

**JMP101:
Basics Of The Lotus
Notes/Domino Architecture**

**George Langlais
Project Leader, Domino Server Team**

Agenda

- Better programming through a better understanding of Notes/Domino architecture
 - should be useful for people other than programmers, too
- Will give a high-level overview of ...
 - 3-level software architecture common to clients and servers
 - Basic features and elements of a Notes database
 - Notes Object Services (NOS)
 - Client and server programs
 - security model
 - directories, domains, and Domino administration
 - messaging
 - programmability

What is Groupware?

- Software which enhances the work performance of a group
 - ... by enabling the group to define, organize, and share information in relevant and timely ways
- Based on shared databases designed specifically for groupware
 - dynamic access to shared databases
 - distributed databases with replication
- Common applications
 - mail / threaded discussions / workflow / training

Considerations When Designing Practical Groupware

- Groups
 - hierarchical organizations having multiple business units
 - distributed knowledge and data with flexible content
 - mobile workforce
- Technology
 - ever-evolving computer, network, OS, and language choices
 - industry standards
- Obvious groupware product requirements
 - security
 - configurability and programmability

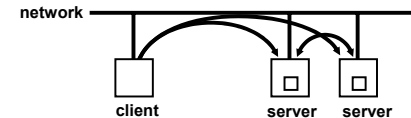
Notes Databases

Were Designed To Support Groupware Needs

- Shared online databases + offline use of replicas
- Products which run on a variety of platforms
 - portable product code
 - portable databases
- Flexible "document-oriented" content
 - create / view / modify documents via different forms
 - rich set of datatypes (simple text, rich text, links, etc.)
 - note response hierarchies
- Security via encryption and Access Control Lists
- Data integrity via transaction logging, cluster replication, ...
- Programmability
 - create your own applications
 - modify existing applications
 - use LotusScript, Java, JavaScript, Formula Language, or C

The Classic Client-Server Model ...

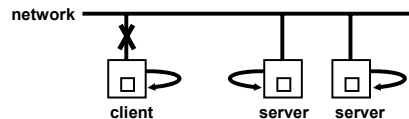
- ... is one of two models for accessing databases supported by Notes/Domino
- Both Notes clients and Domino servers can use a network to access remote databases residing on Domino servers
 - A database is called "remote" if it resides on a computer other than the one running the program that accesses it.



- Supported network protocols:
 - TCP/IP, NETBIOS, SPX/IPX, XPC (dialup), VINES

Clients and Servers Can Also Run Disconnected And Access "Local" Databases

- A database is called "local" if it resides on the same computer as the program that accesses it.

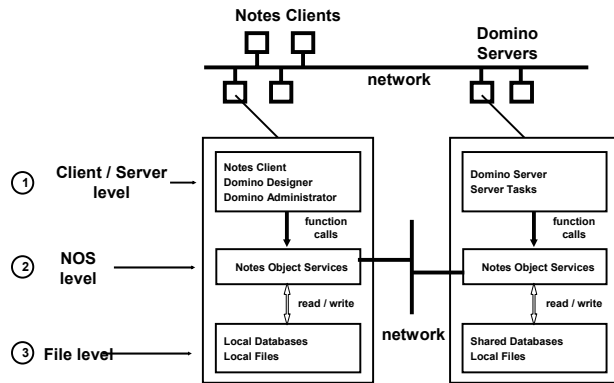


- To do this ...
 - Each client contains the same lightweight, but industrial-strength database software that runs on a server.
 - Each database contains a copy of its design and administrative information so it can be used off-line.
 - Databases can be replicated (resynchronized).

