



集算器

# 敏捷数据计算引擎

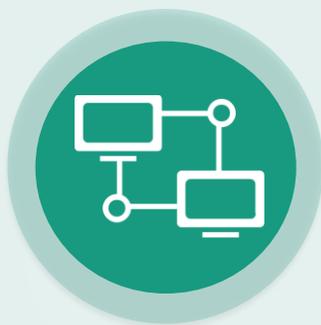
润乾软件出品



# 集算器是什么？



报表数据源



JAVA计算层



离线自由计算

# 集算器解决什么？



## 报表数据源

- ✓ 为报表准备数据的SQL长达数K，写起来难，维护更难
- ✓ 数据量大，报表呈现太慢，搞不好还溢出，用户抱怨
- ✓ ...

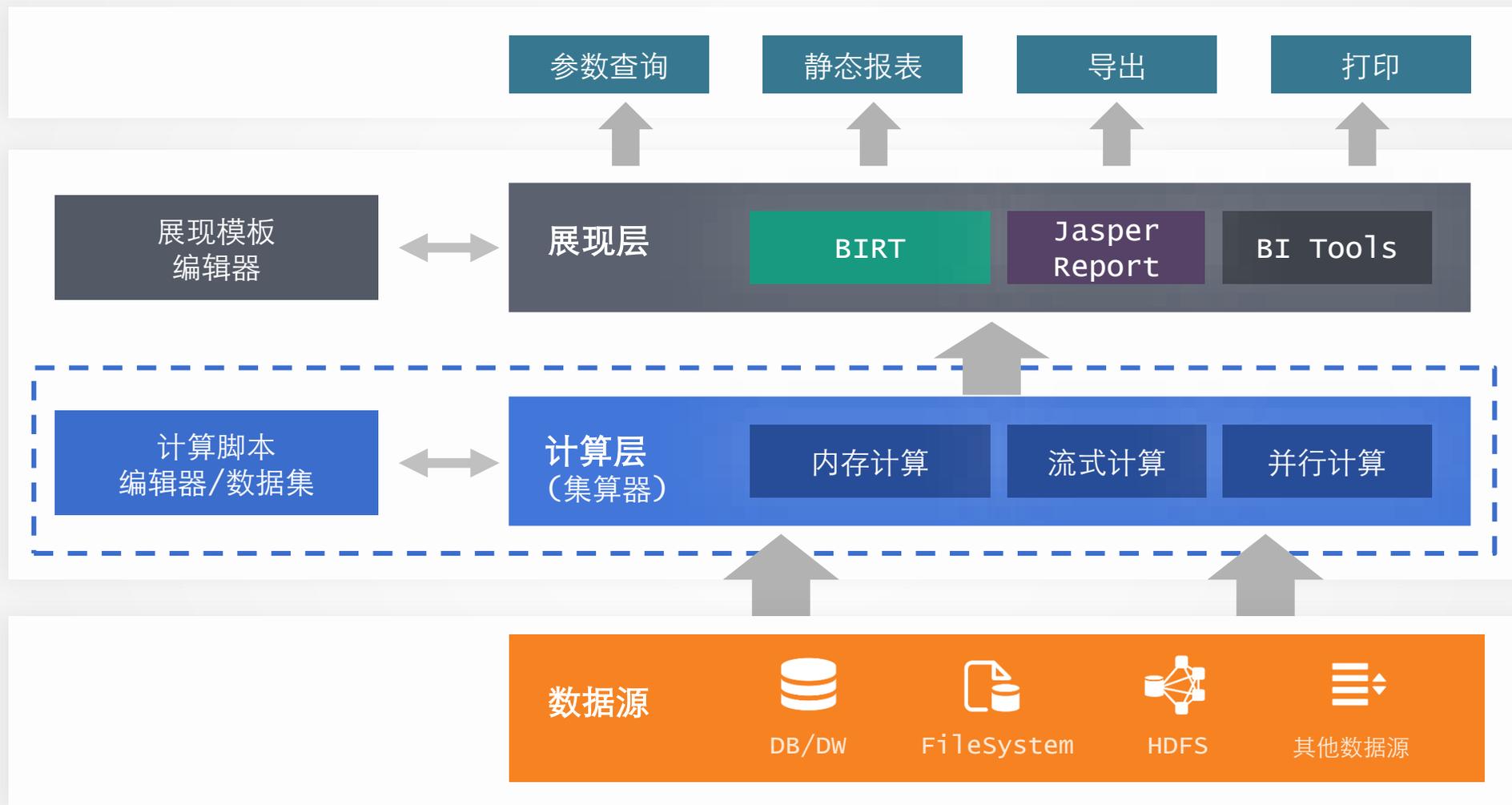
