The problems that report data source faces

Complex logic SQL is hard to write

- SQL is order-less, is not completely set oriented, and is difficult in procedure computing
- Many projects do not allow report stored procedures
- The grammar is not universal and difficult to migratet

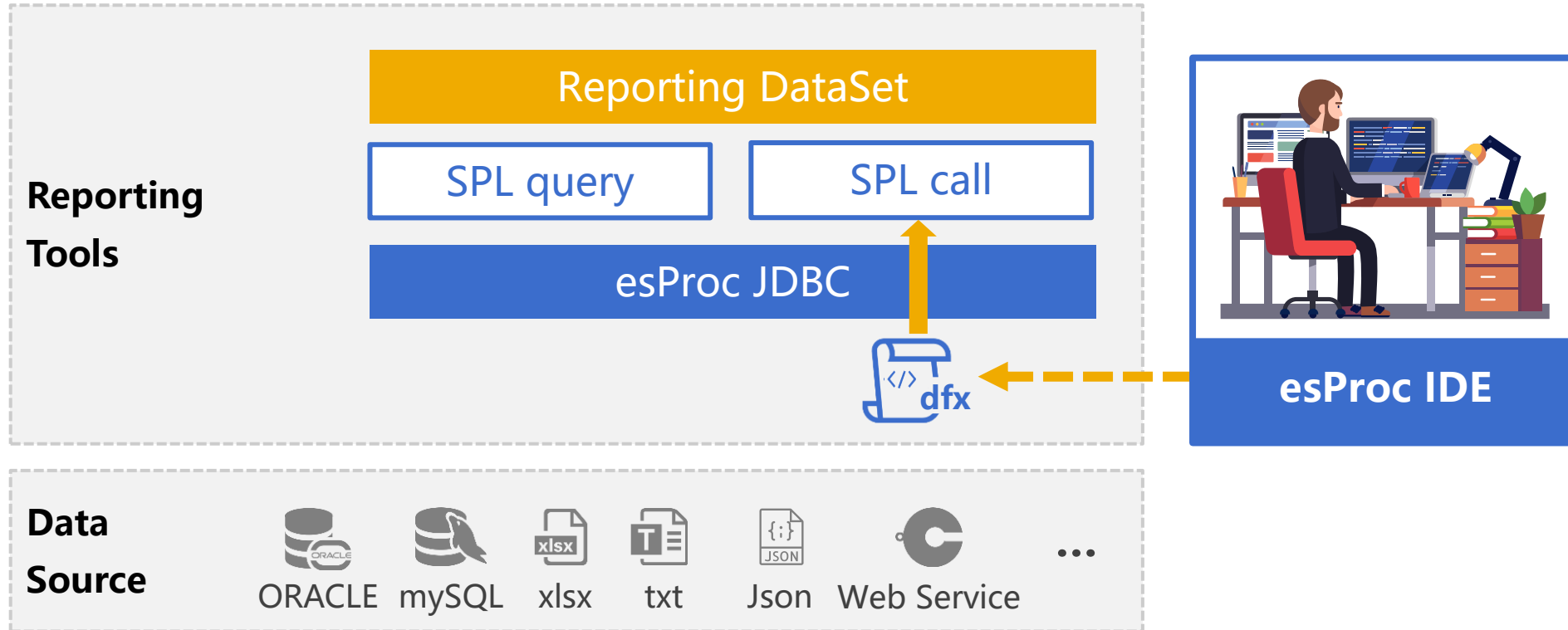## Multiple data sources are difficult to handle

- Special data sources such as Excel and NoSQL
- Cross data source, cross database
- Some reporting tools only support single data source

Custom data source (Java) is more difficult to write

- Lack of structured class library and tedious code
- High coupling and difficult to maintain

Reporting Tools

Reporting DataSet

SPL query

SPL call

esProc JDBC

dfx

esProc IDE

Data Source

ORACLE   mySQL   xlsx   txt   Json   Web Service   ...

Note: SPL is the scripting language used by esProc. DFX is the script file, which is equivalent to an external stored procedure. Report uses the JDBC interface to pass in SPL statements to realize structured calculation or script execution.

Simplify complex SQL / stored procedures

Count the longest consecutively rising trading days for a stock.

