

There is no magic, but there is...

Data Virtualisation

JUDCon 2012

Mark Nolan – Architect, Royal Bank of Scotland

June 2012

Data Virtualisation

Contents

Background

- What is Data Virtualisation?

Data Virtualisation is not a Panacea

- How Data Virtualisation Works
- There is No Magic™

The Opportunity to Think Differently

- Fine-Grained Entitlements
- Business Concepts
- Object DAL

Status of Data Virtualisation in RBS M&IB Finance

- What have we done so far?
- What have we not done?

Conclusions

BUILDING TOMORROW™

Background

What is Data Virtualisation?

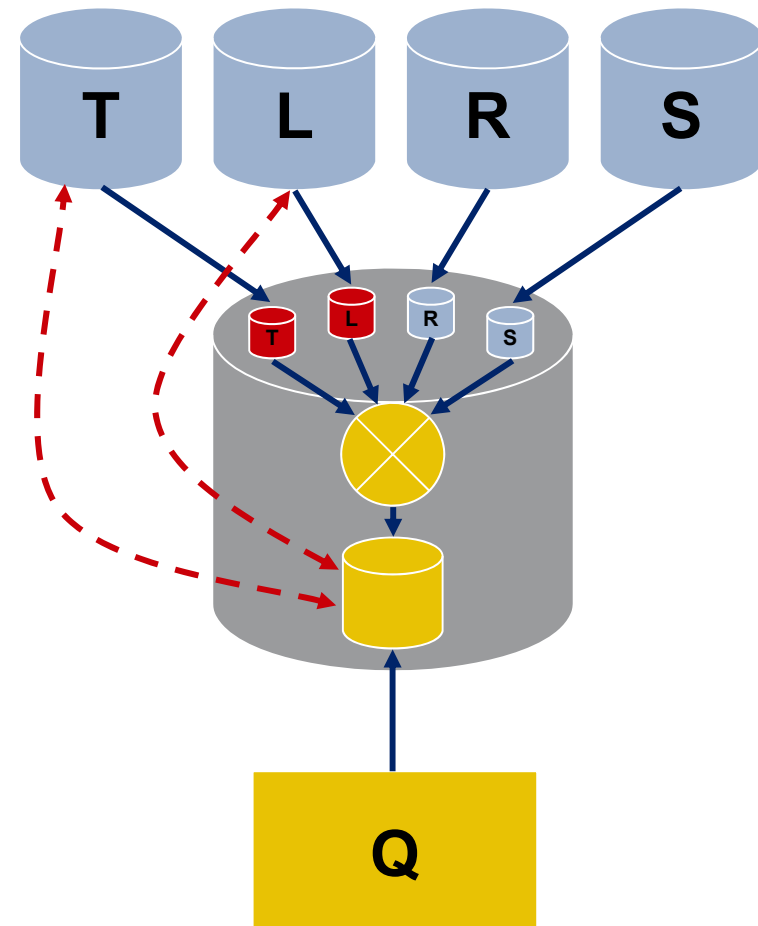
What is Data Virtualisation?

What is the problem?

Some question or process that can only be implemented by using data from multiple sources:

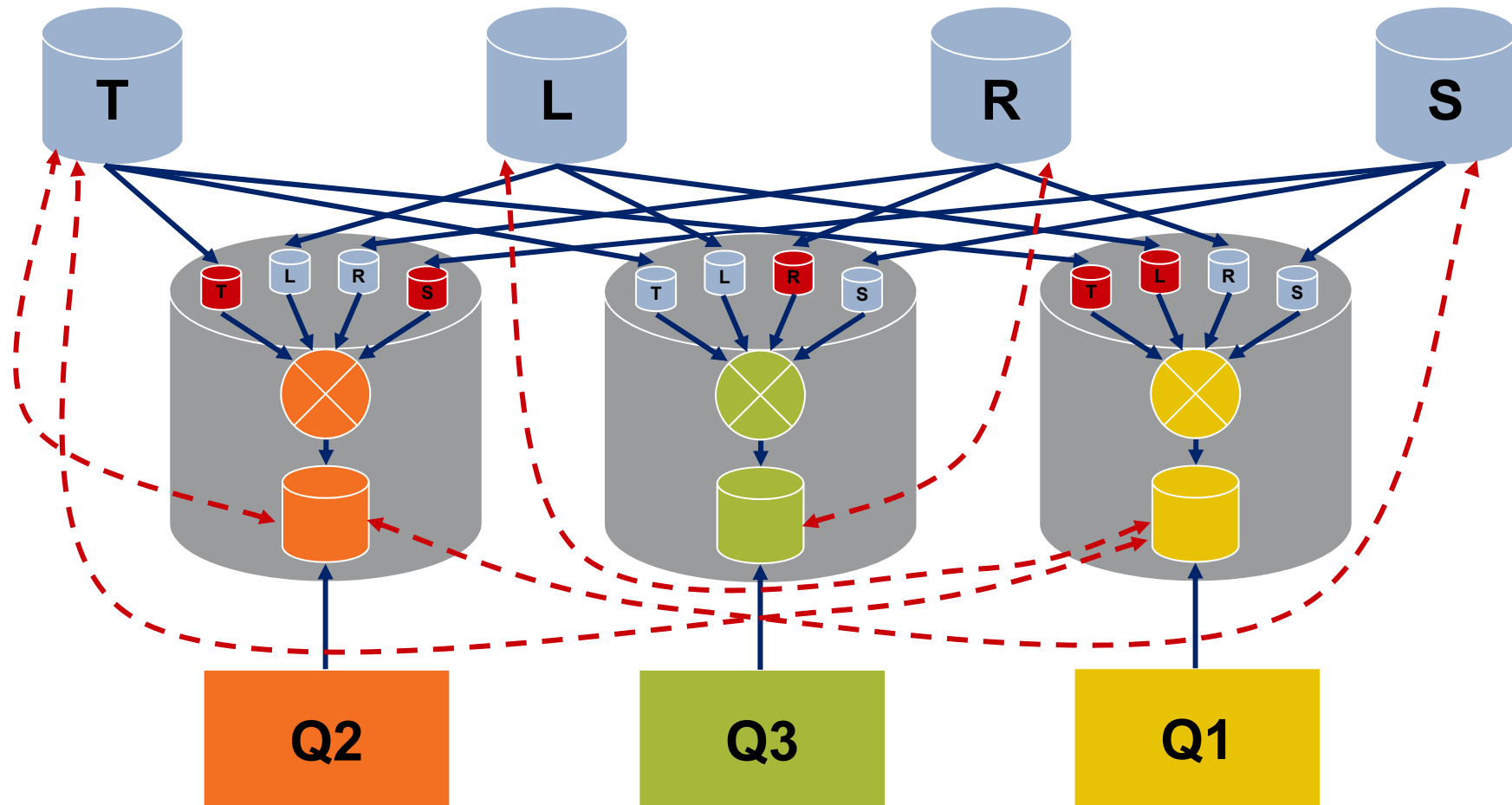
- Transactional
- Ledger
- Risk
- Static

