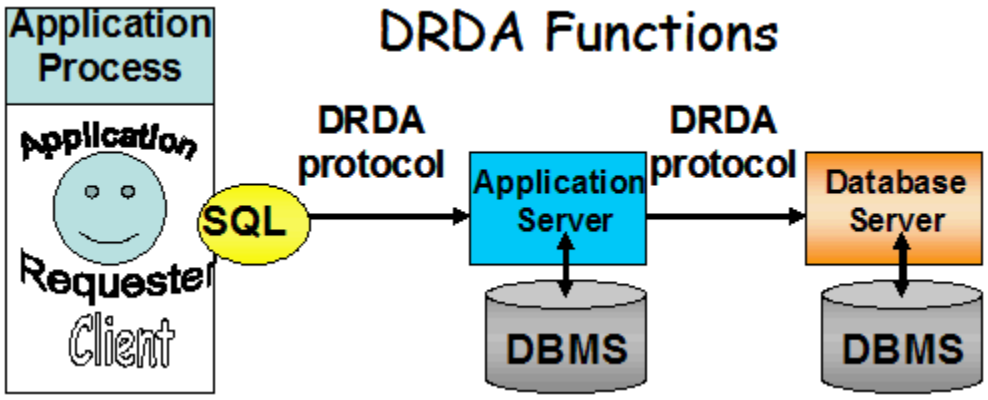
11/12/06 - #03 zTidBits (zIIP's DRDA)

Today’s industry where customers place corporate information on a plethora of remote servers, a need for a common protocol was desirable in the distributed database environment. In the early days, each DBMS vendor developed their products to enable them to communicate within their own line of DB - products. DB2 was no exception. DB2 products used their own private protocols (for example, a DB2 Private Protocol can be used to communicate between DB2 for z/OS subsystems; also, DB2 for Multiplatforms used its DB2RA proprietary protocol up until V7). Starting with DB2 UDB for Multiplatforms, DB2RA is replaced by DRDA, and although the DB2 Private Protocol on DB2 for z/OS and OS/390 is still supported, its functionality has been frozen for a long time.

DRDA is an architecture that was established by IBM in 1989 for the applications accessing relational databases in multiple locations. In 1998, The Open Group adopted DRDA as standard database interoperability protocol. There have been several other attempts to develop a common interface to database access, but DRDA has proven to be a superior architecture in its performance and abundant functions. As an example, DRDA includes two-phase commit protocol, TCP/IP support and support for Stored-Procdures.

Many DBMS vendors have adopted the DRDA architecture in their products. DRDA brings benefits, both to the end-user and database software vendors. DRDA provides a common protocol, so an application program that uses DRDA can access any DRDA-supported databases. In addition, software vendors do not have to develop their own private protocol. If their database management systems support DRDA, the products can access any other database without any special connectivity product.

The major characteristics of DRDA are:
* DRDA is a database interoperability protocol using SQL as the standardized API (both static and dynamic SQL are available).
* Remote bind support.
* Automatic data transformation.
* Unit of work support.
* Stored procedure support.
* Supports XA
* Superior performance, scalability, and availability.



What is z/OS' DDF –
DB2 for z/OS Distributed Data Facility (DDF) is a built-in component of DB2 for z/OS and provides the connectivity to and from other databases (or servers like DB2 Connect) over the network. DB2 for z/OS DDF supports two network protocols, SNA and TCP/IP, as well as two database communication protocols, namely, DB2 Private Protocol and DRDA. DDF implements a full DRDA Application Server and Application Requester.

DDF was first delivered in DB2 for MVS V2R2. At that time, the only supported protocol was the DB2 Private Protocol (over SNA). Since V2R3, both the DB2 Private Protocol and DRDA protocol are available. However, the DB2 Private Protocol functionality has not been enhanced for a long time and, although still supported, it is not encourage to be used, especially when you are just starting with DDF.

DDF is DB2's transaction manager for distributed database connections. Unlike CICS or IMS, connections via DDF are not localized to a few address spaces within z/OS. With DRDA, connections can come from literally anywhere within the bounds of the SNA or TCP/IP network that DB2 is operating in. For this reason DDF has developed very mature thread management strategies to be able to handle thousands of connections from anywhere. DDF runs as an additional address space in the DB2 subsystem. The address space name is *xxxx*DIST, where *xxxx* is the DB2 subsystem name. DDF is an efficient connection handler. It uses SRBs instead of TCBs, which reduces CPU time. MVS enclaves are used in exchanging data across address spaces. This enables proper management by Workload Manager (WLM) of the work coming into DB2 through DDF.

As a side note: SQL/DS (DB2 for VM) provides DRDA remote unit of work application server and application requester support for VM systems.
SQL/DS (DB2 for VSE) also provides DRDA remote unit of work application server support for VSE systems.