Evolving Data Access Methodologies and Standards

John de Longa
Solutions Architect
DataDirect Technologies
John.de.longa@datadirect.com

30 Years of constant change

Relational

SQL

XML

ODBC

ISAM Files

XPath

SQLJ

JDBC

XQuery

OLE-DB

JPA

Entity EJBs

JDO

ADO.NET

OODBMS

Hierarchical

Object Relational Mapping

Database Vendors
Client Access

Past attempts to impose order by Database Vendors

- DBMS Vendor “One Database” programs philosophy
- Up until the early 1990s database access was generally controlled by the Database vendors (and still the case in certain fields!)
- Typically organisations focused on one or two databases and made use of the Database vendors APIs
- Such as
  - Oracle OCI
  - Sybase Open Client
  - IBM DB2 CLI / DRDA
  - Microsoft DB Lib
- Applications tended to be capable of only connecting to one Database

Past attempts to impose order by evolving standards

- From the early 1990s various standards and Frameworks have been introduced to provide more easily Data Access to multiple data stores that includes RDBMS

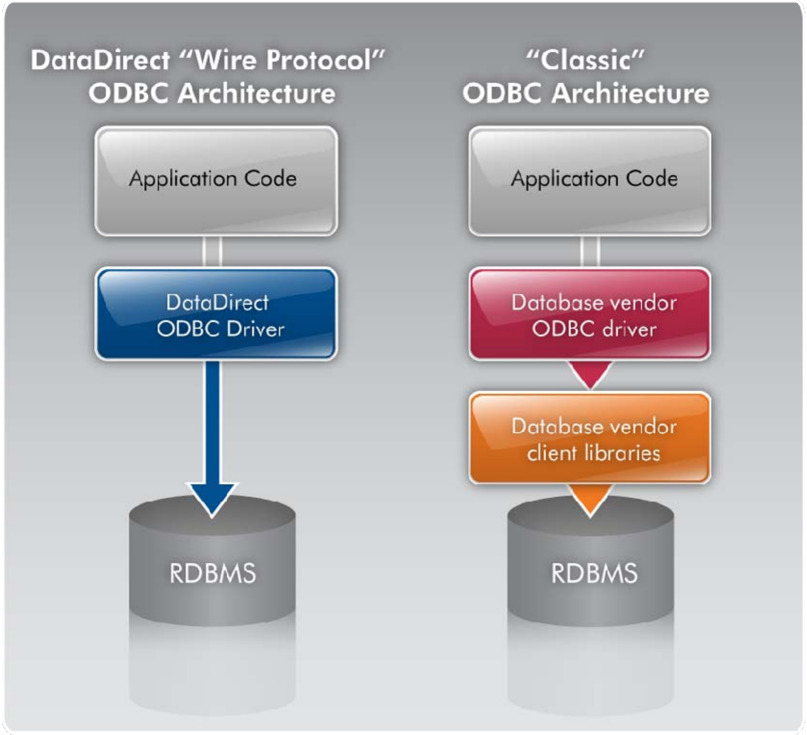
ODBC

The SQL Access Group formulated a standard approach that allowed application to more easily access multiple data stores.

Microsoft published the ODBC standard in 1992

Original ODBC Architecture
Using Database Client Libraries

Evolved Architecture makes use of just Wire Protocol


DataDirect "Wire Protocol" ODBC Architecture
"Classic" ODBC Architecture
Application Code
Application Code
DataDirect ODBC Driver
Database vendor ODBC driver
Database vendor client libraries
RDBMS
RDBMS

