金融大数据整合之路

Thomson Reuters – 深度定制Spark
November 18 2016

自我介绍

沈勇， Raymond Shen

Thomson Reuters

研发经理，研究员

实时增值数据，大数据，机器学习

BEA Systems

高级软件开发工程师

中间件，WebLogic, Tuxedo ， CORBA

MetaData Systems

项目管理负责人

电信级分布式系统

内容

- 数据整合之困
- 概念分析
- 关键技术详解

挑战无处不在

Thomson Reuters是最早的Fin-Tech

The intelligence, technology and human expertise you need to find trusted answers.

the answer company™
THOMSON REUTERS®

挑战无处不在

大量的老旧数据库

- 上百个数据库系统实例，数十个Content Master
- Oracle, MS SQL，Sybase Vector Wise, Oracle Exadata，MySQL
- 数据库之间相互拷贝。
- 复杂的连接，重复的工作

挑战无处不在

复杂的信息模型

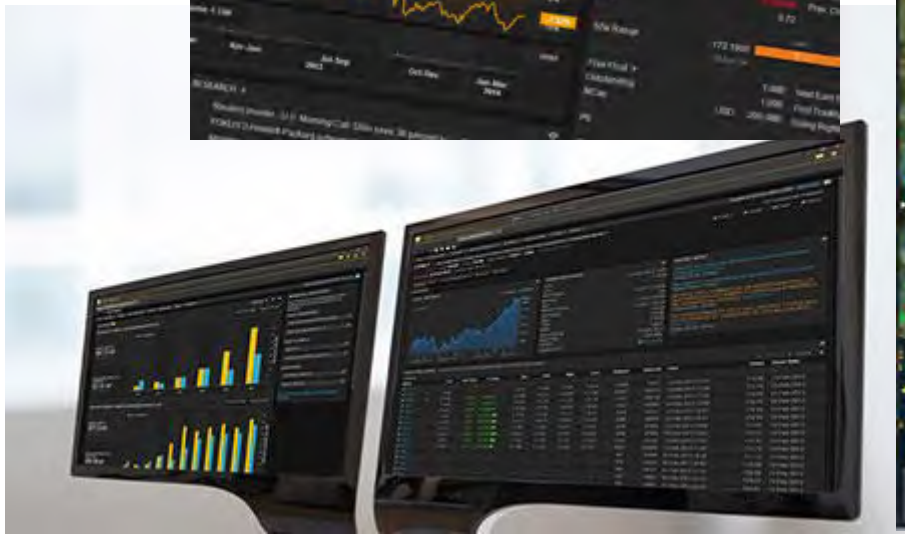
•以结构化半结构化数据为主。

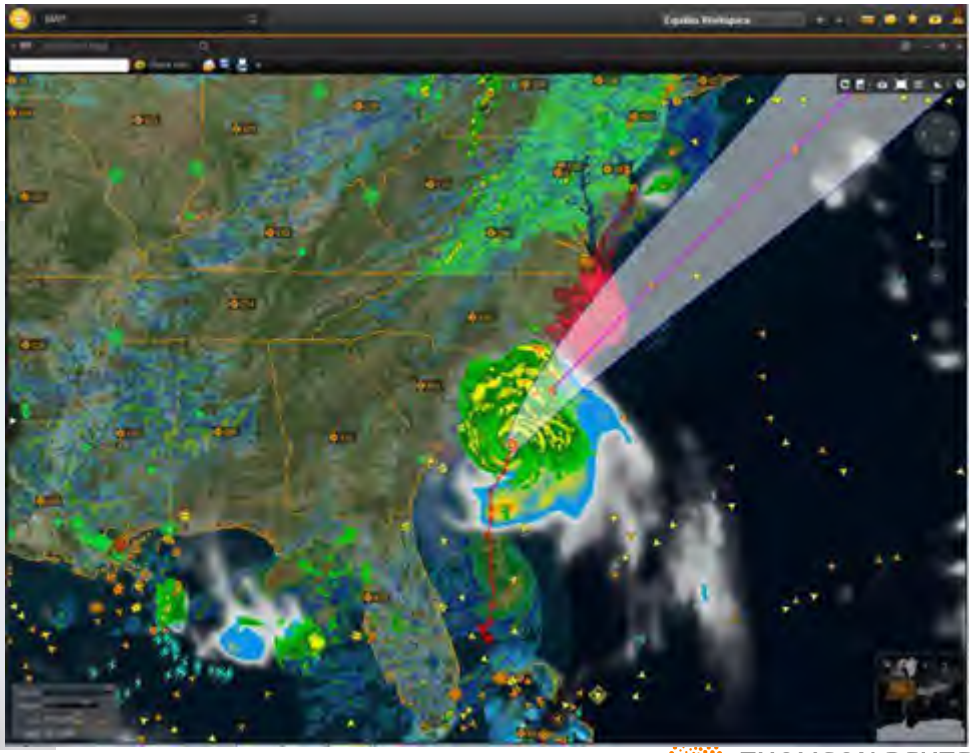
•大量的基于SQL数据库模型。

•针对不同的业务，有不同的数据抽象。

•股票，债券，期权，大宗交易，组织机构信息，企业高管信息，企业基本面，企业并购，宏观经济，法律文本，判决书，新闻，小道消息， 社交媒体，气象…

•复杂的元数据管理，数据格式信息混乱





挑战无处不在

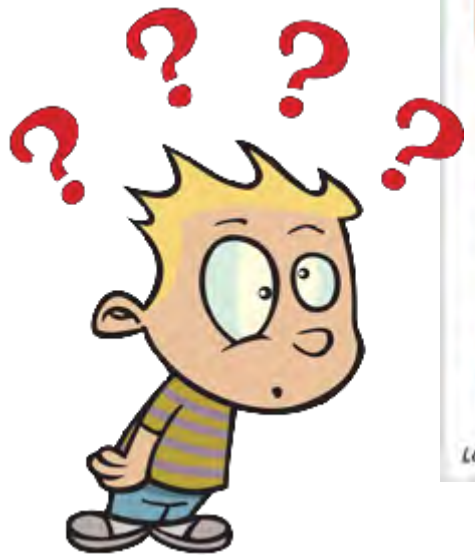
高难度的用户要求

- 数据准确性是前提
- 越来越多的案例需要多种数据源。
- 数据之间的连接成为必须。
- 数据查询的复杂性大大增加
- 数据的实时性逐步提高。
- 快速变化的用户需求。

挑战无处不在

大数据还不完美

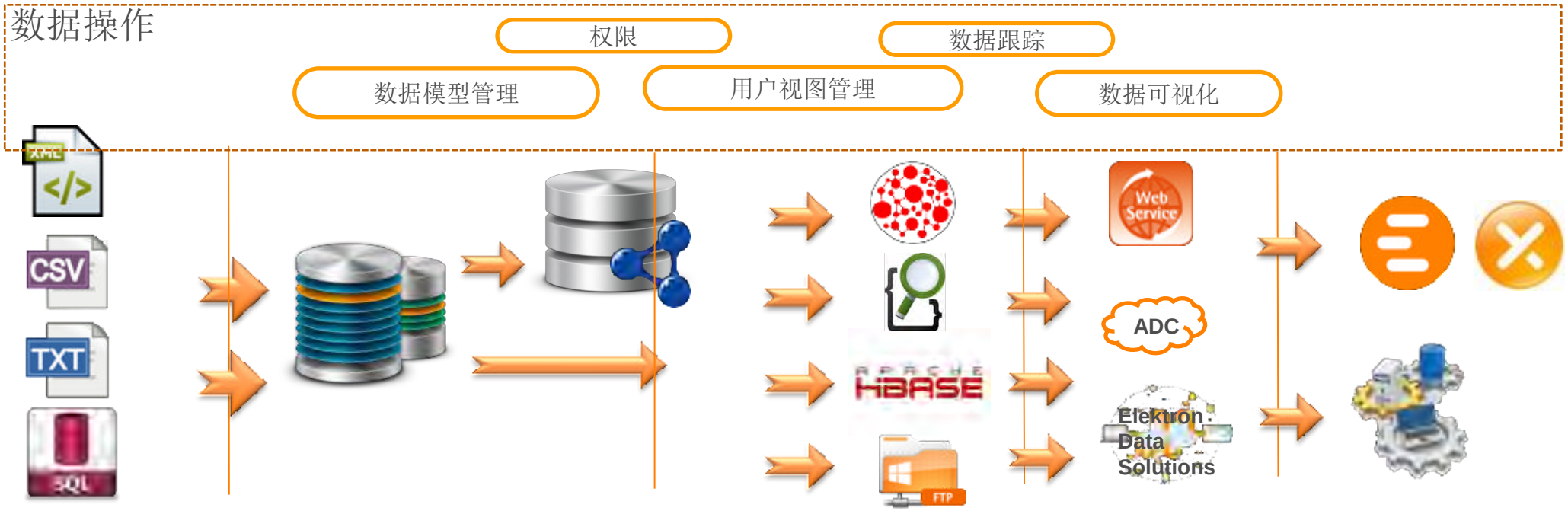
- SQL vs. No-SQL (Columnar, Document, Graph)
- Batch vs. Streaming
- Transaction vs. Eventually Consistent
- Yarn vs. Mesos
- Spark vs. Flink
- HBase vs. Cassandra
- Cloud vs. in-house
- AWS vs. Azure
- Virtualization vs. Container