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

SPL

```
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

Syntax suitable for describing a natural way of thinking!

Can you do it in a more natural way of thinking?

External algorithm reduces stored procedures

Using stored procedure to implement data preparation algorithm will cause the coupling between report and database.

- Stored procedure and report are stored in different locations, which makes it very difficult to be consistent.

- Stored procedure modification needs to allocate corresponding database privileges, and there are security risks.

- Stored procedures are easily used by other applications, resulting in coupling between multiple applications.

- Many projects do not allow the creation of stored procedures for reports for security and permission purposes.

It will greatly reduce the number of stored procedures by replacing stored procedures with esProc to prepare data for report. An algorithm will be stored and managed along with the report template in file system and become a part of the reporting module. This will reduce its coupling with the other parts of the application or other applications.

Avoid complex Java programming

esProc adopts the set-oriented syntax, and the code is shorter than Java which does not directly provide structured computing.

Write faster and shorter

- esProc provides a higher class library and method based on Java.

Easy to understand and troubleshoot

- The proportion of pseudo real code is only about 1:1.5, and most report data preparation algorithms can be displayed in one screen.
- More code can be seen in one page, and the meaning of code can be understood more completely, thus the troubleshooting is easy.

Interpreted execution reduces application coupling

To prepare data for report by using JAVA and esProc has the following differences:

JAVA

Difficulties in modularization

Java programs must be compiled and packaged with the main application, which is highly coupled.

Hot switching is almost impossible

The modified report data preparation algorithm written in Java will lead to the recompilation and deployment of the whole application, and it is difficult to achieve hot switching.

Few class libraries

There are few class libraries in structured and semi-structured data computing in JAVA, and all algorithms need hard coding.

esProc

Simple modularization

The esProc script file can be managed and maintained together with the report template, so that the report function can be modularized.

Hot switching is easy

esProc is the interpretive execution language , and it is easy to do hot switching.

Rich class libraries

There are rich syntax and class libraries, making the computing of structured data more efficient.
