

Distributed Transaction Processing Monitors

What is a TPM?

- middleware that
 - monitors the transactions of applications in a heterogeneous environment
 - ensure that transactions are completed successfully
 - ensure the system is restored to a consistent state if the transaction is unsuccessful
- 출현배경
 - 종래의 transaction 처리 기술에 있어서 주 관심대상은 DBMS
 - but, DBMS가 transaction system의 중요부분이기는 하지만 DBMS의 성능향상만으로 효율적인 transaction processing system 구현에 부족한 점이 많다.
 - Why? OS나 통신 기법, UI등이 전체 시스템의 성능과 용이성에 많은 영향을 줌
 - So, 이러한 부분을 보완한 OLTP Monitor 필요성 대두

DTPM (1)

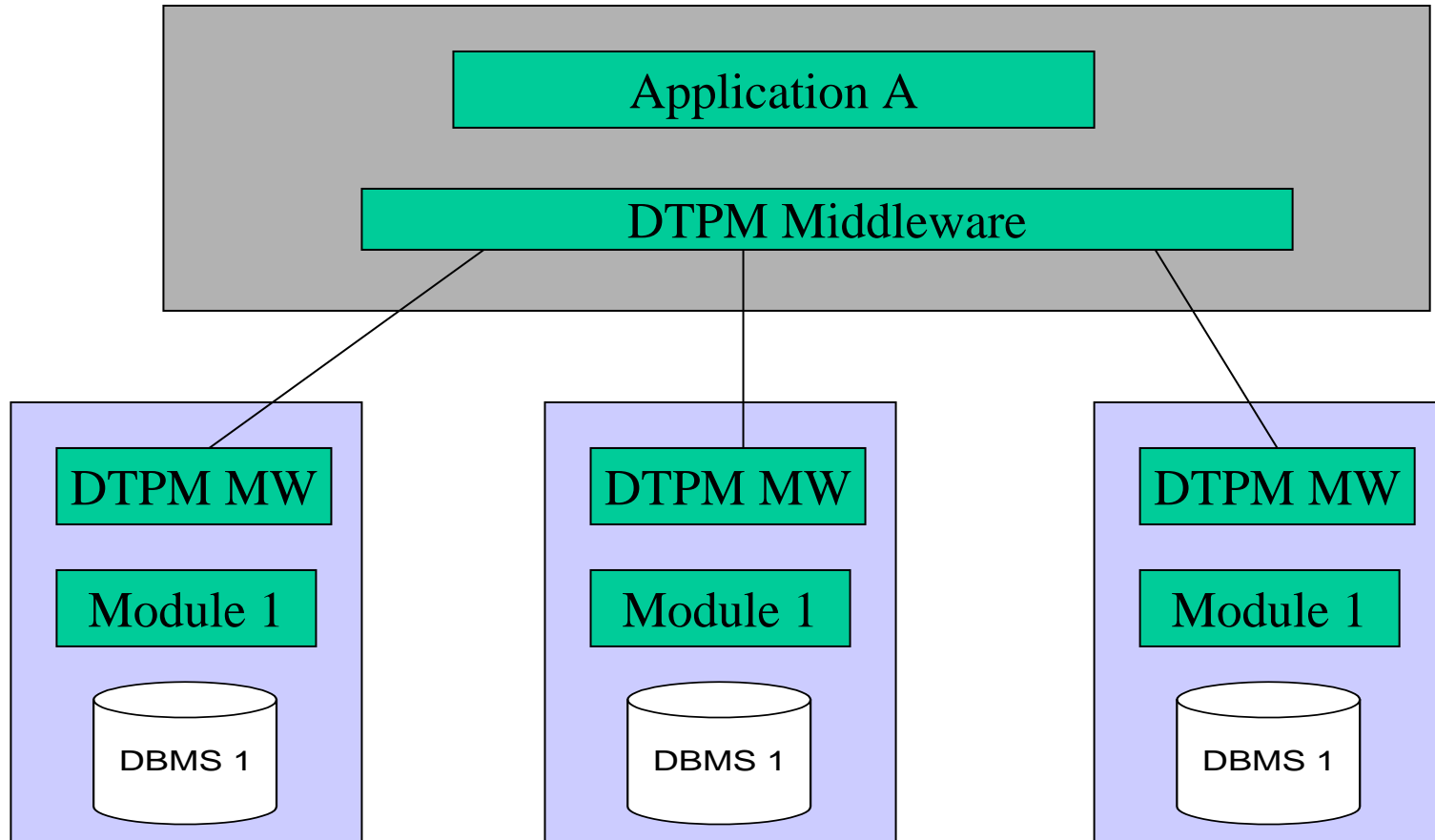
-
- DTPM
 - provide communication and management service
 - used in OLTP systems
 - useful in distributed heterogeneous environment
 - co-ordinates multiple resources and multiple resources managers of different types (DBMS and File systems)
 - What is transaction?
 - Commit and Roll-back
 - committed
 - aborted

