



DAO

Data Access Objects

DAO 1.0 появилась в ноябре 1992 года как API для работы с СУБД Jet.

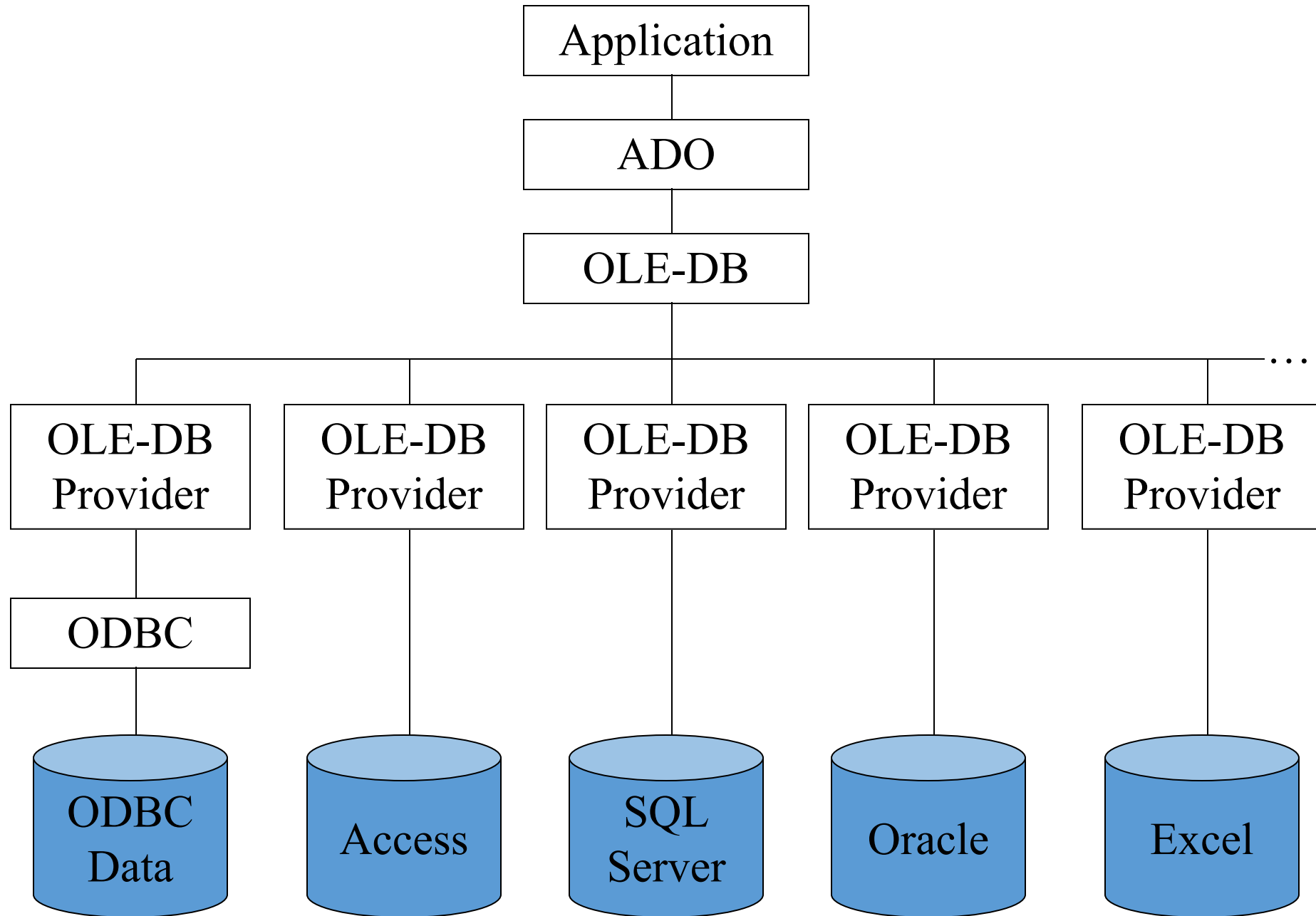
Поддерживала доступ к файлам формата MDB (Microsoft Access), ODBC-источникам данных и к источникам данных ISAM.

