

CHAPTER 3

Accessing IMS

In a broad sense, *accessing* IMS means telling IMS to perform work for you.

- You can write application programs that tell IMS what to do.
- You can use existing application programs that tell IMS what to do.

These application programs can run in many different environments. They tell IMS what to do by calling a number of standard IMS functions through standard APIs: DL/I and JDBC.¹

In This Chapter:

- “Accessing IMS from Application Programs”
- “Accessing IMS from Other Products” on page 22

Related Reading: For information about writing IMS application programs, see Part IV, “IMS Application Development,” on page 215.

ACCESSING IMS FROM APPLICATION PROGRAMS

Application programs that directly use the DL/I interface do so by issuing DL/I calls. Application programs that use the JDBC interface do so by issuing standard SQL (structured query language) calls.

1. The IMS implementation of JDBC supports a selected subset of the full facilities of the JDBC 2.1 API.





Accessing IMS by Using DL/I Calls

DL/I is a standard interface to IMS functions that has been in place since IMS's inception. Most of the IMS application programs that have been written over the years are still providing the service for which they were designed. As business needs have evolved, these application programs have either evolved or have become a base for new application programs to meet the new business needs. In many cases, the application programs that run today's businesses are not individual programs, but are a number of layers of application programs that work together to implement the businesses' information technology (IT) infrastructure.

To illustrate how IMS business application programs have evolved, Figure 3-1 on page 19 shows a simple, hypothetical IMS application program that accesses a checking account database through DL/I. Assume that this application program was written 20 years ago, was written in COBOL, and was designed to support IBM 3270 (non-programmable) terminals. There are many such application programs that still run as originally written and provide this kind of support for the banking industry.

Figure 3-1 illustrates the following processing models:

- **Traditional processing model:** The objects on the left side of Figure 3-1 represent the traditional processing model that includes the following components:
 - **DL/I:** The interface to the IMS modules that access and manipulate the data in IMS databases.
 - **Checking account application program:** This program performs basic operations on checking account balances (such as queries and updates) and issues DL/I calls that cause IMS to actually query and update the checking account data that resides in the IMS checking account database.
 - **Message Format Service (MFS):** A service provided with IMS TM that separates the terminal device characteristics from the application program. For more information, see "Message Format Service" on page 182.
 - **z/OS Communication Server:** A communications package that is part of the z/OS operating system. In Figure 3-1, IMS is communicating with the Virtual Telecommunications Access Method (VTAM[®]) function within the z/OS Communication Server.²
 - **Communications Controller:** A device (hardware and software) that is part of the telecommunications network.
 - **3270-type terminal:** Before personal computers (PCs) were invented, these terminals provided access to the mainframe computer—a bank teller might use such a terminal. In general, these types of terminals have been replaced with PCs that run 3270-emulation application programs.

2. Although IMS uses the network facilities of VTAM, it can also control devices that use the basic telecommunications access method (BTAM) or the basic sequential access method (BSAM). VTAM is the preferred access method for IMS.