## JAVA计算层

- ✓ Java计算太繁琐，随便写个过滤分组都要数百行，太繁
- ✓ 算法逻辑改一下，整个应用要重启，难以提供持续服务
- ✓ ...

## 离线自由计算

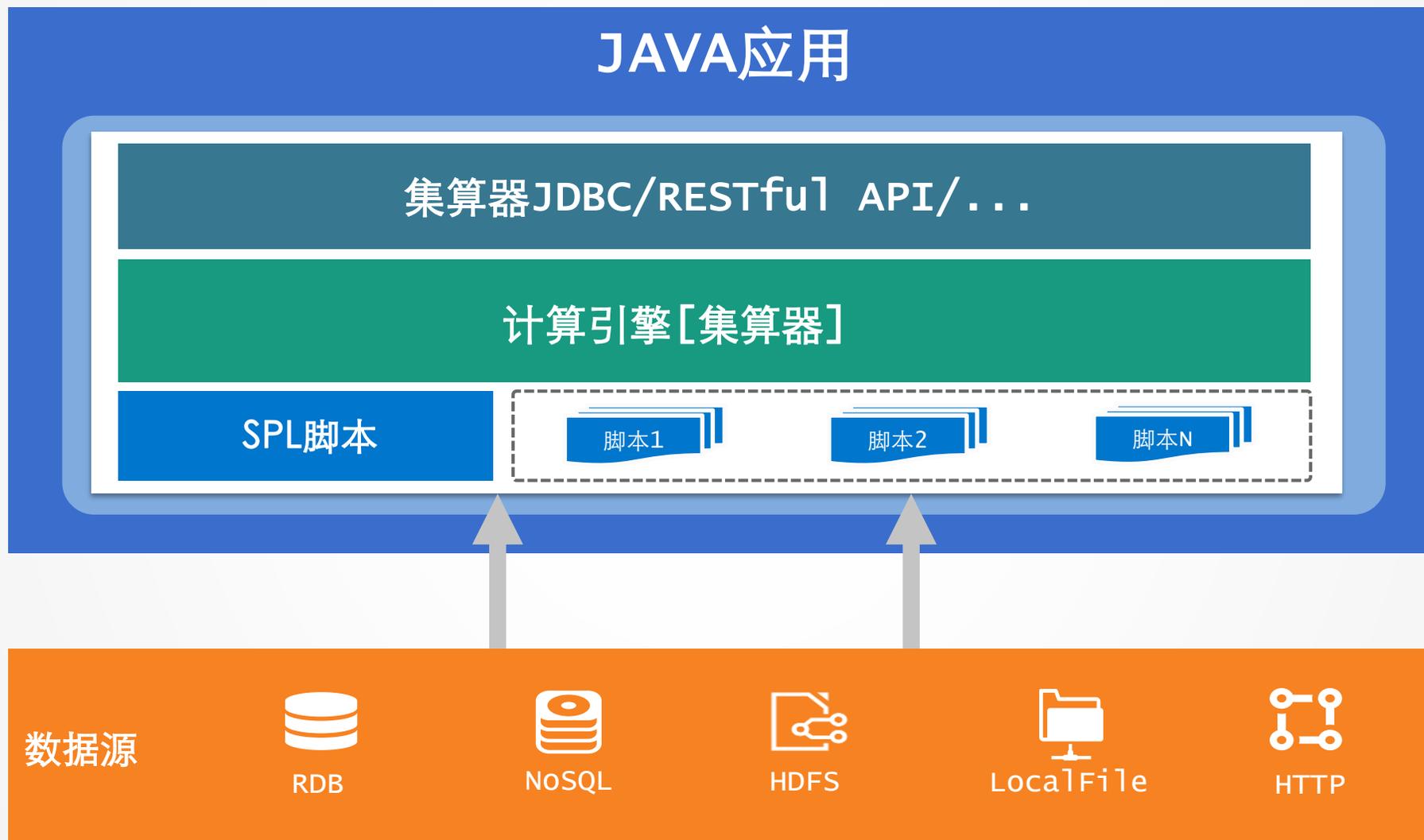
- ✓ 大批TXT/CSV/XLS要处理，又没有SQL可用，做个啥事都很繁
- ✓ SQL不好写，python也难学，总要专业程序员才能做，成本高
- ✓ ...