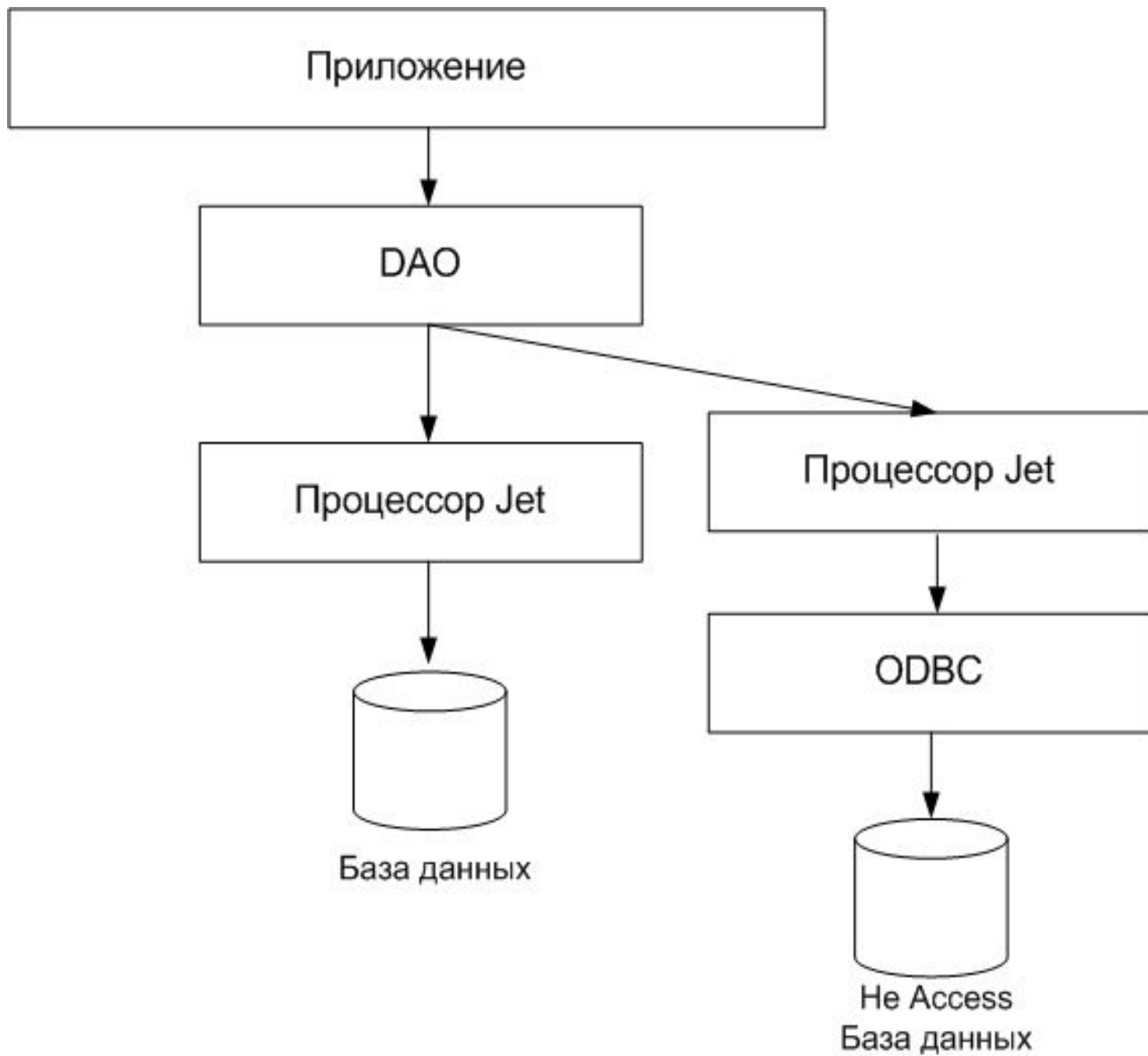
С 1996 года компания Microsoft стала ориентироваться на поддержку другой, более универсальной технологии доступа к данным — OLE DB\ADO.