We have typically solved these problems by creating a new database or adding specialised tables to our Finance Data Mart... but never exact copies!



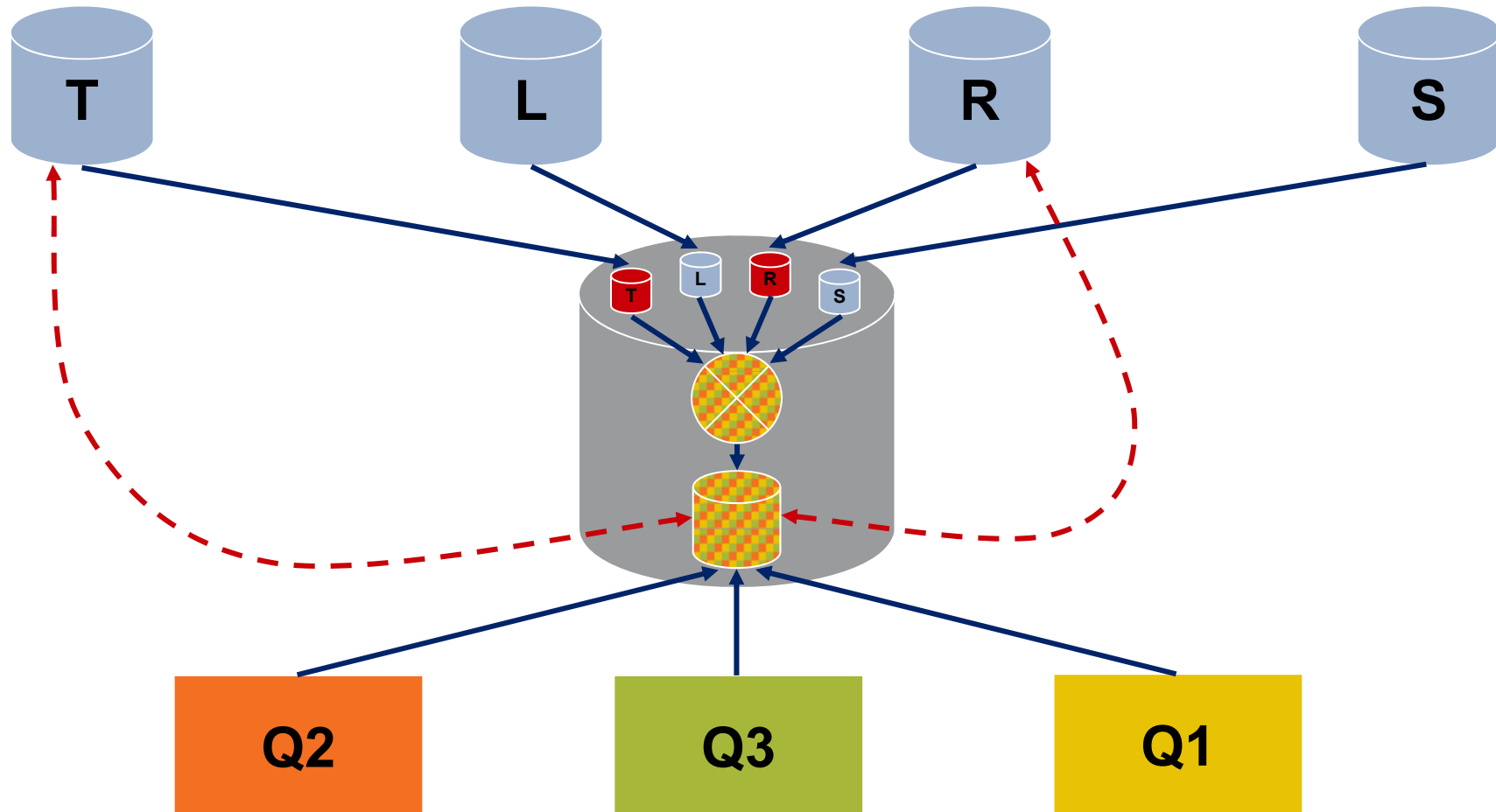
What is Data Virtualisation?

