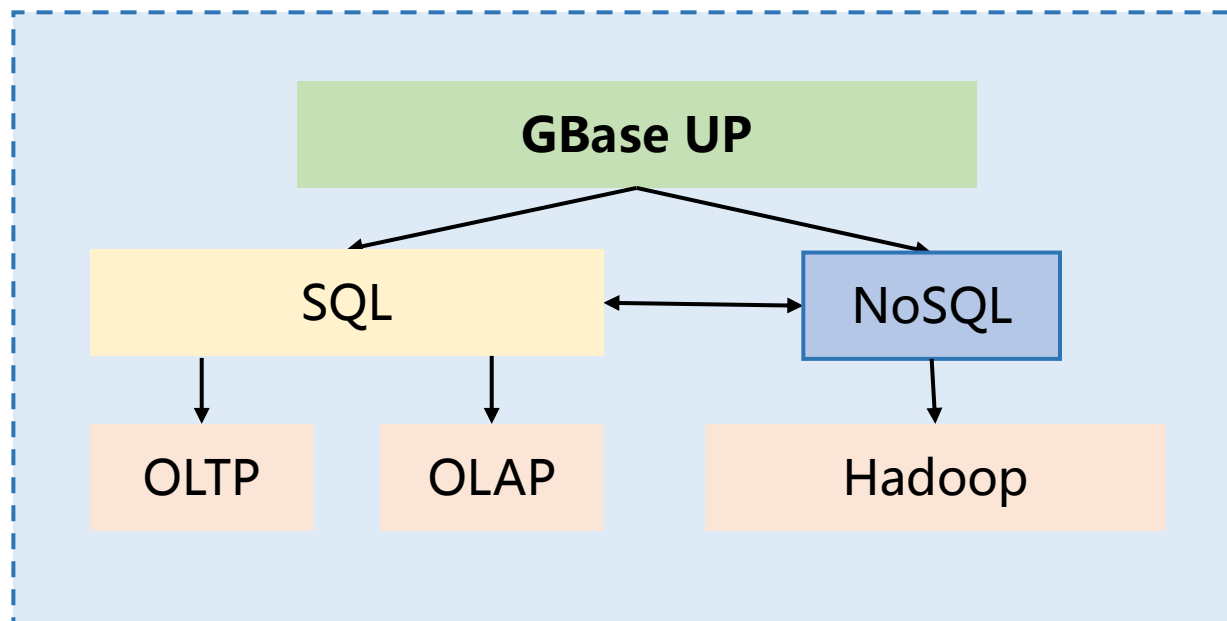


垂直融合方式：

- 部署一套环境，适用于新建大数据环境
- 融合难度大
- 融合效果不佳（HAWQ相对于原型GreenPlum性能下降明显）
- 没有考虑复杂的OLTP应用

OR

SQL and Hadoop!



水平融合方式：

- 充分利用多套环境，发挥各自价值
- 充分保留IT历史资产
- 融合难度小
- 融合效果好
- 融合OLTP，OLAP

VS



目录

- 大数据需求与IT技术架构演变

- SQL (关系) 与Hadoop (非关系) 的融合之道

- GBase UP 介绍及应用案例

- 总结与展望

GBase UP产品定位：企业级大数据融合平台

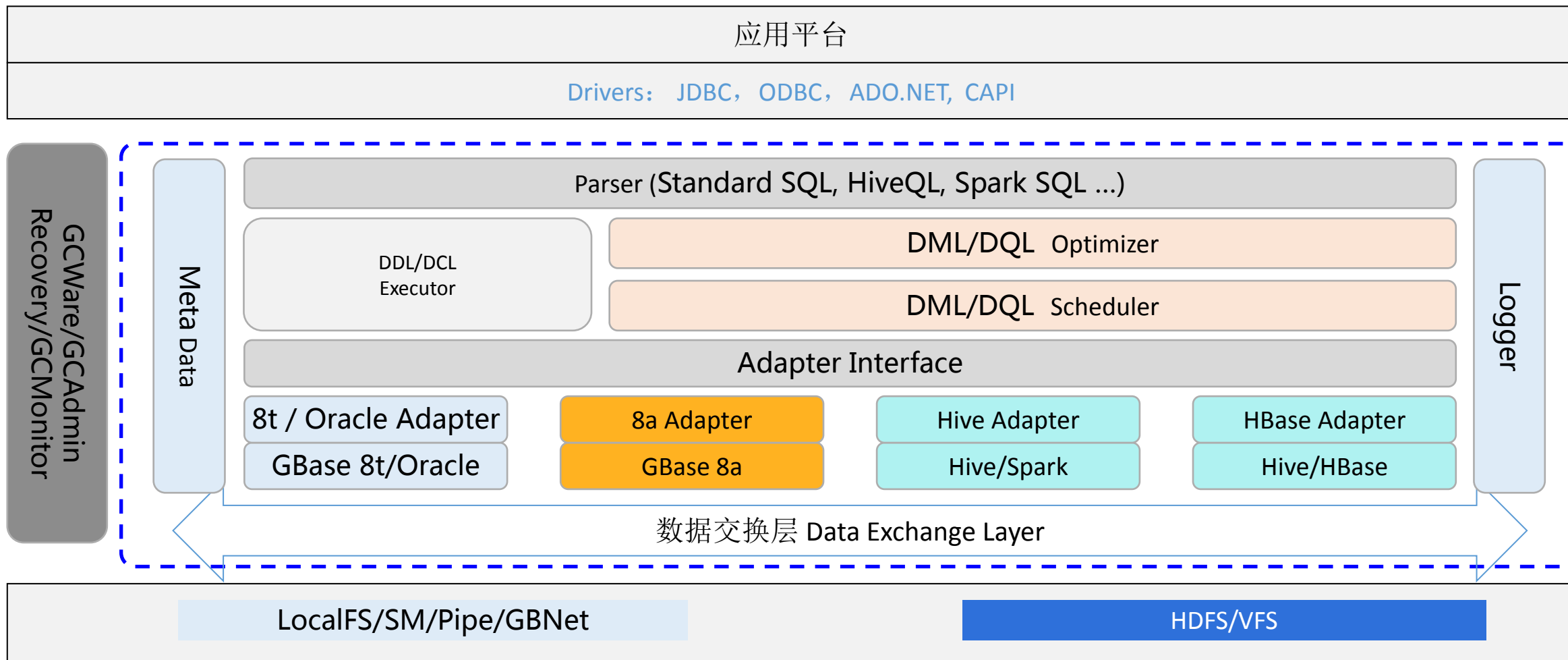
U = 融合、统一

UP

P = 平台、扩展

统一接口：UI	标准的ODBC、JDBC、CAPI
统一语言：UQL	结合SQL92 和 HiveQL等方言
统一元数据：UM	保存完整元数据定义
统一安全：UA	安全认证，避免了多种认证模式
统一事务：UT	支持集群级事务
统一调度：UC	调度引擎间计算和存储
统一日志：UL	保存日志到Hadoop，供日志分析
数据交换	引擎之间建立高速的多对多内部通道；备份恢复、容灾、异地部署
全数据	结构化、半结构化、非结构化
扩展	UDF扩展
系统管理	配置、监控、资源管理
生命周期管理	数据生命周期管理

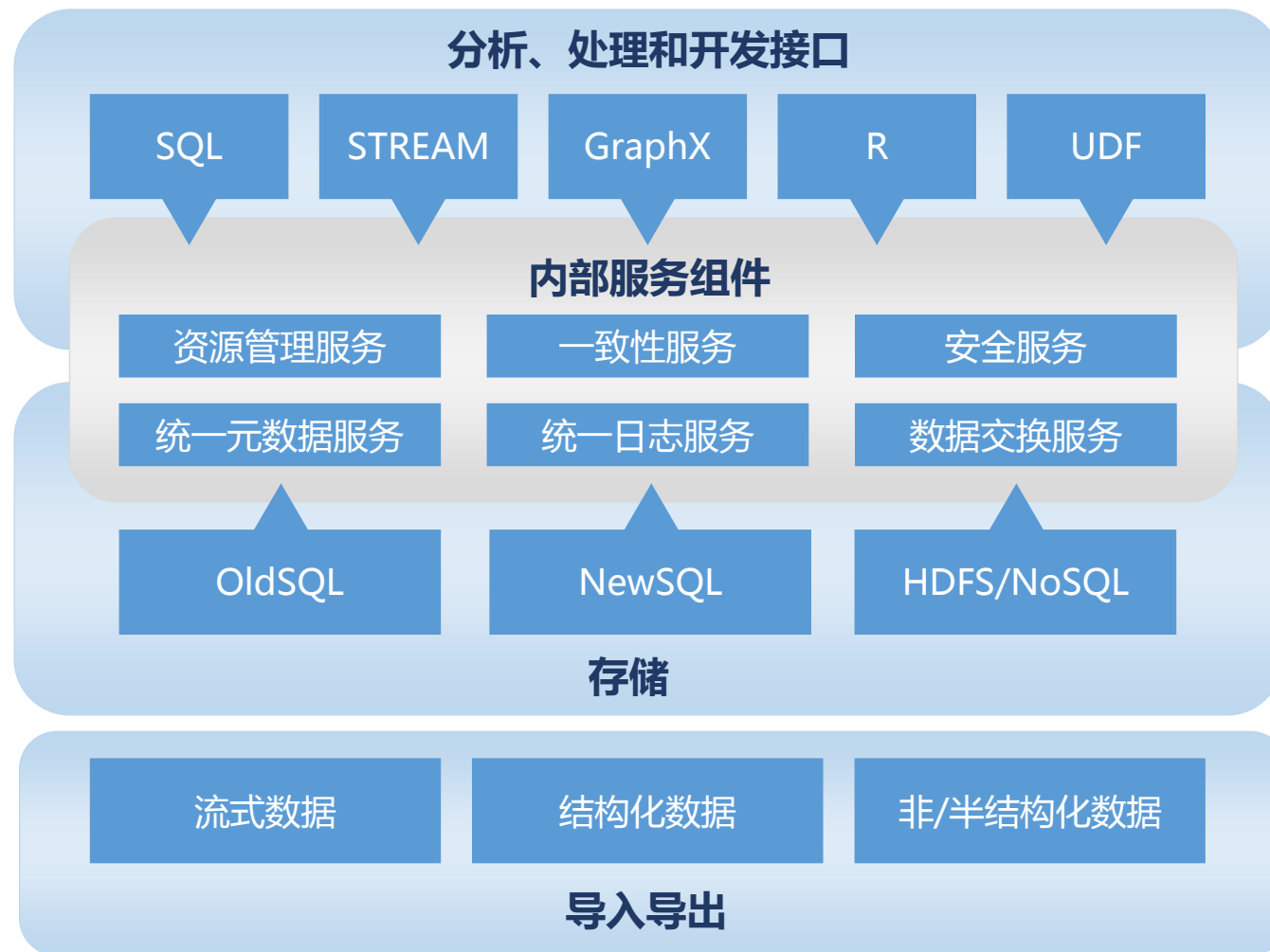
GBase UP产品架构：SQL and NoSQL融合



GBase UP产品特点&价值

Vision	Use Case	Core Values
融合 数据联邦	异构引擎透明访问	简化应用开发，降低数据建模的复杂度
	跨引擎数据交换	高吞吐率的多对多通讯机制
	跨引擎关联查询	实现自动优化的引擎间关联分析
	BLOB on Hadoop	扩展非结构化数据存储和计算能力
简化 数据流通	跨引擎读写分离	支撑大规模数据事务处理和实时BI数据分析
	数据生命周期管理	按不同温度选择最合适的引擎存储数据，降低数据总体持有成本
	PB级备份与恢复	实现在线PB级数据备份与恢复
扩展 数据分析	跨引擎UDF扩展	支持跨引擎UDF函数，灵活扩展系统的计算能力
	机器学习	融合Spark机器学习算法，实现 In-Database Analysis
	R语言	同时适应偏向SQL和偏向R的用户

