

*Oracle EssBase  
интеграция  
между  
приложениями*

*PL-SQL XMLA API*

*Evgeniy A.Rasyuk*

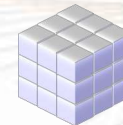
*Oracle Hyperion consultant  
freelancer*

*june 2013*



# *преамбула*

*в данной презентации рассказывается о том  
зачем нужно отказаться от нативных способов  
передачи данных в пользу решения на основе  
PLSQL XMLA API*



*er@essbase.ru*

# постановка задачи

Бюджетирование различных функциональных областей деятельности компании делается в различных приложениях

*Требуется вести учет для следующих видов бизнеса:  
производство  
торговля  
управление финансами  
управление кадрами*

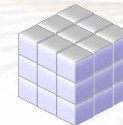
Требуется собрать консолидированный бюджет группы

*Учесть влияние  
внутригрупповых оборотов  
потребности в людских, материальных  
и финансовых ресурсах*



# описание бизнес-процесса

- *Требуется вести учет различных функциональных аспектов деятельности группы компаний.  
(производство, торговля, управленческие расходы)*
- *Требуется собрать информацию по деятельности как каждого конкретного предприятия и головной компании, так и для всей группы (учесть влияние внутригрупповых оборотов)*
- *Каждая функция живет в своем «отдельном мире» аналитик и показателей, которые наиболее точно описывают аспекты ее деятельности.*
- *При старте бюджетного процесса изменения происходят параллельно во всех моделях. Работа осуществляется различными пользователями.*



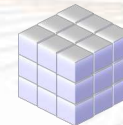
# функциональные решения

Использовать одно приложение для всех задач

*Оптимально на этапе проектирования приложений с использованием приема совмещения аналитик.*

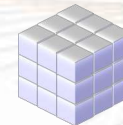
для каждой функции использовать свое приложение

*Позволяет решить проблемы Производительности для приложений с большими и детальными аналитиками, с множеством конкурирующих расчетов, и большим числом интерактивных пользователей*



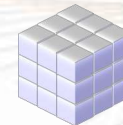
# существующие технологические решения по передачи данных ч1

- **@XREF (XR)**
  - точечный прием данных на «приемнике». Работает от «модели» в правиле расчета
- **@XWRITE (XW)**
  - точечная передача данных на «источнике». Работает от «данных» в правиле расчета
- **Report Script (RS)**
  - выгрузка среза данных на «источнике» в текстовый файл. Работает от «модели» в командной оболочке. Для загрузки требуется писать интерфейс на Load Rules
- **DataExport (DE)**
  - выгрузка среза данных на «источнике» в текстовый файл. Работает от «данных» в правиле расчета. Для загрузки требуется писать интерфейс на Load Rules



# существующие технологические решения по передаче данных ч2

- *Transparent Partition (TP)*
  - *Отображение данных одного куба в другом. Инициализация через командную оболочку*
- *Replicated Partition (RP)*
  - *Копирование данных одного куба в другой. Инициализация через командную оболочку*
- *PLSQL EssBase XMLA API (PX)*
  - *API интерфейс по доступу к многомерным данным EssBase из PL-SQL*



# сравнение методов передачи данных

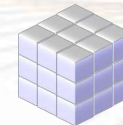
функциональность	PX	XR	XW	RS	DE	TP	RP
управление передаваемым срезом	+	+	+	+	+	-	-
управление на источнике, приемнике	+/+	-/+	+/-	+/+	+/+	-/-	-/-
точечная передача, передача массива	+/+	+/-	+/-	+/+	+/+	+/+	+/+
требуются временные или вспомогательные файлы	-/+	-/-	-/-	+/+	+/+	-/-	-/-
простой системы при внесении изменений	-	-	-	-	-	+	+
требуются дополнительные сетевые ресурсы	+	+	+	+	-	+	+
системное влияние на целостность данных (при исключительных ситуациях)	-	-	-	-	-	+2	+
создает блоки	+	-	+	+	+	-	+
требует CreatOnMissingBlock	-	+	-	-	-	-	-
управление скоростью выгрузки	+	-	-	-	-	+4	-
трансформация показателей	+	-	-	-	-	-	-
<b>Итого баллов ( + - складываем, + - вычитаем)</b>	<b>10</b>	<b>2</b>	<b>4</b>	<b>4</b>	<b>7</b>	<b>-1</b>	<b>-2</b>