Each new question requires a new solution... and don't forget they may not be exact copies.



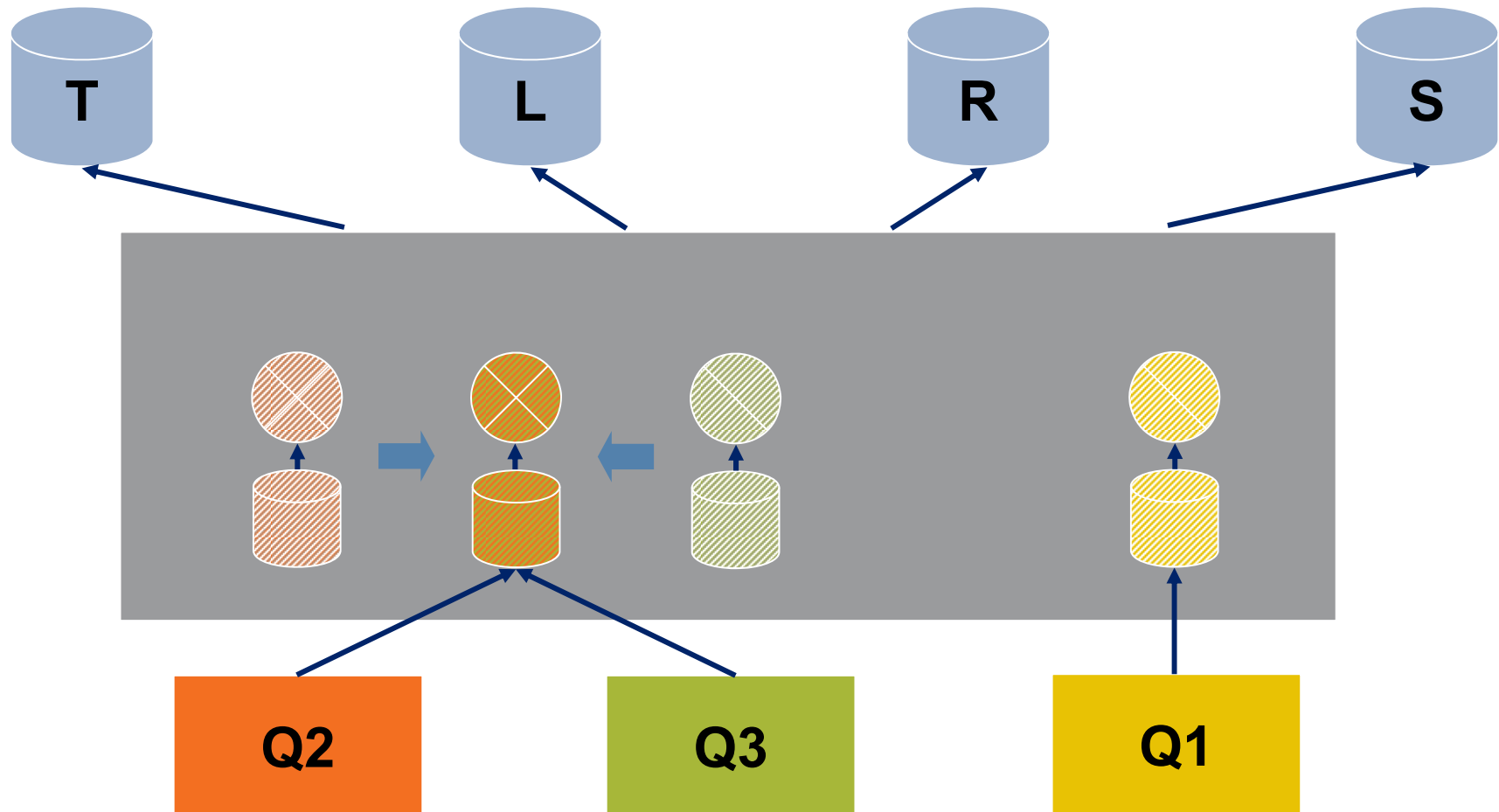
What is Data Virtualisation?

Stop proliferation through an Enterprise Data Warehouse...
but changes are still expensive... and you still need reconciliation



What is Data Virtualisation?

Stop proliferation through Virtual Databases...
changes are quick and easy... and so are transitions



What is Data Virtualisation?

Approaches compared... Virtual Databases chosen...
we call this the Data Access Layer (DAL)

Virtual Databases

Pros

- No data are copied. No reconciliations are required.
- Data model can be different for each requirement (either as transition or permanently)

Cons

- Data are retrieved as needed, meaning sources must be able to support demand and respond in a timely fashion.
- No advantage in optimisation.

