Lambda on OLAP

背景：

随着业务的不断发展，部分业务不满足于只查看历史存量数据，还需要对数据进行实时查询和分析，并与历史数据进行对比。例如：某业务进行一场运营活动，活动期间需要实时的知道一些运营指标。与此同时，还需要和以往的运营活动指标进行对比，用来辅助决策。

关键需求：

快速查询

实时分析

海量数据

对比部分历史指标

Elasticsearch

满足：

- ✓ 快速查询
- ✓ 实时分析
- ✓ 海量数据

不适宜：

长期存储（存储资源需求大，成本较高）

Kylin

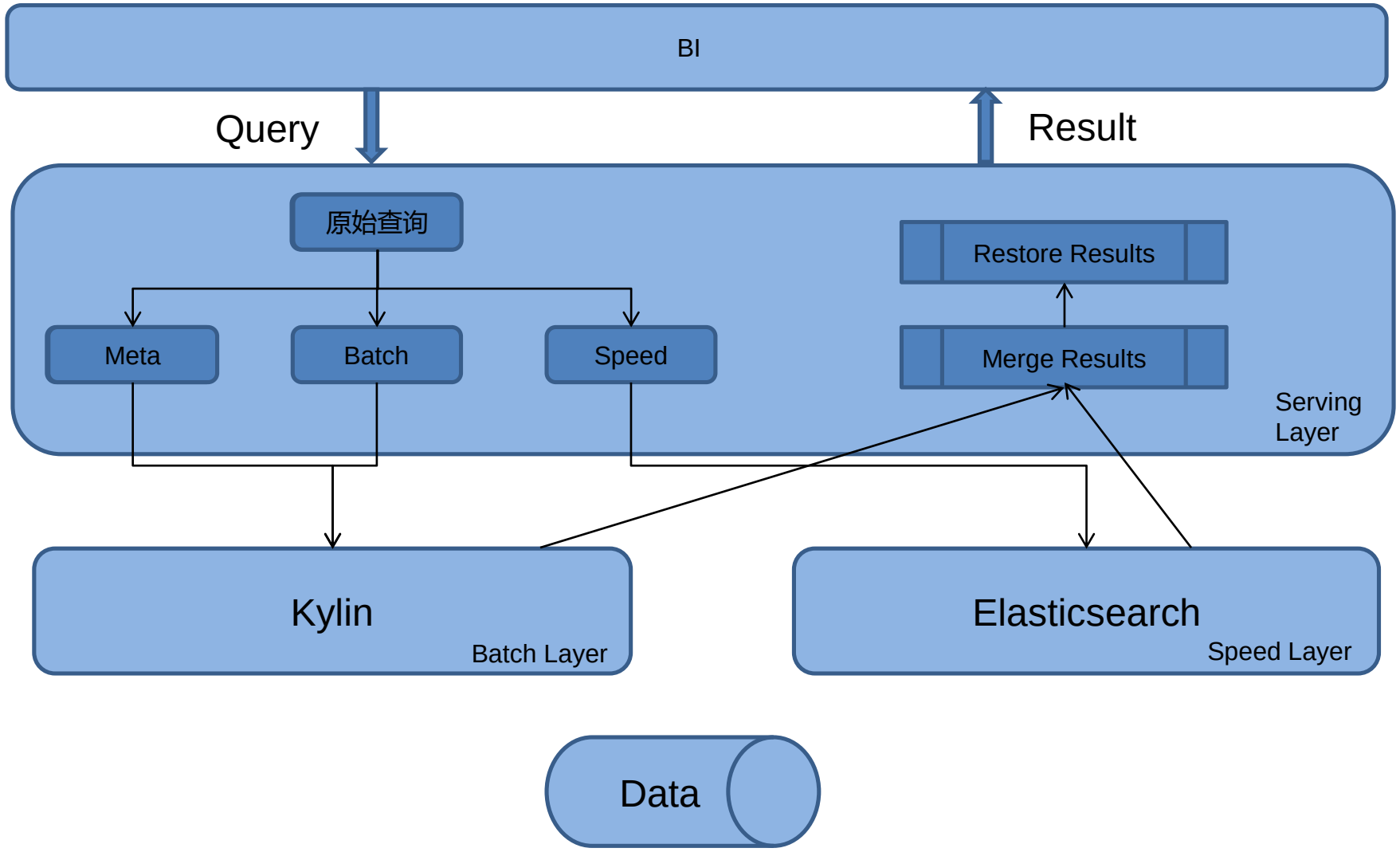
满足：

- ✓ 快速查询
- ✓ 实时分析
- ✓ 长期存储

不适宜：

实时分析

结合公司现有的大数据解决方案，我们设计了如下架构：



主要流程