DTPM (2)

-
- Transaction-processing systems
 - ACID
 - all or nothing
 - isolation
 - data integrity and consistency for the event of failure
 - Why are DTPMs needed?
 - The layers supported by a DTPM
 - Fig next slide
 - DTMPs do not translation

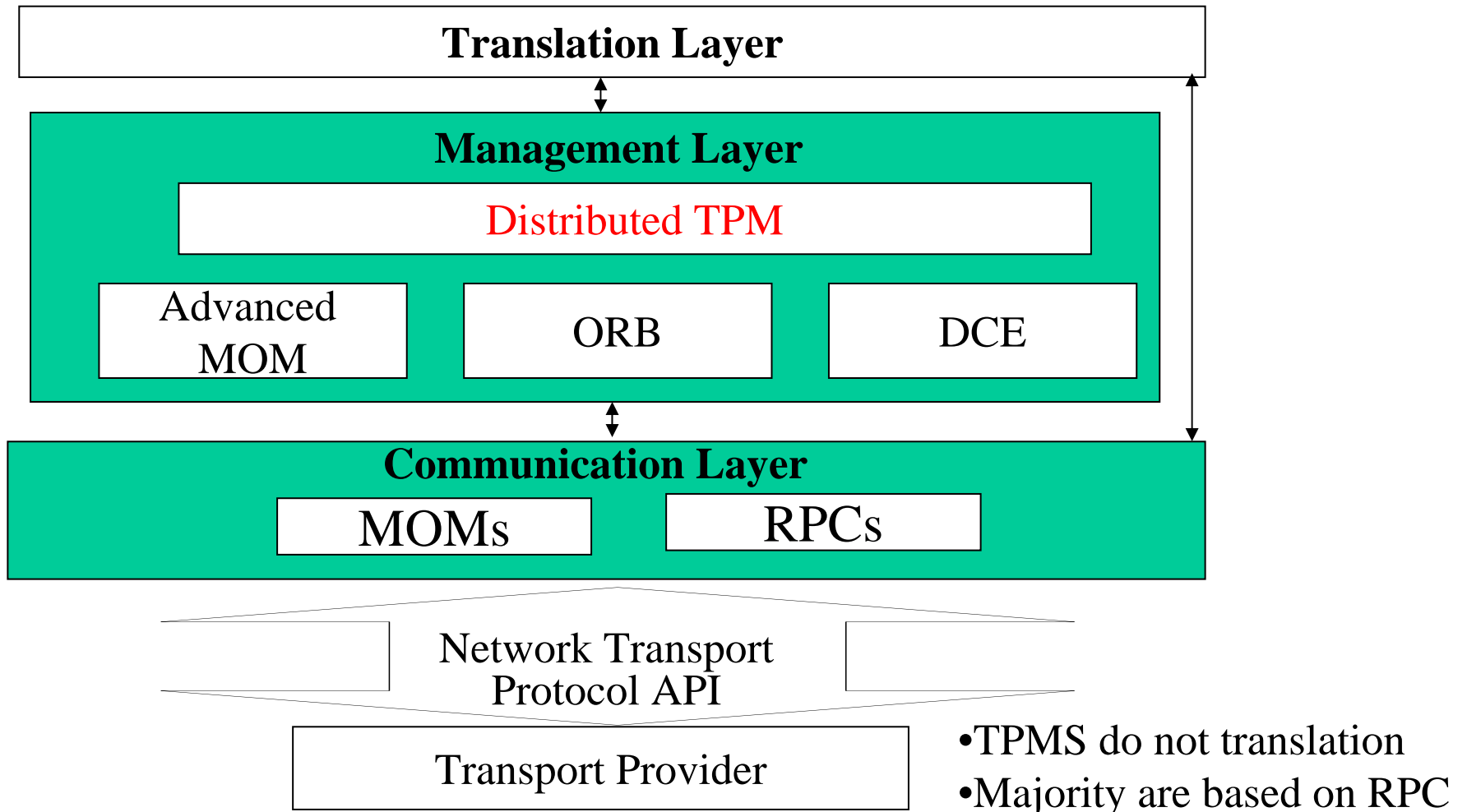
How do DTPMs work?