GBase UP 产品生态



GBase UP 技术架构

Mega SQL Engine (数据联邦) (SQL92 + HiveQL + 8t + Oracle etc)

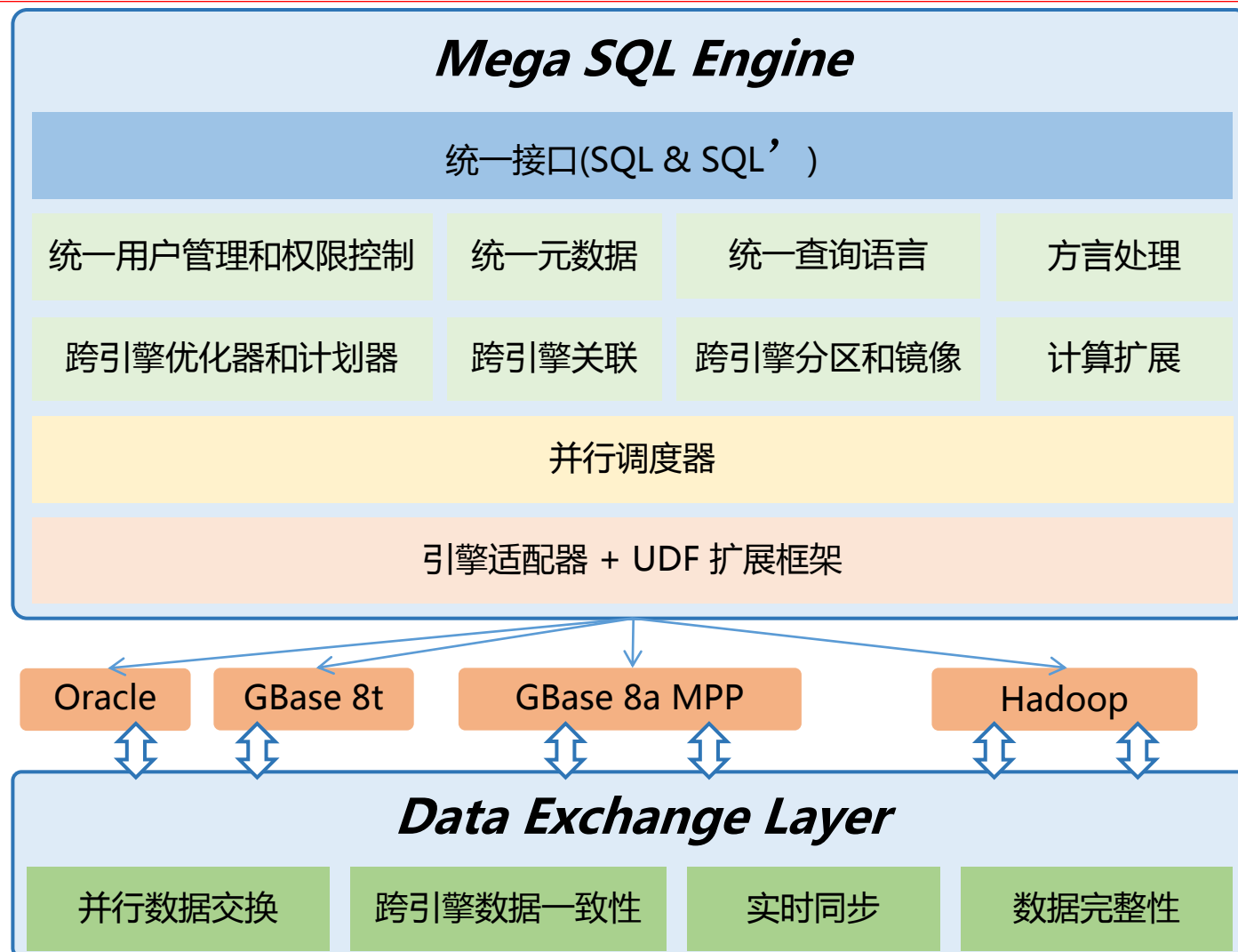
- 统一接口
- 统一查询语言
- 统一用户管理和权限控制
- 统一元数据
- 跨引擎优化器和计划器
- 跨引擎关联查询
- 跨引擎数据分区和镜像
- 并行调度器
- 引擎适配器

UDF扩展框架 (数据分析)

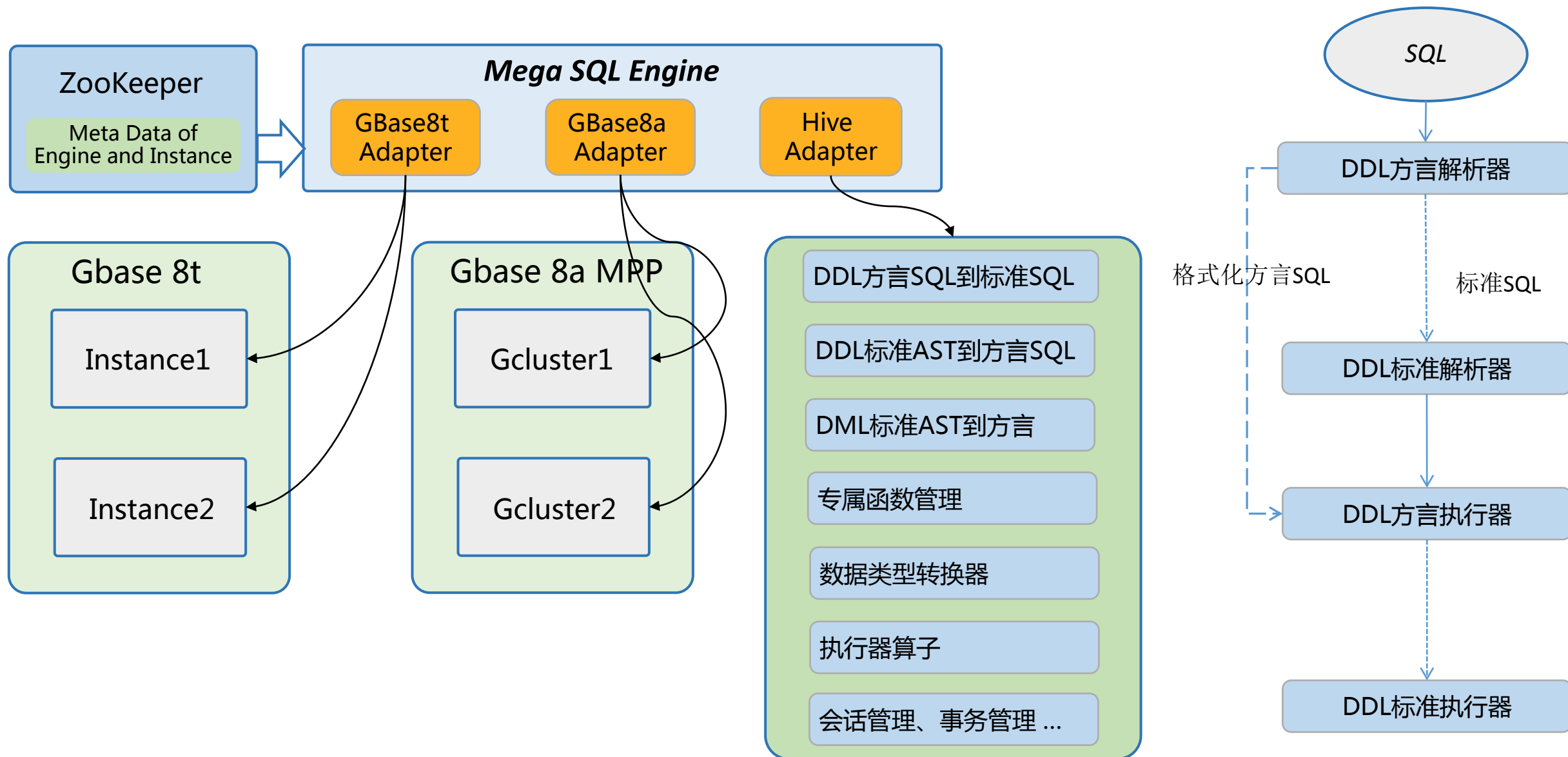
- 扩展数据分析处理功能
- C/C++/Java/Scala/R

Data Exchange Layer (数据流通)

- 跨引擎并行数据交换
- 跨引擎数据一致性、完整性
- 实时同步



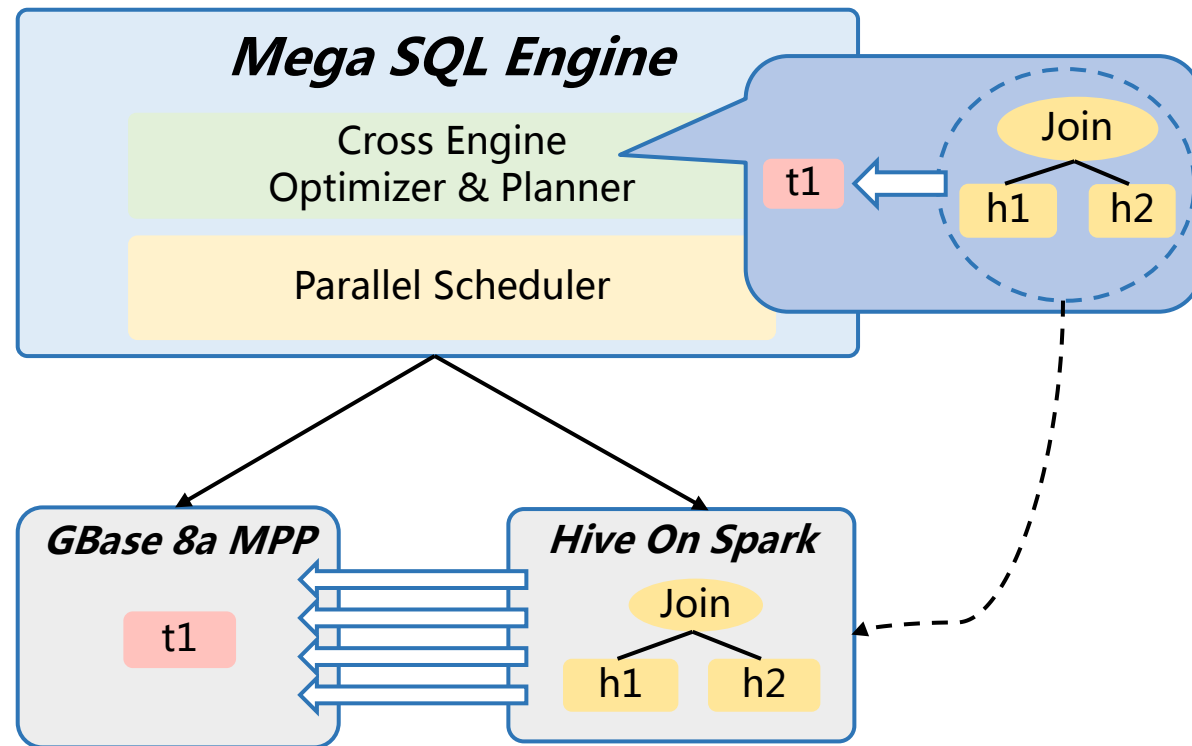
GBase UP 技术组件：多引擎适配器



GBase UP 核心功能：跨引擎数据交换（数据联邦）

```
Create table t1(in_date date, ...) engine= 'GBase8a' ;  
Create table h1 (url varchar(256), ...) engine= 'Hive' ;  
Create table h2 (...) engine= 'Hive' ;
```

```
-- ELT  
Insert into t1 select ... h1, h2 where ... and  
parse_url(url, 'HOST') = 'gbase.cn' ;
```