1.用户发起Query请求到数据服务层

2.数据服务层查询Meta

3.根据查询时间分发请求到Batch Layer和Speed Layer

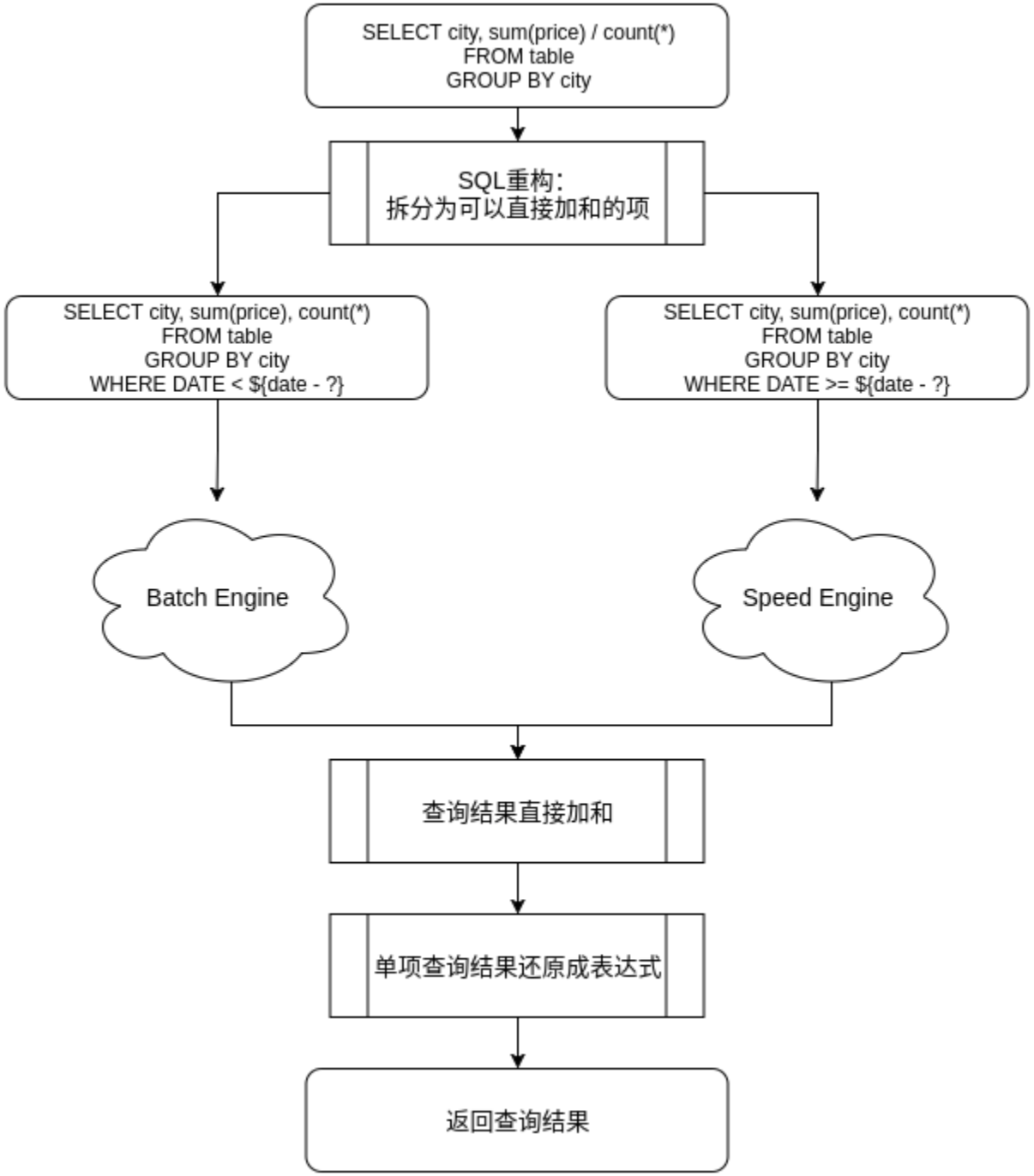
4.收到Batch Layer和Speed Layer返回的结果后，进行合并

5.统一格式后返回结果

服务层

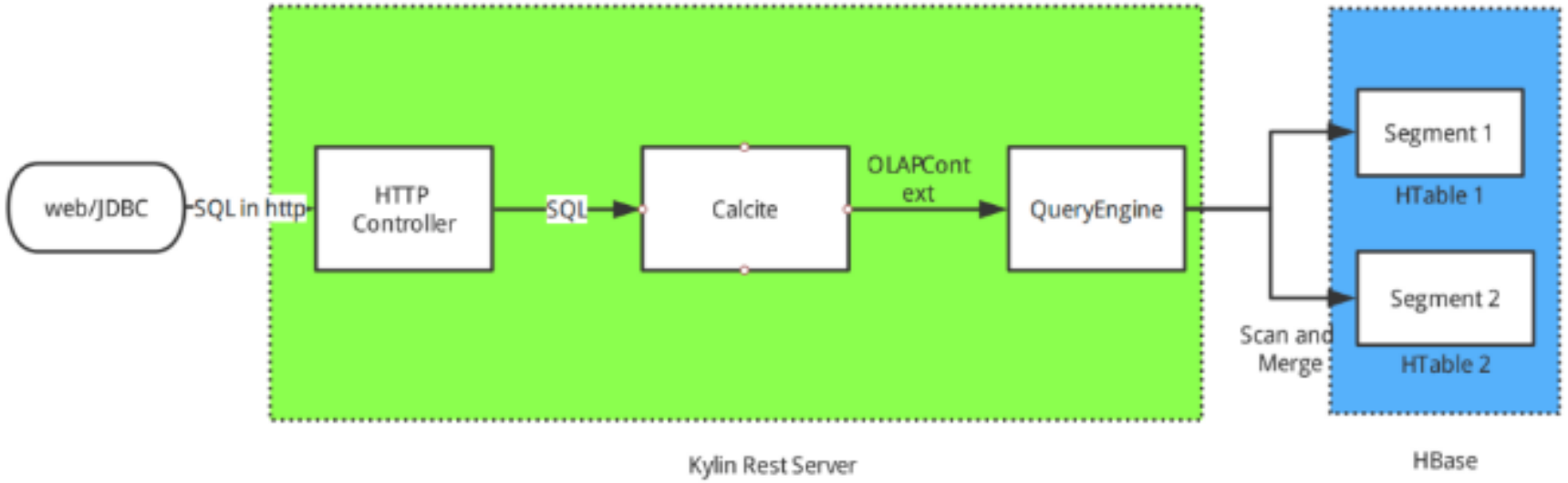
服务层主要处理的是对于SQL的重排以及结果的聚合工作。当服务层接收到一次查询请求时，首先会对SQL进行重构：1. 添加Batch/Speed层的时间限制; 2. 将SQL中无法直接加和的项进行拆分。拆分后的结果会分发给对应的引擎，在引擎返回结果后，查询结果会被格式化成一致的查询结果，随后进行逐项加和。最后会基于单项结果推导原始查询中各项表达式的结果，并以统一的形式返回给用户。

服务层



Batch层

Kylin的核心思想是通过预计算，将查询结果缓存到HBase中。每次查询的时候，只需要去找寻已经计算好的结果进行返回。



Speed层

ES由于有索引支撑，整体查询速度能做到秒级查询。但是由于SQL并非ES原生支持的语法，所以查询语法相对受限，比较适宜一些查询形式相对固定的报表需求。

https://github.com/NLPchina/elasticsearch-sql

其他

Lambda主要在以下方面帮助用户：

1.无需了解多个存储、计算引擎，用简单的SQL语法，就能快速获取最实时的数据；

2.完善的权限管控与访问审计

3.统一了数据统计口径和标准，可以服务多个业务方

谢谢！

专业、垂直、纯粹的 Elastic 开源技术交流社区

https://elasticsearch.cn/