- Two important services
 - two-phase commit
 - recovery and roll-back



Model of middleware and TPM

DTP



TPM의 특성

- Process간의 효율적인 통신
- Server에 대한 부하 조정
- Server에 대한 위치 투명성 제공
- 명시된 시간 제약(dead-line)을 고려하는 transaction processing priority 관리

TPM의 구성요소

-
- transaction 요청 메시지 관리
 - 요청 메시지를 적절한 응용 서버로 mapping시켜주는 name server
 - transaction 서버간의 load balancing
 - 서버의 위치 투명성 제공 module
 - 응용(transaction)서버 관리
 - 원격 서비스 제공 module
 - 시스템 정보 관리 및 우선 순위 큐 관리
 - 제반 통신을 관리하는 module

Why build on TPM monitor?

DTP

- **Interoperability**
 - TPM monitor are database-independent
 - TPM monitor operate well with a variety of DBMS
- **Manageability**
 - management, security, UI, centralized monitoring, dynamic reconfiguration, C/S authentication, data encryption
 - do not need that developers perform any special tasks
- **Scalability**
 - support large numbers of simultaneous users across geographically dispersed networks

Transaction Management

- ACID property
 - atomicity: all changes are made completely (committed) or not at all (rolled-back)
 - consistency : the effects of transaction preserve invariant properties
 - Isolation : intermediate data values are not visible to other transactions
 - durability : the effects of a completed transaction are persistent
- two-phase commit protocol
 - If any of the calls fail, the modules are aborted and the results of the modules which succeeded rolled back.
 - If all the modules succeed the results are committed.

OLTP .vs. DBMS

DTP

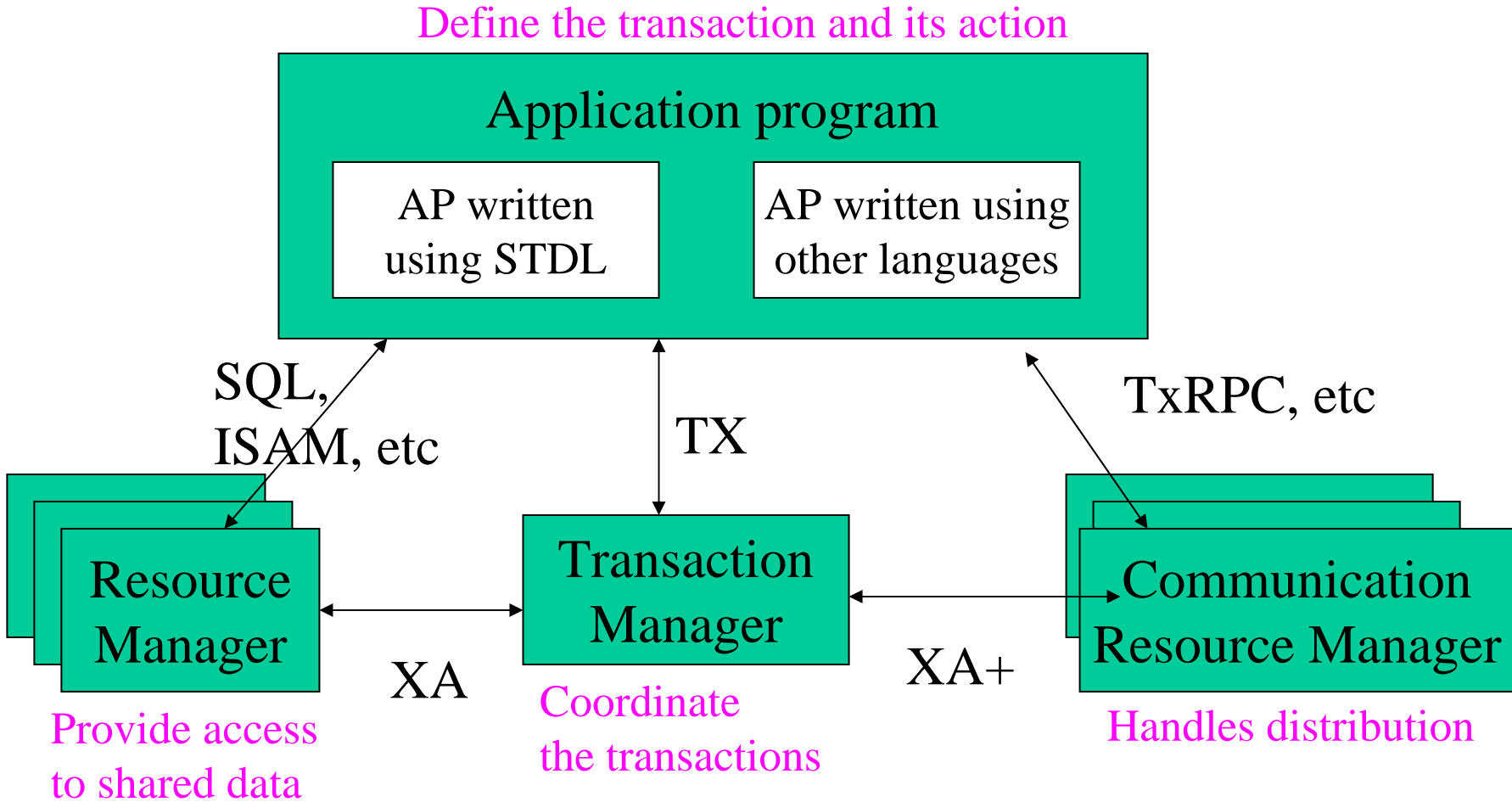
	OLTP	DBMS
security	security tightly integrated into an application	Relies on an OS or an add-on package
scalability	Fast as more users access	slow as more users access
Integration of independently developed application and new types of data	good	bad