# 报表数据源



\*详细内容: <http://c.raqsoft.com.cn/article/1567389285711>

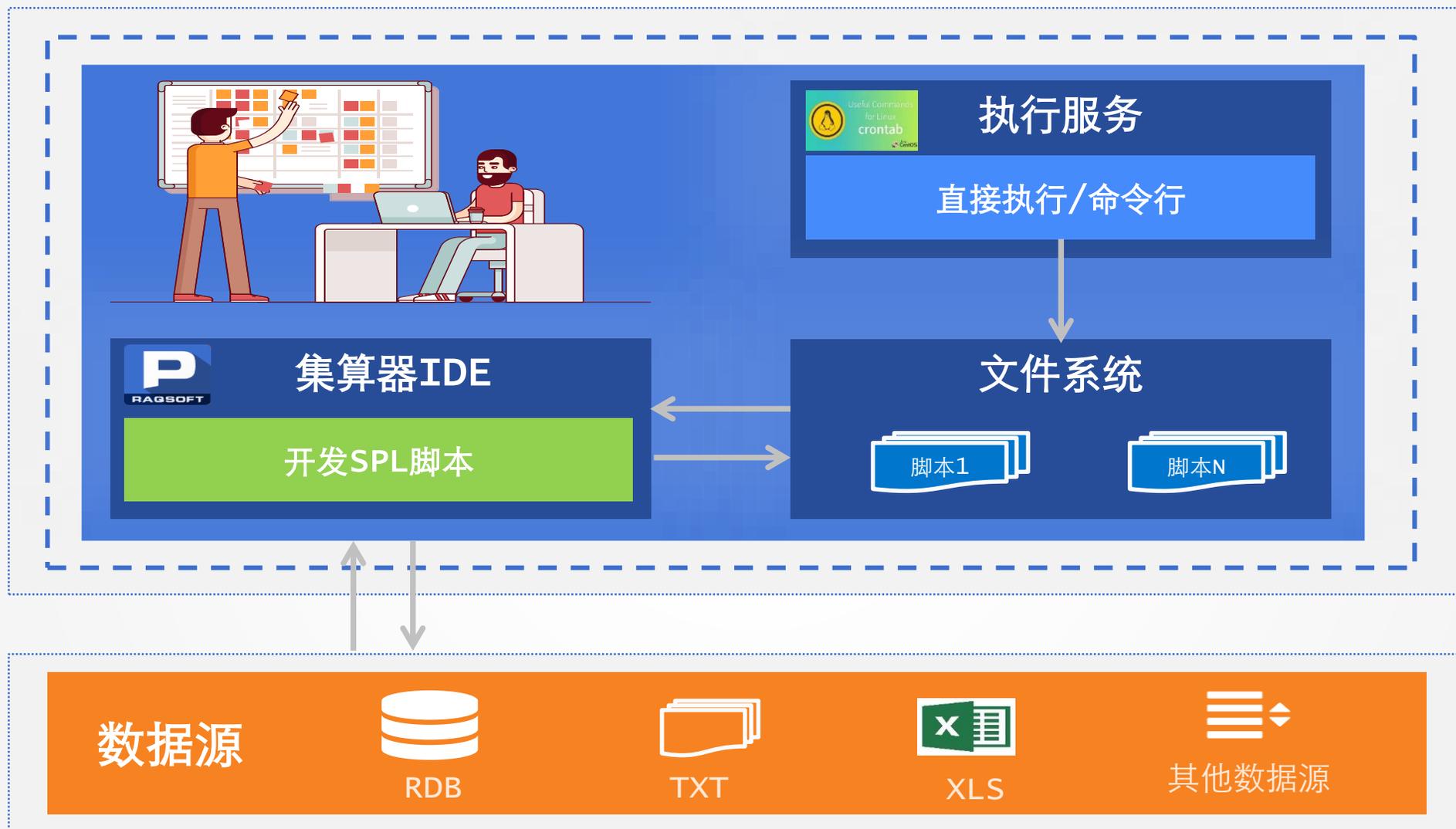
# JAVA计算层



\*详细内容: <http://c.raqssoft.com.cn/article/1548256406872>



# 离线自由计算

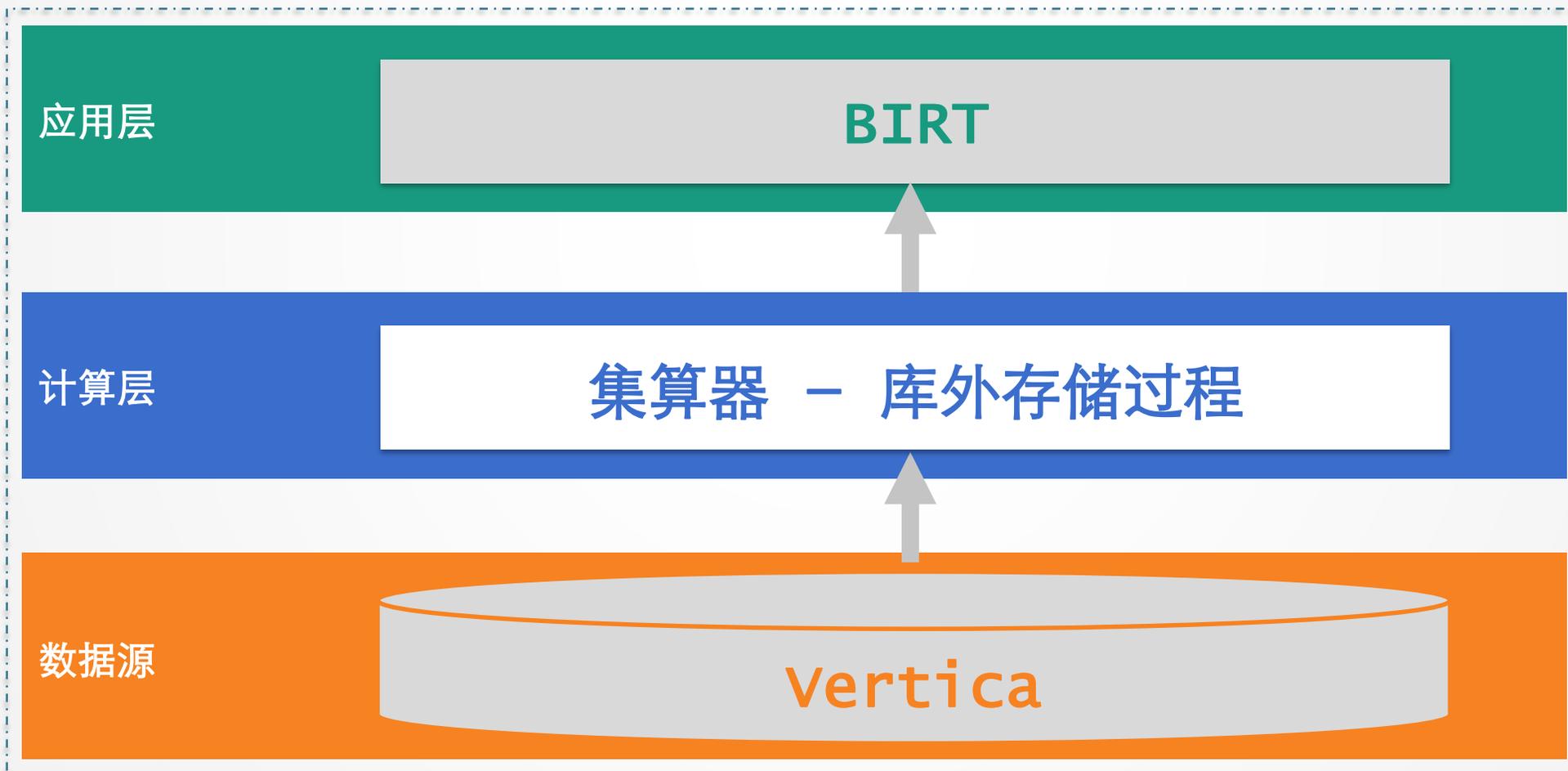


\*详细内容: <http://c.ragsoft.com.cn/article/1567389089201>



# 某保险公司 - 库外存储过程

Vertica不支持存储过程，复杂业务逻辑只能用Java写，很麻烦，难维护





# 应用效果

## 用户评价

“The best use for us is to pass parameters to the Vertica database.”

“ Each cell becomes a data array that are easy to use, compare and manipulate. It is very logical and you have made it user friendly. ”