Enterprise Data Warehouse

Pros

- Data only copied once, so only one reconciliation
- Data can be optimised for reporting

Cons

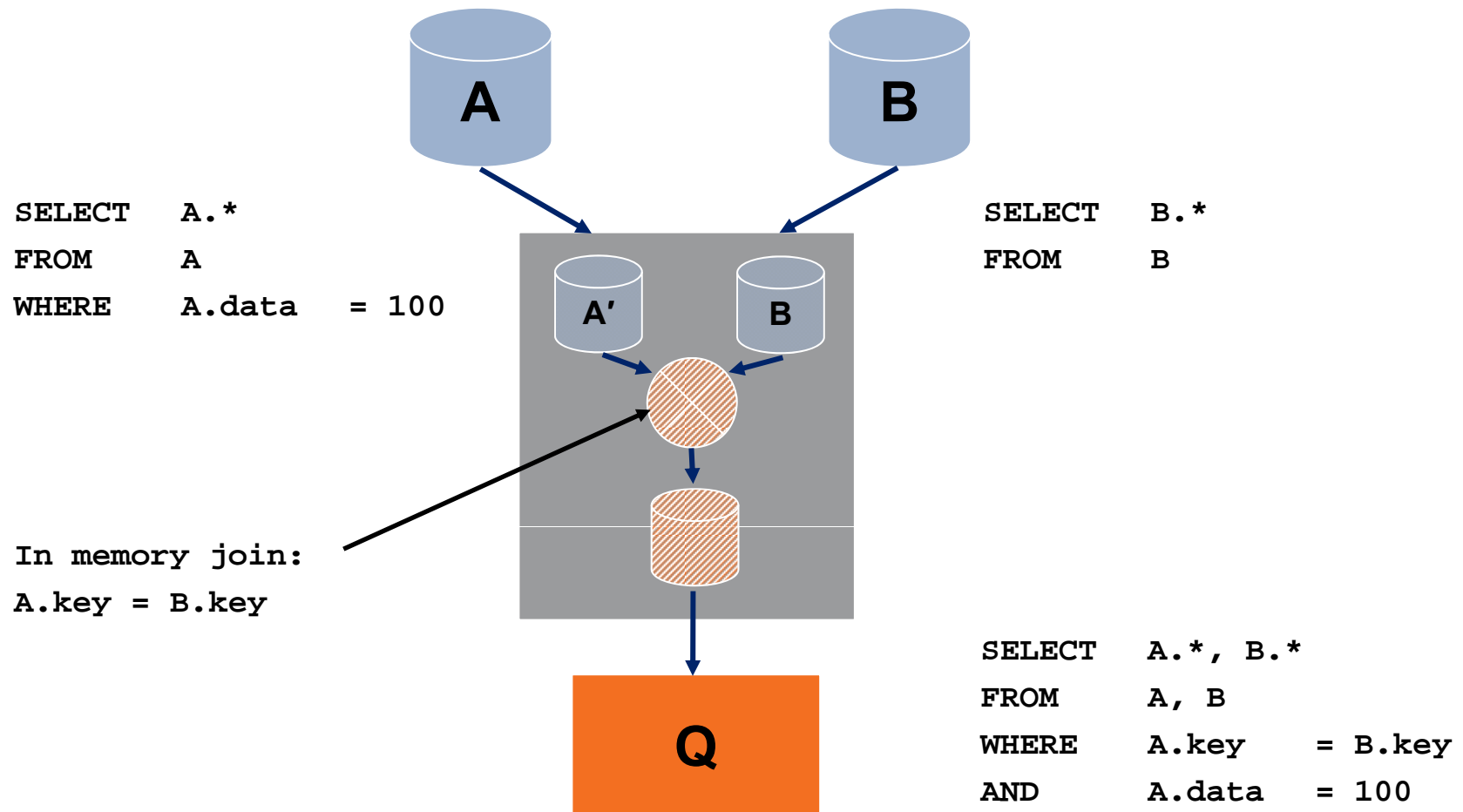
- Data are copied, so reconciliation is required.
- Design must take account of all requirements.
- Changes are expensive.

Data Virtualisation is ^{NOT} a Panacea

When to use it, when not to use it.