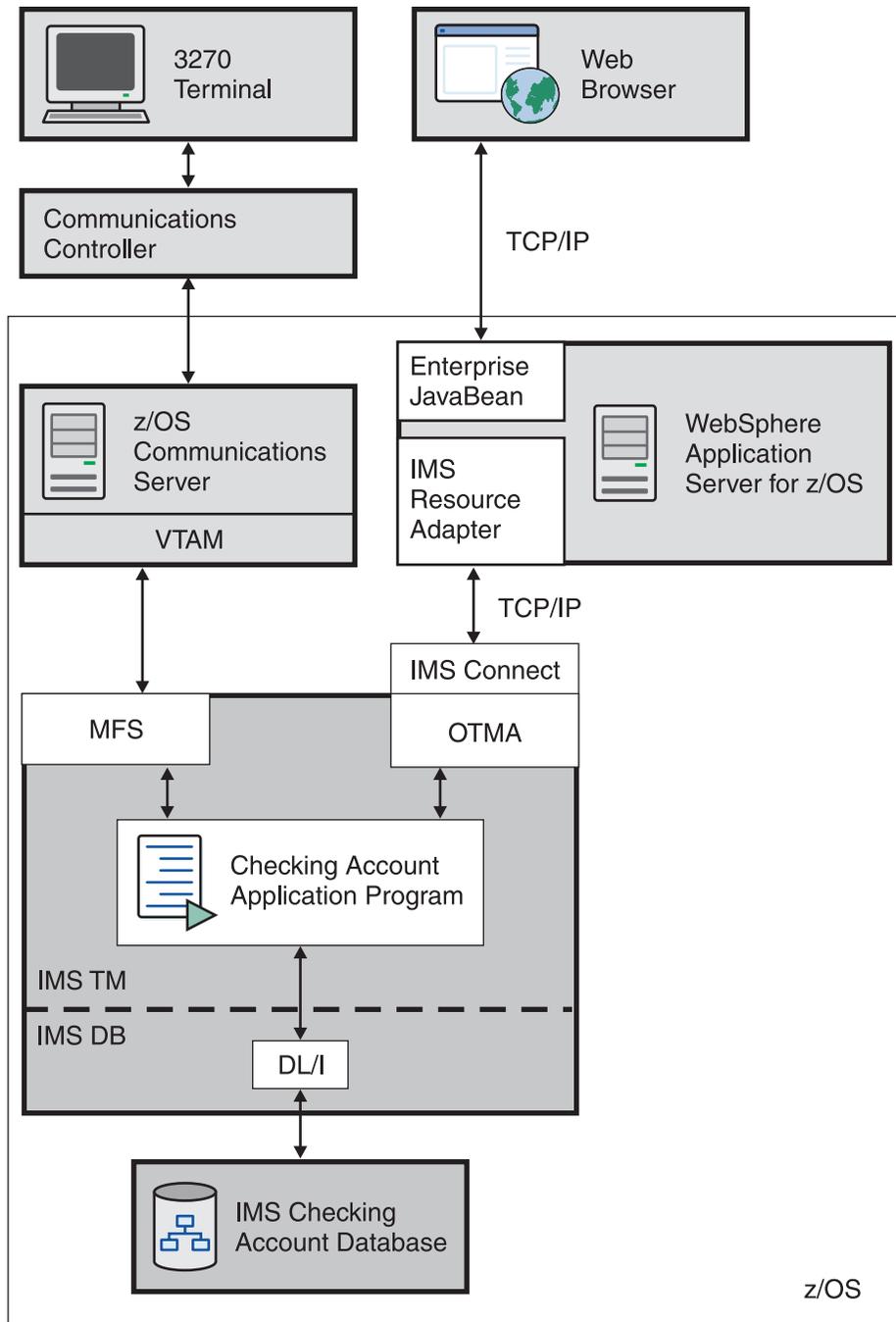
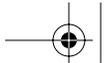


Figure 3-1 Example of a Simple Application That Accesses an IMS Database Through DL/I



- **Updated traditional processing model:** Figure 3-1 on page 19 also shows how the following layers of application programs and protocols have been added to the original application program to enable Web-based personal banking:
 - **OTMA:** Open Transaction Manager Access (OTMA) is an open interface to IMS TM. For more information, see “Open Transaction Manager Access” on page 179.
 - **IMS Connect:** IMS Connect is an optional IMS TM network component that provides high-performance communications for IMS between one or more TCP/IP clients (such as WebSphere Application Server for z/OS) and one or more IMS systems. For more information, see “IMS Connect” on page 187.
 - **WebSphere Application Server for z/OS:** A comprehensive, sophisticated, Java 2 Enterprise Edition (J2EE) and Web services technology-based application platform specifically designed to leverage the qualities of service inherent in the z/OS operating system. For more information about how IMS supports WebSphere Application Server for z/OS, see “WebSphere Application Server for z/OS Applications” on page 317.