Replication Makes Distributed Data/Control Possible

- A process of distributing and resynchronizing databases
- Solves many groupware issues
 - off-line access to data by the mobile or telecommuting user
 - collaboration of groups that do not need constant connection
 - control of administration and other distributed processes
- Works because conflicts tend to be rare
 - often only the original author edits an existing document
 - others tend to add "response" notes

Clients and Servers Both Use the Same 3-level Software Hierarchy



The NOTES.INI File Is Used To Specify Configuration And Operating Parameters

```

***

MailServer=Gump
MailFile=mailjdoe.nsf
Domain=PPS
LAN0=NETBIOS,0,15,2000,,12288,
COM1=XPC,1,15,2000
COM2=XPC,2,15,2000
COM3=XPC,3,15,2000
COM4=XPC,4,15,2000
COM5=XPC,5,15,2000
DisabledPorts=COM1,COM2,COM3,COM4,COM5
DESKWINDOWSIZE=-4 3 356 491

***

```

Contents Of A Notes Database

•Database header

- ODS version number
- database ID (DBID)
- database class
- database instance(DBIID)
- information buffer
- replication settings and ID

•Internal structures to keep track of the database's content

- not directly accessible by any APIs
- specific structures vary from release to release and are identified by the ODS version number in the header

•Replication history (if applicable)

•Notes (if applicable)

- design elements
- administrative notes
- documents

Types Of Notes In A Database

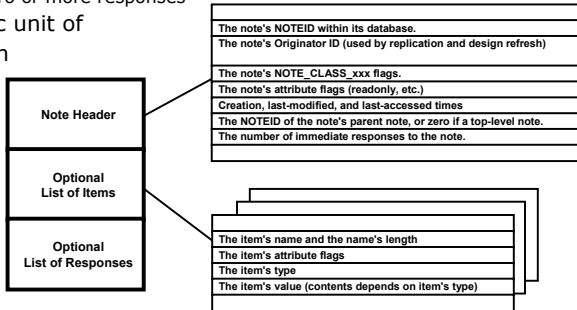
•Categories, classes, and types

Category	NOTE_CLASS_...	Note types
Data	DOCUMENT	Document
Administration	ACL*	Access Control List
	REPLFORMULA	Replication formula
Design Elements	FIELD	Shared field
	FILTER	Agent, Outline, Database script, Script library, etc.
	FORM	Form, Frameset, Page, Subform, etc.
	VIEW	Folder, Navigator, View
	DESIGN*	Design collection (structured like a view)
	ICON*	Icon
	INFO*	Help about information
	HELP	Designer help information
	HELP_INDEX*	Product help index

Note: an asterisk means that there can be at most only one of the indicated type of note in the database

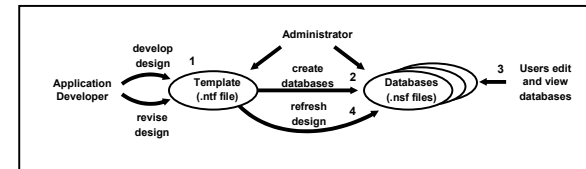
The Note: Structured To Support Collaboration And Replication

- A common structure for holding design, admin, and user data
 - one note header
 - zero or more items
 - zero or more responses
- The basic unit of replication



The Life Cycle Of A Notes Database

- Step 1 - Create an application "template"
- Step 2 - Deploy the application as databases
- Step 3 - Use the databases
- Step 4 - Revise and refresh the application design



Lotus Notes Comes With Many Applications

- Some are part of the basic product
 - mail template, Log template, etc.
- Generic applications useful to all customers
 - discussion databases
- All are customizable by the end user
 - even the product databases, like the mail template
 - use the Domino Designer to modify an application