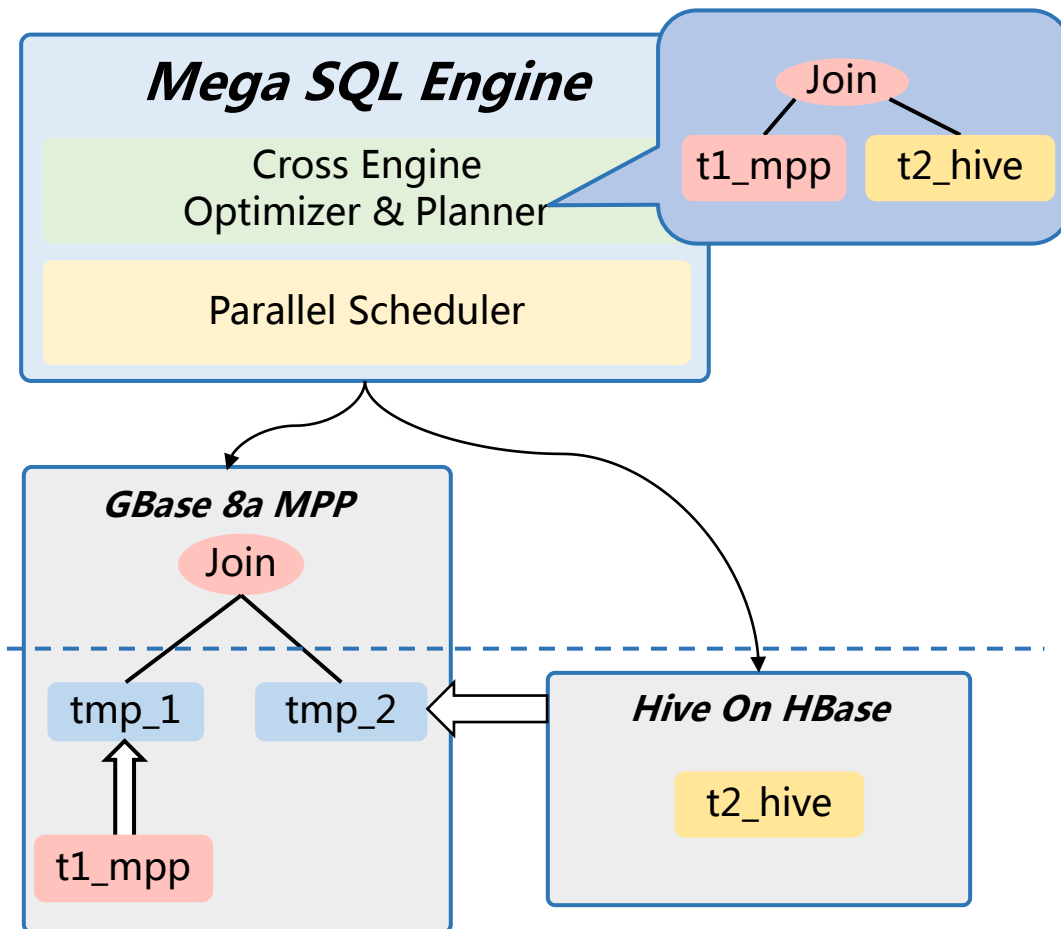
GBase UP 核心功能：跨引擎关联查询（数据联邦）

```
Create table t1_mpp (id number(20), name varchar(100), city  
varchar(100), weichat varchar(2000), oupdate datetime ...)   
engine= 'GBase8a' ;  
Create table t2_hive (key number(20), url varchar(1000), weichat  
varchar(5000), ...) engine= 'Hive' ;
```

```
Insert into t1_mpp ... ;  
Insert into t2_hive ... ;
```

-- 全数据查询案例

```
Select t1.name, t1.oupdate, t2.url from t1_mpp as t1, t2_hive as  
t2 where t1.id=t2.key and t1.city= '北京' and extracts(t2.url,  
'gbase' ) >0 and contains (t1.weichat, '南大通用' )>0 order by  
t1.oupdate limit 10;
```



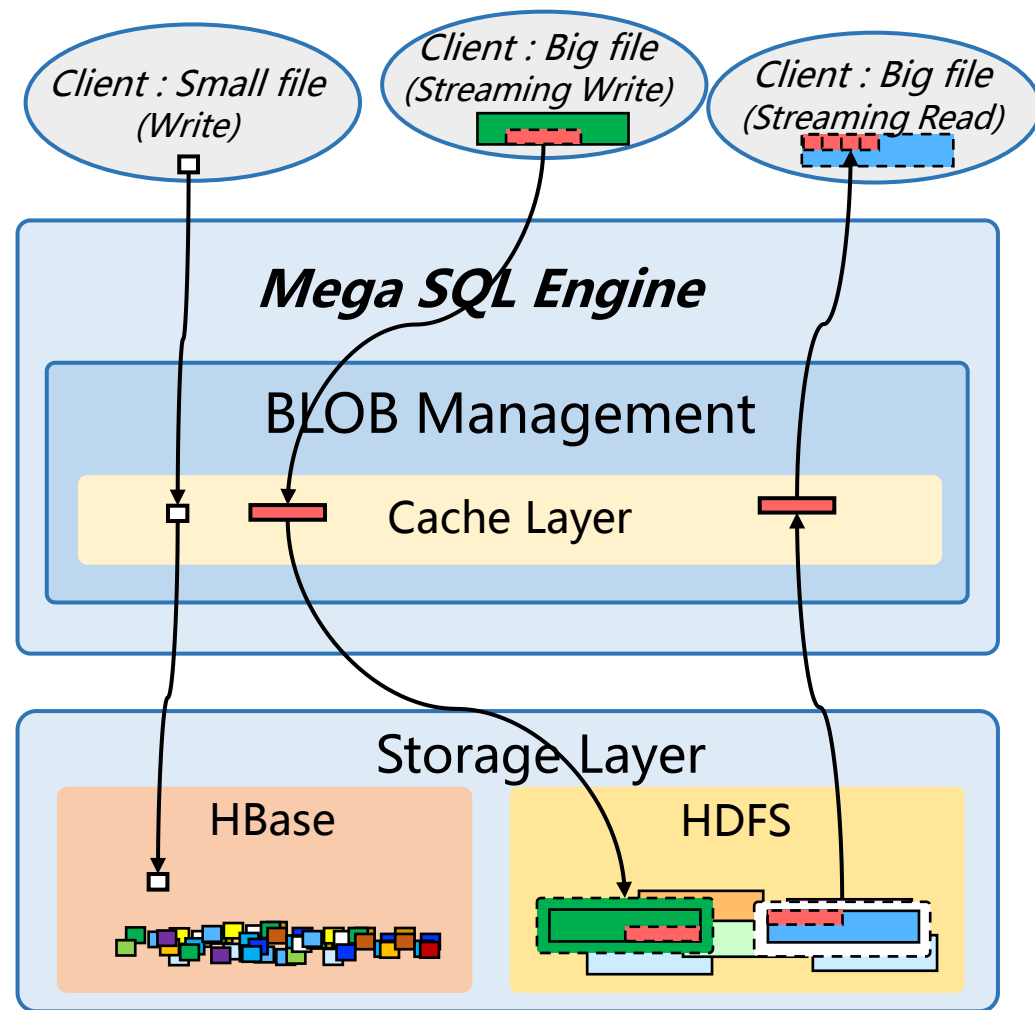
GBase UP 核心功能 : BLOB on Hadoop (数据联邦)

```
Create table email(uid int, send_dt date, content blob  
uri) engine= 'GBase8a' ;
```

```
// Writing BLOB data as stream  
FileInputStream inStream = new FileInputStream(...);  
stm.setBinaryStream(3, inStream);  
stm.executeUpdate();
```

```
// Reading BLOB data as stream  
FileOutputStream outputStream = new  
FileOutputStream(...);  
while(...) {  
    InputStream returnStream= rs.getBinaryStream(3);  
    returnStream.read(userBuffer);  
    outputStream.write(userBuffer);  
}
```

- 1、扩展了MPP数据库非结构化数据的存储能力。
- 2、结合结构化字段和非结构字段，同时检索结构化和非结构化数据。



GBase UP 核心功能：引擎级别读写分离（数据流通）

-- 创建镜像表，镜像方向为GBase8t到GBase 8a MPP

```
Create table t(...) engine= 'Mirror8t8a' ;
```