Figure 3-1 on page 19 includes the IMS Resource Adapter, which is delivered with the IMS Connect function and IBM WebSphere Studio Application Developer Integration Edition.

- **Enterprise JavaBean:** A user-written, object-oriented, distributed, enterprise-level application.
- **Web Browser:** A client program that initiates requests to a Web server and displays the information that the server returns. The Web browser is the actual end-user interface with which bank customers can manipulate their checking accounts.

Accessing IMS Using JDBC

The IMS Java function provides IMS’s implementation of the JDBC API. Application programs that use the JDBC interface to access IMS issue SQL calls and might run on a non-z/OS platform.

The IMS Java function allows you to write Java application programs that access IMS databases and IMS message queues from many different locations:

- Within IMS
- Any environment that supports the JDBC API
- IBM WebSphere Application Server for z/OS
- WebSphere Application Server running on a non-z/OS platform
- IBM Customer Information Control System (CICS) Transaction Server for z/OS
- IBM DB2 Universal Database for z/OS stored procedures



For more information about writing Java applications for IMS, see Chapter 18, “Application Programming in Java” on page 311.

Figure 3-2 illustrates an Enterprise JavaBean (EJB) that can perform work on the IMS checking account database. Notice that in Figure 3-2, IMS does not require an application program to issue DL/I calls.