概念

- 系统整体流程
- 数据标准化
- 数据增强
- 图数据库转换

系统整体流程



原始数据

- HDFS
- 从外部获取数据
- 内部数据库
- 自动发现格式
- 可以通过标注建立更多的数据关系

核心数据库

- 二进制存储
- 元数据
- 三元组存储

衍生数据库

- 产生
  - 子图
  - 文档数据库
  - 文件服务 (HDFS, FTP，S3)
  - 列数据库
- 数据连接和增强

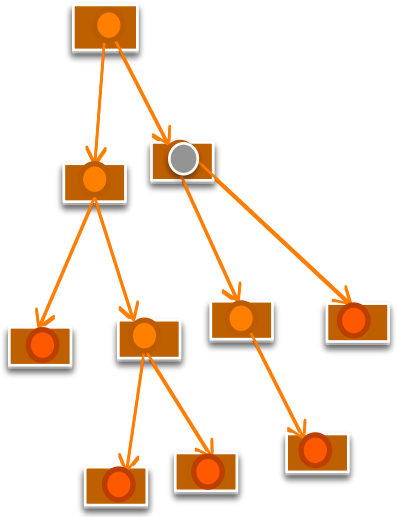
服务层

- Web Service
- 流
- 数据云API
- EDS 2.0

用户

- 终端
- 应用程序

数据标准化存储

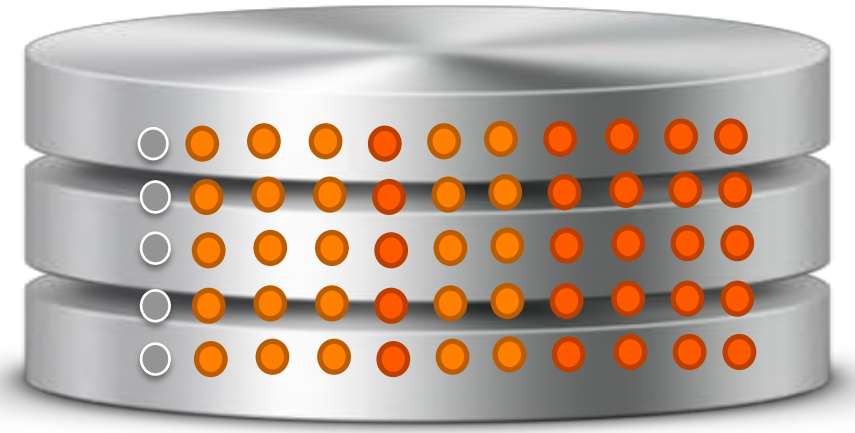


原始数据
- 弱类型
- 层级关系
- 树形结构

二进制的元数据
- 保留层级关系
- 自动探测数据类型

标注
- 主键
- 外键
- 具体数据类型

二进制数据
- 支持层级结构
- 串行化的
- 强类型的

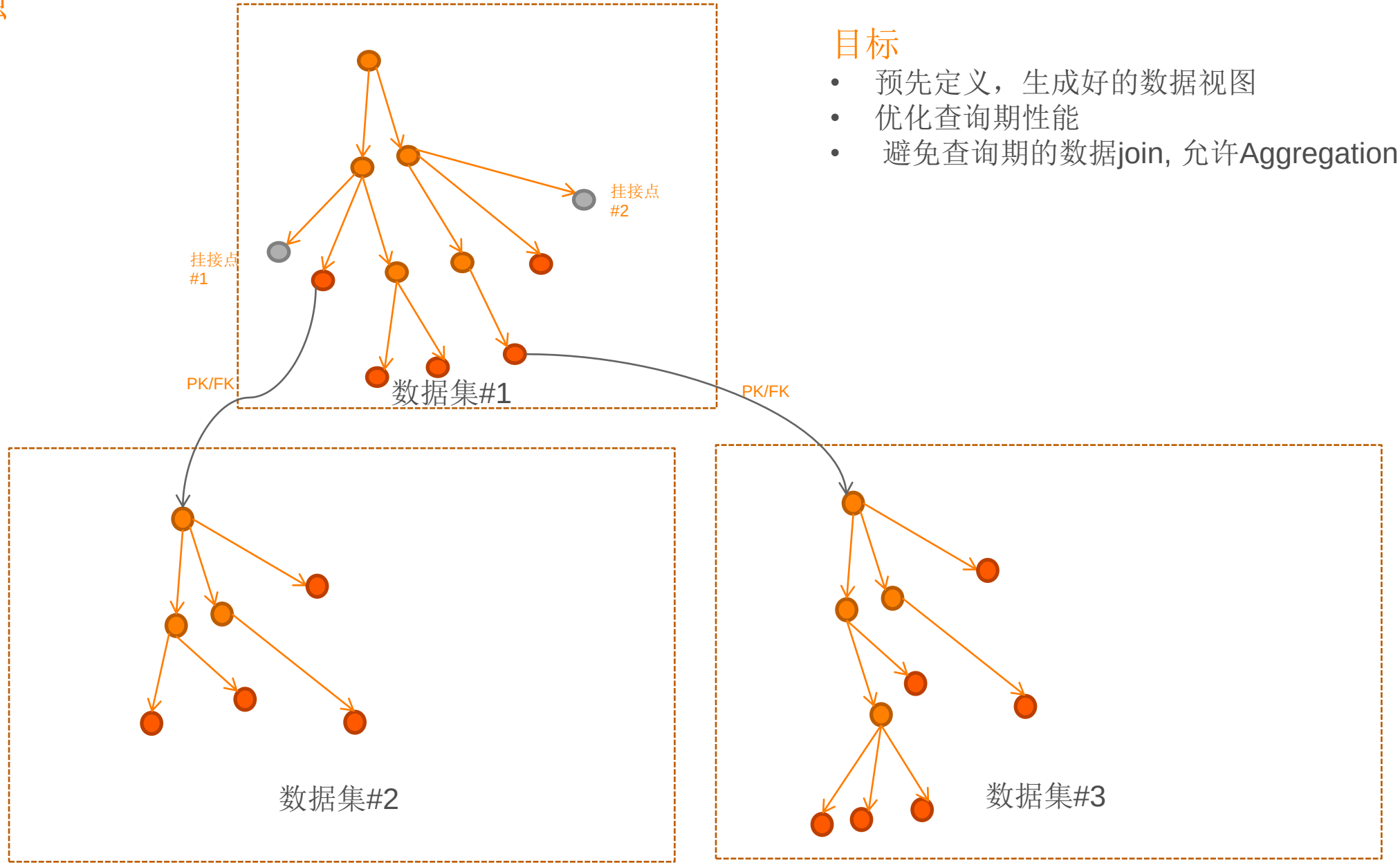


二进制数据存储
- HBase
- Avro Data 序列化饭序列化
- 根据标注产生主键
- 可检索的数据

目的
- 从原始数据中提取数据的类型和结构信息。