-- 写操作用8t引擎

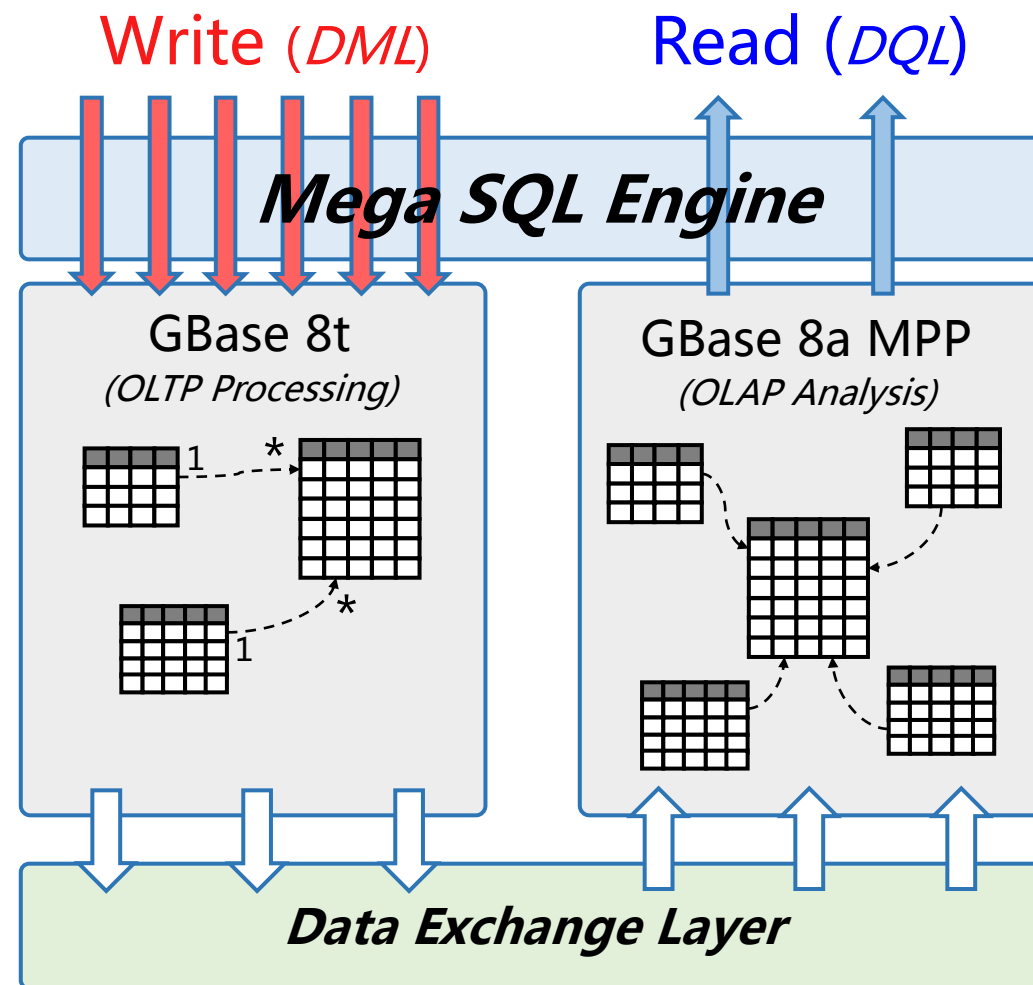
```
Insert into t values(...);
```

-- 分析型查询用8a引擎

```
Select avg(...) from t group by ...;
```

通过透明的mirror table实现真正的实时交易和实时分析的数据流通：

- （小）交易在OLTP引擎实现（> 100万tpm）
- 交易数据实时同步到OLAP引擎
- UP将查询自动路由到OLAP引擎



GBase UP功能：数据生命周期管理（跨引擎分区表，数据流通）

-- 创建分区表，按热、温、冷分别存储在三个数据引擎

```
Create table t_part (... , in_date date) partition by range(in_date)
(partition p_hive values less than (date_sub(current_date(), interval 1 month)) engine= 'Hive' ,
partition p_8a values less than (date_sub(current_date(), interval 1 week)) engine= 'GBase8a' ,
partition p_8t values less than MAXVALUE engine= 'GBase8t' );
```

-- 实时数据

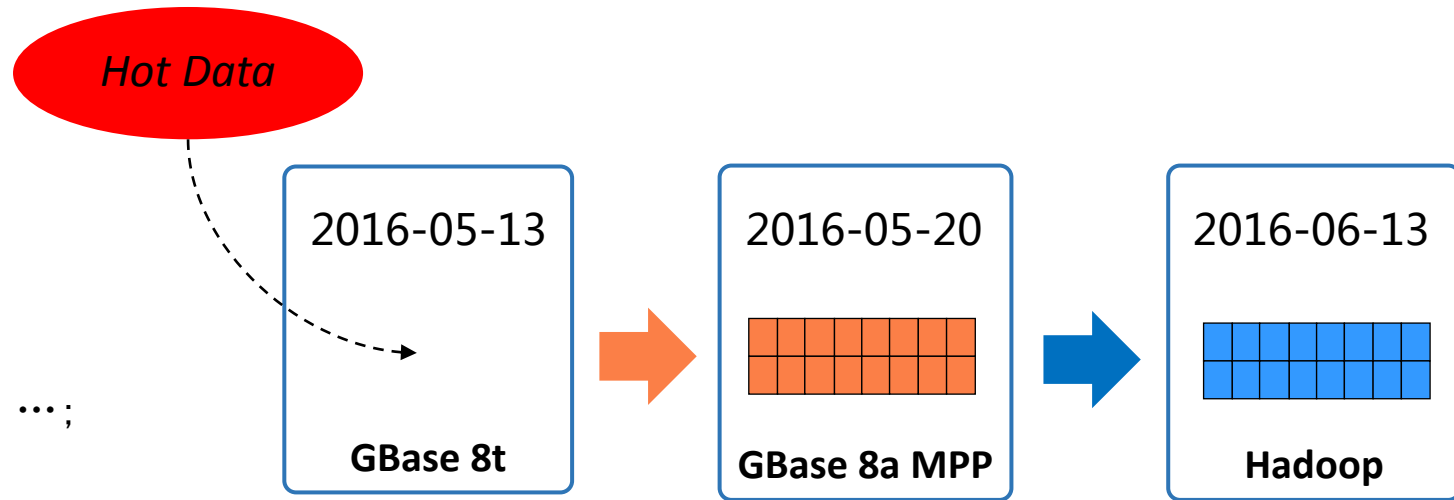
```
Insert into t_part values(... , '2016-05-13' );
Update t_part set ... where in_date = '2016-05-13' ;
```

-- 近期数据分析

```
Select count(0) over (partition by ...), ... from
t_part where ... and in_date
between '2016-05-01' and '2016-05-13' ;
```

-- 历史数据分析

```
Select count(a) , ... from t_part
where ... and in_date <= '2016-05-13' group by ...;
```



GBase UP 核心功能：PB级备份与恢复（数据流通）

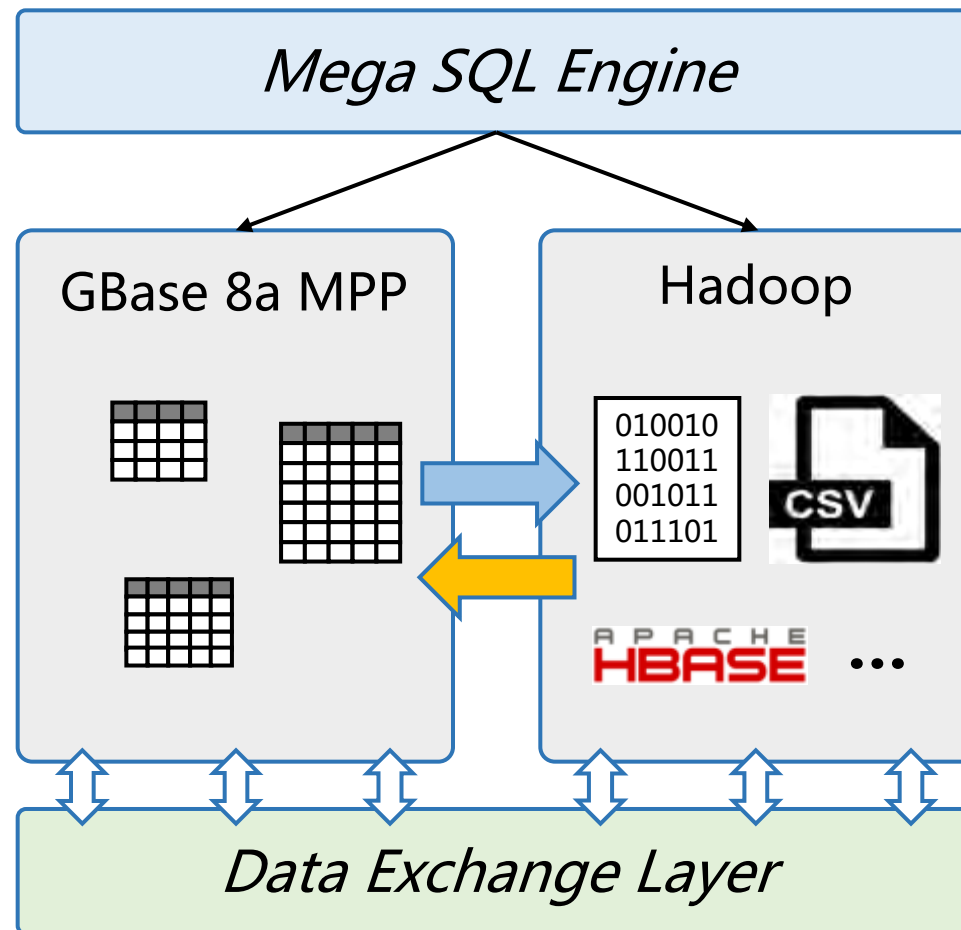
```
Create table t1_8a(cust_key bigint, order_stat int,  
total_price decimal(18,2), order_dt date)  
distributed by ('cust_key') engine= 'GBase8a';
```

-- 备份到HDFS

```
Select ... from t1_8a into outfile 'hdp://...'  
format = binary | csv | hbase | ...;
```

-- 从HDFS恢复

```
Load data infile 'hdp://...' into table t1_8a;
```



GBase UP 核心功能：跨引擎UDF扩展（数据分析）

```
Create table t1_oltp(website varchar(200), clickcount  
number(10)...) engine= 'GBase8t' ;  
Create table t2_hive(key bigint, url varchar(1000), weichat  
varchar(5000), ...) engine= 'Hive' ;
```

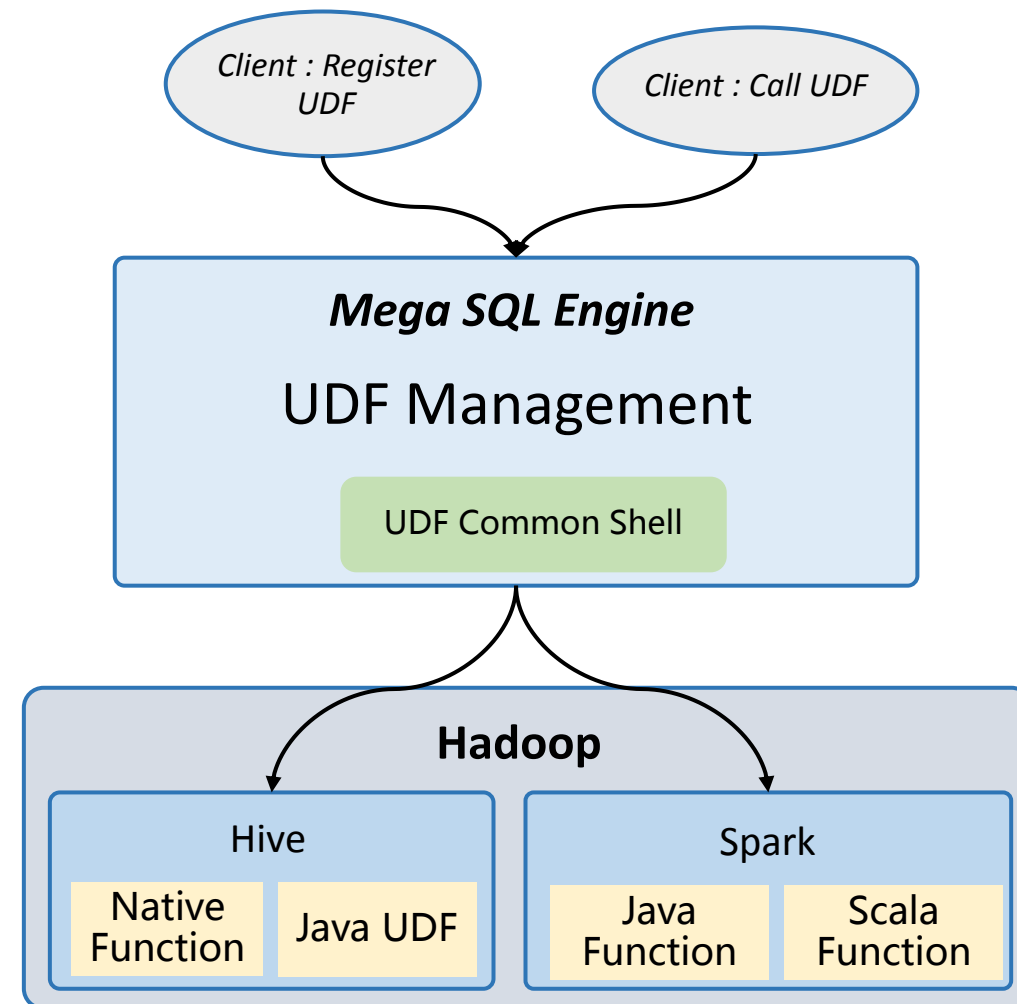
```
Insert into t2_hive ... ;
```

-- 创建用户自定义函数

```
Create function extractwebsite returns string soname  
'hive_common.so' ;
```

