Case Study in Directory-Enabling an Application

AddressIPWorks

Session ES139

Anton Okmianski
[Andy Bennett]

PROCESS SOFTWARE
CORPORATION

Our Application

Domain Name System (DNS) Server
Name to IP Address Translation

Dynamic Host Configuration Protocol (DHCP) Server
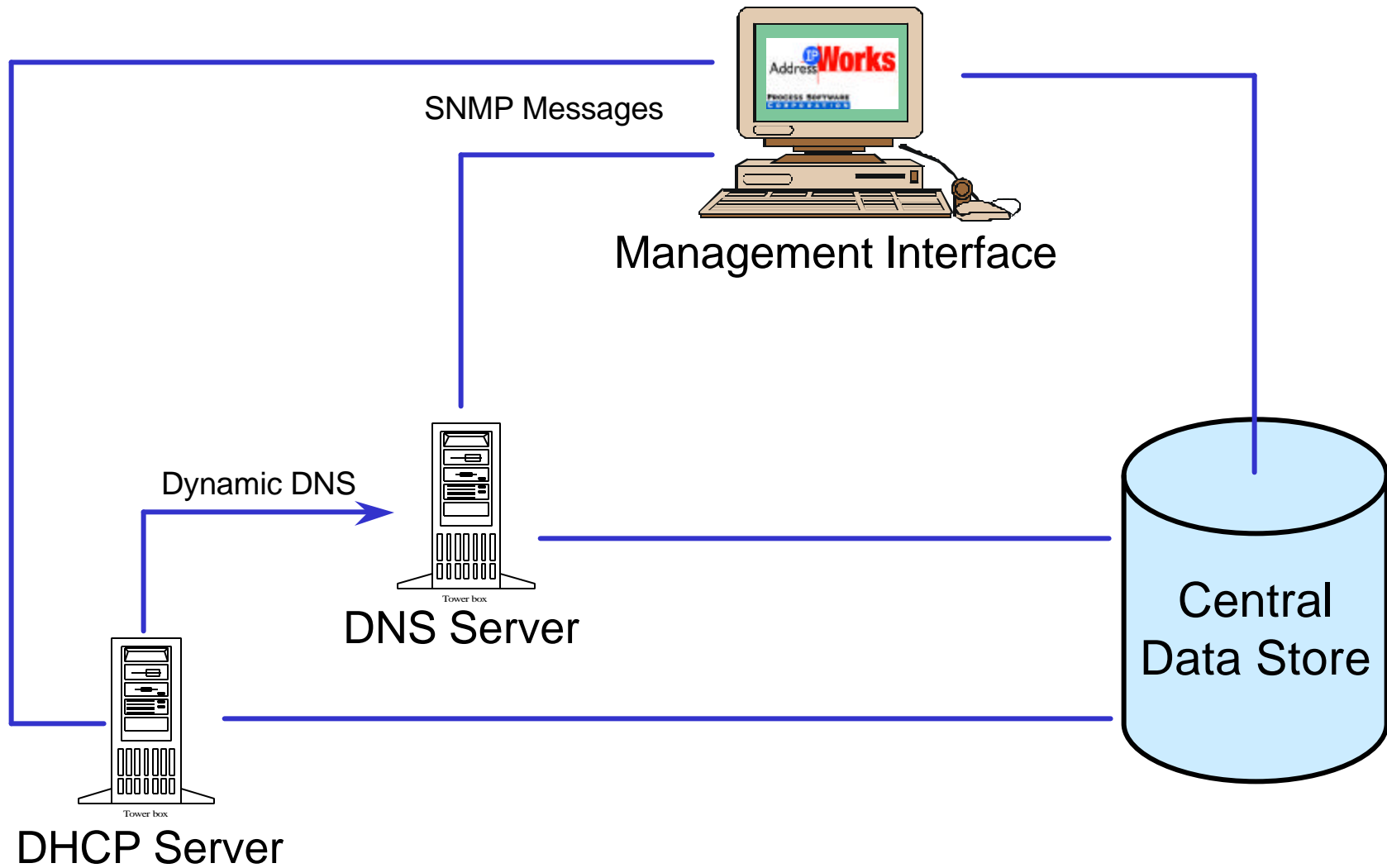Dynamic IP Address Assignment