- 弱类型到强类型。类型标注
- 统一存储格式，
- 统一查询方法

文档增强

二进制数据

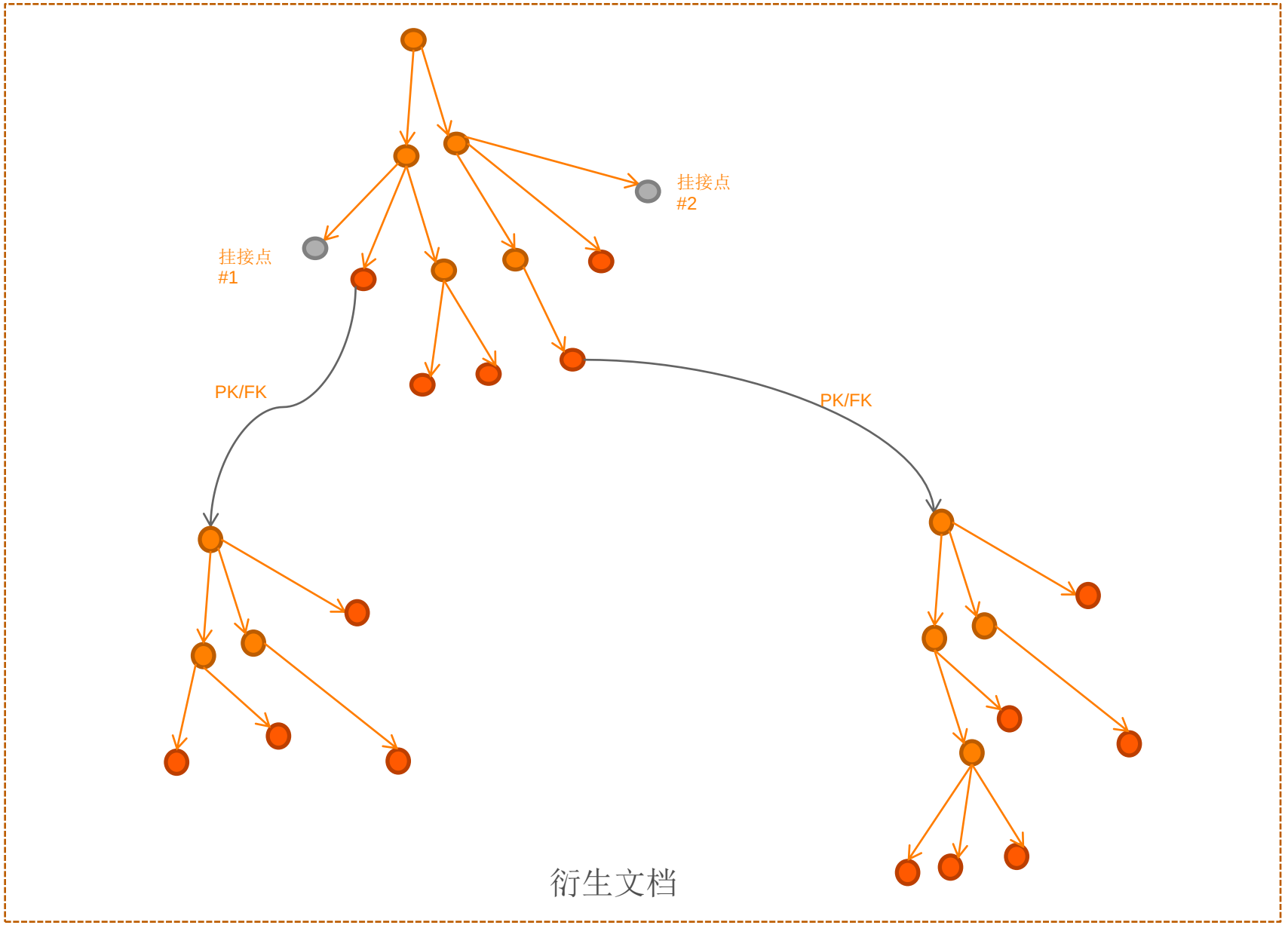


目标

- 预先定义，生成好的数据视图
- 优化查询期性能
- 避免查询期的数据join, 允许Aggregation

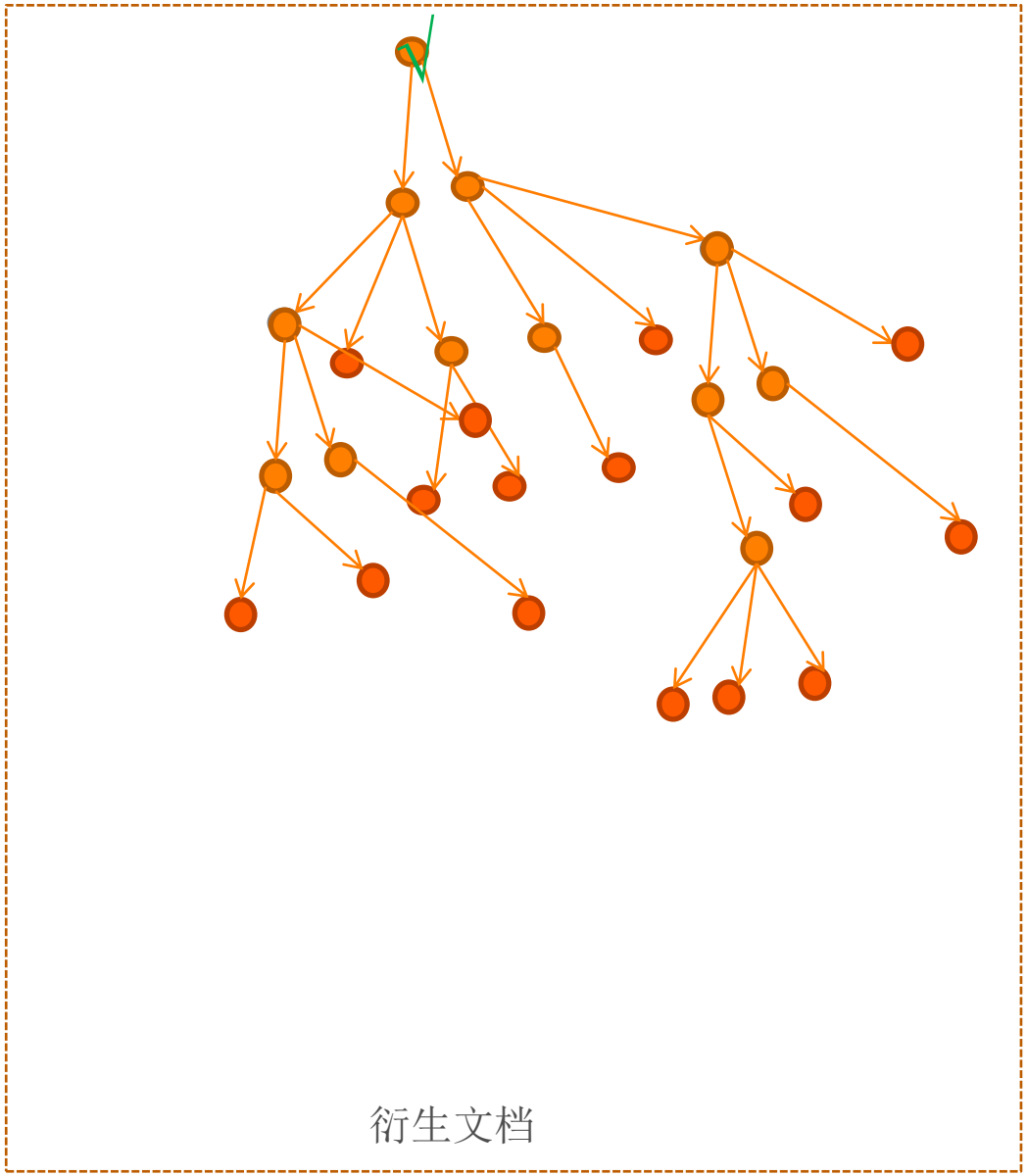
文档增强

文档连接



文档增强

连接后的操作



衍生文档

过滤

- 通过过滤条件判断文档是否保留

属性选取

- 根据预先定义的条件保留某些属性值

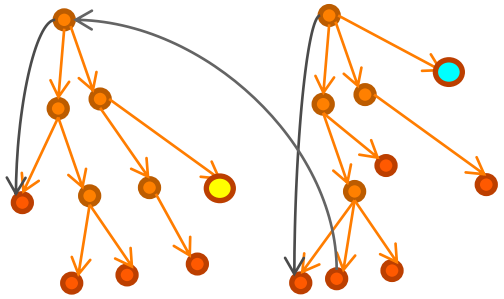
格式转换

- 重新组织文档结构
- 使用用户自定义函数进行特定的格式转换

生成三元组

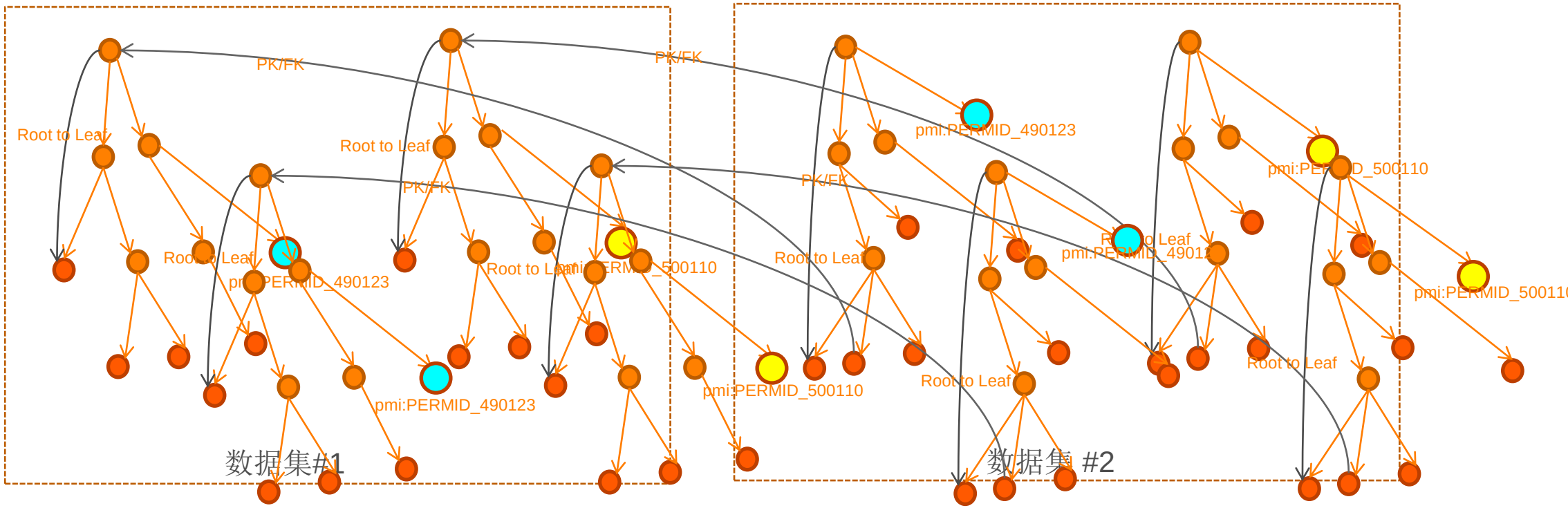
三元组= (主题, 谓词, 对象)

- 转换树形结构到图数据库结构
- 数据连接基于，固有层级关系，跳级关系，主键外键连接，Perm ID连接
- 可以跨越数据集之间进行广泛的连接。



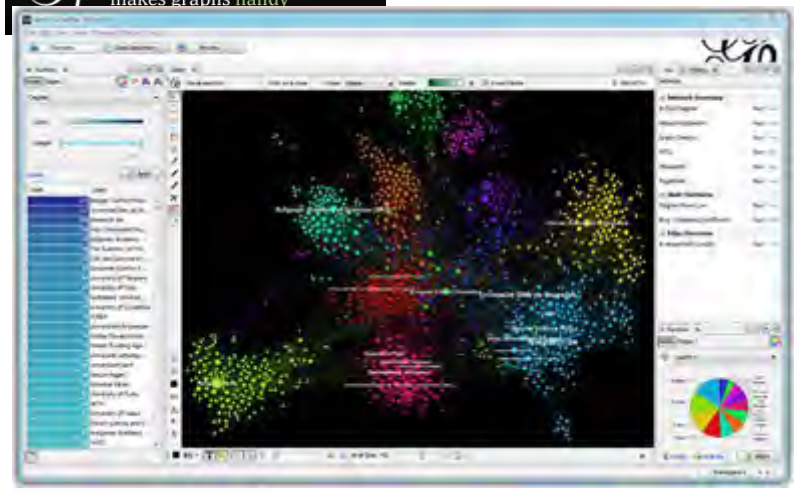
图元数据

- 通过二进制存储的元数据，
- RDFS , OWL – 类, 属性, 限制