-- SQL中调用自定义函数

```
Insert into t1_oltp(website , clickcount) select  
extractwebsite(url), count(*) from t2_hive;
```



GBase UP 核心功能：机器学习算法（数据分析）

-- 创建模型

Call

```
create_model( 'moive_rec' , ' als' , ' t_setting' );
```

-- 训练电影推荐模型

```
Call train_model( 'moive_rec' , ' t_train' );
```

-- 评估模型

```
Call evaluate_model( 'moive_rec' , ' t_test' );
```

-- 计算预测结果

Call

```
predict( 'moive_rec' , ' t_predict' , ' t_result' );
```

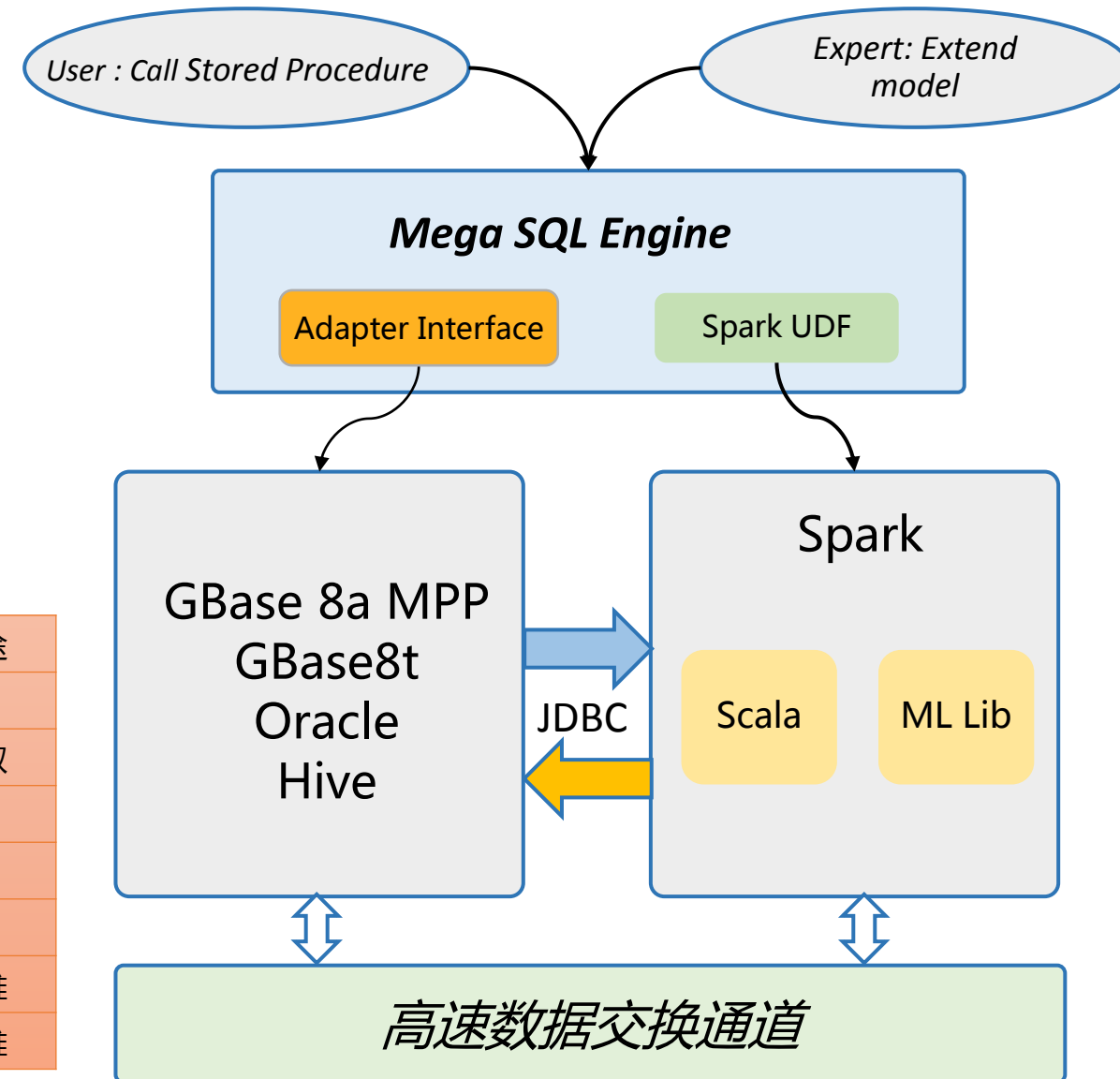
```
Select * from t_result;
```

-- 删除模型

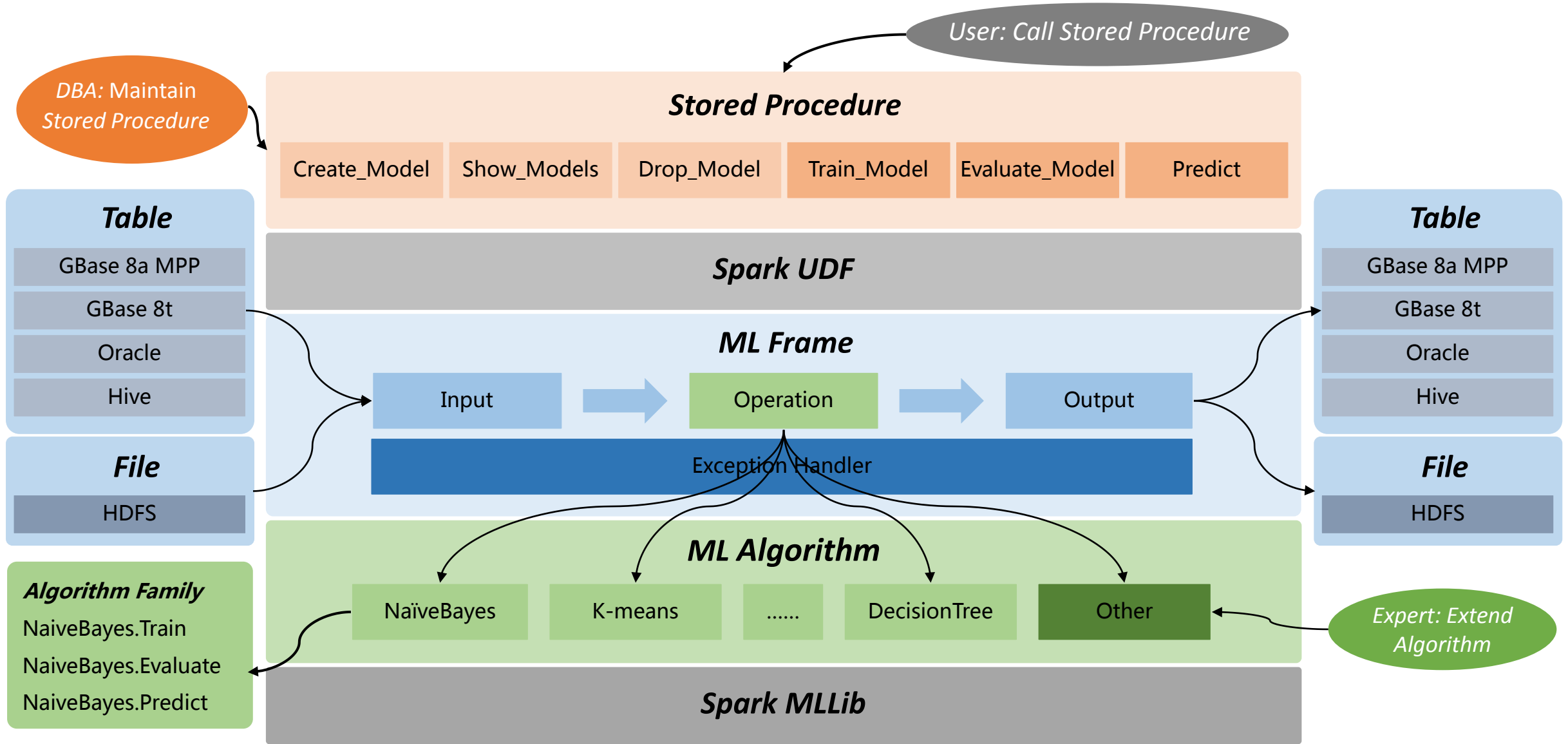
```
Call drop_model( 'moive_rec' );
```

- 数据源：表、视图或文件
- 输出结果：表或文件
- 通过jar包扩展算法

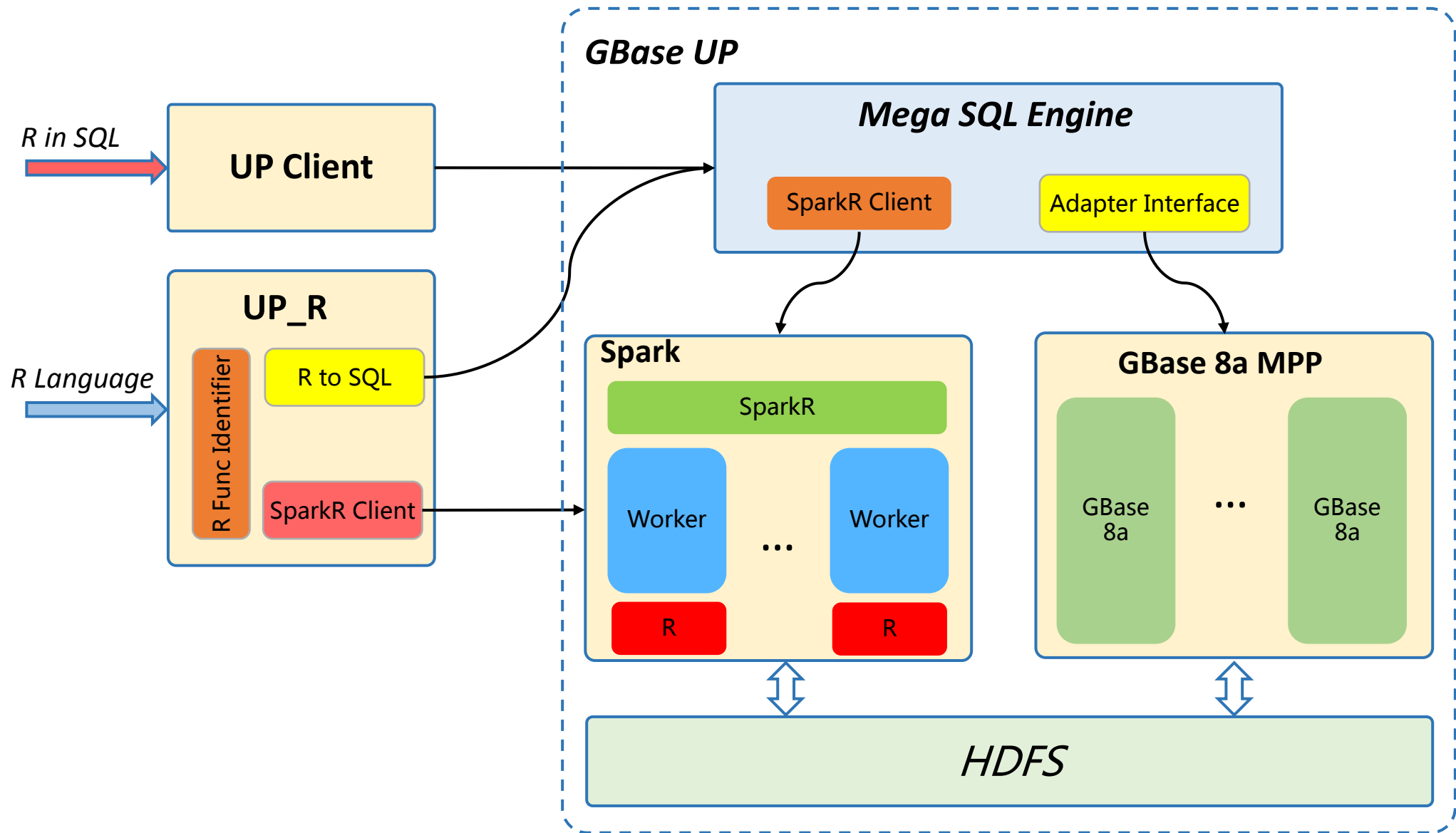
内置算法	主要用途
最小二乘法 (ALS)	推荐
词频-逆文本频率(TF-IDF)	特征提取
支持向量机 (SVM)	分类
朴素贝叶斯 (Native Bayes)	分类
K-均值 (K-means)	聚类
主成分分析法 (PCA)	数据降维
奇异值分解法 (SVD)	数据降维



GBase UP 核心功能：机器学习流程（数据分析）



GBase UP 核心功能：R语言的融合（数据分析）