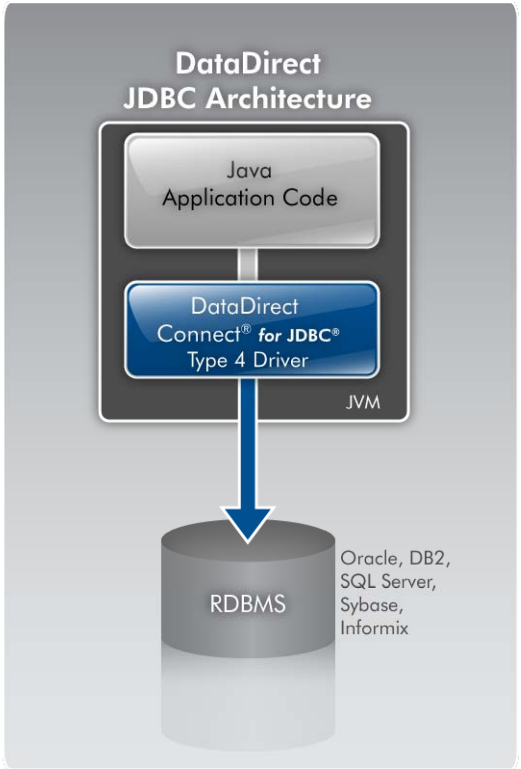
Past attempts to impose order

- From the early 1990s various standards and Frameworks have been introduced to provide more easily Data Access to multiple data stores that includes RDBMS

JDBC

SUN in 1995 added JDBC to Java to allow direct access from Java JVMs to data stores and Databases.

Over time JDBC Type 4 has become the JDBC Driver of choice. The database access Wire Protocol is embedded in the JDBC driver



Past attempts to impose order

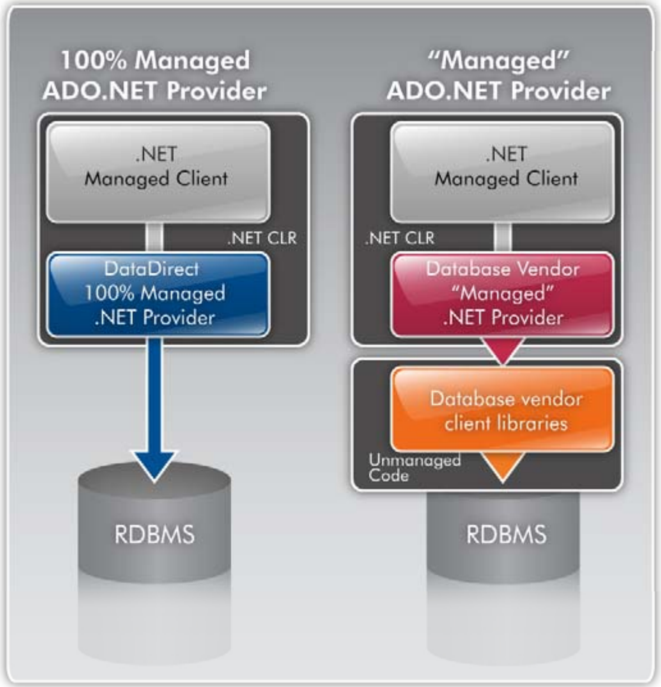
- From the early 1990s various standards and Frameworks have been introduced to provide more easily Data Access to multiple data stores that includes RDBMS

ADO.NET

Microsoft over time have introduced additional standards such as ADO OLE/DB.

More recently the ADO.NET Providers associated with the .NET Framework has become well accepted.

ADO.NET 100% Managed Providers remove the need for Client Libraries


100% Managed ADO.NET Provider
.NET Managed Client
.NET CLR
DataDirect 100% Managed .NET Provider
RDBMS
"Managed" ADO.NET Provider
.NET Managed Client
.NET CLR
Database Vendor "Managed" .NET Provider
Database vendor client libraries
Unmanaged Code
RDBMS

Key Development Phases

- Stove Pipe
  - Where applications were single database centric via database vendor's APIs
- Client Server
  - Initially single database focus
  - With advent of ODBC, JDBC etc applications could more easily access multiple databases and related stores
- Web / Intranet based solutions
  - Application server hosted applications became popular in both the Windows and Java worlds
- XML
  - XML evolving as a set of standards to store data locally and for the transferring data between organisations
- Enterprise wide architectures
  - ESB, SOA and Web Services have all become popular in larger organisations

SOA

- SOA is an architectural style that encourages the creation of loosely coupled business services

- Loosely coupled services that are interoperable and technology-agnostic enable business flexibility

- An SOA solution consists of a composite set of business services that realize an end-to-end business process

- Each service provides an interface-based service description to support flexible and dynamically re-configurable processes

SOA 2007

+ Allows reuse of existing software assets

+ Provides architecture for disparate IT systems

+ Meets goals of abstracted business processes, programming paradigms, architectures, etc.