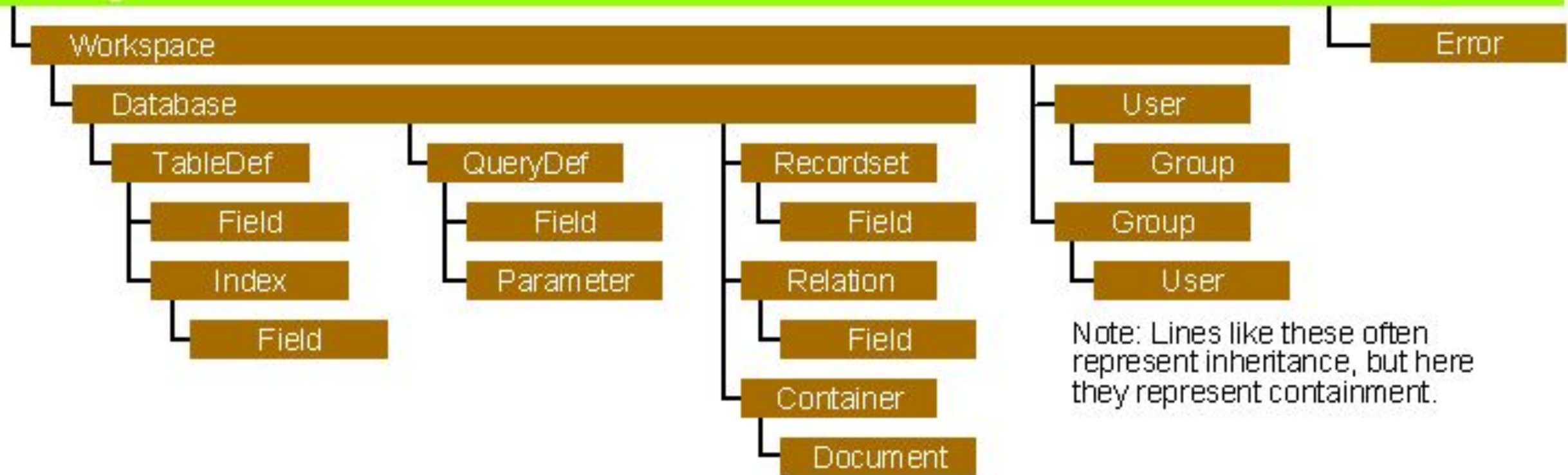
Тем не менее технологии Jet\DAO продолжали развиваться до 2001 года, когда Microsoft анонсировала прекращение поддержки этих технологий.

Основные версии DAO и продукты, в составе которых они поставлялись:

- версия 1.0, Visual Basic 2.0, Visual Basic 3.0, Microsoft Access 1.1
- версия 2.0, Microsoft Access 2.0
- версия 2.5, Visual Basic 4.0 (16-bit only)
- версия 3.0, Visual Basic 4.0, Microsoft Access 95, Microsoft Excel 95, Visual C++ 4.03
- версия 3.5, Microsoft Access 97, Visual Basic 6.0
- версия 3.6, Microsoft Access 2000, XP







ADO Object Model

4 ОСНОВНЫХ ОБЪЕКТА:

- * **Connection** – объект, управляющий соединением между приложением и хранилищем БД.
- * **Command** – запуск SQL запросов к БД
- * **Recordset** – содержит набор записей как результат выполнения команды-запроса.
- * **Stream** – позволяет оперировать с данными, содержащихся на web-ресурсах (например: обработка HTML)