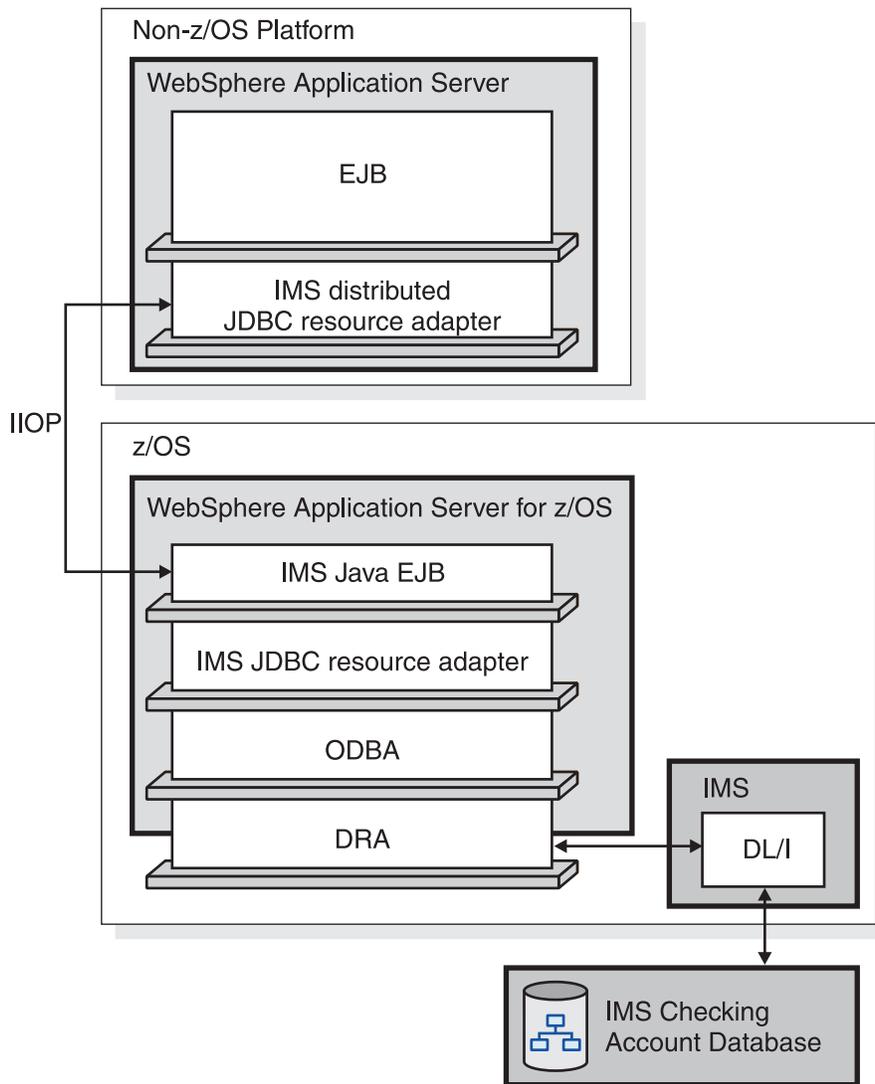


Figure 3-2 Example of an EJB That Accesses an IMS Database Through the JDBC Interface

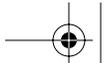


Figure 3-2 includes the following components:

- **DL/I.**
- **DRA:** The database resource adapter (DRA) is the bridge between ODBA and IMS.
- **ODBA:** Open Database Access (ODBA) is the IMS callable interface for access to IMS DB.
- **IMS JDBC resource adapter:** The IMS JDBC resource adapter that is deployed on the z/OS platform. The IMS JDBC resource adapter is delivered with the IMS Java function.
- **IMS Java EJB:** One of two IMS Java-supplied EJBs is the host-side component that facilitates communication with and passes transaction information to the IMS JDBC resource adapter. These EJBs act as listeners for remote requests.
- **WebSphere Application Server for z/OS.**
- **IIOIP:** Internet Inter-ORB Protocol (IIOIP) is the protocol that can be used between WebSphere Application Server for z/OS and WebSphere Application Server running on another platform. IIOIP allows the servers to exchange data. Data is securely transferred across the Internet using the SSL (Secure Sockets Layer) protocol.
- **IMS distributed JDBC resource adapter:** The resource adapter that is deployed on the non-z/OS platform, which contains a type-3 JDBC driver and is delivered with the IMS Java function.
- **EJB:** The EJB enterprise application that contains your business logic and is deployed on WebSphere Application Server.
- **WebSphere Application Server:** WebSphere Application Server on which the client application runs.
- **non-z/OS platform:** The operating system that hosts WebSphere Application Server.

ACCESSING IMS FROM OTHER PRODUCTS

The scope of this book is not large enough to fully explain all the products and protocols that can access IMS. Instead, some of these products and protocols are briefly mentioned in the following sections to demonstrate that you can access IMS from almost anywhere in the IT world.

Some products are shown in both Figure 3-3 on page 23 and Figure 3-4 on page 25, which indicates that these products can access both IMS DB and IMS TM.

Accessing IMS DB

Figure 3-3 shows the main interfaces to IMS DB (IMS TM, the ODBA interface, and the DRA interface), along with some of the products that use the ODBA and DRA interfaces.