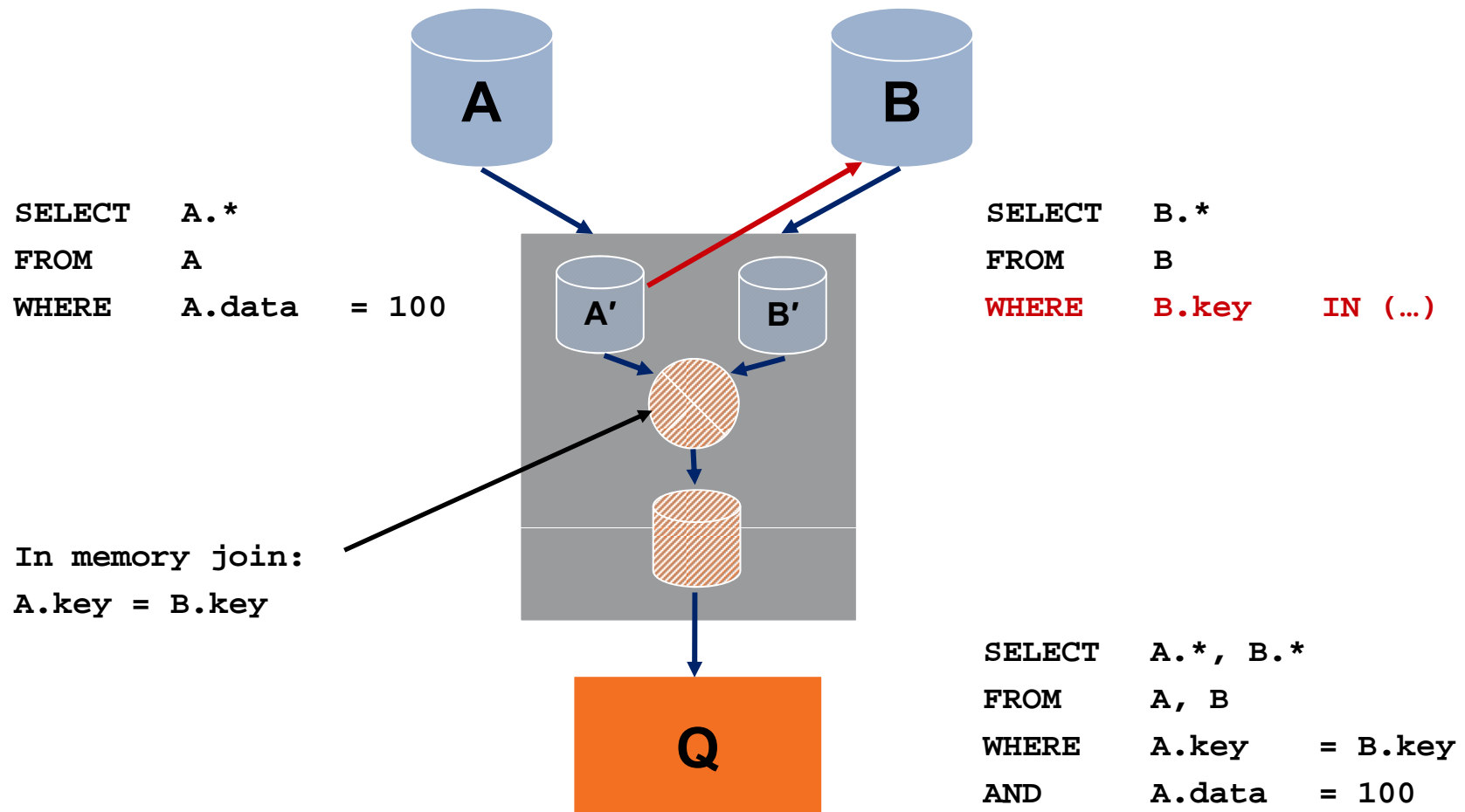
How Data Virtualisation Works

Naïve Query Federation



How Data Virtualisation Works

Better Query Federation



Data Virtualisation is not a Panacea

There is no “Magic”™

“You cannae change the laws of physics”:

- In general, we will move more data at query time than in a highly tuned data warehouse or data mart query and we will move it further, but we will not move data “just in case”.

Database design remains important:

- Poorly designed databases that don’t meet the query requirements will perform slowly and there will be no opportunity to fix it in a separate layer.
- This is not a substitute for every Data Mart we might create, but it should hide such optimisation decisions from most consumers.

Some data sources will not want to be “hit” directly:

- These sources will need to create a “reporting database”.
- Ideally, this will have the same ownership as the data source (including responsibility for data consistency, timeliness & the preferred data representation).

Data Virtualisation is not a Panacea

There is no “Magic”™

Data Quality is vital:

- We are effectively removing ETL, so existing opportunities to “fix up” are eliminated.
- While this presents a challenge, it is really a “good thing”.
- There is no excuse for not using Common Keys or for having multiple representations of the same data item. Get rid of all those UPPER(), LOWER() & DECODE() calls.

This approach relies on good quality data analysis:

- We are exposing our Corporate Data Model & Data Dictionary quite directly.
- Need a sound Data Architecture, defining the names, definitions and domains (sets of values) for all data items, starting with keys.