## 应用效果

使Vertica支持库外存储过程

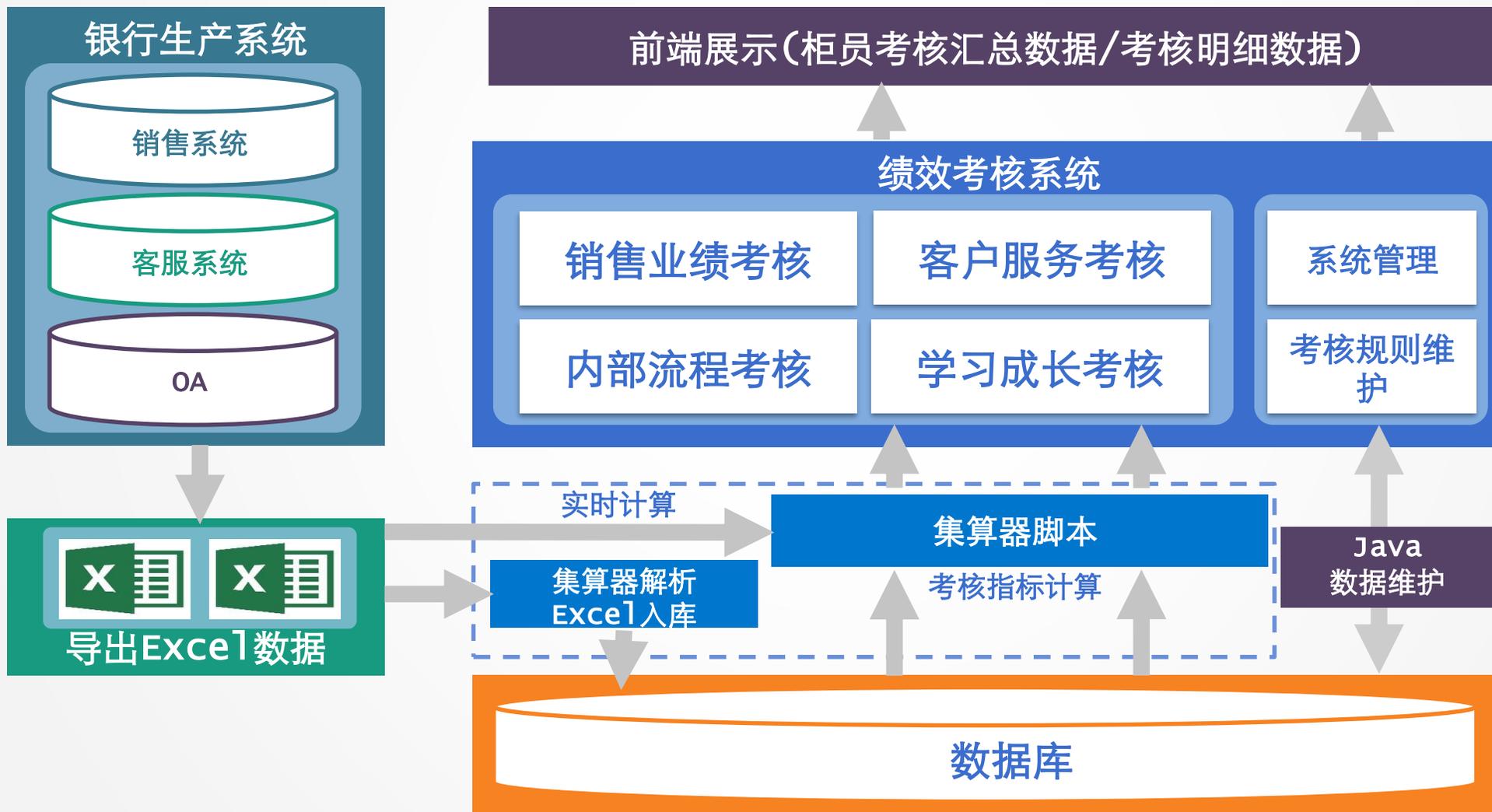
过程化算法实施简单，友好

BIRT数据准备简单高效



# 某银行 - 计算中间件

Excel种类多达44种，由Java解析数据入库，开发周期长，硬编码难度高



# 应用效果



## 降低开发成本

全部44种Excel解析由原来的32人天下降到6人天

## 提升开发效率

每种Excel解析代码量由100行变为3行

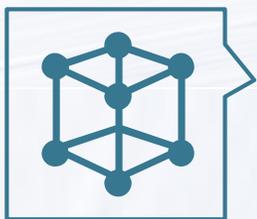
## 降低维护成本

除代码量简短易维护外；脚本热部署即修改即生效

对比指标	JAVA编程	集算器(SPL)	提升
代码量	4400行	132行	33倍
工作量	32人天	6人天	5倍
可优化	无	容易	质变
耦合性	高	低	模块化、工具化
维护性	较差	好	易管理、热切换

\*案例详情: <http://c.raguoft.com.cn/article/1560728208862>

# 集算器凭什么？



## 开放

不依赖于数据库的计算能力  
= 体系结构更轻



## 语言

新型语法适应复杂过程计算  
= 开发编码更简



## 性能

易于实现高性能算法及并行  
= 运算能力更强



## 应用

标准接口可实现嵌入式集成  
= 应用部署更易



# 独立计算引擎

不依赖于数据库的计算能力，拥有完整的敏捷语法体系

计算过程中不需要数据库

不是转换成SQL交由数据库执行

可以针对任何数据源实施计算



# 敏捷程序语言



## SPL (Structured Process Language)

- 结合了SQL、Java的共同优势
- 精心设计的丰富库函数和一致性语法
- ✓ 适应集合批量运算 (SQL优势)
- ✓ 比python更易掌握并且性能更好
- ✓ 擅长分步过程计算 (Java优势)

? 某支股票最长连续涨了多少交易日?

```
1 import pandas as pd
2 def iterate(col):
3     prev = 0;
4     res = 0;
5     val = 0;
6     for curr in col:
7         if curr - prev > 0:
8             res += 1;
9         else:
10            res = 0;
11            prev = curr;
12            if val < res:
13                val = res;
14            return val;
```

```
1 select max(continuousDays)-1
2 from (select count(*) continuousDays
3        from (select sum(changeSign) over(order by tradeDate) unRiseDays
4              from (select tradeDate,
5                    case when closePrice>lag(closePrice) over(order by tradeDate)
6                      then 0 else 1 end changeSign
7                    from stock) )
8        group by unRiseDays)
```

SQL

	A	
1	=stock.sort(tradeDate)	SPL
2	=0	
3	=A1.max(A2=if(closePrice>closePrice[-1],A2+1,0))	