Graphical and CLI Management Interface

IP AddressWorks
Centralized IP Address Management

PROCESS SOFTWARE
CORPORATION

IP AddressWorks Architecture

Management Interface

SNMP Messages

Dynamic DNS

DNS Server

Tower box

DHCP Server

Tower box

Central
Data Store

PROCESS SOFTWARE
CORPORATION

What is a directory?

- Specialized database

- Massively distributed database

- Appears to be virtually centralized

- Optimized for read performance

- Attribute-value based

- Global hierarchical namespace

- Entries with unique global distinguished name (DN)

- Security based on global hierarchy

Why directory-enable application?

- Get access to data already in directories

- Provide standard way of accessing our data

- Provide simple way of accessing data

- Easier integration with other network management products

- Directory Enabled Networking (DEN)
  - policy-based routing, QoS, COS, policy networking, etc

- High performance read operations

Access Protocol - LDAP

- Lightweight Directory Access Protocol

- Simplified TCP/IP version of X.500 DAP

- Responsible for popularizing directories

- Widely supported by directory vendors

- Most popular way of accessing directories

- Connection-based protocol

- Basic operations: *bind/authenticate, read, search, add, delete, update*

- LDAPv3 has ability to read/push schema

- RFC 1487, RCF 2251 (LDAPv3)

- Widely available and free client SDKs

PROCESS SOFTWARE
C O R P O R A T I O N

LDAP Directory Vendors

- **University of Michigan SLAPD**

  http://www.umich.edu/~dirsvcs/ldap/index.html

- **Sun Directory Services**

  http://www.sun.com/solstice/telecom/LDAP.html

- **Netscape Directory Server (our choice)**

  http://developer.netscape.com/tech/directory/index.html

- **IBM SecureWay Directory**

  http://www-4.ibm.com/software/network/directory/

- **Novell Directory Service (NDS)**

  http://www.novell.com/products/nds/index.html

- **Siemens DirX**

  http://www.siemens.de/directory/en/metadir/index.htm

- **Innosoft Directory Services (IDS)**

  http://www3.innosoft.com/index.html

- **Microsoft Active Directory**

  http://www.microsoft.com/windows/server/Technical/directory/ADarch.asp

- **ISOCOR Global Directory Server (GDS)**

  http://www.isocor.com/prodsol/PSdrgds.htm

LDAP Client SDKs: Abundant and Free

- Netscape SDK for LDAP (C and Java)

  http://developer.netscape.com/tech/directory/index.html

- Sun's Java Naming & Directory Interface (JNDI)

  http://java.sun.com/products/jndi/index.html

- MS Active Directory Service Interface (ADSI)

  http://www.microsoft.com/Windows/server/Technical/directory/adsilinks.asp?RLD=407

- Novel Directory Service (NDS) SDK

  http://www.novell.com/products/nds/index.html

- IBM SecureWay LDAP Client SDK (for C)

  http://www-4.ibm.com/software/network/directory/library/publications/31/program_ref/ldap.htm

- Isochrone LDAP SDK (Java)