DAO Connections

DBEngine

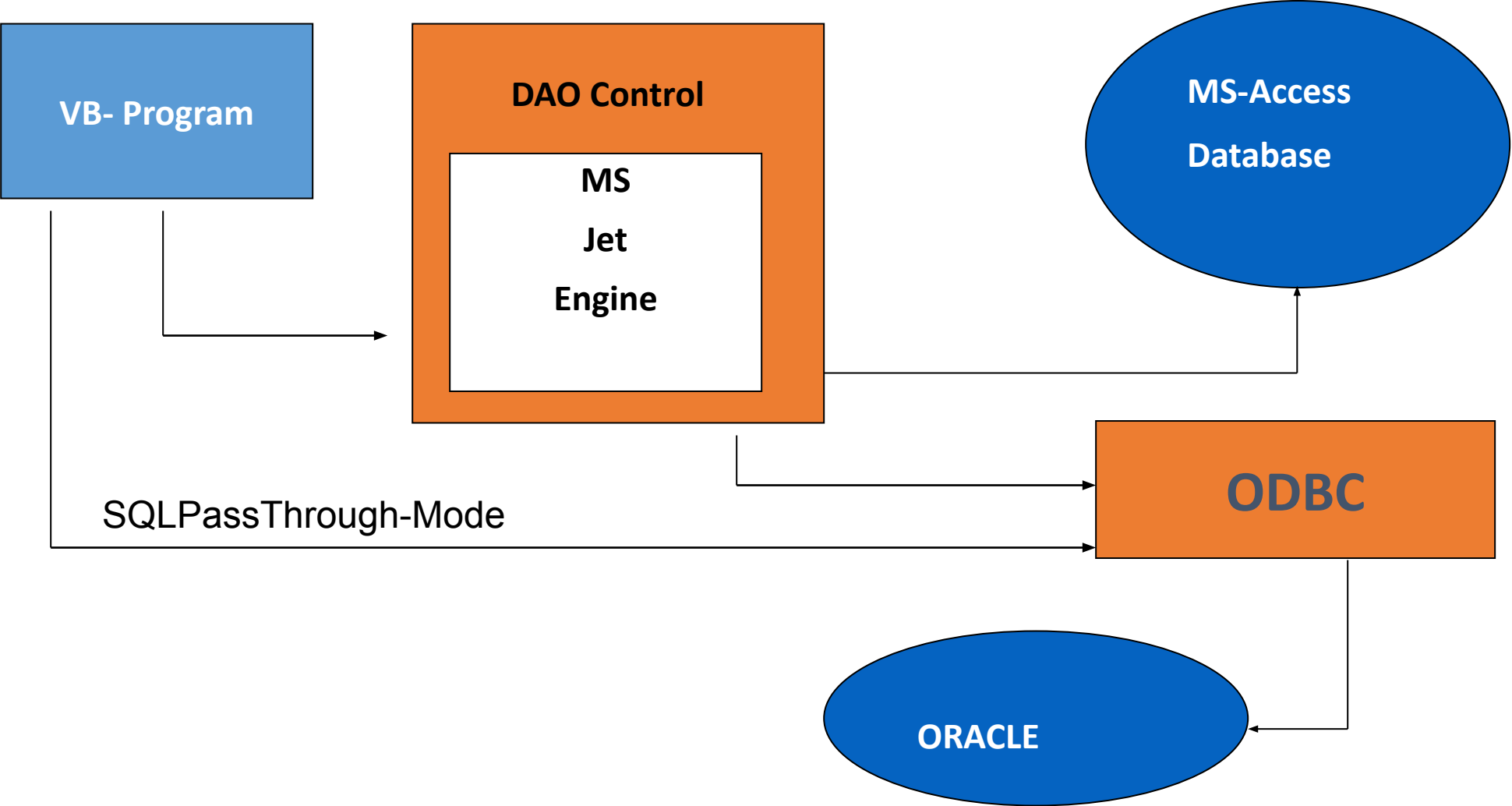
Class	DAO object	Remarks
CDaoWorkspace	Workspace	Manages a transaction space and provides access to the database engine.
CDaoDatabase	Database	Represents a connection to a database.
CDaoTableDef	Tabledef	Used to examine and manipulate the structure of a table.
CDaoQueryDef	Querydef	Used to store queries in a database. we can create recordsets from a querydef or use it to execute action or SQL pass-through queries.
CDaoRecordset	Recordset	Used to manage a result set, a set of records based on a table or selected by a query.
CDaoException	Error	responds to all DAO errors by throwing exceptions of this type.


```

1 Dim dbs As Database
2 Dim tdf As TableDef
3 Dim fldID As Field
4 Dim fldName As Field
5 Dim fldResponse As Field
6 Dim fldClass As Field
7
8 Set dbs = OpenDatabase("C:\MYTEST.MDB")
9 Set tdf = dbs.CreateTableDef("Marketing Survey")
10
11 set fldID = tdf.CreateField("ID", dbInteger)
12 fldID.Required = True
13
14 Set fldName = tdf.CreateField("Name", dbText)
15 fldName.Required = True
16 fldName.Size = 40
17 fldName.AllowZeroLength = True
18 fldName.DefaultValue = "Unknown"
19
20 Set fldResponse = tdf.CreateField("Response", dbMemo)
21 set fldClass = tdf.CreateField("Class", dbText, 10)
22 fldClass.Required = True
23 fldClass.ValidationRule = "in('A','B','X')"
24 fldClass.ValidationText = "Enter of of A, B, or X"
26 tdf.Fields.AppendFldID
27 tdf.Fields.Append fldName
28 tdf.Fields.Append fldResponse
29 tdf.Fields.Append fldClass
30 dbs.TableDefs.Append tdf
31 dbs.Close