- Business Logic and Data Access Logic not fully separated

- Data management guidelines are sparse and are typically formulated by SOA experts and not by data management authorities

SOA Data Management Uses Classical Data APIs

- Defined interfaces include:
  - JDBC
  - ADO.NET

- These all roughly do the same thing
  - Provide APIs to connect to data, issue queries, return data
  - Reuse legacy business knowledge

Characteristics of Traditional Data Access APIs vs. SOA Characteristics

Traditional Data Access	SOA
• Tightly coupled	• Loosely coupled
• Complex State Machine	• Stateless
• Connection based	• Message based
• Well defined API	• “RPC” model
• Mostly synchronous	• Synchronous / Asynchronous
• Relational model driven • SELECT then Fetch model	• XML data interchange

Today’s Data Access for SOA

- Data is all over the place
  - Many heterogeneous data sources
  - Fragmented data across data sources

- Data access is complex
  - Multiple programming languages and data access APIs for various clients

- Data centric strategies embraced by SOA
  - Master data management MDM
  - Customer Data Integration CDI
  - General Data Governance

- Developing applications is expensive
  - Costly to build and maintain
  - Hard to enforce Data security consistently

Need for a Data Access Standard for SOA era

- A group of leading software vendors have committed to develop further a group of Standards that makes data access a loosely connected data service.

- The OSOA Collaboration represented by eighteen leading technology vendors announced that key specifications had completed incubation and has be submitted to OASIS. Some of the leaders include IBM, Oracle, Progress SAP and Software AG

- One of the new standards is Service Data Objects (SDO)

- Related to new standards is another standard being formulated called Data Access Service (DAS)

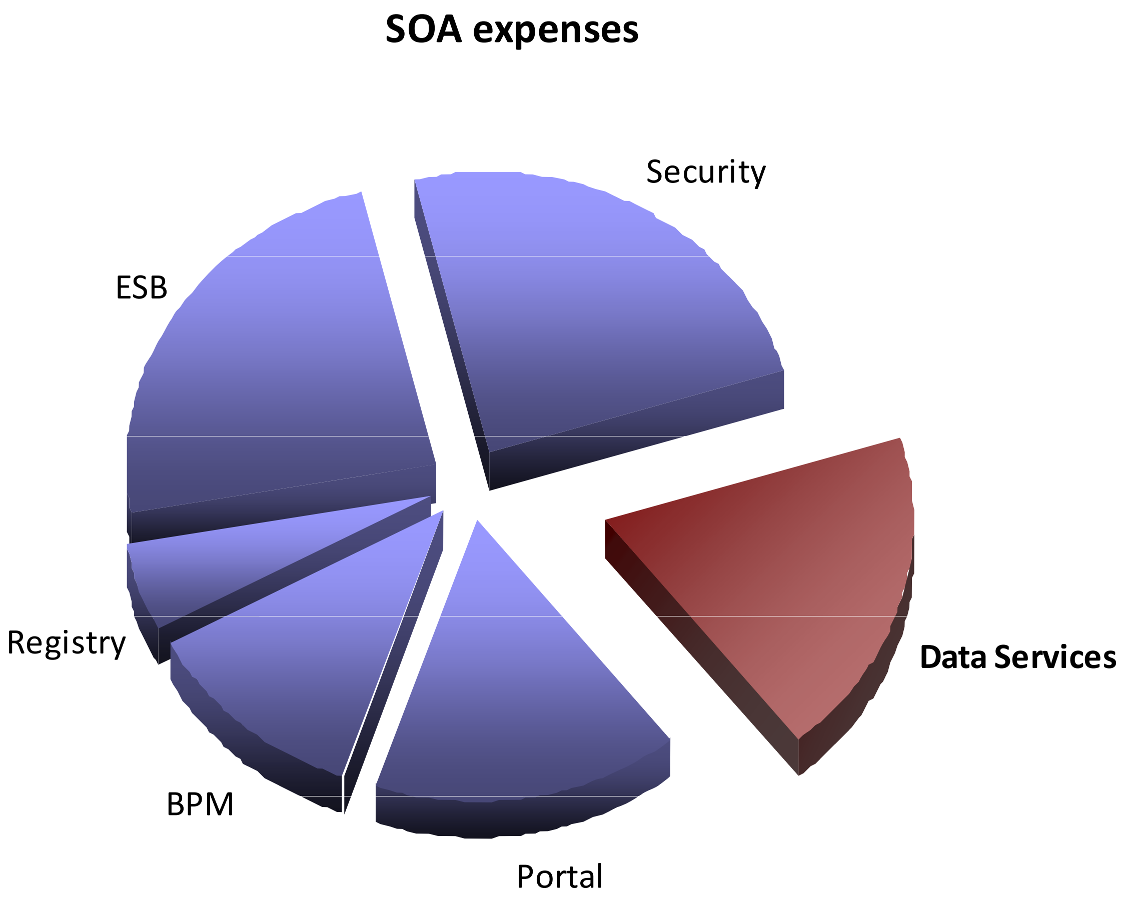
- The rest of the presentation will be concentrating on SDO and DAS