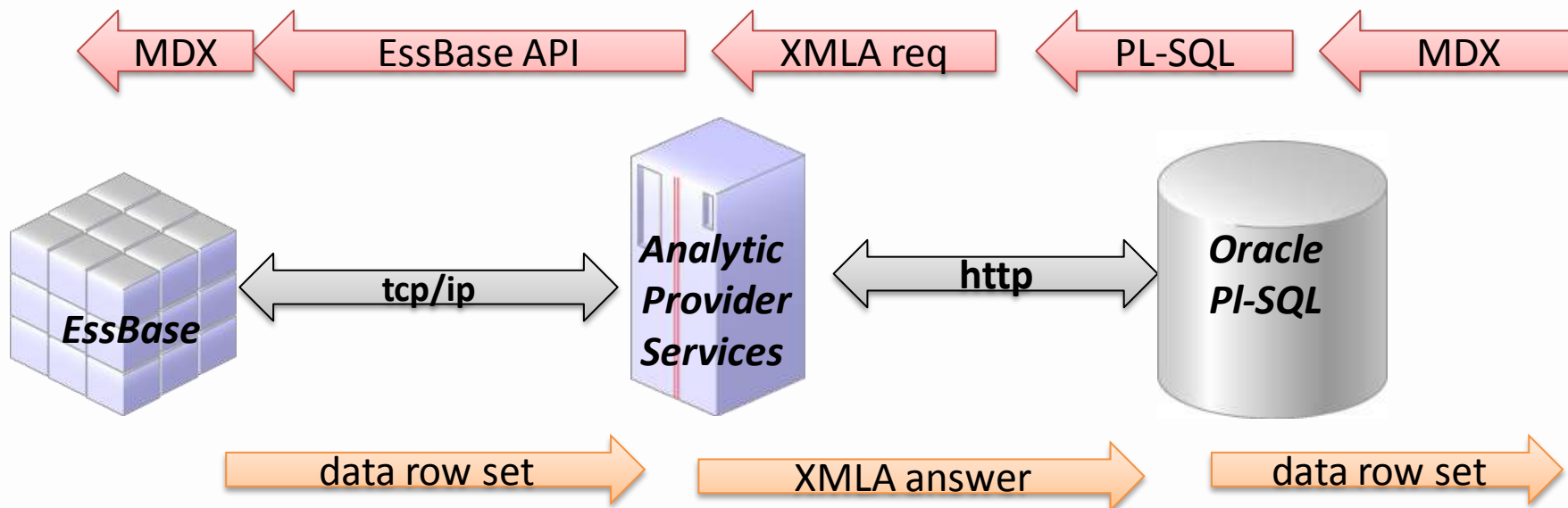
# **описание преимуществ PLSQL XMLA API (PX) по сравнению с DataExport (DE)**

*Эти два способа по сумме баллов конкурентные между собой ,  
но есть различия, которые могут быть нивелированы  
разработчиком.*

*Основное преимущество PX – это в том , что значения  
оказываются в БД и у разработчика появляются  
множество вариантов по управлению ими.*

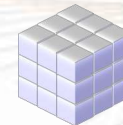


# PL-SQL XMLA API: системная архитектура



# Essbase MDX: пример запроса

```
SELECT NON EMPTY {( [Actual] )} ON COLUMNS,  
NON EMPTY  
CrossJoin (CrossJoin ([Product].children, [Market].children),  
CrossJoin([Year].Children, [Measures].children))  
ON ROWS  
FROM Sample.Basic
```