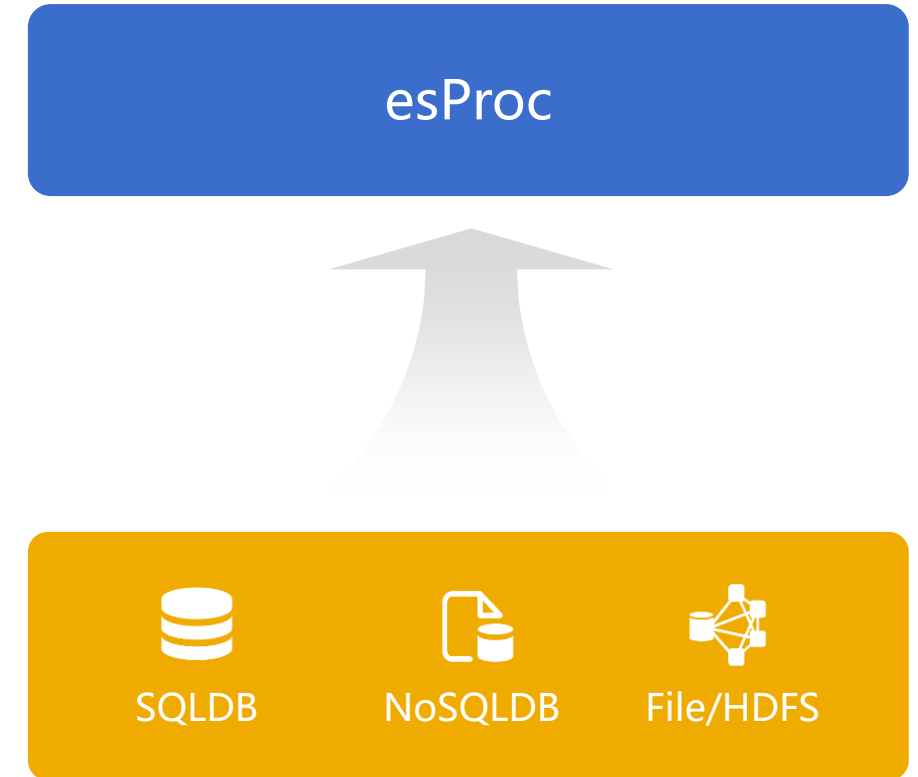
Support multiple data sources

The computing power of reporting tools are not competent for diverse data sources.
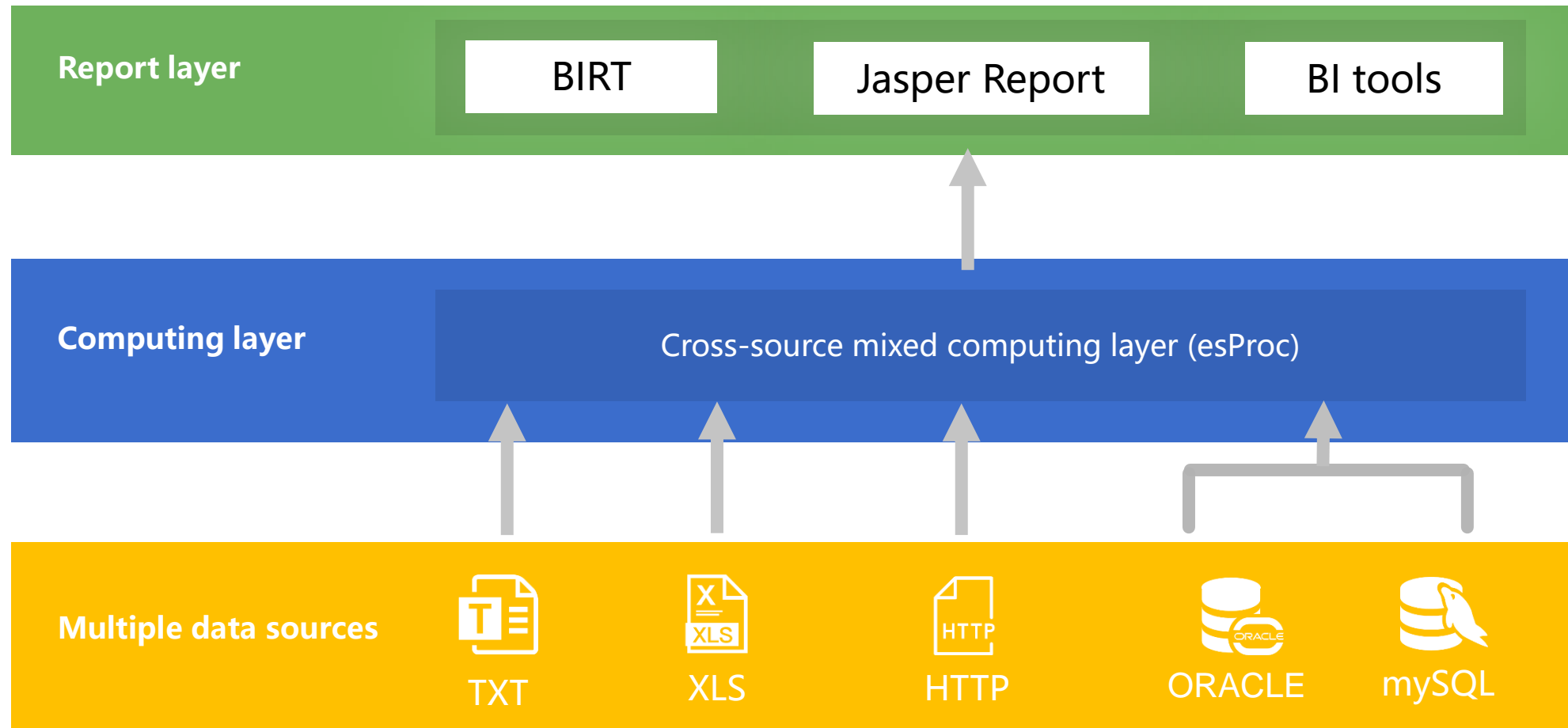
Computing layer processes diverse data sources:

- RDBMS：Oracle, MS SQL Server, DB2, Informix, MySQL, PostgreSQL

- NOSQL：MongoDB, Redis, Cassandra, ElasticSearch

- Hadoop：HDFS, HIVE, HBase

- Application software：SAP ECC, BW

- File：Excel, Json, XML, TXT

- Others：Http Restful, Web Services, OLAP4j , …

esProc

SQLDB   NoSQLDB   File/HDFS

Direct use of multiple data sources and cross database computing

Report layer

BIRT	Jasper Report	BI tools

Computing layer

Cross-source mixed computing layer (esProc)

Multiple data sources

TXT XLS HTTP ORACLE mySQL

Dynamic Data Source/Set

Dynamic Data Source

Switch database dynamically according to parameters ${pds}.query("select * from T where F=?",pF)

Dynamic Data Set

General code to implement algorithm, no need to modify during migration

Generate algorithm dynamically. Concat dynamic SQL.

	A	
1	=sums.array().("sum("+~+") as "+~).string()	/sum(a) as a, sum(b) as b
2	=db.query("select G,"+A1+" from T group by G")	

Capacity control for result set

	A	B	
1	=db.cursor("select * from T")	=A1.fetch(1000)	
2	if B1.fetch@0(1)	>B1.insert(0, "Continue")	/Insert markers when uncompleted
3	>A1.close()	return B1	

Parallel access improves performance

Databases generally have a bad JDBC performance, but the reporting performance relies heavily on the data retrieval performance. esProc can create multiple database connections to retrieve data in segments by using multithreaded parallel processing, and the performance can be increased by several times.

	A	B	C
1	fork 4	=connect(db)	/There are four threads, and each connection is established separately.
2		=B1.query@x("select * from T where part=?" ,A1)	/Retrieve each segment separately
3	=A1.conj()		/Merge the results

BIRT representation

Showing page 1 of 1							
dataset_date	measurement	managedelement	content	last_minute_cp	last_5_minute	last_30_minute	last_hour_cpu
2015-05-26 21:45:00.0	cpqHoCpuUtilTa	nltcom04	20:7:5:5	20	7	5	5
2015-05-26 21:45:00.0	cpqHoCpuUtilTa	nltcom04	17:8:5:5	17	8	5	5
2015-05-26 21:45:00.0	cpqHoCpuUtilTa	nltcom01	20:4:7:7	20	4	7	7
2015-05-26 21:45:00.0	cpqHoCpuUtilTa	nltcom03	8:12:5:5	8	12	5	5
2015-05-26 21:45:00.0	cpqHoCpuUtilTa	nltcom02	11:5:5:6	11	5	5	6

esProc calculation

	A
1	=file("D:\\DummyData.csv").import@t(dataset_date,measurement,managedelement,content:string;,",")
2	=file("D:\\Mapping.csv").import@t(;,",")
3	=A2(1).content.split(":")
4	=A3.to(2,).("cols("+string(#+1)+"):"+~).string()
5	=A1.derive((cols=string(content).split(":"))(1):${A3(1)},${A4})

Data sources

DummyData.csv