Notes Databases Are Portable

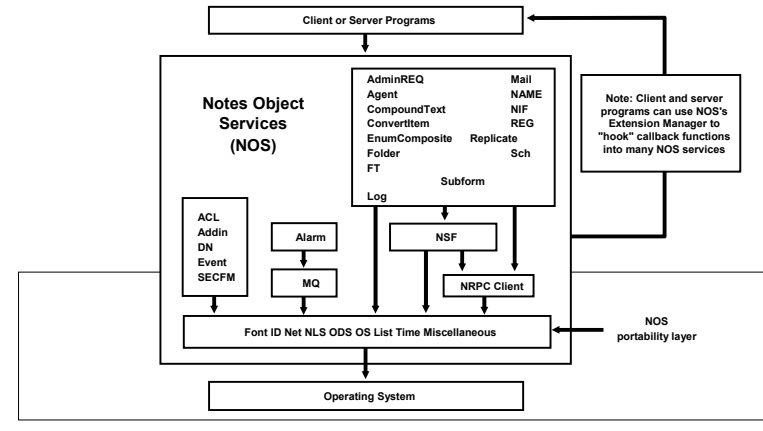
- On-Disk-Structure (ODS) format is used on all platforms
 - complete binary compatibility regardless of where created
 - create a database on an NT server
 - move it to an IBM 390 server to balance your system
 - replicate it to your MAC and then work off-line
- Embedded programs interpreted the same on all platforms
 - Formula language
 - LotusScript
 - Java
 - JavaScript

Notes Object Services (NOS)

- The key software component used by the rest of Notes/Domino
 - contains the core Notes/Domino software facilities
 - portability services interface to platform operating systems
 - Notes Storage Facility (NSF) creates/maintains databases
 - Notes Indexing Facility (NIF) maintains database indices
 - replication services
 - etc.
- Key to Notes/Domino multiplatform capabilities
- Key to client-server and server-server interaction

Inside Notes Object Services (NOS)

- NOS services are programmatically available via the C API, LotusScript, Java, and Corba



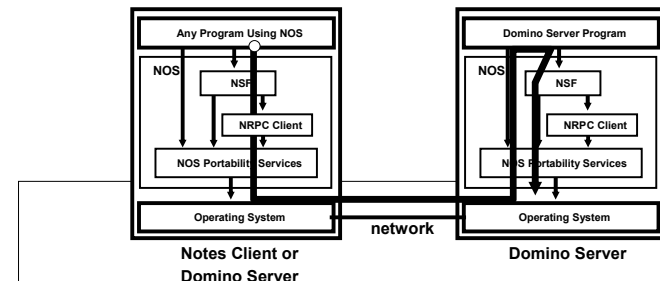
NOS Is Key To Notes/Domino Multiplatform Capabilities

- NOS's portability layer makes it easier to port NOS's clients
- NOS itself has been ported to many platforms
 - a representative sample of past and present configurations

Operating System	Processor	NOS	Client Programs	Server Programs
Mac OS	Power PC			
OS/2	Intel X-86			
Windows 95/98	Intel X-86			
Windows NT	Intel X-86, DEC Alpha			
Solaris	Intel X-86, Sun Sparc			
AIX	IBM RS6000			
HP-UX	HP PA			
OS/400	IBM AS/400			
OS/390	IBM 390			
Linux	Intel X-86			

The Notes Remote Procedure Call (NRPC) Service

- Simple access to NOS services running on remote servers
 - Programming is no different whether service is local or remote
- NRPC "client request" logic is NOS
- NRPC "server response" logic runs only on the server

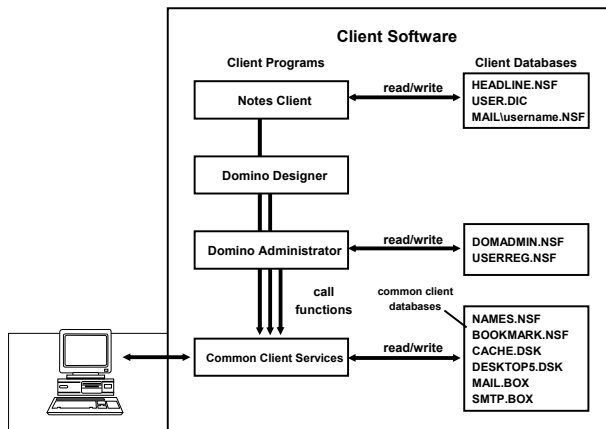


Notes/Domino Client Programs

Notes/Domino Client Programs Are Used To Design/Access Databases and Administer Domains