Standards

DTP

-
- CICS
 - de facto standard in the IBM world
 - X/Open DTP Model
 - the most popular framework outside the IBM world
 - framework to build heterogeneous, distributed transaction processing systems
 - X/Open's DTP API has about 350 calls and is being used as the basis of a number of products
 - STDL
 - API used by Digital in its ACMSxp product
 - With STDL, application developers describe application

X/Open DTP Model



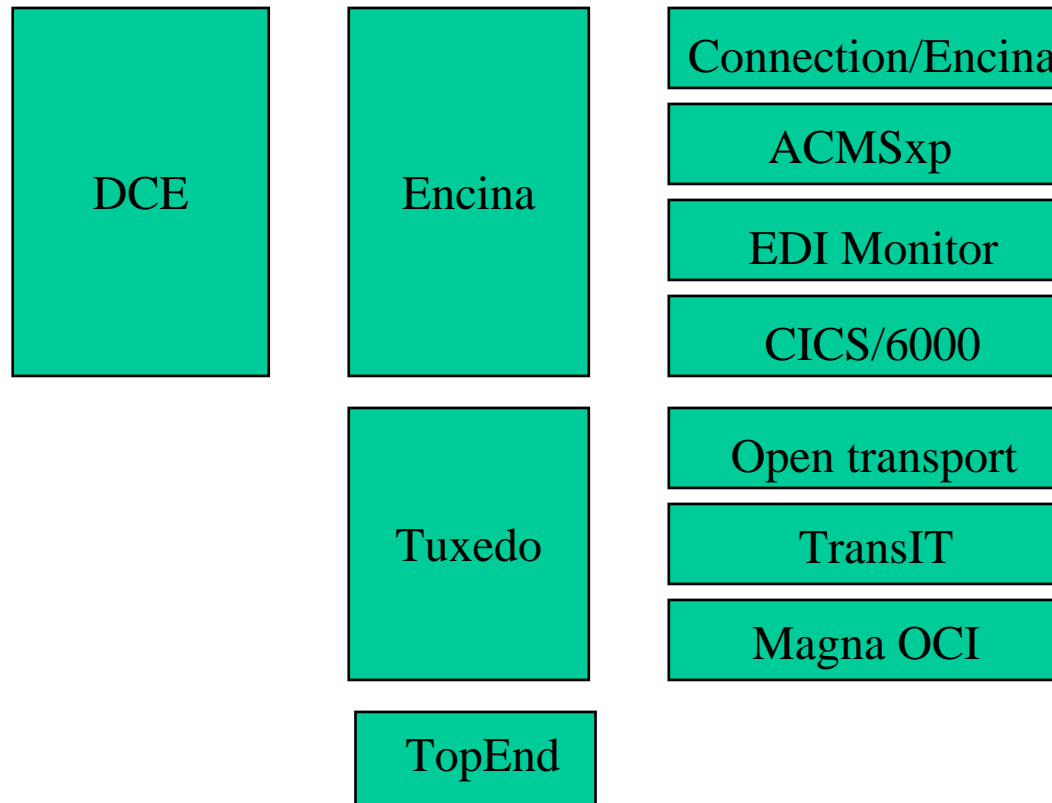
Product

DTP

- BEA's Tuxedo
- Transarc's Encina
- BEA's TopEnd (1998,7 NCR로부터 인수)
- 메인프레임 업체 중심으로 메이커 각사가 독자적으로 개발한 제품
 - CICS/6000
 - TPS
 - BeTRAN

Product (2)