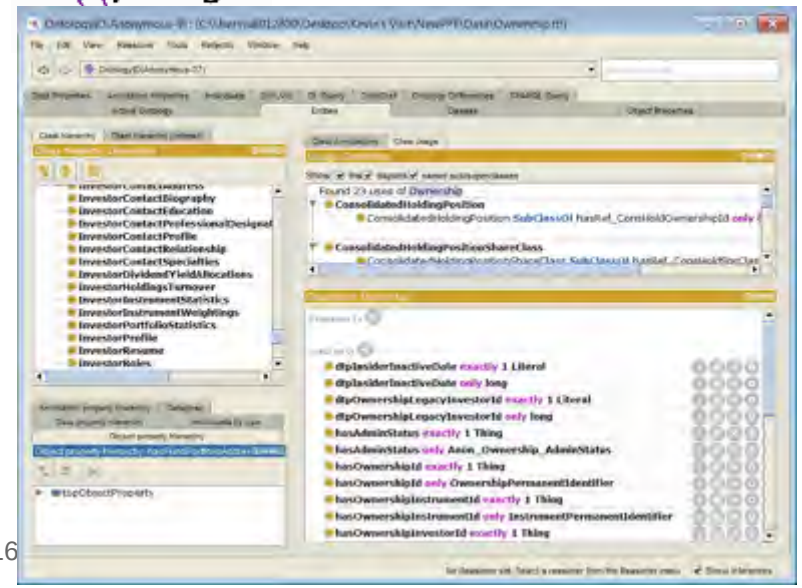


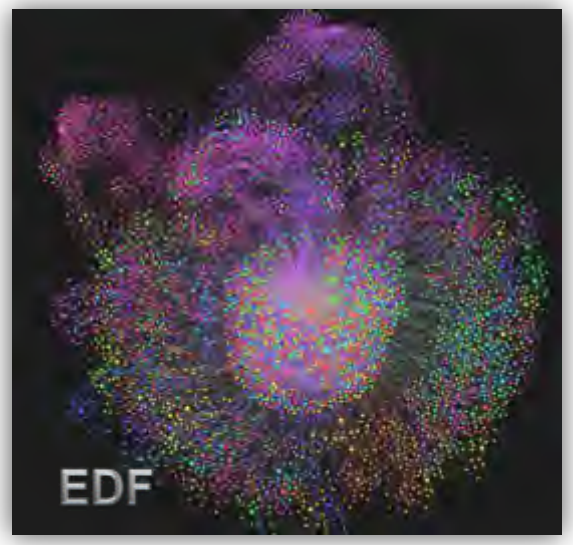
图数据库工具

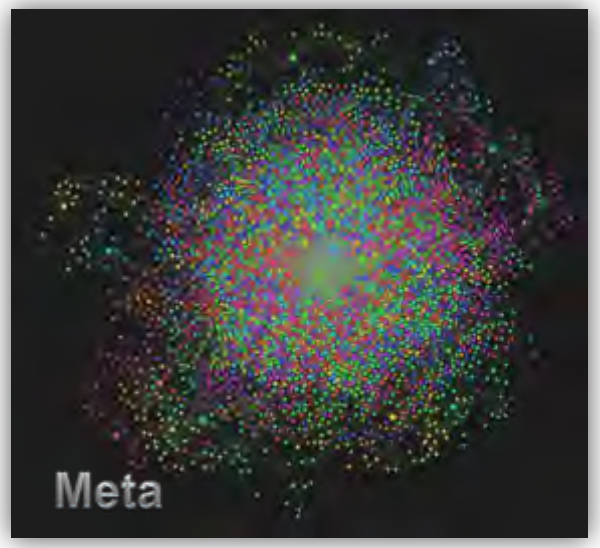
Gephi
makes graphs handy

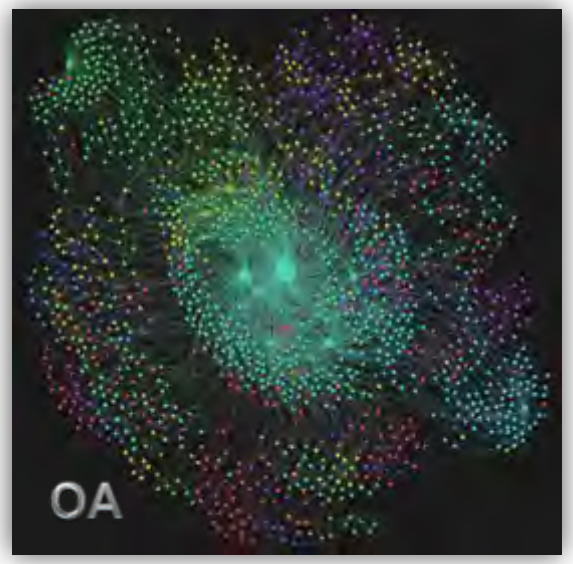


protégé









子图选取

SPARQL

目的

- 批量导出数据到具体的图数据
- 缩减数据量，以快速响应查询

方法

- 元数据相关
  - 谓词选取
  - 类型选取
- 数据值相关

#谓词选取

```
PREFIX own: <http://www.thomsonreuters.com/ownership#>

CONSTRUCT {
    ?s1 own:dtpRTLInvestorType ?o1.
    ?s2 own:dtpRTLInvestorFullName ?o2.
}
FROM <demo>
WHERE
{
    {?s1 own:dtpRTLInvestorType ?o1 .} UNION
    { ?s2 own:dtpRTLInvestorFullName ?o2. }.
}
```