What exactly is SDO? The SDO Goals …

- Unified and Consistent Data access to heterogeneous data sources
  - Simplified programming model for application programmers
  - Enable Tools and Frameworks to work consistently across heterogeneous data sources

- To become the standard for Data Access in a Service Oriented Environment (SOA)
  - ODBC is the standard for C++ just like SDO will be the standard for SOA

Growth of Data Services in SOA environments



Source: GCR 2006

Thinking of Data Services like a database

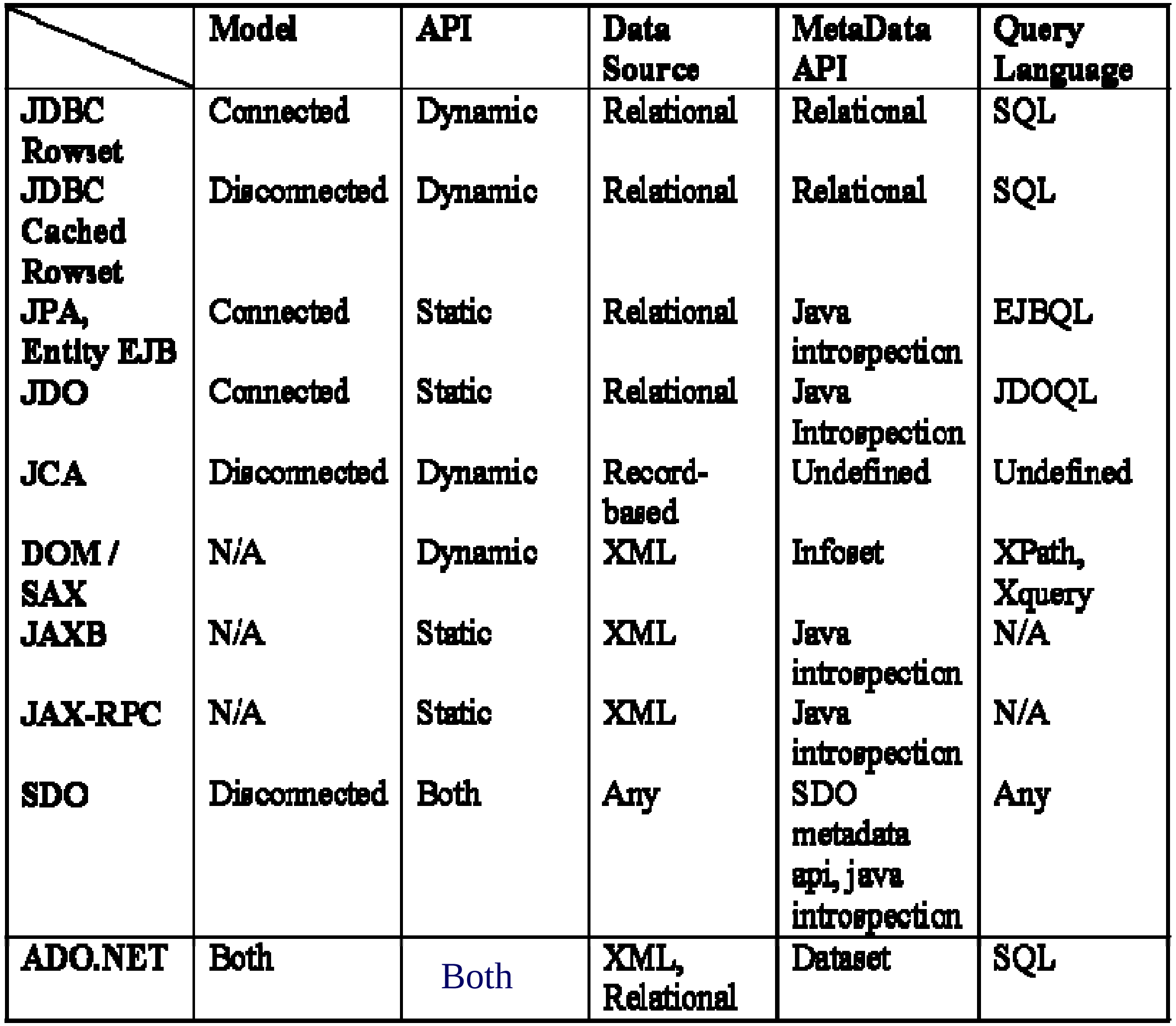
An analogy: Think of a Data Service Platform as:

A DBMS, complete with tools for application developers and administrators, without its own permanently stored data files, but with the ability to access all corporate data.

- Data Modelling language
- Data Manipulation language
- Data Query Language
- Data structures (in-memory only, no persistent storage)
- Transaction mechanism
- Backup and replication
- Rule enforcement
- Security
- Computation
- Change and access logging
- Automated optimization
- Metadata repository

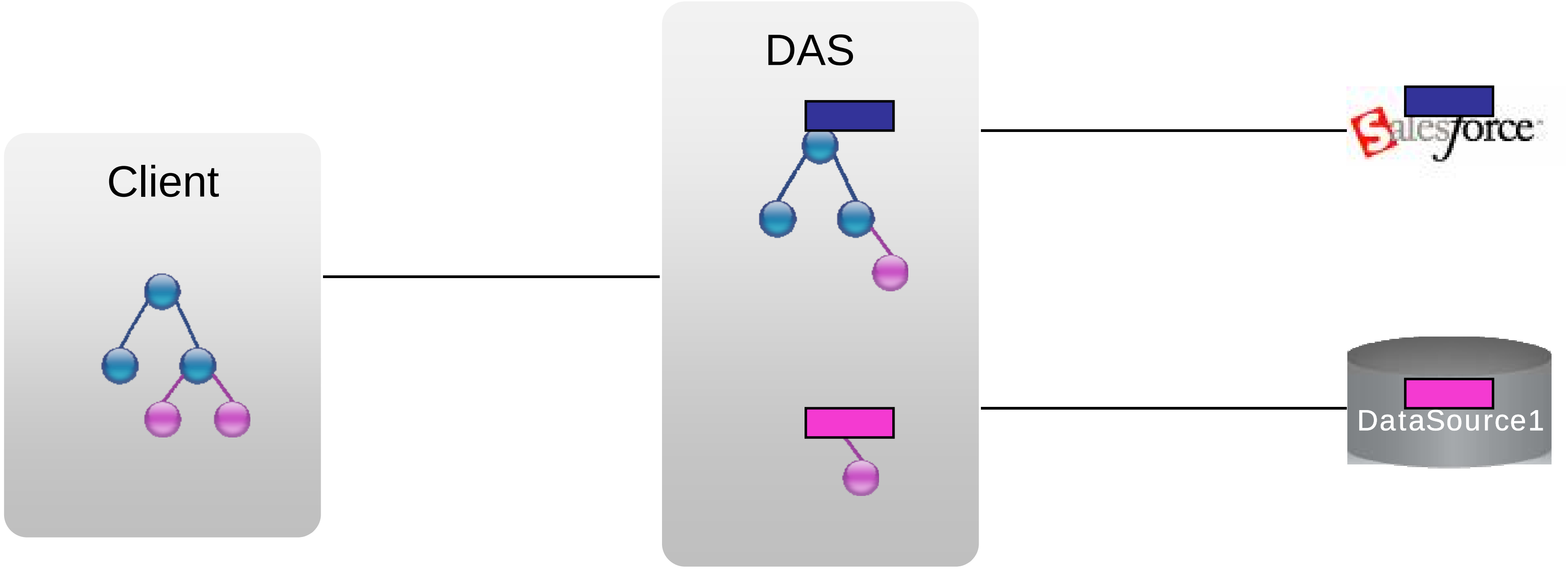
SDO: Relationship to other APIs/Standards