GBase UP应用案例1 - 混合业务处理（数据联邦）

-- 创建镜像表和Hive表（镜像方向为GBase 8t到GBase 8a MPP）

```
Create table t_mirror(...)
```

```
engine= 'Mirror8t8a' ;
```

```
Create table t_hive(...)
```

-- 写操作

```
Insert into t_mirror values(...);
```

-- 加载Flume流式数据到Hive

```
bin/flume-ng agent
```

```
--conf-file conf/hivesink.conf
```

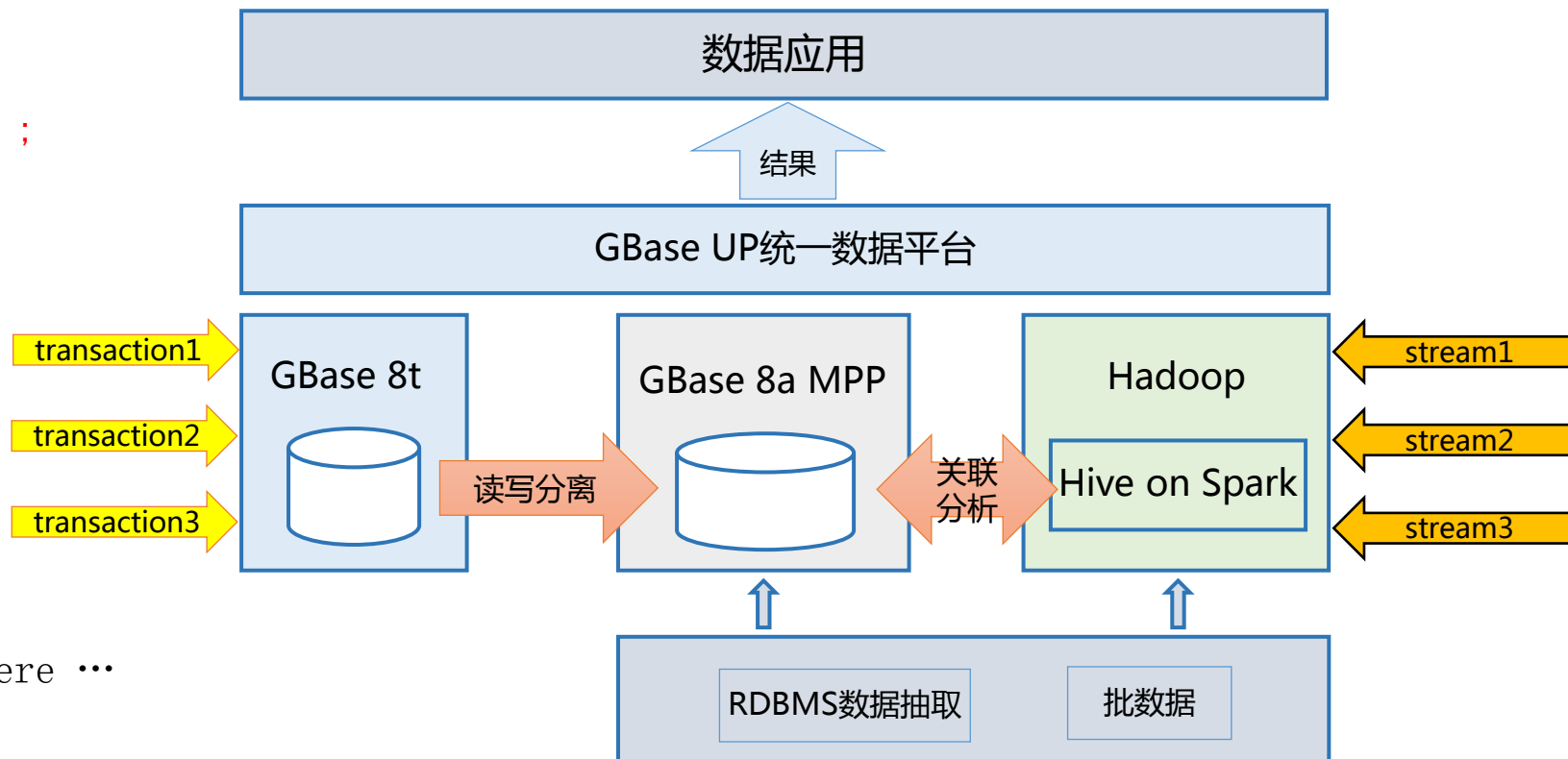
```
--name agent1 ...
```

-- 关联分析

```
Select avg(...)
```

```
from t_mirror,t_hive where ...
```

```
group by ...;
```



GBase UP应用案例2 - OLTP , OLAP混合处理 (数据联邦)

-- 创建各引擎的表

```
Create table t_8t(tag varchar(10))
engine='GBase8t';
Create table t_8a(sender varchar(100),
receiver varchar(100), send_dt datetime)
engine='GBase8a';
Create table t_ft(title text, content text
url) engine='GBaseFt';
```

-- 创建统一视图

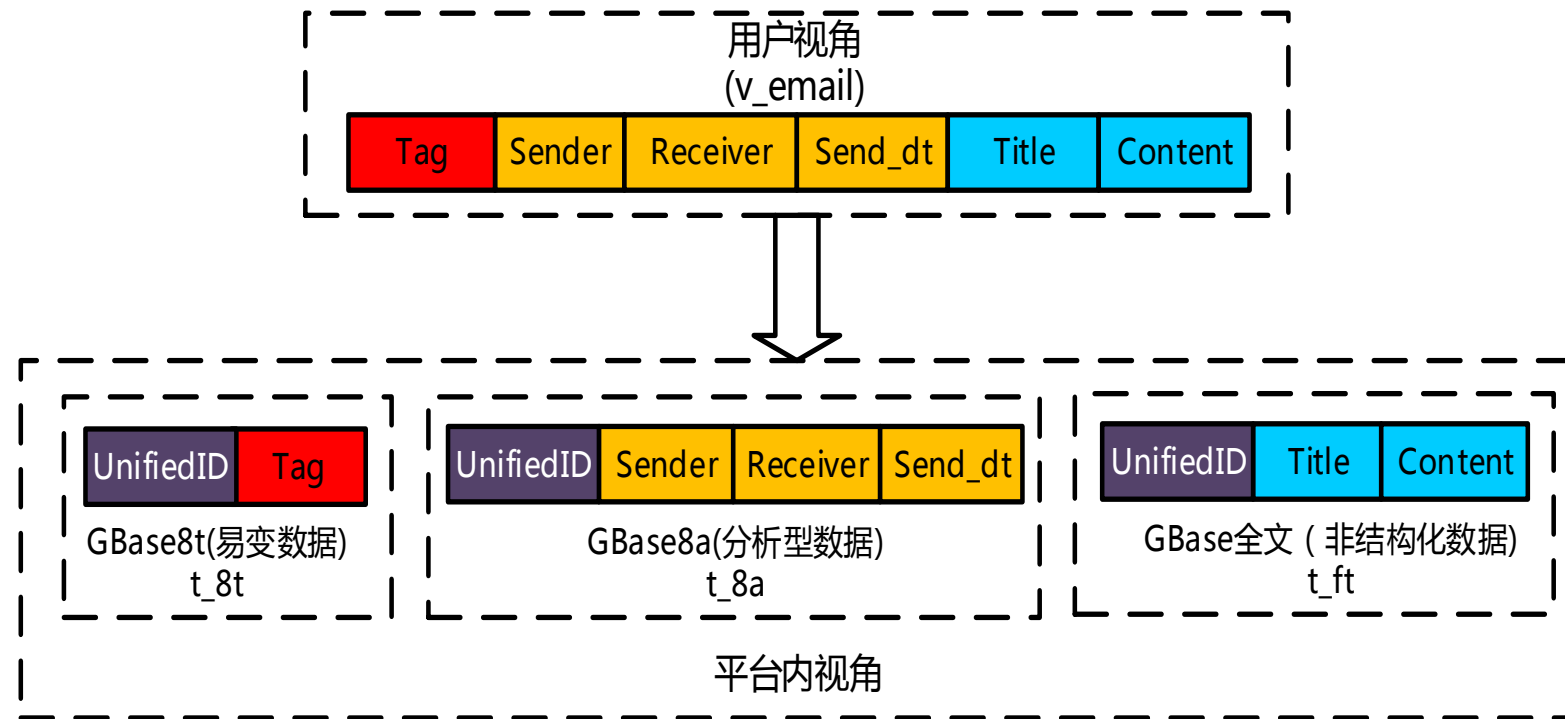
```
Create unified view v_email with table
(t_8t, t_8a, t_ft);
```