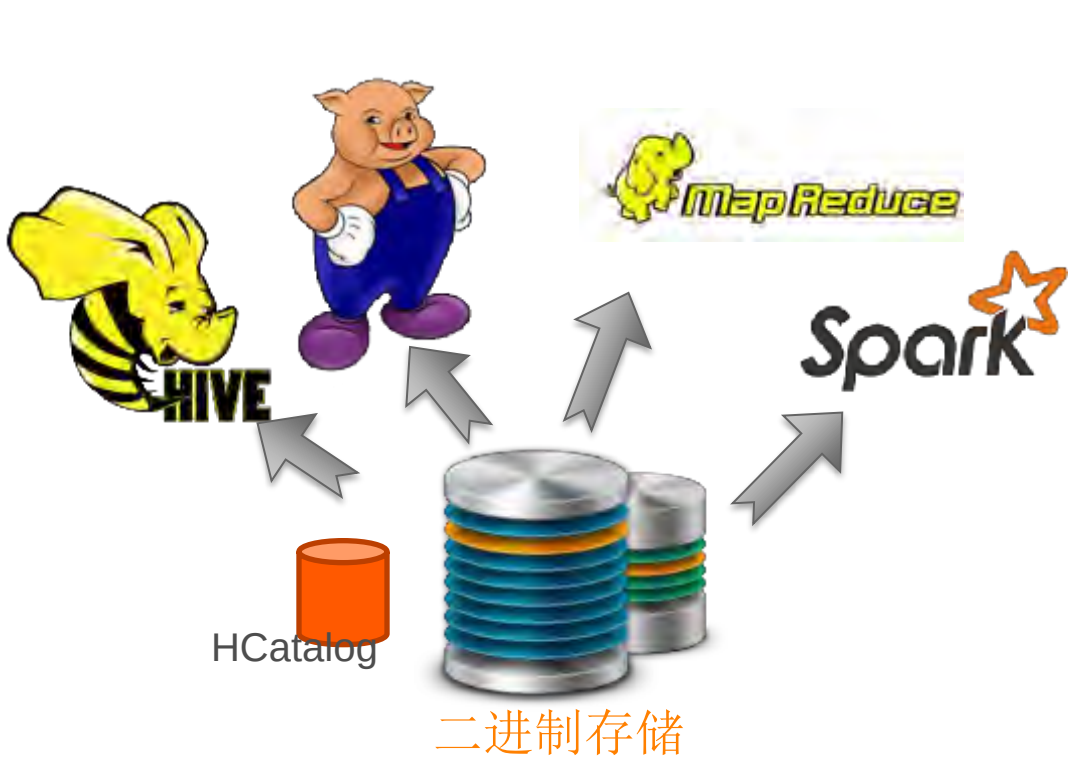
#类型选取

```
PREFIX rdfs:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

CONSTRUCT {
    ?s ?p ?o.
}
FROM <demo>
WHERE
{
    { ?s rdfs:type pmi:InstrumentPermanentIdentifier.}
    { ?s ?p ?o. }.
}
```

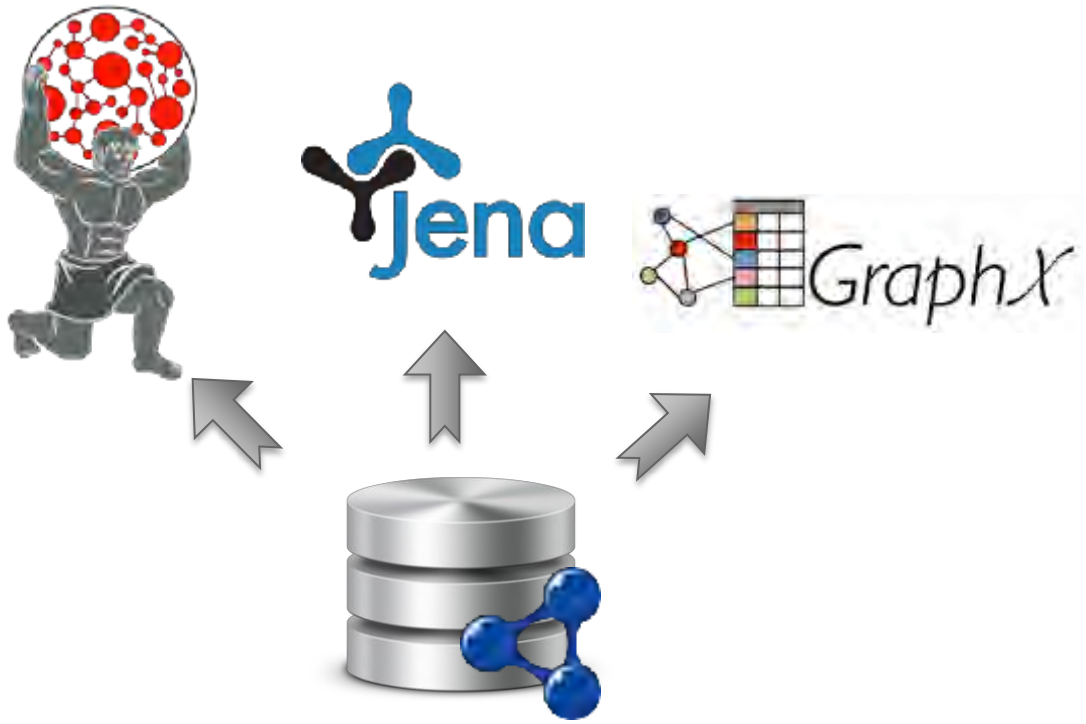
#值相关的过滤

```
PREFIX own: <http://www.thomsonreuters.com/ownership#>

CONSTRUCT {
    ?s ?p ?o .
}
FROM <demo>
WHERE
{
    ?s ?p ?o .
    ?s own:dtpConsolidatedHoldingPosition_EffectiveFrom ?d
    FILTER ( ?d > "2000-01-01"^^xsd:date)
}
```