:

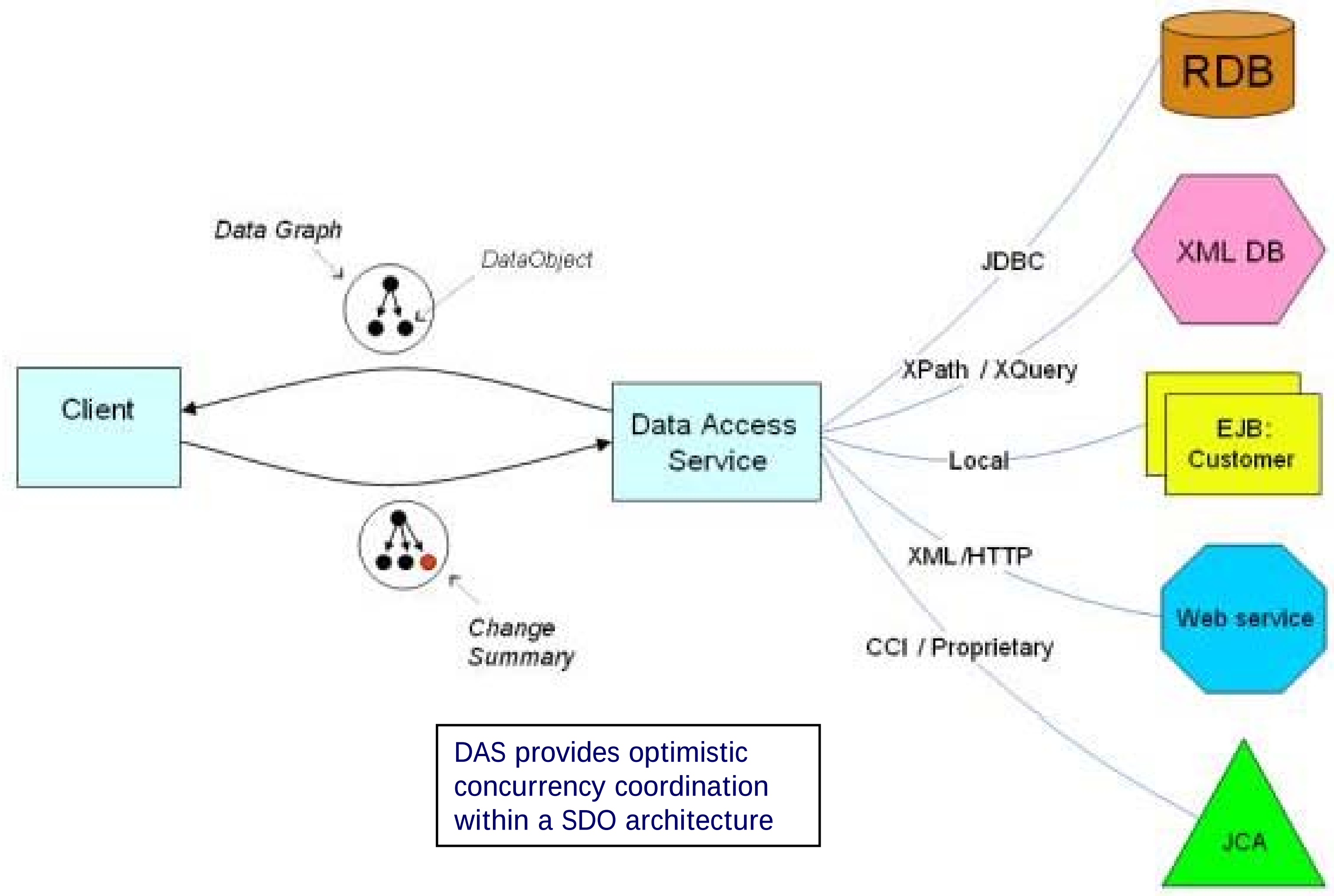


	Model	API	Data Source	MetaData API	Query Language
JDBC Rowset	Connected	Dynamic	Relational	Relational	SQL
JDBC Cached Rowset	Disconnected	Dynamic	Relational	Relational	SQL
JPA, Entity EJB	Connected	Static	Relational	Java introspection	EJBQL
JDO	Connected	Static	Relational	Java Introspection	JDOQL
JCA	Disconnected	Dynamic	Record-based	Undefined	Undefined
DOM / SAX	N/A	Dynamic	XML	Infoset	XPath, Xquery
JAXB	N/A	Static	XML	Java introspection	N/A
JAX-RPC	N/A	Static	XML	Java introspection	N/A
SDO	Disconnected	Both	Any	SDO metadata api, java introspection	Any
ADO.NET	Both	Both	XML, Relational	Dataset	SQL

SDO Heterogeneous Fetch & Update



SDO Heterogeneous Data Access