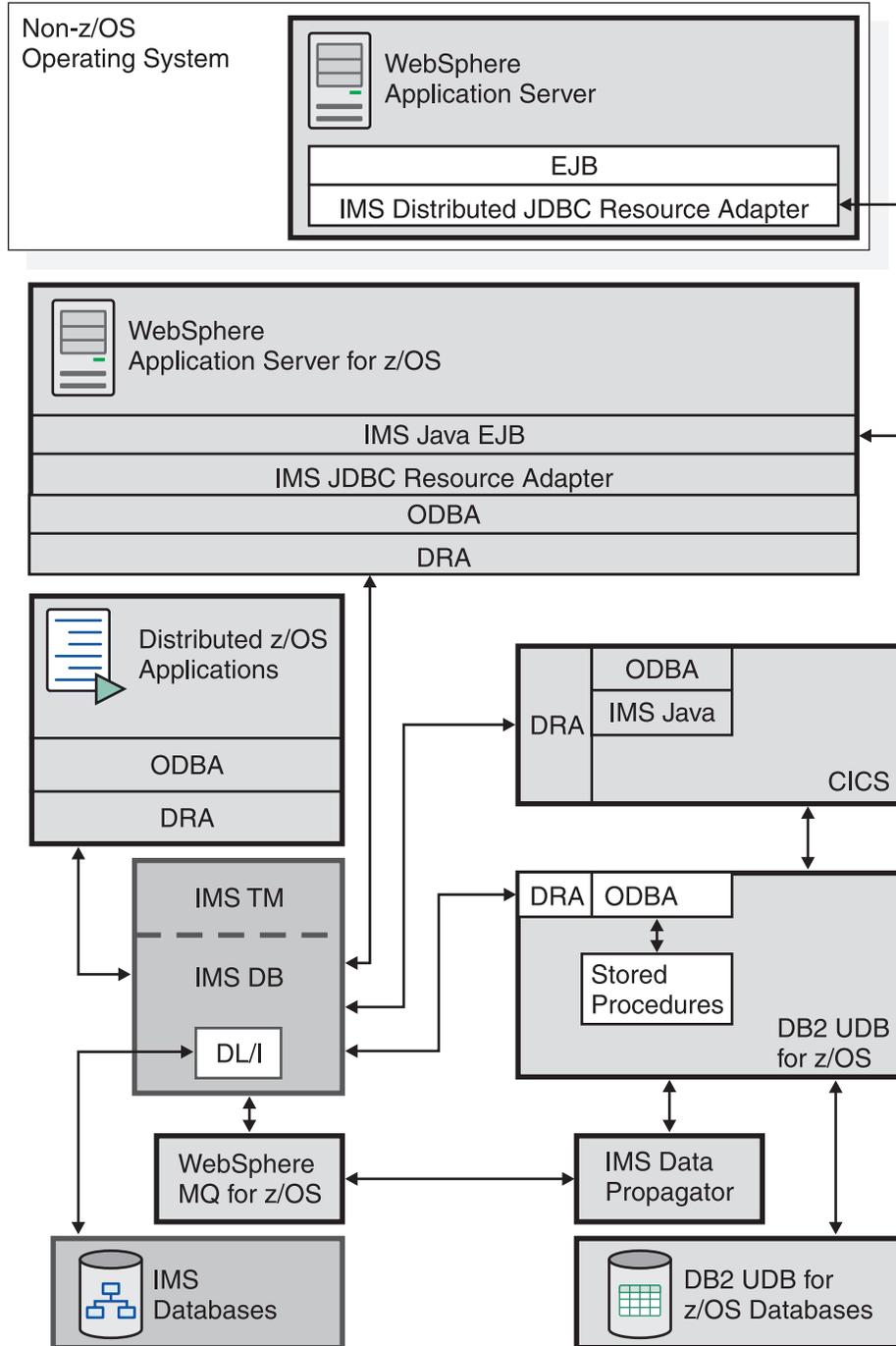
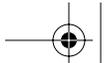


Figure 3-3 Accessing IMS DB from Other Programs



The following components and products are shown in Figure 3-3:

- **DL/I.**
- **DRA:** The DRA comes with IMS DB, but is run where it is used, not within IMS.
- **ODBA:** ODBA comes with IMS DB, but, like the DRA, is run where it is used, not within IMS.
- **Distributed z/OS applications:** Any z/OS application that follows the guidelines and rules associated with the ODBA interface can access IMS.
- **WebSphere Application Server for z/OS.**
- **CICS:** IBM CICS Transaction Server for z/OS, an IBM licensed program that provides online transaction-processing services and management for business applications. CICS can be a transaction manager for either IMS DB or DB2 UDB for z/OS. In Figure 3-3, CICS application programs that need to access IMS DB can do so in two ways:
 - Directly through the DRA by issuing CICS EXEC DL/I calls.
 - Through the DRA using the IMS Java function and issuing the supported SQL calls.
- **DB2 UDB for z/OS:** IBM's premier database manager for relational databases. Stored procedures that run in DB2 UDB for z/OS can access IMS DB through ODBA.
- **IBM WebSphere MQ for z/OS:** A z/OS subsystem that supports the transfer of messages across z/OS address spaces and to other WebSphere MQ products on both IBM and non-IBM platforms. The platforms include AIX[®], iSeries[™], OS/2[®], VSE/ESA[™], HP-UX, Solaris, Windows, Compaq NonStop Kernel, Compaq Tru64 UNIX, Compaq OpenVMS (Alpha), and Linux, using the IBM SNA LU 6.2 or TCP/IP communications protocols.
- **IBM IMS DataPropagator[™]:** Figure 3-3 also shows WebSphere MQ for z/OS working in conjunction with IMS DataPropagator and DB2 UDB for z/OS to replicate IMS data across the DB2 family of databases.