```
1  dataset_date,measurement,managedelement,content
2  2015-05-26 21:45:00.0,cpqHoCpuUtilTable,nltcom04,20:7:5:5
3  2015-05-26 21:45:00.0,cpqHoCpuUtilTable,nltcom04,17:8:5:5
4  2015-05-26 21:45:00.0,cpqHoCpuUtilTable,nltcom01,20:4:7:7
5  2015-05-26 21:45:00.0,cpqHoCpuUtilTable,nltcom03,8:12:5:5
6  2015-05-26 21:45:00.0,cpqHoCpuUtilTable,nltcom02,11:5:5:6
```

Mapping.csv

```
1  managedelement,content
2  nltcom,last_minute_cpu_utilization_kpi:last_5_minute_cpu_u
   tilization_kpi:last_30_minute_cpu_utilization_kpi:last_hou
   r_cpu_utilization_kpi
```

Jasper representation

| Designer | XML | Preview | | | 1 | | 100% ∨ | |

OID	EID	NAME	DEPT	AMOUNT
10262	8	Megan	Marketing	583.199999392
10265	2	Ashley	Finance	1170.0
10268	8	Megan	Marketing	1098.0
10276	8	Megan	Marketing	400.0

esProc calculation

Cross-database associative filtering

	A
1	=mysql.query("select OID, EID, AMOUNT from orders")
2	=pg.query("select EID,NAME,DEPT from employee")
3	=A1.switch(EID,A2:EID)
4	=A3.select(["Marketing","Finance"].pos(EID.DEPT))
5	=A4.new(OID,EID.EID:EID,EID.NAME:NAME,EID.DEPT:DEPT,AMOUNT)

Data sources

MySQL table:orders(partial field)

OID	EID	AMOUNT
10248	5	428.0
10249	6	1842.0
10250	4	1523.4999898076...
10251	3	624.94999976083...

PG table:employee(partial field)

EID	NAME	DEPT
1	Rebecca	R&D
2	Ashley	Finance
3	Rachel	Sales
4	Emily	HR

JDBC class stored procedure calls SPL script file

```
…
Connection con = null;
Class.forName("com.esproc.jdbc.InternalDriver");
con= DriverManager.getConnection("jdbc:esproc:local://");
// Calling stored procedures , CountName is the file name of dfx
st =(com. esproc.jdbc.InternalCStatement)con.prepareCall("call
CountName()");
// Execute stored procedures
st.execute();
//Get  result set
ResultSet rs = st.getResultSet();
…
```

JDBC query files directly using SQL

```
…
Connection con = null;
Class.forName( "com.esproc.jdbc.InternalDriver" );
con= DriverManager.getConnection( "jdbc:esproc:local://" );
// Calling stored procedures , CountName is the file name of dfx
st =(com. esproc.jdbc.InternalCStatement)con.createStatement();
//Query files using SQL,get result set
ResultSet rs =  st.executeQuery( "$select name,count(*) from
/home/user/duty.txt group by name");
…
```

Resource link

- SPL codes of common calculation http://doc.raqsoft.com/
- Dynamic Grouping of crosstab in BIRT http://c.raqsoft.com/article/1560999918897
- How to implement irregular month statistics in BIRT http://c.raqsoft.com/article/1555577300442
- Solve some difficult problems in reporting tools such as Jasper http://c.raqsoft.com/article/1555587476131
- Making a More Efficient iReport with esProc http://c.raqsoft.com/article/1574662582751
- Some examples of solving BIRT special layout http://c.raqsoft.com/article/1574671682673
- Some examples of solving BIRT dynamic data source http://c.raqsoft.com/article/1577243026905

- How to Call an SPL Script in BIRT http://c.raqsoft.com/article/1544410181816

- How to Call an SPL Script in JasperReport http://c.raqsoft.com/article/1544413715675

- Installation and free authorization http://c.raqsoft.com/article/1573787506233