分析系统集成



二进制存储



三元组存储

关键技术分析

运动的数据集

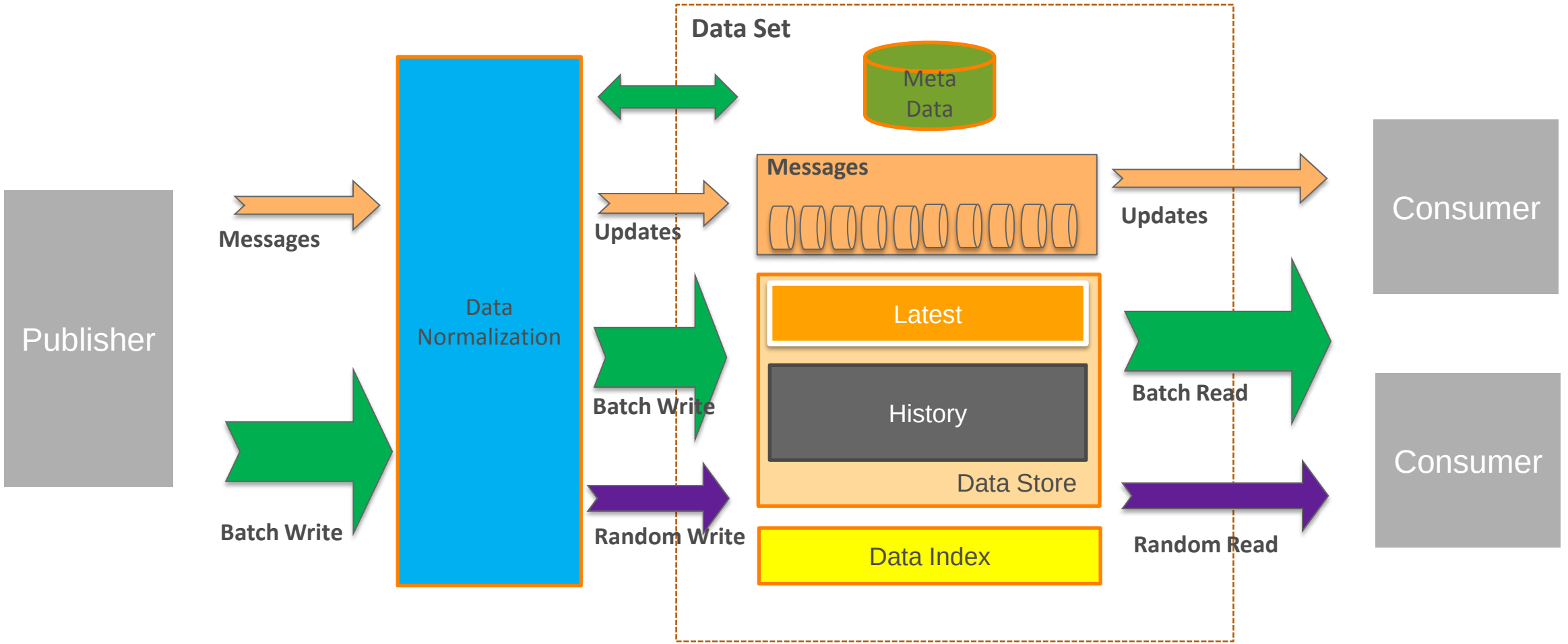
流式文档连接

自助服务框架

Spark 之上的编程语言

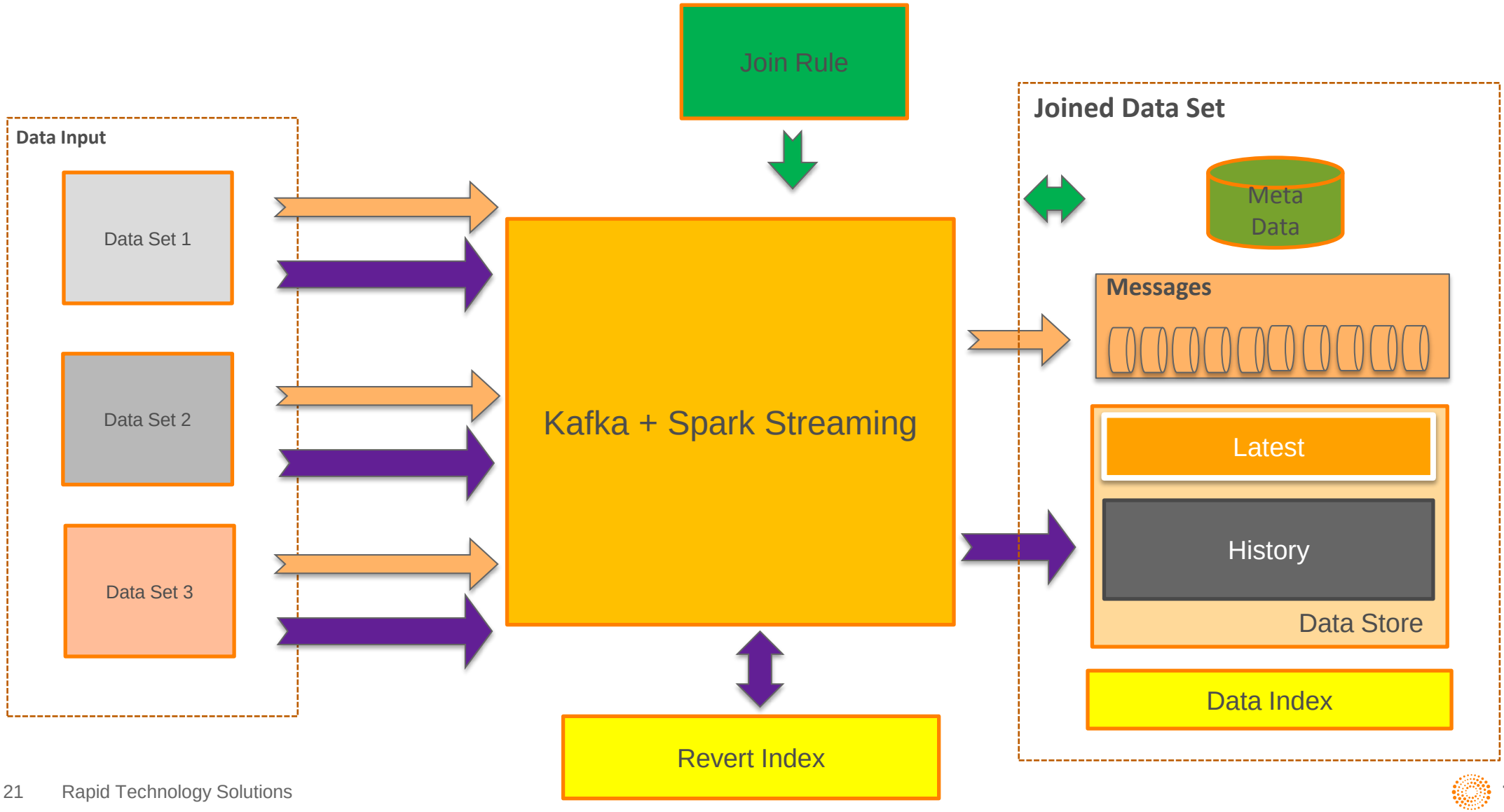
数据获取和分发

运动的数据集



基于流的连接

实体化视图， 比Spark 2.0更强的流处理



文件链接

基于Spark SQL

A.a 链接 B.b

a, b 都是简单值

b 是主键

Left Join

```
select *
from A left join B
    on A.a = B.b
```

A.a 链接 B.b

A.k 是A的主键

a, b 是简单值

b是B的主键

Left Join + Group By + Collect List

```
select A.*, cl(B.*)
from A left join B
    on A.a = B.b
group by A.k
```

A.a 链接 B.b

A.k 是A的主键

b 是简单值

a 是数组

b是B的主键

Left Join + Group By + Explode + Collect List

```
select A.*, cl(B.*)
from A lateral view explode('a') 'a1'
left join B on a1 = B.b
group by A.k
```

文档连接

基于JSONiq

A.a 链接 B.b

a, b 都是简单值

b 是主键

Left Join

```
for $a in A,
    $b allow empty in B
[$a.a eq $b.b]
return {
    "a": $a
    "b": $b
}
```

A.a 链接 B.b

A.k 是A的主键

a, b 是简单值

b是B的主键

Left Join + Group By

```
for $a in A,
    $b allow empty in B
[$a.a eq $b.b]
group by $a.k
return {
    "a": $a
    "b": [$b]
}
```

A.a 链接 B.b

A.k 是A的主键

b 是简单值

a 是数组

b是B的主键

Left Join + Array In + Group By

```
for $a in A,
    $b allow empty in B
                [ $b.b in $a.a[[]] ]
group by $a.k
return {
    "a": $a
    "b": [$b]
}
```