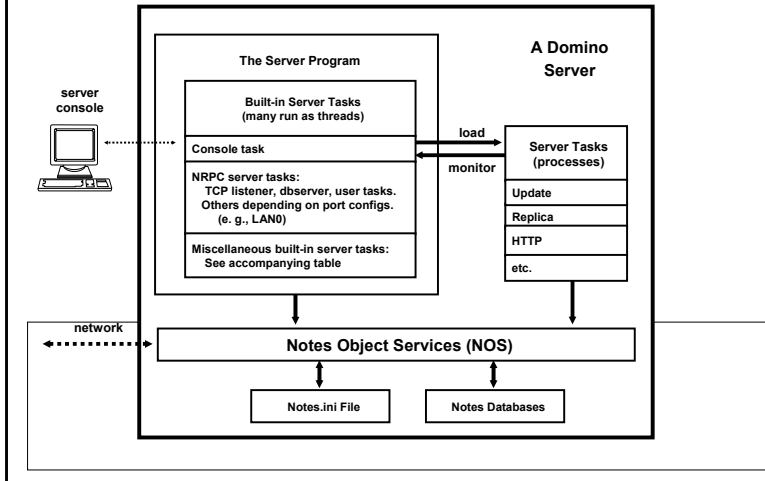
- The **Domino Designer**
 - create and revise Notes applications
- The **Notes Client**
 - create and modify local and remote databases
- The **Domino Administrator**
 - add/remove/modify server and user information
 - control overall system operations and security

Client Programs And Their Supporting Databases



Notes/Domino Server Programs

Server Work Is Done Using Threads And Tasks



Server Program And Server Tasks ...

- ... Support the care and feeding of shared databases

Server tasks that ...	Tasks
Maintain databases	Agent manager, cataloger, designer, directory cataloger, replicator, update, updall
Manage server and administrative activities	Admin process, cluster directory database manager, cluster replicator, cluster admin process, map generator, server console
Manage mail, calendaring, and scheduling	Calendar connector, IMAP, POP3, router, schedule manager, SMTP
Manage protocols	DIOP, DECS, HTTP, LDAP, NRPC
Monitor server activities	Billing, database statistics, events, ISPY, mail tracking, reporter, statistics collector, stats

The Server Uses A Pool Of NRPC Threads For TCP/IP

- A relatively small set of threads handle up to thousands of NRPC sessions connected through TCP/IP
 - for non-TCP/IP connections, there is one thread for each NRPC session
- A performance/scaleability improvement introduced in R5
 - prior to R5, the server created one thread for each NRPC connection, which affected the scaleability of large systems

Miscellaneous Server Tasks Implemented Using Threads

- Threads are often used for core server features that normally run on every server

Built-in Server Task	Description
Cluster Manager	Probes availability of cluster members and updates cluster statistics
Console	Manages server console input and display
Db Cache Ager	Ages cached databases (those pending for closing)
Db Cache Processor	Reopens cached databases and closes fully-aged databases
Db Fixup	Fixes inconsistent and/or bad databases listed in the fixup queue
Loadmon	Updates server load-balancing statistics
Listener	Listens for connection requests by NRPC clients (one per NRPC port)
Name Server	Maintains a list of networked servers and the services they provide
Poll	Performs periodic housekeeping - for example, scheduling other tasks to run
RM Checkpoint	Recovery Manager task prepares transaction-log checkpoint record
RM Flush	Recovery Manager task flushes cached DB info in order to free log space
UBMIO	"Cleans" modified database information cached in the UBM by writing DBs
UBM Cleaner	Ages modified database information in the UBM and schedules cleaning
User	Manages a user's connection to the server (one per NRPC user)