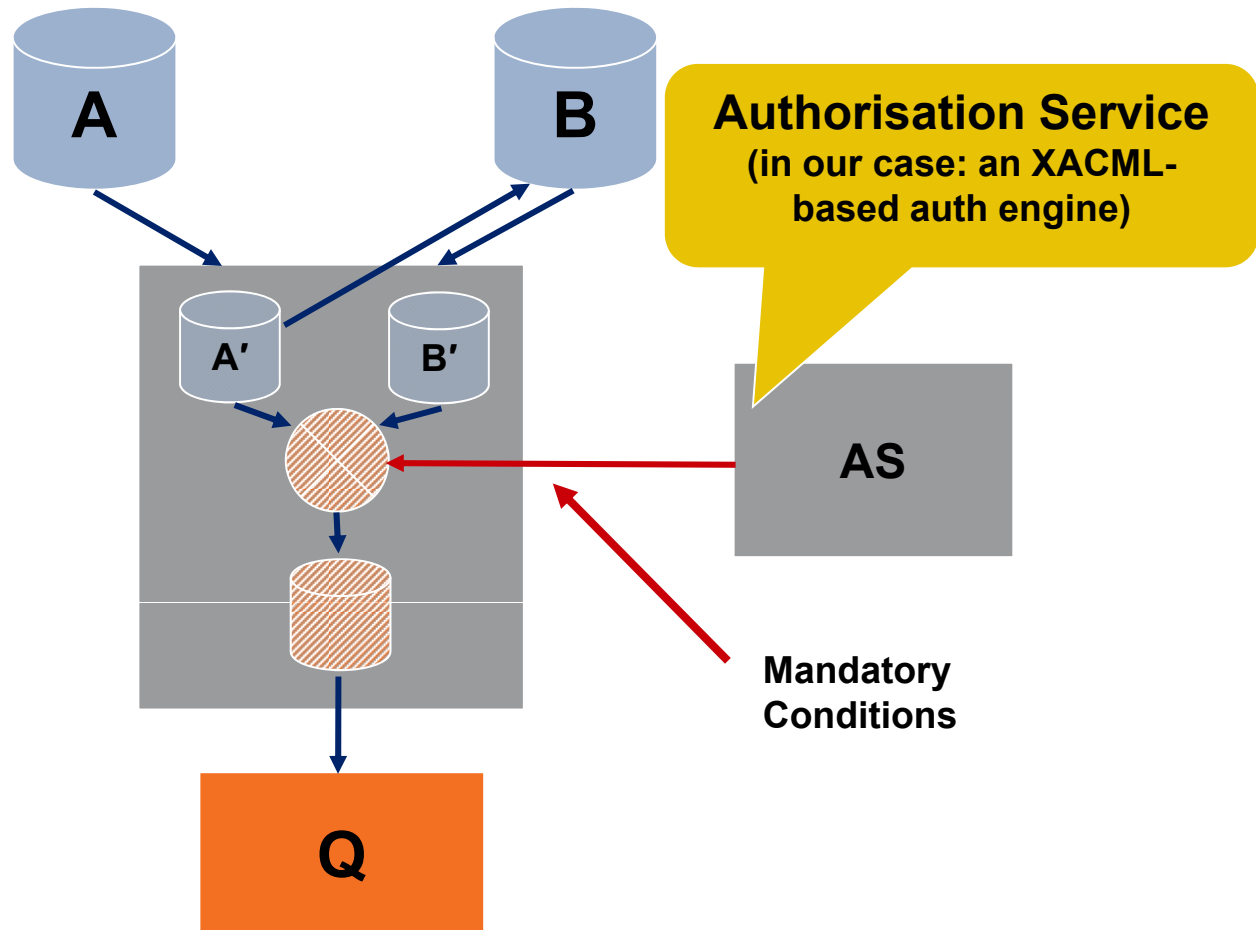
In general, Data Federation is best suited to situations where there are good quality identifying joins that reduce the resulting data sets significantly. It is definitely not suitable where extensive transformation is required.

The Opportunity to Think Differently

Beyond Simple Virtualisation.