Accessing IMS TM

Figure 3-4 on page 25 shows the main interfaces to IMS TM (the OTMA interface, the IMS Connect function, VTAM, and APPC), along with some of the products and protocols that use these interfaces.

The following components and products are shown in Figure 3-4:

- **DL/I.**
- **Advanced Program-to-Program Communications (APPC):** An implementation of the SNA LU 6.2 protocol that allows interconnected systems to communicate and share the processing of programs. For more information about IMS TM and APPC, see “APPC/IMS and LU 6.2 Devices” on page 185.



Accessing IMS from Other Products

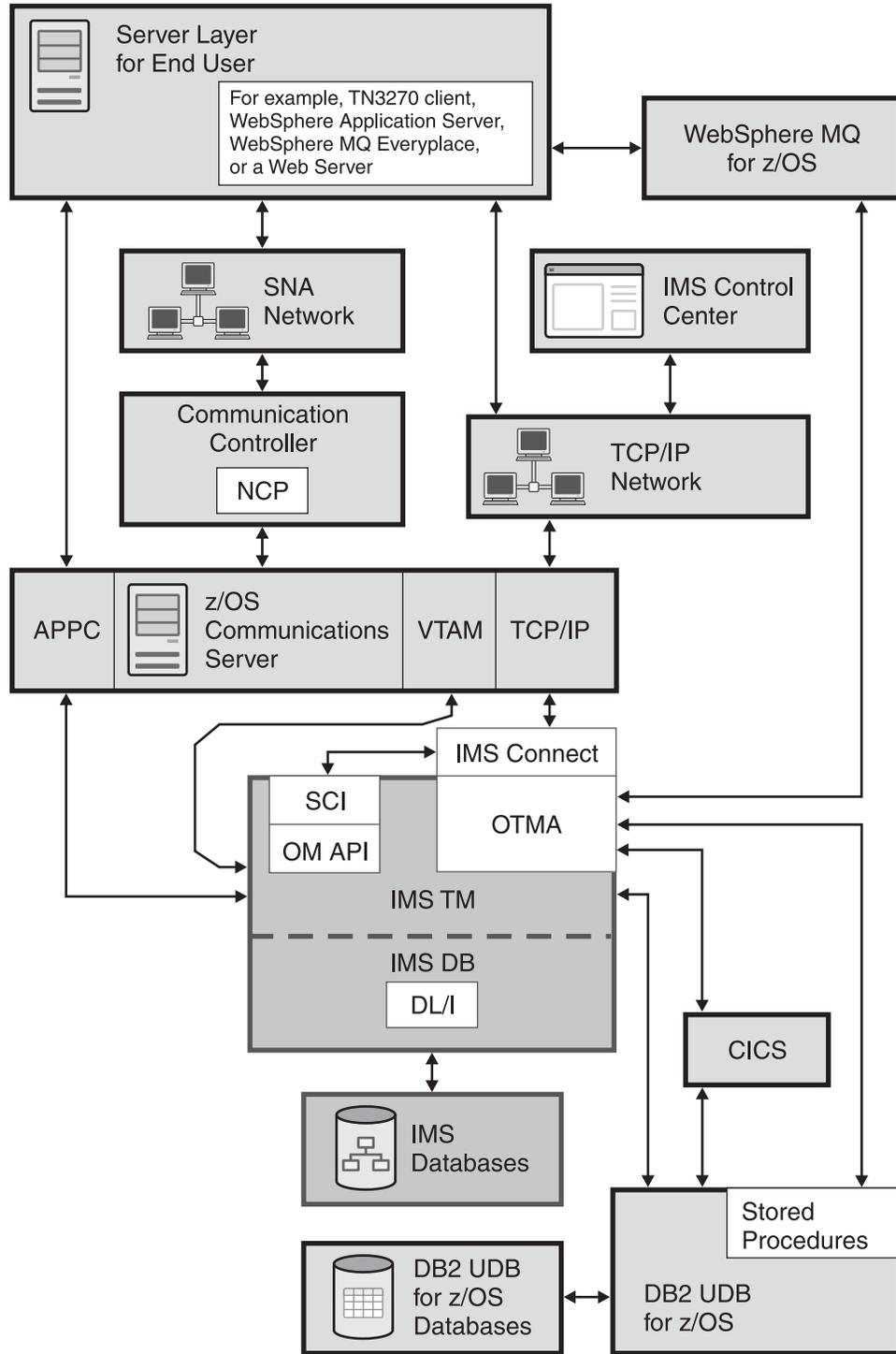


Figure 3-4 Accessing IMS TM from Other Programs



- **z/OS Communications Server:** z/OS Communications Server provides both System Network Architecture (SNA) and Transmission Control Protocol/Internet Protocol (TCP/IP) networking protocols for z/OS. The SNA protocols are provided by Virtual Telecommunications Access Method (VTAM) and include Subarea, Advanced Peer-to-Peer Networking, and High Performance Routing protocols. For more information on z/OS Communications Server SNA protocols or TCP/IP networking protocols for z/OS, see *z/OS V1R4.0 z/OS Communications Server SNA Network Implementation Guide*.
- **Communication Controller and Network Control Program (NCP):** A combination of hardware and software that provides support for single-domain, multiple-domain, and interconnected network capability. The NCP is an IBM licensed program.
- **SNA Network:** Existing (traditional) network that implements the SNA protocols³ to connect terminals and other devices.