Notes/Domino Domains, Directories, Administration

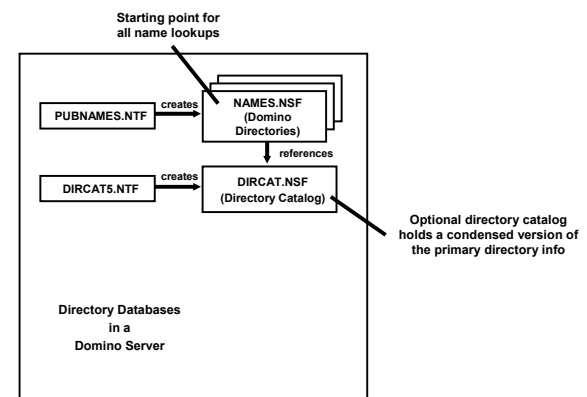
Domains, Directories, And Administration

- A **domain** is a set of related entities
 - servers, users, groups of servers and/or users, databases, ...
- A **directory** is a database that defines a domain
 - contains information about each entity in the domain
 - every server in the domain gets a replica copy of the directory
- Administration programs** manage directories and database administration information
 - add/remove servers, user's, etc., from directories
 - specify server and database security, replication schedules, ...
 - migration of users and mail from other systems

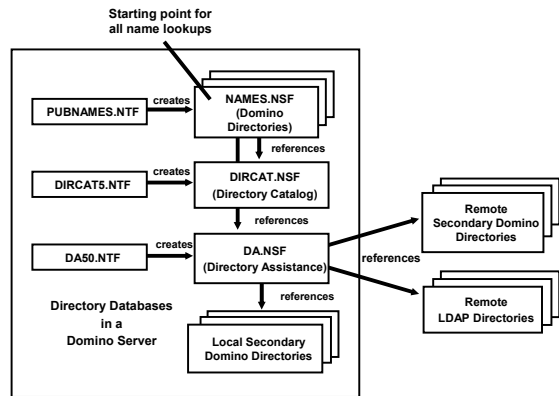
Directory Information Can Be In Several Databases

- Primary directory info for a domain is in ...
 - a replica of the Domino Directory database on each server
 - an optional, compacted Directory Catalog database
- An optional Directory Assistance (DA) database is used to access
 - directories for other ("secondary") Domino domains
 - LDAP directories
- See upcoming slides for diagrams

Directory Databases For The Local Domain



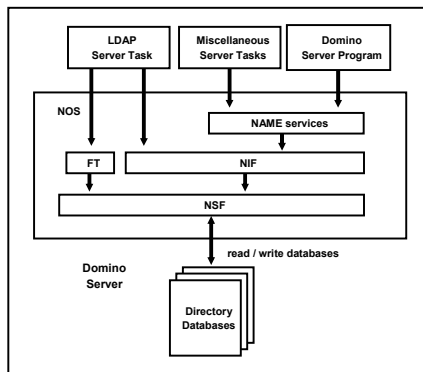
Additional Directory Databases For Other Domains



Domino Can Provide Directory Information In Several Ways

- In response to LDAP requests fielded by the LDAP server task
- In response to calls to NOS NAME services
 - by local programs or in response to remote NRPC requests
 - information returned may come from ...
 - one of the domain's Domino Directories on the server
 - the Directory Catalog
 - local or remote secondary Domino directories
 - a remote LDAP server
- See next slide for a diagram