-- 更新标签

```
Update v_email set tag = 'checked'
where send_dt < '2016-03-01';
```

-- 统一视图分析

```
Select count(*), sender, receiver
from v_email
where tag = 'checked' and
contains(content, '南大通用')
group by sender, receiver;
```



GBase UP应用案例1 - 用户位置轨迹计算（数据流通）

- **平台执行**

```
Create table t_h (...) engine= 'Hive' ;  
Create table t_8a (...) engine= 'GBase8a' ;
```

- **通过UP的批量处理**

- 云ETL数据入库

```
Load data infile  
'HDP://...' into table t_h fields  
terminated by ...;
```

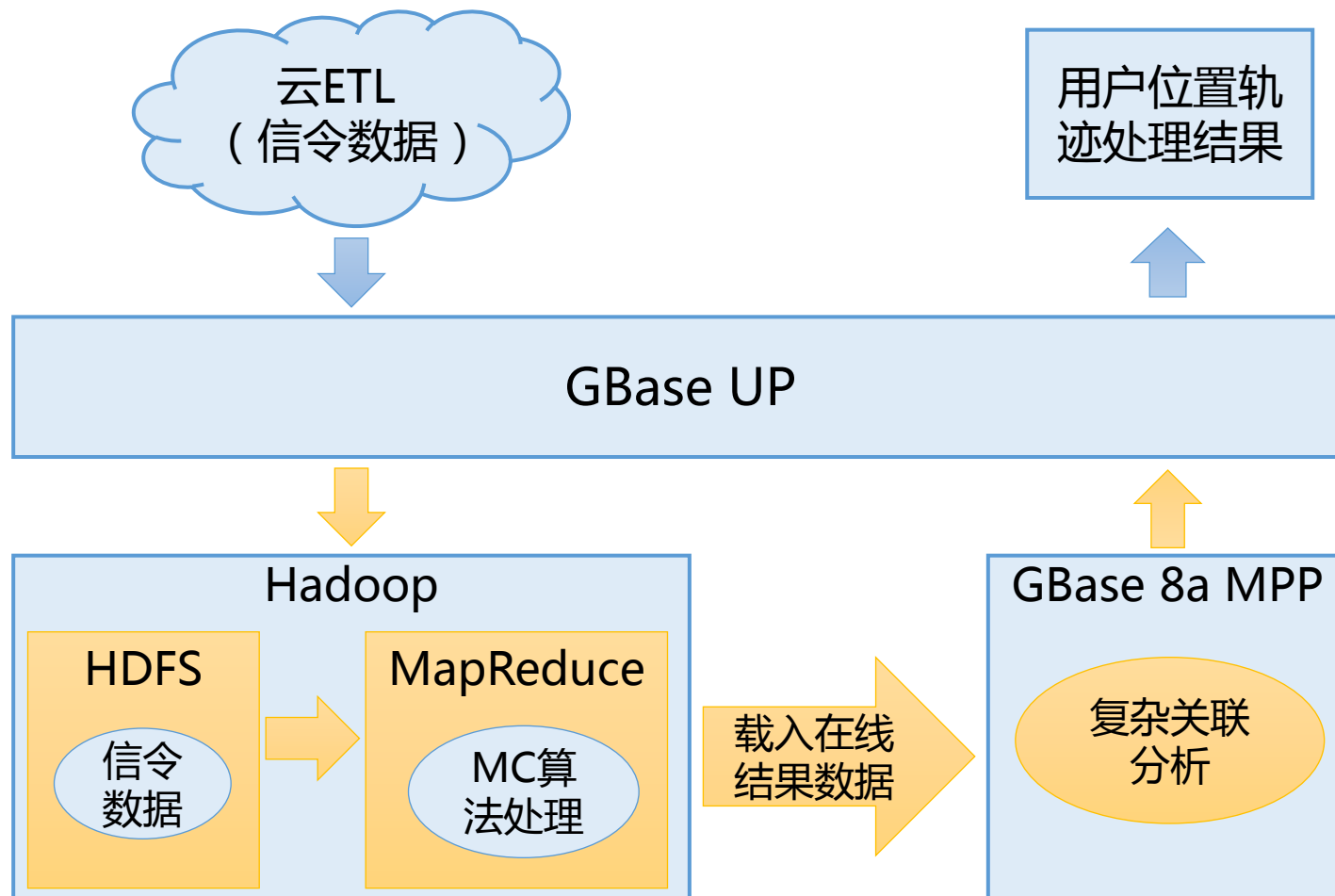
- 通过UDF调用MC标签算法

```
Insert into t_8a select mc(...) from  
t_h ...;
```

- **前端业务**

- 获取用户位置轨迹处理结果

```
Select ... from t_8a where ...;
```



GBase UP应用案例4 - 某电信运营商项目（数据流通）

平台执行

```
create table MCC_VOICE_USAGE...engine=Hive  
create table MCC_SMS_USAGE...engine=Hive  
create table MCC_DATA_USAGE...engine=Hive  
create table .....
```

通过UP的批量处理

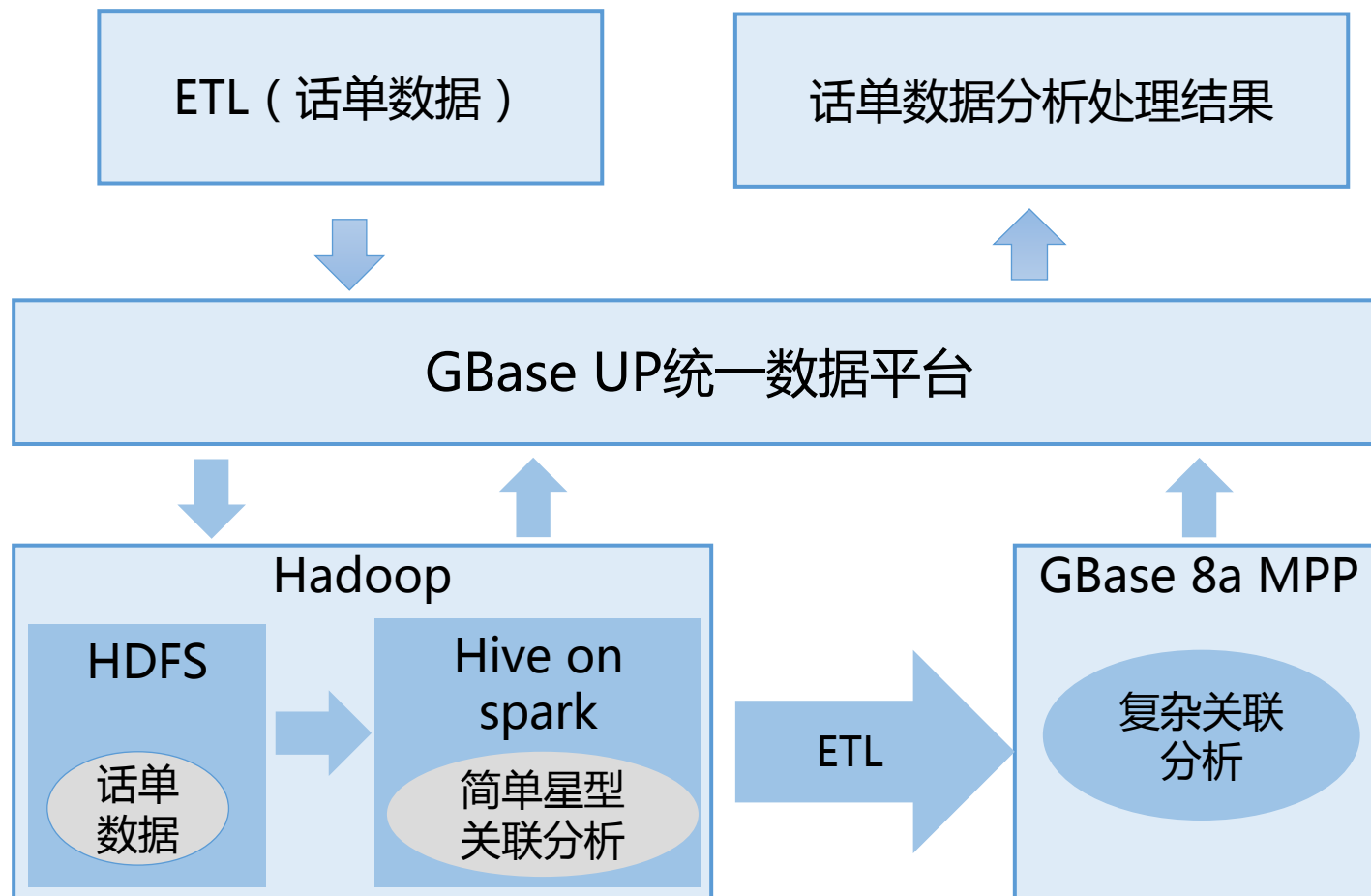
--ETL数据入库示例

```
Load data infile 'HDP://...' into table  
MCC_DATA_USAGE ...;
```

前端业务示例

-- 话单星型关联查询、时段清单查询、账目明细统计、复杂数据加工等业务操作

```
Select ... case when ... from ... left join  
where ... group by ...;
```



目录

- 大数据需求与IT技术架构演变

.....

- SQL（关系）与Hadoop（非关系）的融合之道

.....

- GBase UP 介绍及应用案例

.....

- 总结与展望

.....