DAS provides optimistic concurrency coordination within a SDO architecture

Service Data Objects (SDO) platforms

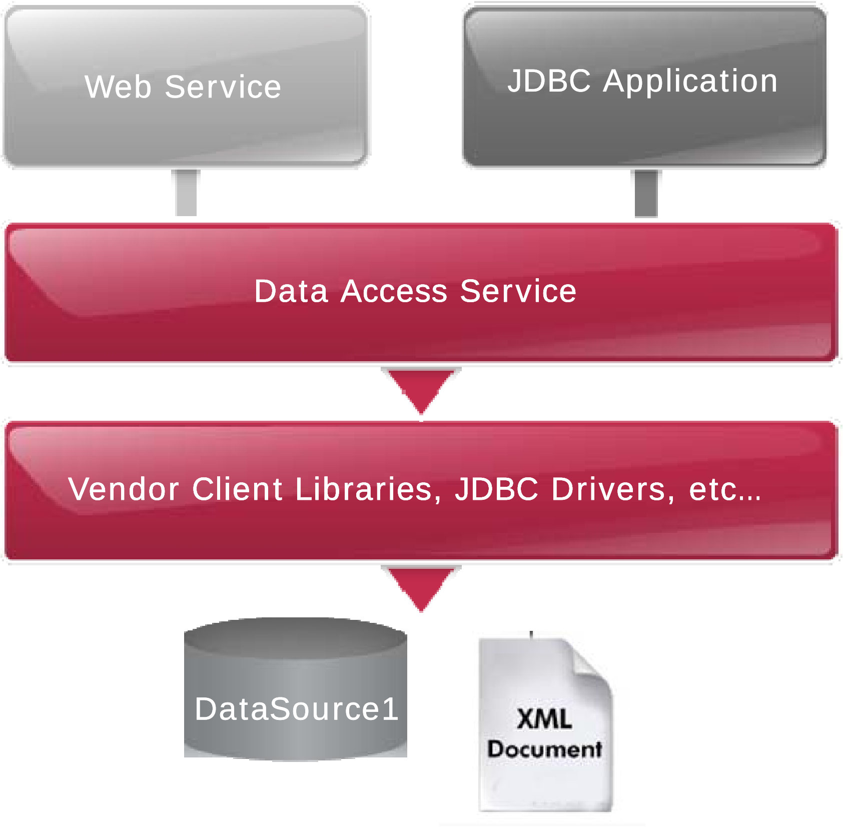
- Service Data Objects (SDO) are designed to simplify and unify the way in which applications handle data.
- Using SDO, application programmers can uniformly access and manipulate data from heterogeneous data sources, including relational databases, XML data sources, Web services, and enterprise information systems.
- These specifications can be downloaded from the
- Open SOA Collaboration.
- For more information, see the SDO v2.1 white paper.
- SDO for Java and C++
- SDO for PHP
- SDO for C
- SDO for COBOL

What do the analysts say?