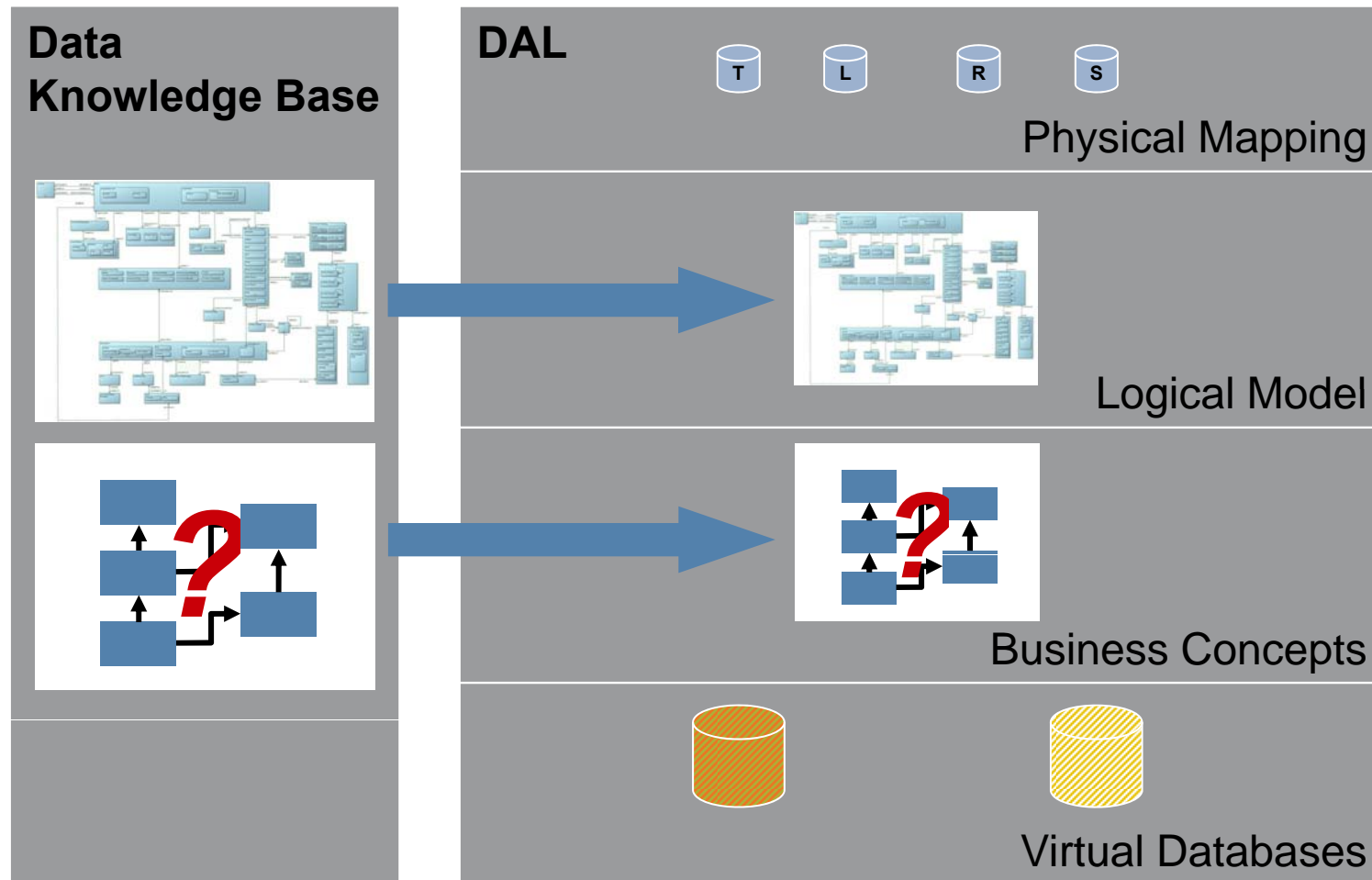
The Opportunity to Think Differently

Implementing Fine-Grained Entitlements



The Opportunity to Think Differently

Create a new model to hold Business Concepts



Business Concepts

A new way of looking at our data through virtual databases

Think about the data first

- Don't confuse "function" and "data"
- Expose "concepts" rather than "entities"

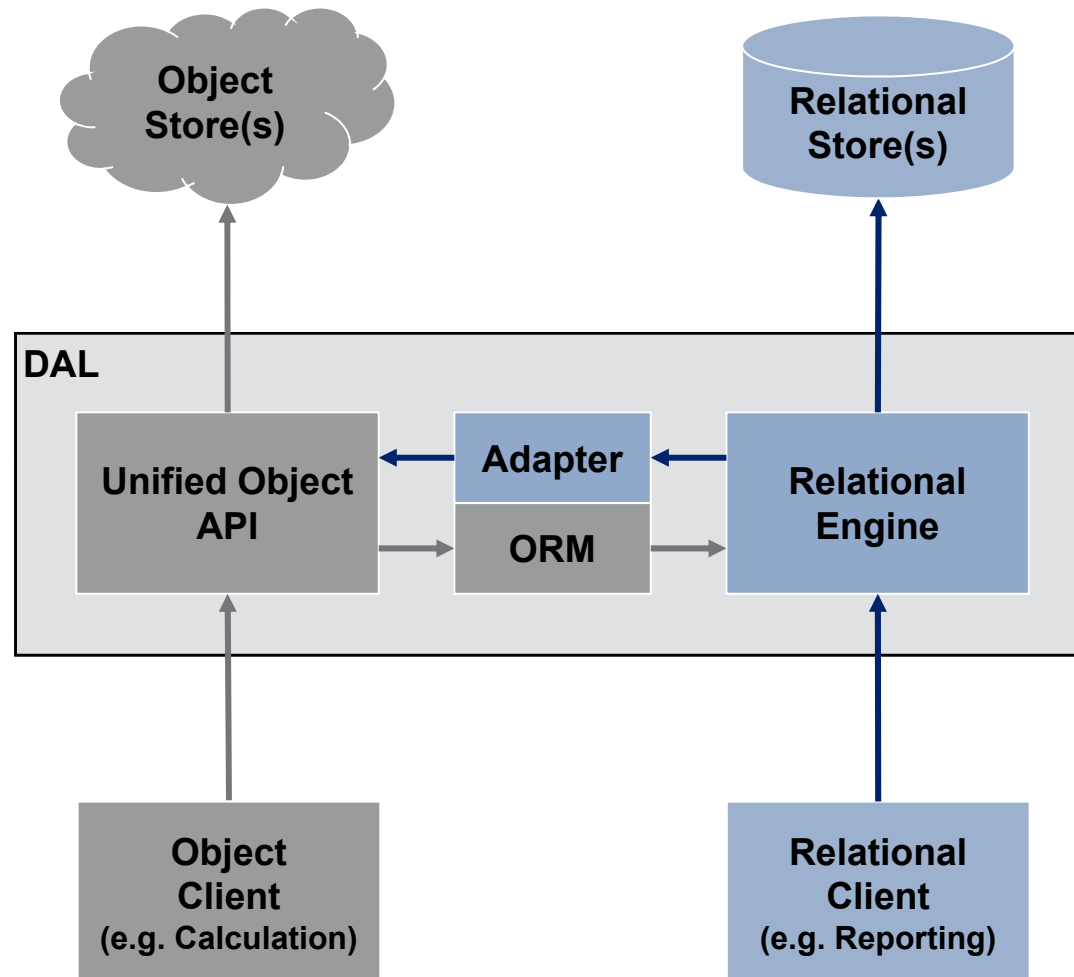
```
SELECT * FROM FINANCIAL_ACCOUNT_ITEM
WHERE  accounting_date = '2011/11/11'
AND    (
        account_type IN (3,4)
OR     account IN ('9100', ...)
      )
```



```
SELECT * FROM PROFIT_AND_LOSS
WHERE  accounting_date = '2011/11/11'
```