- How TPMS products are based on other products

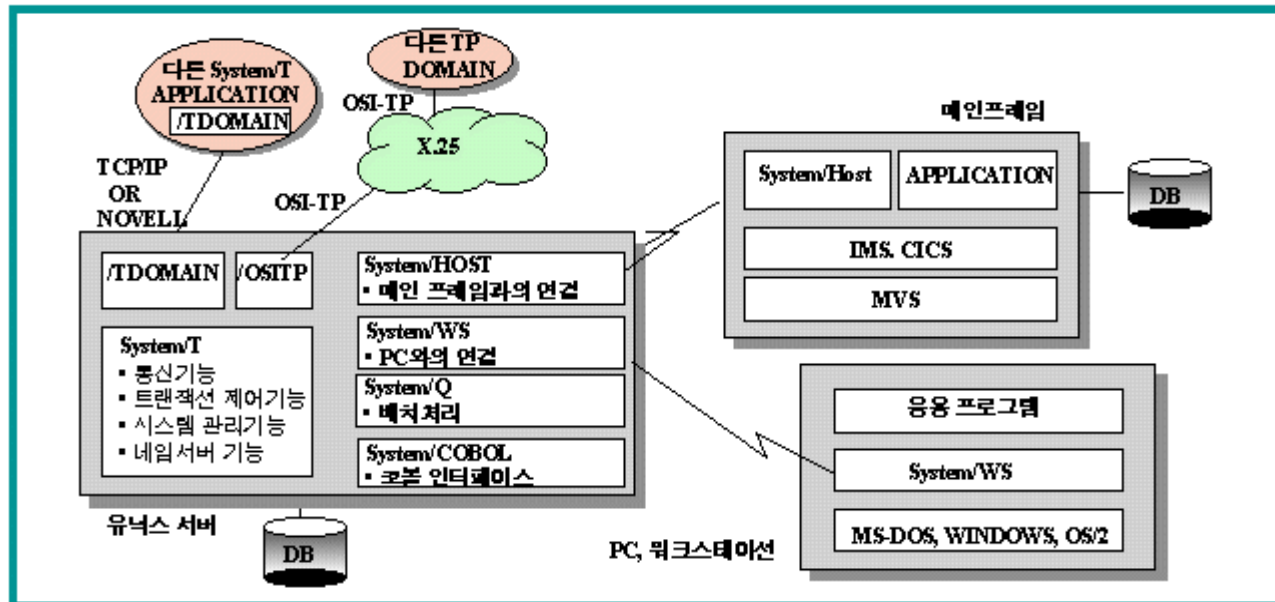


BEA Tuxedo

- 전 세계적으로 가장 많이 배포
- 특징
 - :X/Open API인 ATMI(Application-to-Transaction Manager Interface)와, X/Open의 TxRPC 표준을 따른 TxRPC로 구성
 - 클라이언트/서버 쉽게 구축가능
 - 35개 이상의 보편적인 4GL, 3GL 및 CASE 중 하나를 선택할 수 있다.
 - 일곱 가지의 상이한 통신 모드 지원
 - Asynchronous, Synchronous, Conversational, Forwarding, Broadcast, Queuing, Events-based Publish and Subscribe
 - 의사결정 지원 어플리케이션을 위한 대화형 서버(conversational servers) 지원.
 - 어플리케이션 인코딩/디코딩 및 메모리 복사를 최소화하는 typed buffer의 사용으로 어플리케이션의 성능 향상.
 - 최적화된 어플리케이션 처리를 위한 forward service request.
 - 서비스 요청의 우선 순위 설정.
 - dynamic data-dependent routing