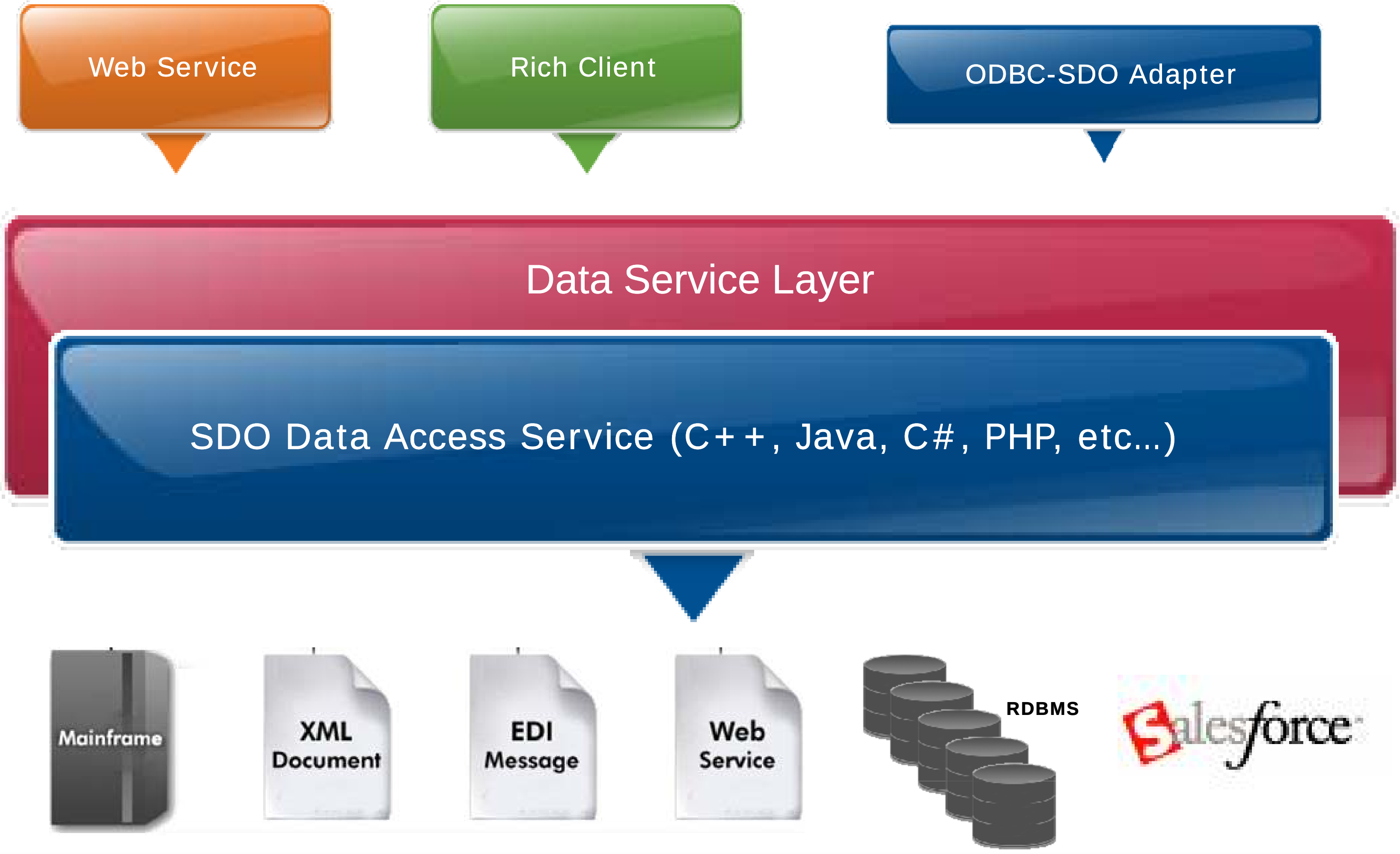
- **Gartner Hype Cycle for Data Management**
  - “On the rise: data service architectures”
  - “Data services are, by their nature, a new style of data access strategy that replaces the data management, access and storage duties currently deployed in an application-specific manner.
  - Data services architecture is merely a sub-class or category of SOA that does not form a new architecture, but brings emphasis to the varying services that exist within SOA.”

Traditional and Web based Services embraced by DAS

Typical DAS Uses Client Libraries or JDBC Drivers to access Data Sources
Both Web Services and Connection based applications can make use of DAS



A Data Services Strategy



Web Service

Rich Client

ODBC-SDO Adapter

Data Service Layer

SDO Data Access Service (C++, Java, C#, PHP, etc...)

Q&A

DataDirect®
TECHNOLOGIES