文档连接

JSONiq扩充

link: 新的关键字

第一个数据集，根节点

链接条件

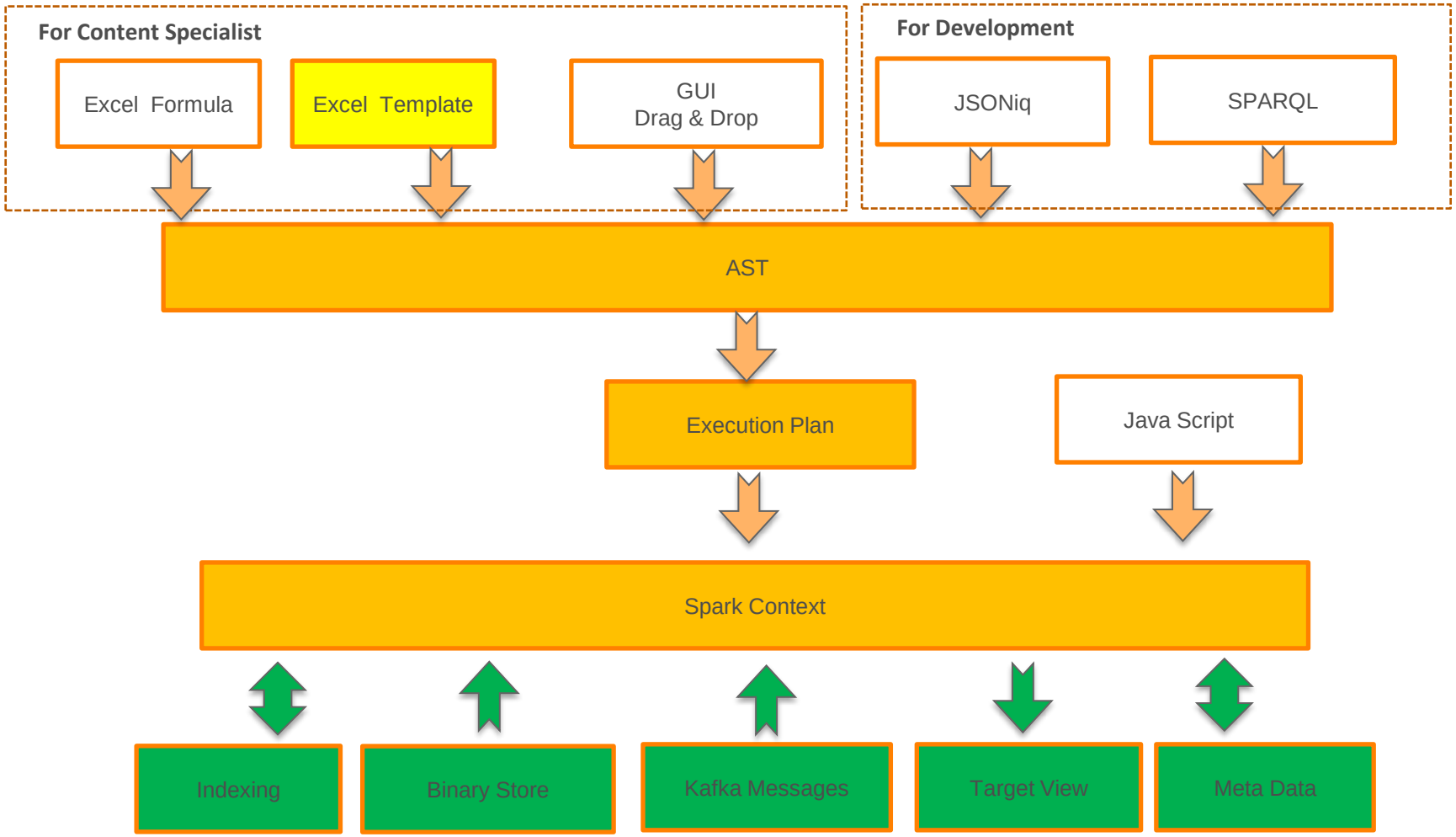
挂接点

allow empty: 左连接

```
link $a in A [$a.a eq  ”A sample”],
   $b allow empty in B [$a.a eq $b.b] at
$a.b,
   $c in C [$a.b.c eq $c.c and $a.a eq $c.d]
return {
   <formatter works>
}
```

自助服务软件框架

如何在Spark的基础上构建自己的DSL



分布式JSONiq查询引擎

JSONiq应用案例

- 通用的支持板结构化数据查询语言
- 改善定义文档增强的方便性
  - 同时支持批处理和流处理

复用已有的工具

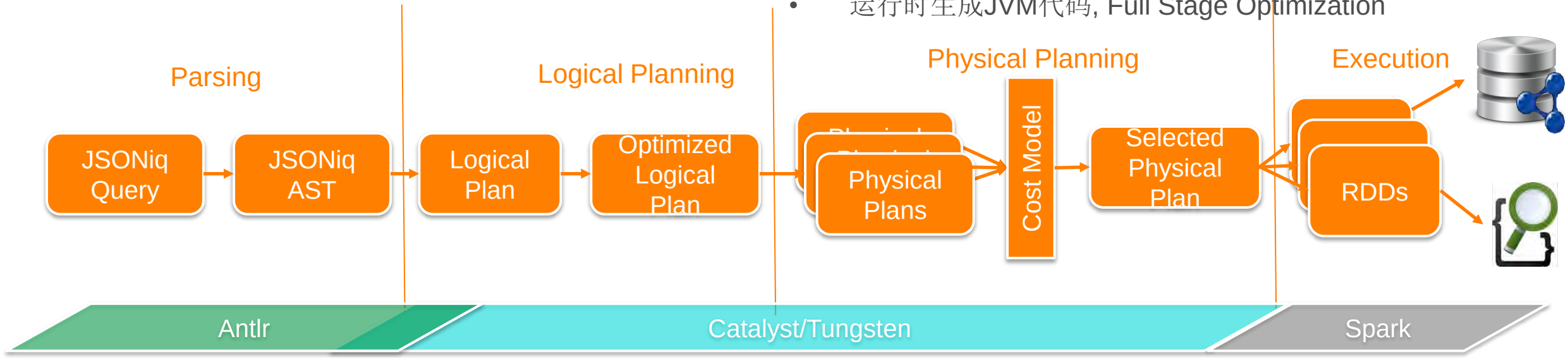
- Antlr, Spark, SchemaRDD, Spark SQL (Catalyst, Tungsten), HBase/Co-processor, Elastic Search, Scala Macro