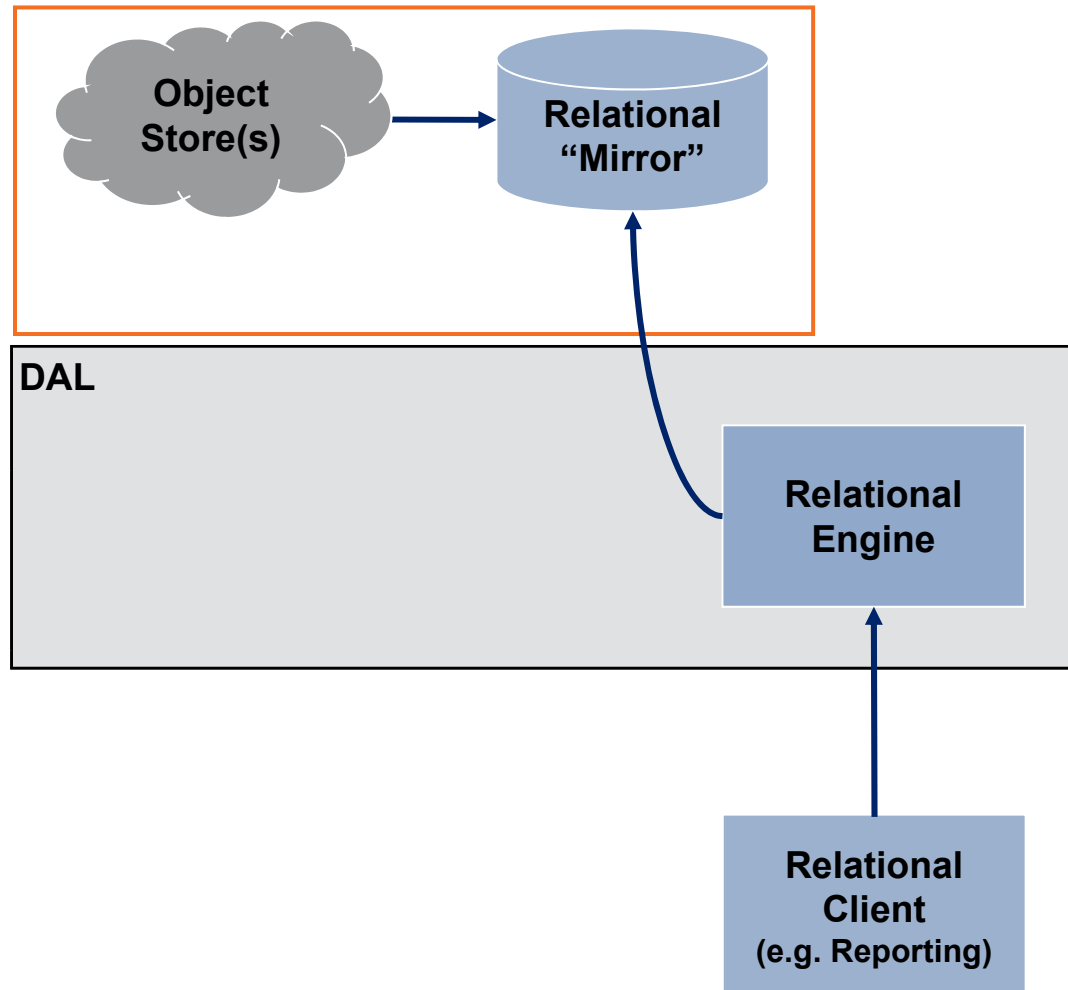
Object Interface

Extend Data Virtualisation to Objects



Object Interface Ownership

Alternative model for relational access to object stores



What have we done so far?

Status of Data Virtualisation in RBS M&IB Finance

DAL History

2011

Q1: we undertook a Proof of Concept

- Selected Red Hat JBoss EDS (was MetaMatrix until bought by Red Hat)

Q2-Q4: we built DAL v1 for a significant new Finance application

- Connects to:
 - Core Financial data (Ledger) by SQL
 - Enterprise Static data by SQL
 - Front Office view of P&L by custom XML interface

But there were conscious and unconscious compromises that had to be remediated.

2012

Q1: we worked to eliminate technical debt

- Established DAL as a standalone component

Q2: we have been introducing our first version of row-level security

- Somewhat integrated with FAS (Finance Authorisation Service)

For the rest of the year, we will be on-boarding new projects and extending the scope beyond Finance

DAL History

What we haven't done

Business Concepts

- Although we have a layer for these in the model, their implementation is rudimentary and we're not sure whether we really want the DAL to occupy this role.

Object Interface

- We have proved the viability of an object interface to our key object store (Operational Data Cache), but no solutions are live using this interface.
- We have not created an Object DAL.

Conclusions

What we've learned, Data Virtualisation is an option

Conclusions

What we've learned

Experience with Red Hat:

- Great support from enthusiastic people, but limited product knowledge in UK
- Free 5 day evaluation, including architect from US
- Very responsive to feature requests
- Have managed to “fit in” additional bug fixes late in release cycle

Experience with JBoss EDS:

- Some surprising bugs from such an established product
- IDE is slow and hard to use, may stem from port to Eclipse
- Documentation definitely needs an overhaul and needs some additional examples of advanced cases

Conclusions

Data Virtualisation is an Option

- Consider Data Virtualisation (& Data Federation) over traditional ETL/ELT techniques
- Can't magically fix underlying problems