# PL-SQL XMLA API : функциональная архитектура ч1

call CDF function

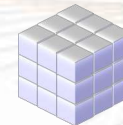
```
FIX (Dim1_NA,Dim2_NA,...,DimN_NA)  
Account_NA(  
Account_NA=jRunCMD( CalcRelational_01.xml);  
)  
ENDFIX
```

MaxL load data with rule  
file

```
Import DataBase app.db  
data connect as Login identified by Password  
using rules_file DataLoad.rul  
on error write to DataLoad.err
```

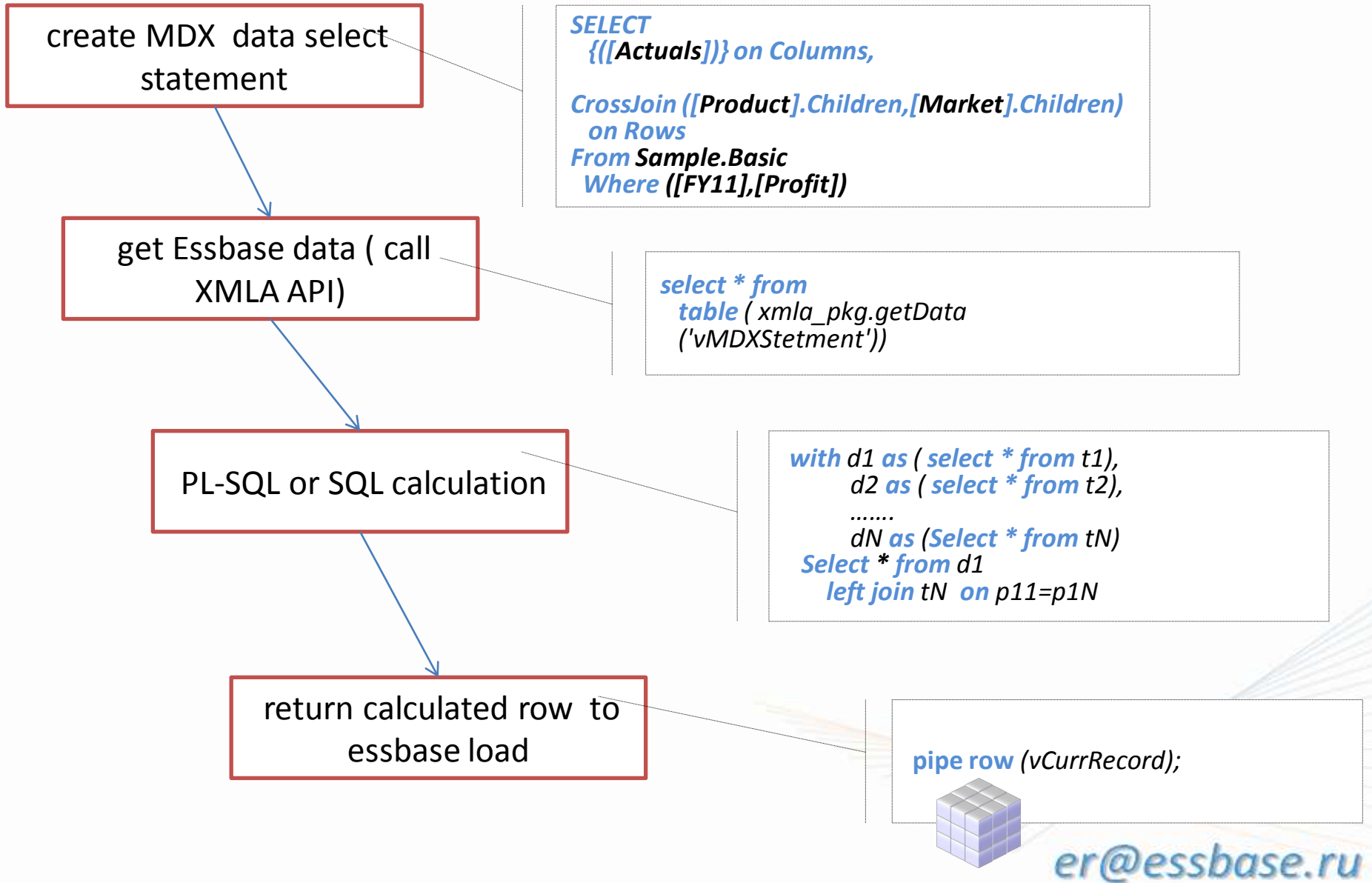
PL Sql pipeLine  
calculation

```
select * from  
table ( xmla_pkg.currCalc  
('&vParam1','&vParam2')  
)
```