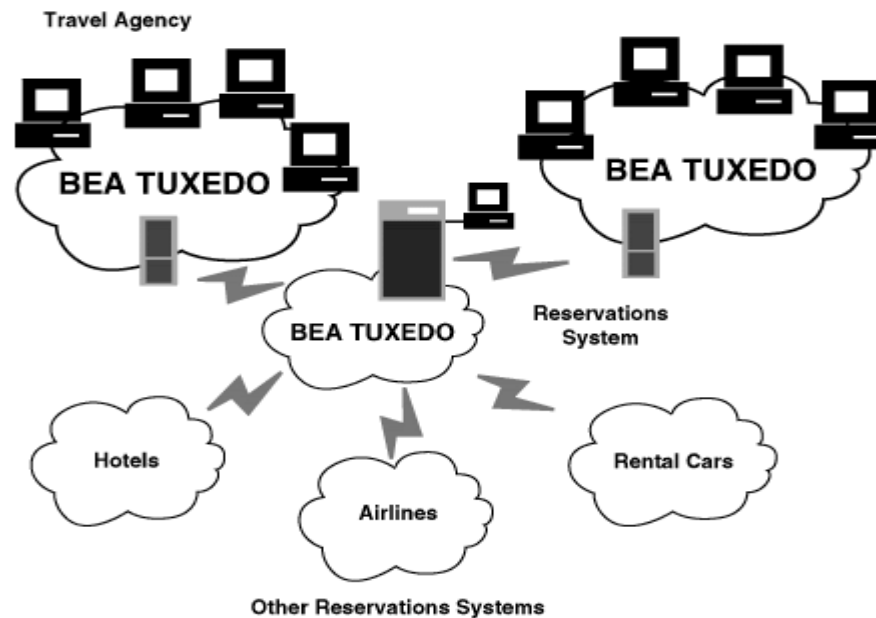
Tuxedo 제품구조

- System/T : 트랜잭션 관리/제어
- System/WS : PC나 워크스테이션과 유닉스 서버와의 연결성 제공
- System/Host : 메인프레임의 CICS나 IMS 어플리케이션과 연결성 제공
- System/Q : batch job 처리기능 제공
- System/COBOL : COBOL interface 제공
- System/Tdomain : 서로 다른 TP 어플리케이션군간의 인터페이스
- System/DCE : DCE환경에서 TUXEDO 운용