Gartner预测市场发展趋势：传统数据仓库模式在下降

● 数据库访问趋势

- ❑ 持续数据加载需求增长26%
- ❑ 日常In-database分析需求增长48%
- ❑ 访问非关系型数据库数据需求增长40%

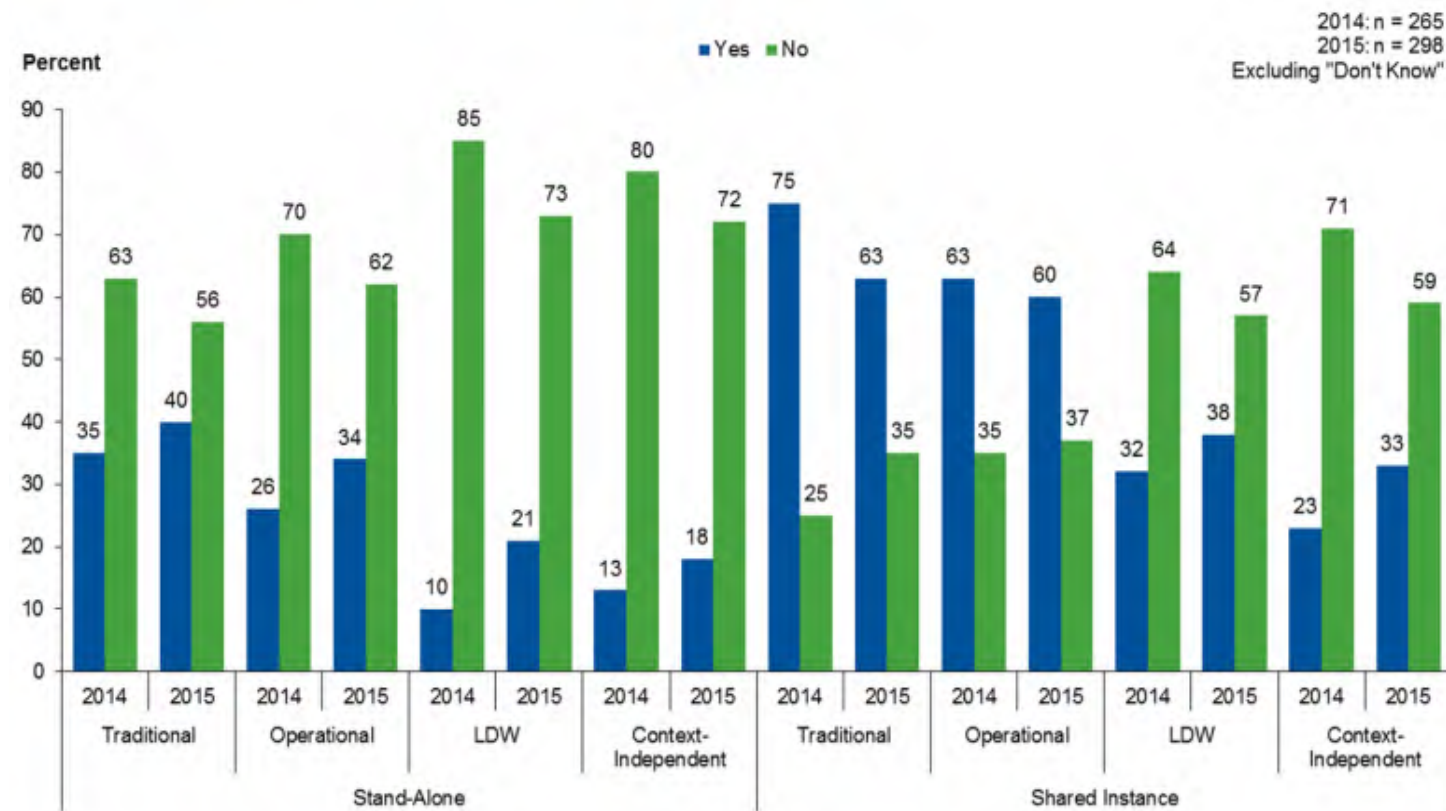
● 数据库应用模式发展趋势

- ❑ LDW的整体需求增长88%
- ❑ 专用Operational DW需求从26%增长到34%
- ❑ Context-independent DW需求增长接近50%
- ❑ 传统数据仓库需求比例略有下降，表明用户需求向着其他应用模式发展

● 总体趋势

- ❑ 数据融合
- ❑ 数据流通
- ❑ 数据分析
- ❑ **与GBase UP产品定位高度一致**

Figure 1. Adoption of Data Warehousing and DMSA Use Cases



Survey question: How is the vendor's DBMS being used for the indicated use case?

Source: Gartner (June 2016)

Gartner预测应用模式 1：LDW (逻辑数据仓库)

● 解决问题

- 支持多个数据源以及多种类型数据的综合分析场景
- 双峰 (bi-modal) 工作模式

从 查询单一数据仓库	到 查询多个数据仓库及数据源
从 SQL	到 SQL、Java、Perl、Ruby、Python、R
从 结构化数据	到 结构化及多元结构化数据 (XML、JSON、博客)
从 业务用户	到 业务用户及开发人员
从 报表/整体查询	到 报表/整体查询, 以及数以千计的数据库内分析 算法

● 关键技术

- **数据虚拟化** (Data Virtualization)
 - 通过在多个数据库之上建立一个中间层, 实现了统一接口统一方式访问数据源, 达到了对上层应用展现为统一数据视图的效果
- **数据联邦** (Data Federation)
 - 通过**联合查询** (Federate Query) 技术从每个数据源并行抓取数据, 完成跨数据源访问

● GBase UP相关技术

- GBase UP通过Mega SQL Engine实现了统一接口统一访问, 对用户呈现统一视图
- GBase UP原生融合OLTP, OLAP以及Hadoop系统, 支持多个数据源以及多种数据类型

Gartner预测应用模式 2 : Operational DW(运营数据仓库)

● 解决问题

- 新数据能够持续地，实时地加载到数据库中
- 可以对新数据进行实时分析并马上看到结果
- 实时指导业务运营

● 关键技术

➤ In-Database 大数据分析

- 通过数据库上的分析功能直接分析数据，不需要将数据拉到分析工具里分析（比如SAS等）
- 数据库通过加载UDF函数扩展机器学习算法能力

➤ 流数据分析

- 实时接入流数据（Kafka等工具）并根据数据时间窗口进行分析

● GBase UP相关技术

- GBase UP内置Spark MLlib机器学习算法，并通过高速数据交换通道使得深度分析算法可以作用在任意数据源上
- GBase UP无缝连接流数据源（Kafka等工具），可持续接入流数据并实时分析
- GBase UP通过“UDF扩展框架”支持数据运营分析能力的可持续化提升

Gartner预测应用模式3：Context Independent DW(无模型数据仓库)

● 解决问题

- 传统数据模型无法有效，实时地支持到深度数据挖掘
- 数据价值需要通过复杂机器学习算法才可以获取

● 关键技术

- 基于R语言进行无模式数据的复杂机器学习和关联分析
- 支持进行文本分析挖掘的计算引擎和算法库（比如Spark MLlib）
- 支持进行图分析的的计算引擎的算法库

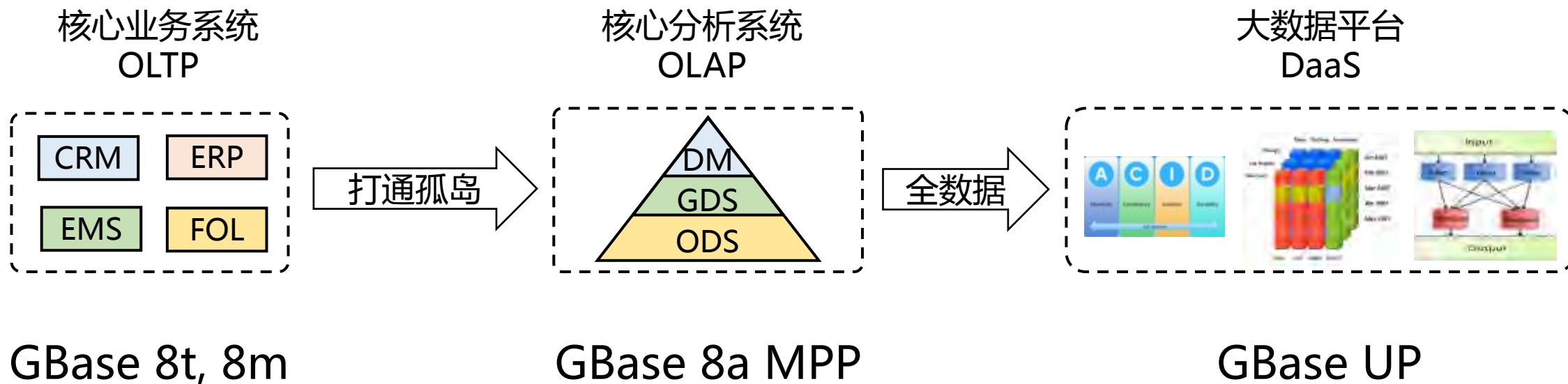
● GBase UP相关技术

- GBase UP内置Spark MLlib机器学习算法，可以进行文本等非结构数据的深度分析
- GBase UP自主开发的全文分析引擎“GBaseFt”，将SQL功能和文本分析功能完美融合，极大简化用户使用方式
- GBase UP集成R语言分析功能，支持非常丰富的非结构化数据数据能力
- GBase UP通过“UDF扩展框架”提供理论上可无限扩展的机器学习算法能力

Gartner预测应用模式总结

数据库模式	解决问题	核心技术	GBase UP大数据平台
LDW (逻辑数据仓库)	<ul style="list-style-type: none"> <input type="checkbox"/> 多个数据源以及多种类型数据的综合分析场景 <input type="checkbox"/> 双峰 (bi-modal) 工作模式 	<ul style="list-style-type: none"> <input type="checkbox"/> 数据虚拟化 (Data Virtualization) 实现统一接口, 统一访问方式 <input type="checkbox"/> 数据联邦 (Data Federation) 实现跨数据源数据访问 	✓ 支持
Operational DW (运营数据仓库)	<ul style="list-style-type: none"> <input type="checkbox"/> 实时加载数据 <input type="checkbox"/> 实时分析数据 <input type="checkbox"/> 实时指导业务运营 	<ul style="list-style-type: none"> <input type="checkbox"/> In-Database 大数据分析 <input type="checkbox"/> 流数据分析 	✓ 支持
Context-independent DW (模型无关数据仓库)	<ul style="list-style-type: none"> <input type="checkbox"/> 深度数据挖掘 <input type="checkbox"/> 复杂机器学习 	<ul style="list-style-type: none"> <input type="checkbox"/> R语言无模式数据的深度机器学习 <input type="checkbox"/> 文本分析挖掘的计算引擎和算法库 <input type="checkbox"/> 图分析的计算引擎和算法库 	✓ 支持
Scalable traditional DW (高扩展传统数据仓库)	<ul style="list-style-type: none"> <input type="checkbox"/> 数据量大 <input type="checkbox"/> 数据库上应用多 	<ul style="list-style-type: none"> <input type="checkbox"/> MPP数据库 <input type="checkbox"/> Scale-out计算和存储能力扩展 	✓ 支持

总结：GBase新的视野 - DaaS数据即服务



GBase 产品的发展轨迹

公司简介：自2004年聚焦数据库与大数据解决方案

- 2015年7月在新三板挂牌：**“通用数据”**，**股票代码：“833056”**
- 主营业务：**数据库产品、大数据平台研发与销售，数据服务，数据工程**
- **注册资金1.087亿，员工670人**
- **2015年收入2.11亿元，净利润4000万元**
- **2010至2015年连续获评“国家规划布局内重点软件企业”**
- **股东：创业团队、员工 + 中兴、国投、天创、达晨、东华软件、银信长远、荣之联等**

公司简介：超过1000个客户，13个国家

GBase 8a在国内MPP数据库市场占有率第一，运行节点数量超过其他厂商总和，并开始走向世界。



节点数 > 2300个

总数据量 > 15PB

用户覆盖13个国家

- 软件发掘数据价值 -

谢谢

GBASE

天津：中国天津华苑产业区海泰发展六道6号海泰绿色产业基地J座

电 话：022-58815678 传 真：022-58815679

北京：北京市朝阳区太阳宫中路12号太阳宫大厦10层1008室

电 话：010-88866866 传 真：010-88864556

网 址：<http://www.GBASE.cn> E-mail:info@GBASE.cn

技术支持热线：400-817-9696