# PL-SQL XMLA API : функциональная архитектура ч2



# PL-SQL EssBase XMLA API все вместе

```
select
  *
from (
  table( xmla_get_data_PKG.getMdxValue
    (
vMdxQuery =>'SELECT NON EMPTY {( [Actual])} ON COLUMNS,
          NON EMPTY
          CrossJoin (CrossJoin ([Product].children, [Market].children),
                    CrossJoin([Year].Children, [Measures].children))
          ON ROWS
          FROM Sample.Basic' ,
vApsConnect=>'http://hypadmin:hyperion@localhost:13080/aps/XMLA' ,
vEsbConnect=>'localhost'
    )
  )
)
```

*MDX select statement*

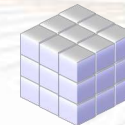
*Essbase TCP-IP address*

*XMLA web services address*



# PL-SQL EssBase XMLA API

<i>function name</i>	<i>description</i>
<i>getXmlaData</i>	<i>получить строки данных из ответа EssBase XMLA</i>
<i>getMdxValueColumn</i>	<i>разобрать вывод из функции getXmlaData на колонки с данными и метаданными</i>
<i>getDimensionList</i>	<i>получить список аналитик из базы данных Essbase</i>
<i>getApplications</i>	<i>получить список приложений из сервера Essbase</i>
<i>getDataBase</i>	<i>получить список баз данных из приложения Essbase</i>
<i>getHierarshyList</i>	<i>получить значение аналитики из бд EssBase</i>





# Essbase server tuning

```
AGENTTHREADS 250  
SERVERTHREADS 250
```

*discard reverses host name lookup*

```
NO_HOSTNAME_LISTCONNECT TRUE
```

*Generate "error" flow if odbc exception happens*

```
;Controls whether an Essbase error is generated when fetching data  
;from a SQL database during a data load
```

```
SQLFETCHERRORPOPUP TRUE
```

```
DATAERRORLIMIT 100
```

```
;Sets the maximum amount of time a query can use to retrieve and deliver  
;information before the query is terminated.
```

```
QRYGOVEXEETIME 60
```

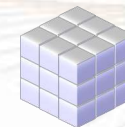
*limit for MDX queries*

```
CALCPARALLEL 1
```

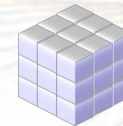
```
CALCTASKDIMS 1
```

```
;When set to true, prevents the server from going beyond 31 formula execution levels.
```

```
CALCLIMITFORMULARECURSION TRUE
```

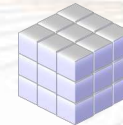


?



[er@essbase.ru](mailto:er@essbase.ru)

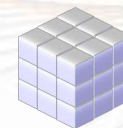
***Thank you !***



[er@essbase.ru](mailto:er@essbase.ru)

# ***послесловие***

- *работаю с Hyperion с 2004*
- *сертифицированный по различным версиям Oracle Hyperion Planning, Essbase и Financial Management*
- *Oracle DBA 10g*
- *Оказываю услуги для консалтинговых компаний:*
  - *обучение ( семинары, вебинары)*
  - *аудит и рекомендации по настройке производительности*
  - *сопровождение проектов*



*er@essbase.ru*