```

DAO Connections



Example: Obtaining Information About Querydefs

```
// pDB is a pointer to a CDaoDatabase object
// Allocate a CDaoQueryDefInfo object to
// receive the information
CDaoQueryDefInfo queryinfo;
int nQueries = pDB->GetQueryDefCount( );
for ( int i = 0; i < nQueries; i++ )
{
    pDB->GetQueryDefInfo( i, queryinfo );
    if ( queryinfo.m_strName == "Senior Students" )
    {
        // Get other information about the query ...
        // ...
        break;
    }
}
```

Quick SQL Pass-Through Queries

```
// pdb is a pointer to a CDaoDatabase object
// (an .MDB database)
// Set up the connection string
// Security Note: Hard-coding a password is a major security weakness.
// Either query for a user ID and password or specify a trusted connection
// (which uses Windows NT integrated security) instead.
CString strConnect = "ODBC;DSN=ntstuff;Trusted_Connection=Yes;APP=App
Name;WSID=MyComputer;DATABASE=pubs;TABLE=dbo.authors;";
pdb->SetConnect( strConnect );
// Use SQL pass-through in a recordset
// Set up the SQL and open the recordset
CString strSQL = "whatever";
CDaoRecordset rs( pdb );
try
{
    rs.Open( dbOpenSnapshot, strSQL, dbSQLPassThrough );
    // ...
}
// ...
```