- **Server Layer for End User:** One or more middleware software products that interact with the end user, for example, TN3270 client (TN3270 Telnet server acts as an interface between IP and SNA networks), WebSphere Application Server, or WebSphere MQ Everyplace®.
- **IMS Control Center:** A workstation graphical interface from which you can manage your IMS systems. The IMS Control Center connects to IMS through a TCP/IP network, IMS Connect, and the Structured Call Interface (SCI). For more information about SCI, see Chapter 28, “IMSplexes” on page 495. For more information about the IMS Control Center, see “Operating an IMSplex” on page 499. For more information about IMS Connect, see “IMS Connect” on page 187.
- **OTMA:** An open interface to IMS TM through which a z/OS or TCP/IP application program can send a transaction or command to IMS without using SNA or VTAM. For more information about OTMA, see “Open Transaction Manager Access” on page 179.
- **IBM WebSphere MQ for z/OS:** A z/OS subsystem that supports the transfer of messages across z/OS address spaces and to other WebSphere MQ products on both IBM and non-IBM platforms. The platforms include AIX, iSeries, OS/2, VSE/ESA, HP-UX, Solaris, Windows, Compaq NonStop Kernel, Compaq Tru64 UNIX, Compaq OpenVMS (Alpha), and Linux, using the SNA LU 6.2 or TCP/IP communications protocols.
- **CICS:** In Figure 3-4 on page 25, CICS application programs that need to access IMS TM can do so by using the OTMA callable interface.
- **DB2 UDB for z/OS:** IBM’s premier database manager for relational databases. IMS TM can provide transaction processing services for DB2 UDB for z/OS.

3. Network access to IMS Transaction Manager was originally provided by IBM’s communications systems, which evolved into the System Network Architecture (SNA), as implemented in VTAM.