Domino Can Provide Directory Information In Several Ways



How Domino Administration Works

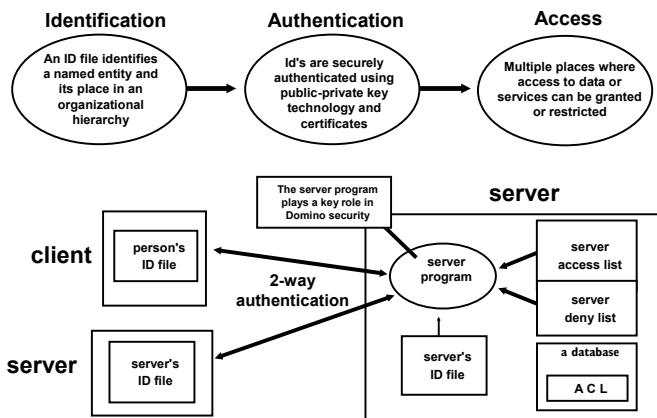
- Operations affecting a single database can often be done immediately
 - for example, add an entry to one ACL (Access Control List)
- Those affecting many databases (remove a user) are usually done by the ADMINP server task running on each server
 - requires that a database with replicas on many servers specify one of those servers to be its "admin server" in order to avoid replication conflicts
 - ADMINP requests are first posted to the ADMIN4.NSF database of the server being administered. The request is then replicated to other replicas of ADMIN4.NSF on every server
 - the ADMINP task on a given server processes ADMIN4.NSF requests for databases that designate the server as their "admin server," and it ignores requests that pertain to other databases

ADMINP Example: Remove A User From The Directory

- ADMINP takes full advantage of Domino facilities
 - implemented as a server task running on every server
 - uses replication to propagate requests in a distributed system
- The flow of control ...
 - post request to the ADMIN4.NSF on the server being administered
 - the request eventually replicates to the directory's admin server
 - ADMINP on that server removes the user and posts a request to its own replica of ADMIN4.NSF to clean all database ACLs
 - the directory deletion eventually replicates throughout the domain
 - the ACL request eventually replicates to all ADMIN4.NSF replicas
 - the ADMINPs eventually clean up the ACLs

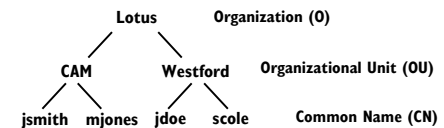
Notes/Domino Security

Notes/Domino Security is Based on Three Principles



Every Entity Has A "Distinguished" Name

- An entity's name encodes its organizational position
 - fully distinguished name: **CN=jsmith /OU=CAM /O=Lotus**
 - abbreviated name: **jsmith/CAM/Lotus**
- Typical names include these components
 - one top-level organization name (O)
 - up to 4 organizational-unit names (OU)
 - a common name (CN)



You Are Who Your ID File Says You Are

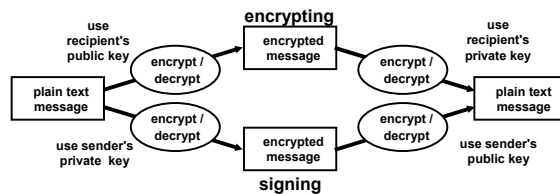
- An ID file contains info for securely communicating with others
 - the entity's name
 - the entity's public and private keys
 - certificates which validate the entity's name and public key
- An ID file can (and should) be password protected
 - the password encrypts/decrypts the contents of the ID file
- Once compromised, an ID file should be destroyed and replaced
 - just changing the password won't repair the damage done

Certificates Form The Basis Of A "Trust" System

- A certificate validates a public key of a named entity in your domain
- A cross-certificate validates a public key of a named entity in another domain
- Certificates are trusted if issued by ...
 - an organizational ancestor
 - entities having organizational ancestors in common with you
 - entities having organizational ancestors in common with a entity validated by a cross certificate

Public-Private-Key Technology Is Used To Secure Both Data And Communications

- Encode using recipient's public key, decode with private key
 - used to securely send information to a known entity
- Encode using sender's private key, decode with public key
 - used to sign information and to authenticate
- The same algorithm both encrypts and decrypts



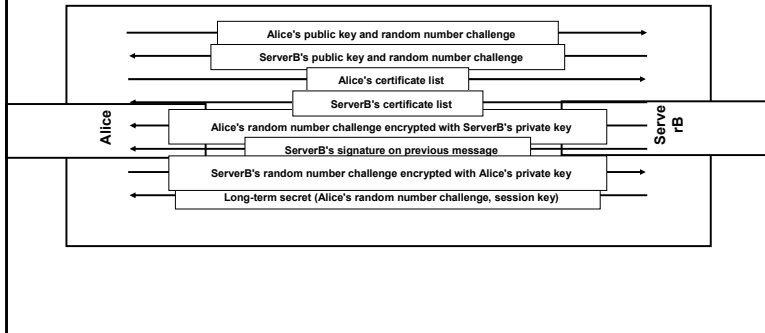
Use A Digital Signature To Indicate Approval Or Validity Of Data