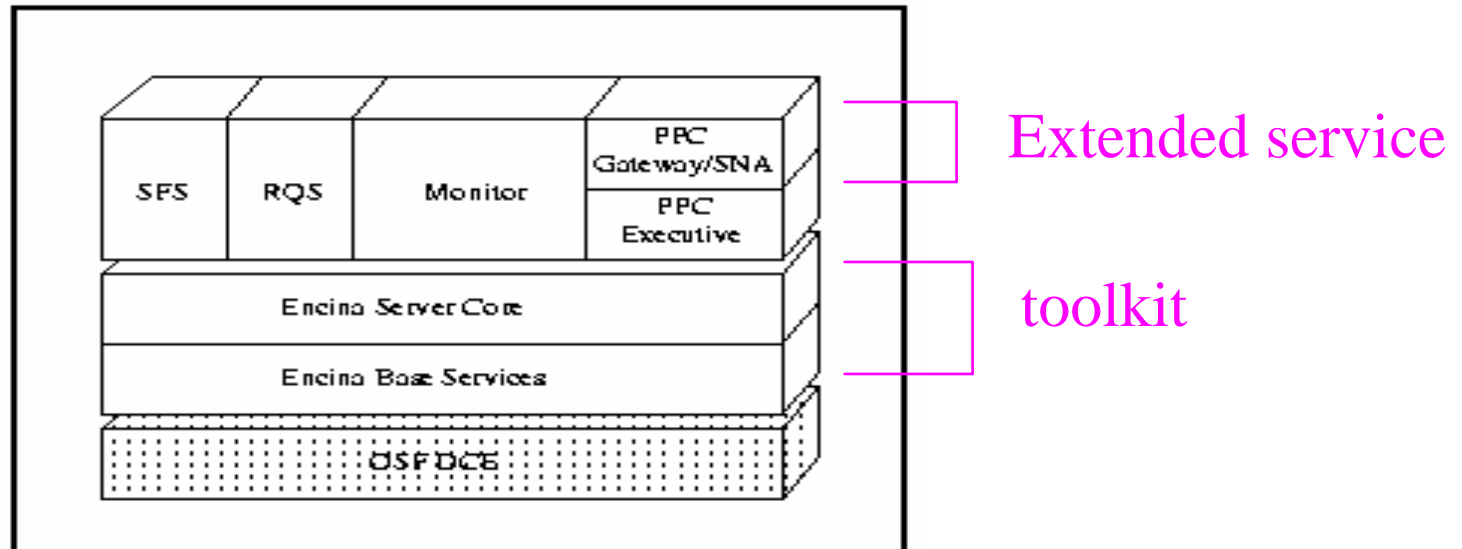
Example -Reservation system

- 여행대리점
 - 항공사,호텔,렌트카 회사 등과 같은 다양한 예약 시스템에 접근 필요
 - => 문제점) 각 시스템들이 다른 협약, 인터페이스 사용
 - 공통의 단순한 사용자 인터페이스를 가짐으로써 여행 대리인에게 교육 최소화
 - 각 회사별로 가진 서로 다른 데이터베이스 통합 필요
- 여행대리점 시스템과 중앙 예약 시스템과의 연결



**Solution
: Tuxedo!!**

Transarc Encina -Architecture



- Encina toolkit :
base service, server core
- Encina extended service :
monitor,SFS,RQS,PPC Gateway/SNA,PPC Executive

Transarc Encina -Encina Toolkit

-
- encina base services
 - provide services that permit a node to initiate, participate in, and commit distributed transactions
 - include "Transactional-C", a high-level API that provides for transaction demarcation and concurrency management
 - nested transactions : a feature that provides failure containment and simplifies the application development task

 - encina server core
 - provide facilities for managing recoverable data
 - include a locking library, X/Open XA interface to permit the use of XA-compliant resource managers(such as Oracle, Informix,..)

Transarc Encina

- Encina extended service

DTP

- Encina Monitor
 - full-featured transaction processing monitor which provides a powerful, reliable environment for the development, execution, and administration of distributed transaction processing applications
 - allows application servers to be replicated to increase availability and performance
 - support automatic load balancing, restart of failed application server
 - automatic authorization checking for security
- Encina Structured File Server(SFS)
 - record oriented file system that provides full transactional integrity, high performance, and log-based recovery for fast restart
 - provides both X/Open and ISAM compliant and VSAM-like interfaces
- Encina Recoverable Queuing Service(RQS)
 - enables the transactional enqueueing and dequeuing of data
 - provide multiple levels of priority
- Encina PPC Executive
 - support transactional peer-to-peer communications
 - enables cooperating Encina systems to participate in transactions
- Encina PPC Gateway/SNA
 - provides transactional interoperability over an SNA protocol

Programming the Encina Monitor

- Interface specification

DTP

- Encina is based on a RPC, so server(module) looks like RPC specification
- specify a syntax for interface files called IDL(Interface Definition Language)
 - => looks like a ".h" file in C

```
void debit([in]long int account,  
          [in] long int amount,  
          [out] long int * newBalanceP);  
void credit([in]long int account,  
           [in] long int amount,  
           [out] long int * newBalanceP);  
long int balance([in] long account);
```

IDL

```
[transactional] void debit([in]long int account,  
                           [in] long int amount,  
                           [out] long int * newBalanceP);  
[transactional] void credit([in]long int account,  
                            [in] long int amount,  
                            [out] long int * newBalanceP);  
[transactional] long int balance([in] long account);
```

TIDL(Transactional IDL)

Programming the Encina Monitor

- Client program

DTP

- initialization call

```
mon_InitClient("mss","test-cell");
```

"mss" : user of the client program

"test-cell" : administrative entity that is controlling the resource used by the client. Encina environment is based on "cells".

in internal, following steps are executed

- 1) establish the network protocols needed for communication
- 2) check with the directory service to locate the desired servers
- 3) create binding handles for the appropriate servers for each protocol
- 4) specify desired security characteristics for each handle
- 5) specify desired performance characteristics for each handle