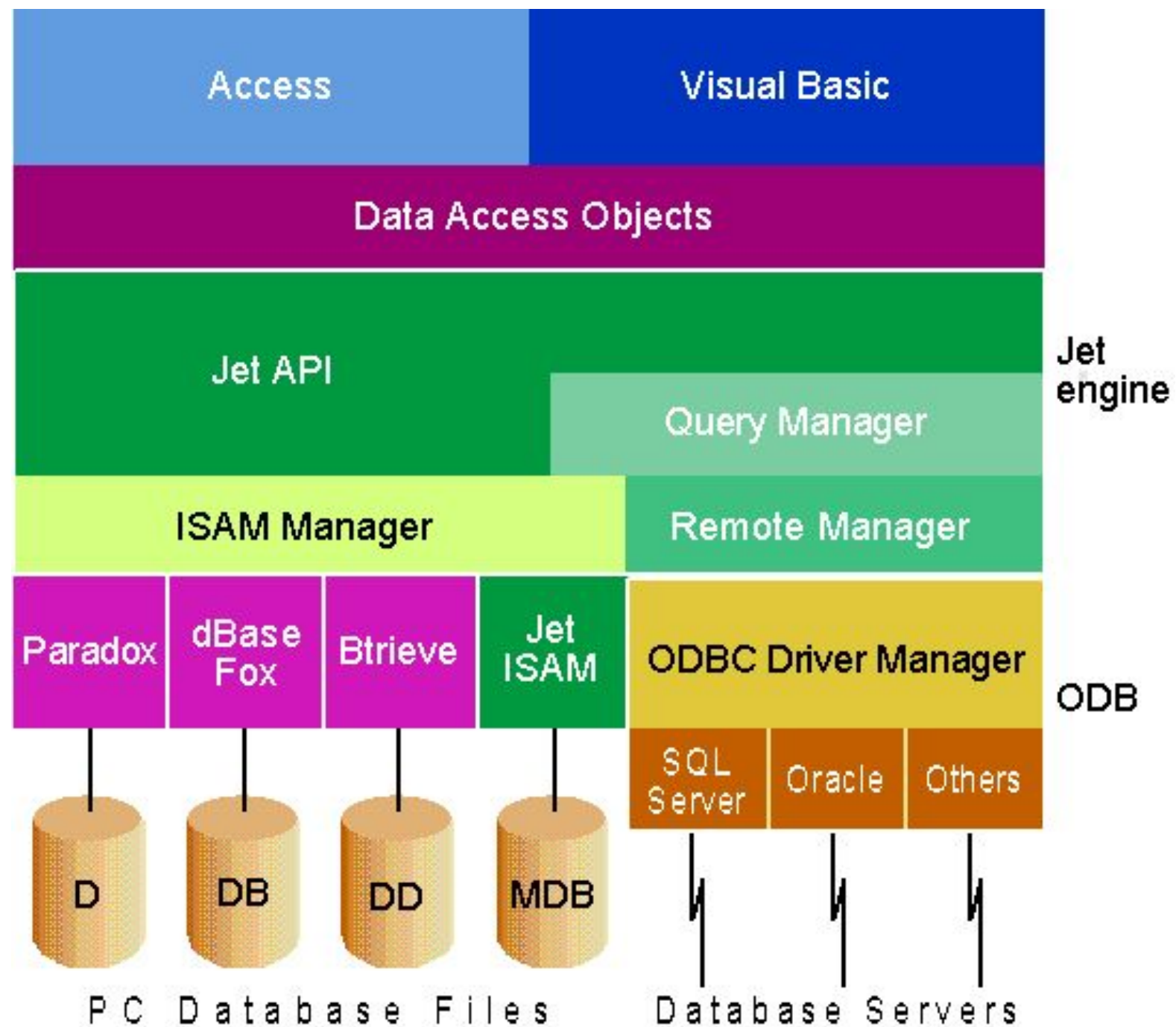
RDO

Remote Data Objects

RDO (сокр. от англ. Remote Data Objects) — технология доступа к базам данных компании Microsoft.

Представляет собой набор COM-объектов, инкапсулирующих ODBC API, а также клиентскую курсорную библиотеку.