- To sign information ...
 - hash the information into a smaller value
 - encrypt the hash value using your private key
 - use the encrypted hash value as the signature
- To verify a signature ...
 - hash the information into a smaller value
 - decrypt the signature using signer's public key
 - the decrypted signature should match the hash value

Notes Server Authentication Is A Two-Way Process

• Client authenticates the server and server authenticates the client

- exchange of public keys and certificates
- exchange of random numbers encrypted with private keys
- switch to a more-efficient session encryption key



Once An ID Is Authenticated ...

• ... Security is based on passing through a series of roadblocks

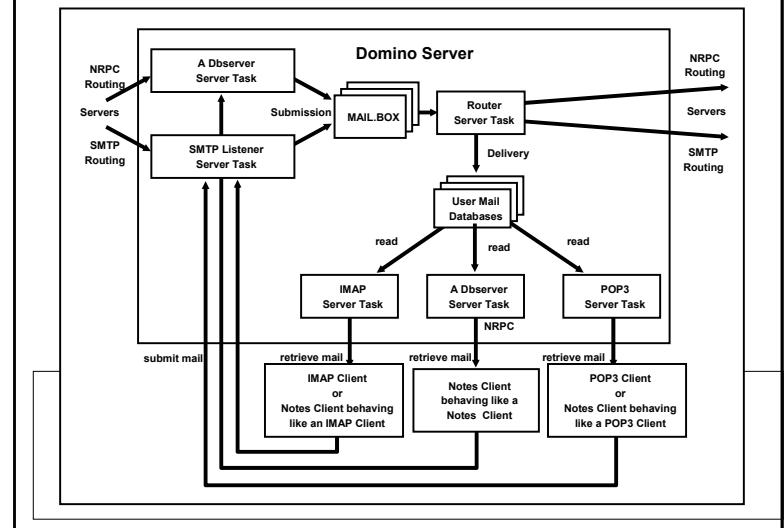
- Server and database services enforce various access control lists
- server inclusion and exclusion lists
- database Access Control List (ACL)
 - manager, designer, editor, author, reader, depositor, no-access
- document author and reader lists
- etc.

• General rule:

- the most-restrictive access applies if there are ambiguities

Notes/Domino Messaging

Basics of Domino Messaging



Notes/Domino Programmability

Programmability: Tailoring Notes To Your Needs

- Numerous ways to access NOS services programmatically

Domino Designer	Design the precise look and feel of your Notes/Domino application and use portable logic written in LotusScript, JavaScript, Java, and Formula Language.
C and C++ Toolkits	Develop Domino server addins, use the Extension Manager to customize core NOS behaviors, and/or use NOS services by a non-Notes application.
LotusScript Extension (LSX) Toolkit	Define your own LotusScript Domino classes.
Connector Toolkits	Use a connector toolkit like Domino Enterprise Connector Services (DECS) to interface Domino to external data sources.
Database Drivers	Use a database driver like NotesSQL to make Domino resemble other backend data sources.
DIIOOP Server Task	Access NOS services through CORBA.

Other Sources Of Information

- Notes/Domino document set (yellow books) and on-line help
- "Inside Notes: The Architecture of Notes and the Domino Server"

-<http://www.notes.net/notesua.nsf>

-Select "Search Documentation Library" then search for "Inside"

-the Lotus UA pedestal has it on a CD along with other technical documentation, too

Questions and Answers

- JMP101: Basics of the Lotus Notes/Domino Architecture

Please --

Don't forget to pass in your evaluations.

-- Thanks