Programming the Encina Monitor

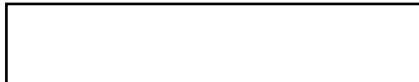
- Client program(2)

- transactions execute

```
transaction{ // is defined using Transactional-C(extension to C provided as part of Encina)
    credit(creditAccount,cmount,&newCreditBalance);
    debit(debitAccount,amount,&newDebitBalance);
} /* may or may not be on the same machine or not,
    But, programmer need not be concerned with these issues */
OnCommit { // executed if the transaction complete successfully
    printf("Transfer successful. New balance in %d: %d\n",
        creditAccount, newCreditBalance);
}
OnAbort { // executed if the transaction aborts
    printf("Transaction aborted: %s\n",abortReason());
}
```

- resource release

```
mon_ExitClient(0);
```



Programming the Encina Monitor

- Server program

- initialization

```
server_Init(arc,argv)
```

- service procedure

```
void debit(account,amount,newBalanceP)
```

```
    long int account;  
    long int amount;  
    long int *newBalanceP;  
    {  
        /* Implementation specific code */  
    }
```

- termination

```
void server_Terminate() { // to release resource,such as database  
    /* Nothing specific needs to be done */  
} // no arguments and returns no value
```



BEA Top End

DTP

- 필수 제품
 - Base Service
 - global administration
 - ISD(Interactive System Definition)
- add-on
 - Security services
 - RTQ(Recoverable Transaction Queing)
 - Remote Client Services
 - IBM LU6.2 Connectivity
 - Remote Server Services
 - Cryptographic Services
 - Services Interface Repository

BEA Top End(2)

- **Base Service**
 - BEA TOP END 노드 Manager 포함
 - 다른 BEA TOP END 컴포넌트에게 컴포넌트 개시 및 종료
 - 노드간 보안
 - 신뢰할 수 있는 메시지 전달
 - 분산 트랜잭션 프로세싱
 - 고도의 가용성
 - 애플리케이션 서비스
- **Global Administration**
 - BEA TOP END GUI 기반 관리 콘솔
 - 노드Manager와 모든 분산 애플리케이션을 포함해
모든 BEA TOP END 컴포넌트를 위한 아이콘, 풀다운메뉴 제공
=>단일 시스템 이미지 제공
- **Interactive System Definition (ISD)**
 - BEA TOP END 시스템의 정의를 보유하고 있는 SQL 기반 저장소 포함

제품별 장점

-
- Tuxedo
 - easy to grasp and prototype
 - OSI/TP domain
 - Encina
 - distributed componentry : LOG,VOL,LOCK,REC,TRAN,SFS
 - nested transactions
 - Top End
 - feature rich : sync,async,shared conversational,life cycle tools
 - good administration : tools

Product Comparison

DTP

Product Feature	Encina	TopEnd	Tuxedo
asynchronous	Possible	Yes	Yes
Nested transaction	Yes	No	No
threads	Yes	No	No
Admin tool	some	good	few
platforms	~30	~7	~5

Limitation & Future trends

- **Limitation**
 - no oo support
 - no DB support for nested transactions
 - insufficiency of operational support tool
 - scalability : at least not at the high end
- **Future trends**
 - ORB/TPM integration
 - TPM enabled tools

Reference

-
- “Middleware: Achieving Open System for the Enterprise”, Paul Korzeniowski, Computer technology research corp
 - “Middleware: the key to distributed computing”, Rosemary Rock-Evans edited by Eric woods, ovum reports, 1995
 - “Product overview”, “an introduction to programming to the Encina Monitor”,
<http://www.transarc.com/>, white paper
 - “inter-application transaction processing with BEA domains”
<http://www.beasys.com/>, white paper
 - <http://www.dakas.com/channels/index.htm>
 - <http://www.dstc.edu.au/events/TP/intro.html>
 - “TP Monitor”, infobridge 96.11,
http://www.posdata.co.kr/pos_kor/info13/infor13.html#A6
 - “Do I need TP Monitor and a Database?”,
<http://www.software.ibm.com/ts/cics/library/whitepapers/dbtmgrs/#tier3>, whitepaper
 - <http://www.kcc.co.kr/Plaza/Product/oltp.html>
 - <http://lgcom.lge.co.kr/NEWS/news/0401.html>
 - “Transaction Processing today”, DBMS and Internet Systems magazine 1998.1,
<http://www.dbmsmag.com/9801d13.html>