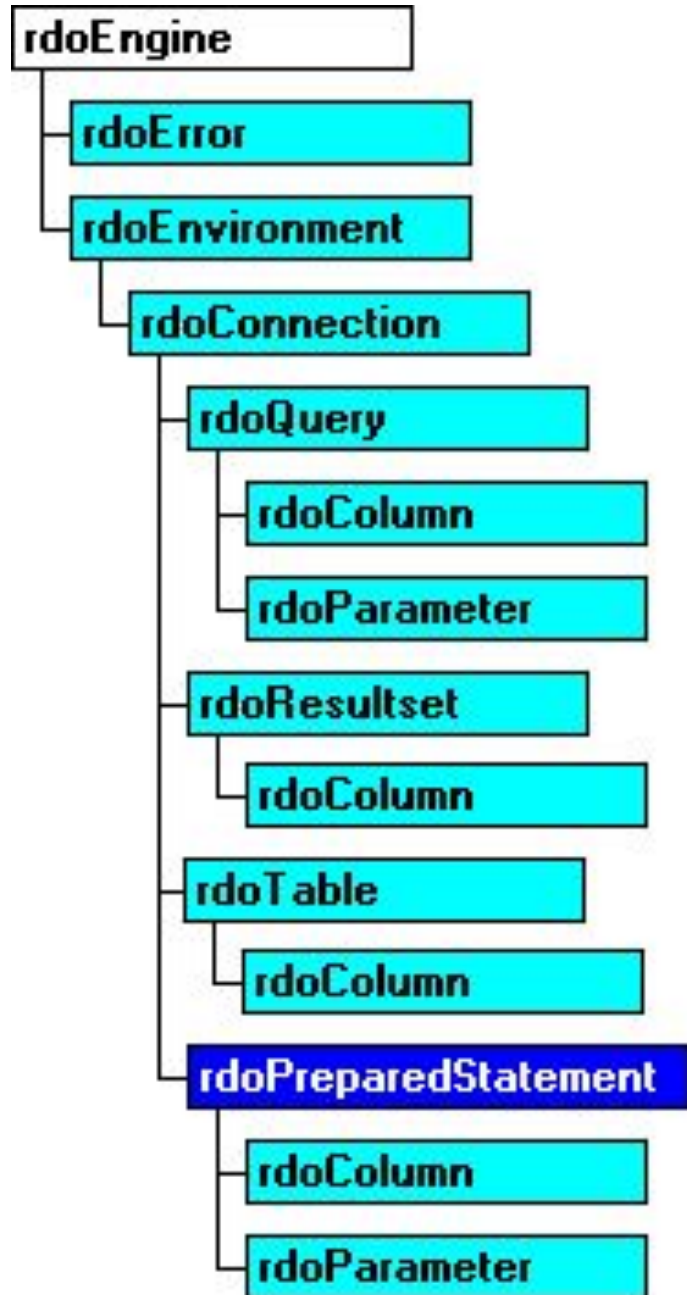
Технология RDO появилась в 1995 году одновременно с выходом продукта Visual Basic 4.0.



Remote Data Objects and their DAO/Jet Equivalents

RDO object	Equivalent DAO/Jet object
rdoEngine	DBEngine
rdoError	Error
rdoEnvironment	Workspace
rdoConnection	Database
rdoTable	TableDef
Not Implemented	Index
rdoResultset	Recordset
Not implemented	Table-type
Keyset-type	Dynaset-type
Static-type (r/w)	Snapshot-type (r/o)
Dynamic-type	(none)
Forward-only – type	Forward-only-type
(cursorless)	(none)
rdoColumn	Field
rdoQuery	QueryDef
rdoParameter	Parameter
Not Implemented	Relation
Not Implemented	Group
Not implemented	User

RDO object model



RDO object	Description
rdoEngine	The base object. Created automatically when you first access RDO in your application.
rdoError	Used to handle all ODBC errors and messages generated by RDO. Created automatically.
rdoEnvironment	Defines a logical set of connections and transaction scope for a particular user name. Contains both open and allocated (but unopened) connections, provides mechanisms for simultaneous transactions, and provides a security context for data manipulation language (DML) operations on the database. rdoEnvironments(0) created automatically.
rdoConnection	Represents an open connection to a remote data source and a specific database on that data source, or an allocated but as yet unconnected object, which can be used to subsequently establish a connection.
rdoTable	Represents the stored definition of a base table or an SQL view.
rdoResultset	Represents the rows that result from running a query.
rdoColumn	Represents a column of data with a common data type and a common set of properties.
rdoQuery	An SQL query definition that can include zero or more parameters.
rdoParameter	Represents a parameter associated with an rdoQuery object. Query parameters can be input, output, or both.

Features of Remote Data Objects

- Достижение большей производительности при работе с удаленными ODBC ресурсами
- Управление несколькими наборами результатов
- Запуск нескольких запросов одновременно
- Запуск запросов асинхронно
- Уменьшение занимаемой памяти
- Возможность лимитировать количество строк, которые будут возвращены или обработаны