优化方向

- 数据准确性
- 超大数据集的处理
- 流处理的快速处理

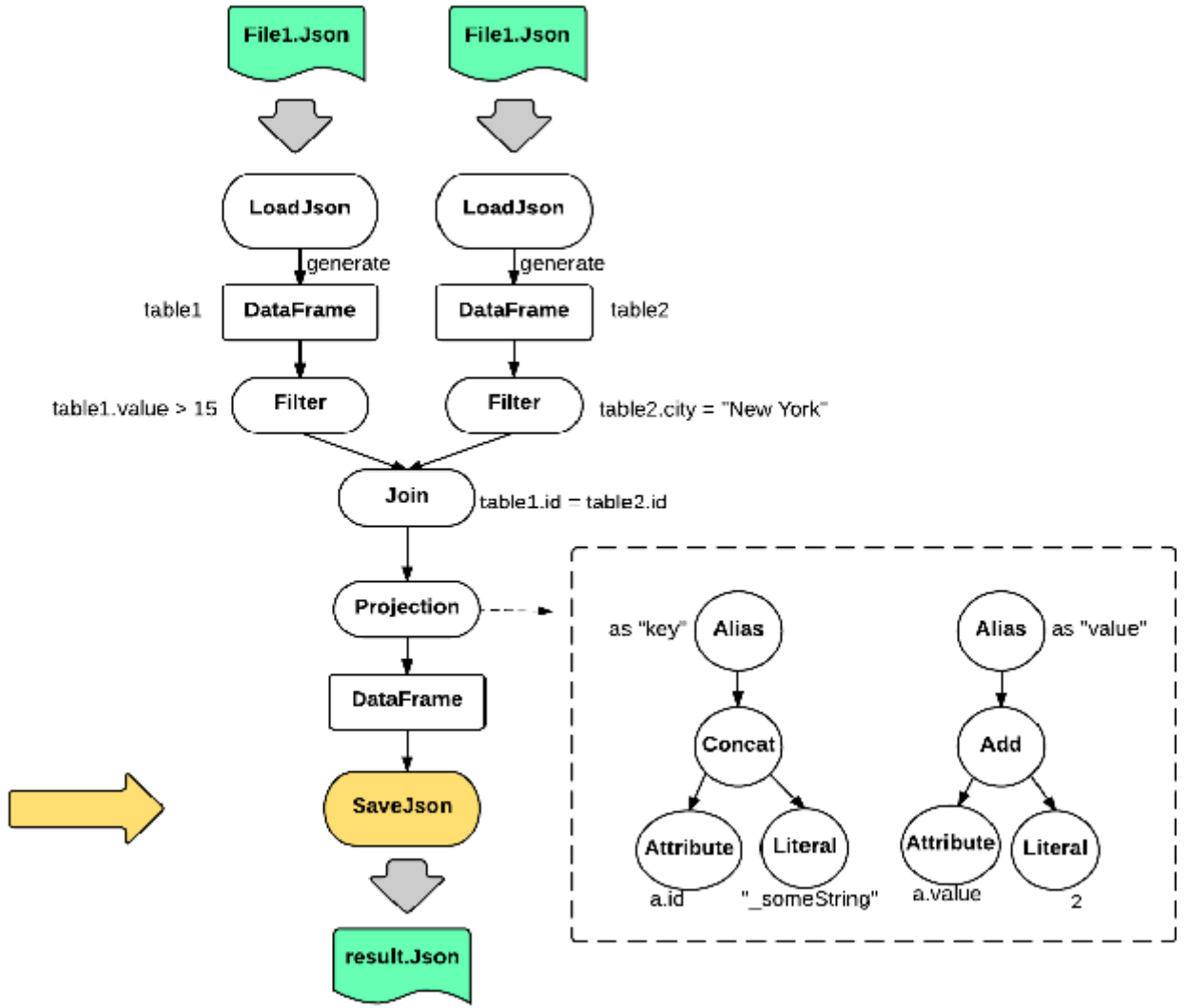
优化策略

- 使用ES做建立HBase外部索引
- 减少不必要的数据处理
  - 尽早处理过滤条件
  - 在HBase/Elastic Search里执行过滤条件
- 基于花费的优化策略，估计HBase数据集的大小
- 运行时生成JVM代码, Full Stage Optimization



Spark逻辑计划生成

```
declare variable $result :=

let $table1 := parsejson("data/file1.json")
let $table2 := parsejson("data/file2.json")


for $a in $table1[$$.value > 15],
    $b in $table2 [$$.city = "New York"]
where
   $a.id = $b.id


return {
          "key": $a.id || "_someString",
          "value": $a.value + 2
};


savejson($result, "data/result.json")
```



分布式SPARQL查询引擎

SPARQL应用案例

- 通用的视图查找
- 产生子图的衍生数据,
- 基于规则和推理增强现有数据

复用已有的工具

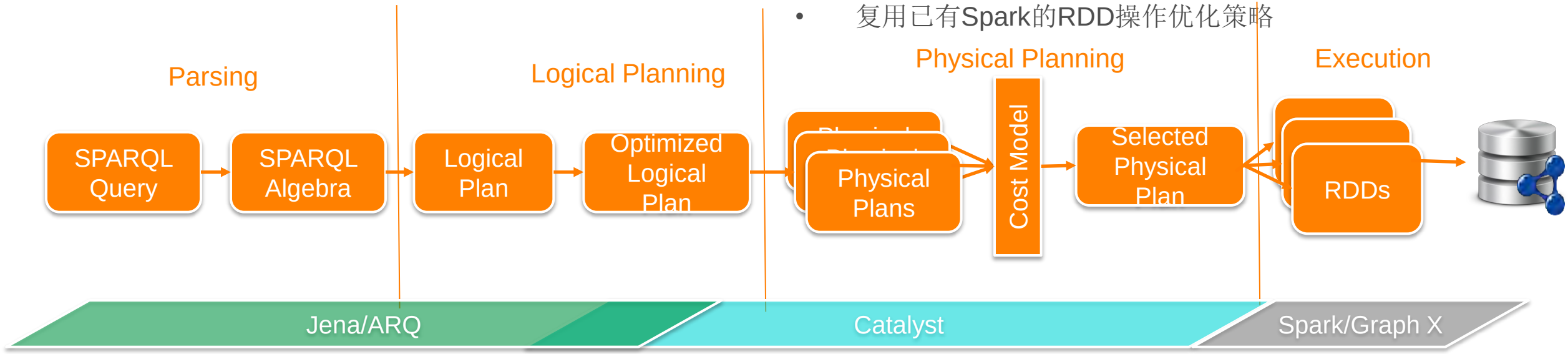
- Apache Jena, Catalyst, Spark, Graph X, HBase/Co-processor, Elastic Search

优化方向

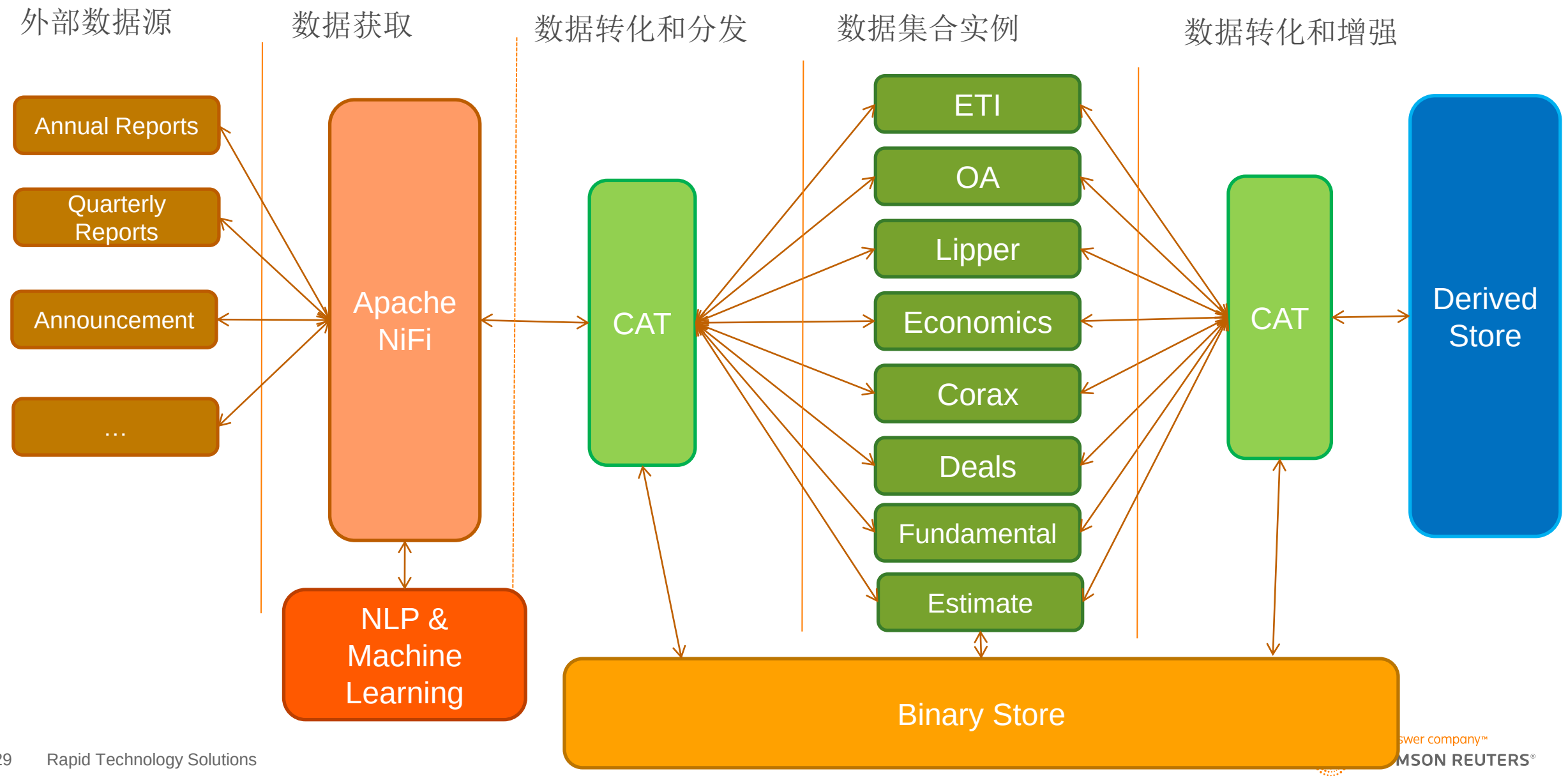
- 非常大(超过十亿）的三元组数据库查找
- 大规模的并行处理

优化策略

- 对已有数据建立索引。SPO, POS, OSP。适当的数据冗余，以空间换时间
- 降低数据查询开销
  - 基于索引的数据提取
  - 使用HBase的已有Range Scan功能
- 复用已有Spark的RDD操作优化策略



数据获取和分发



REUTERS / Darrin Zammit Lupi

Q/A

非常感谢

联系我：

raymond.shen@thomsonreuters.com