Python

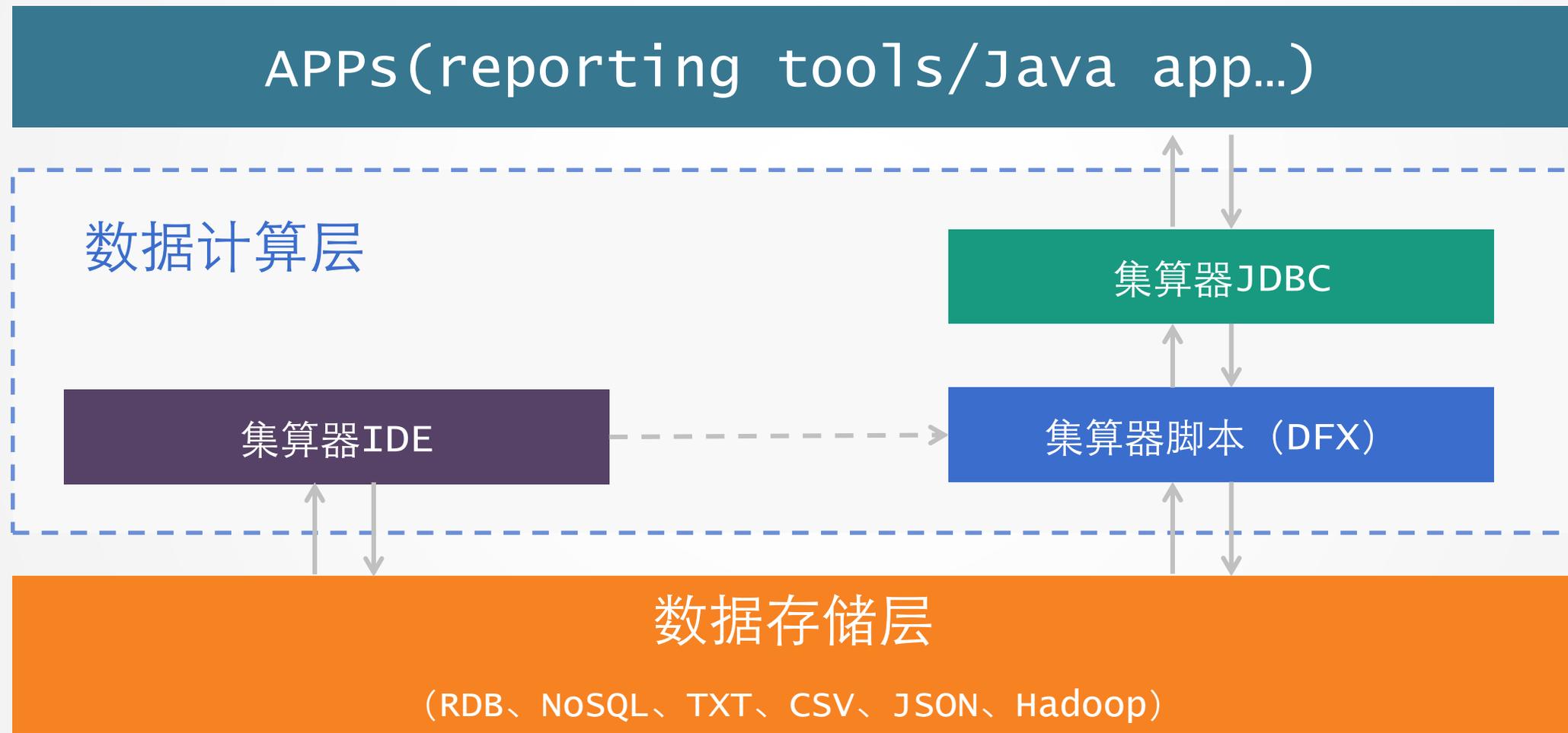
语法体系更容易描述人的自然思维!

集合运算领域专业语法，同样过程代码更简洁!



# 应用集成与管理

集算器提供标准JDBC接口，可无缝集成到Java应用中



# 创新技术 推动应用进步!