  http://www.isochrone.com/products/ldapsdk/ldapsdk.htm

- Perl Net::LDAP Library

  http://search.cpan.org/search?module=Net::LDAP

- Innosoft LDAP Client SDK
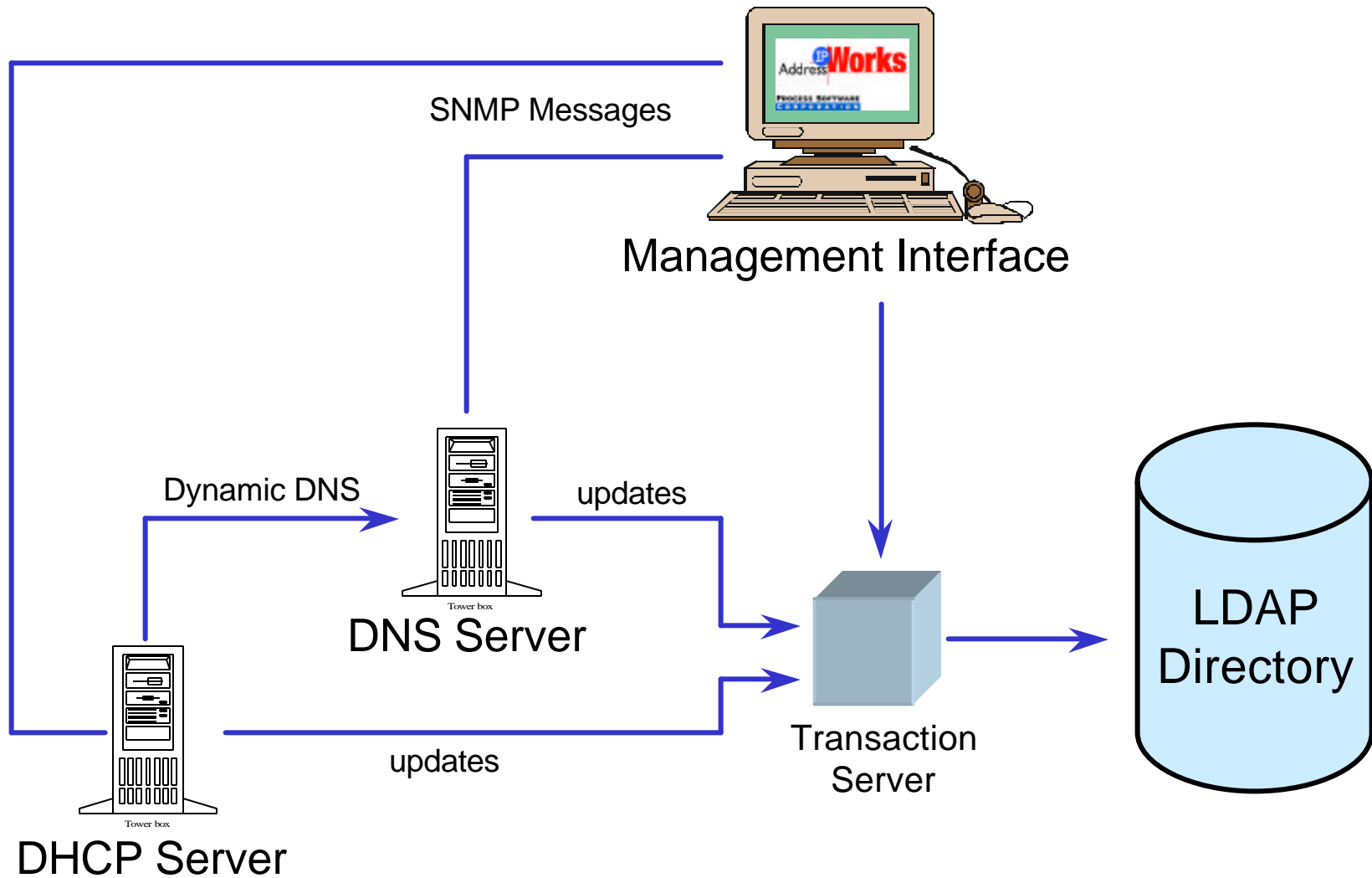
  http://www3.innosoft.com/ldap-sdk.html

Challenges With LDAP Directories

- **LDAP Protocol Limitations**
  - No support for transaction across data entries
  - No support for relationships across data entries
  - No locking mechanism

- **Lack of standards for besides client-server interaction**
  - No interoperability between vendors
  - Vendor specific schema definition
  - Vendor-specific extended operations
  - Vendor-specific LDAP extensions

Lack of Support for Transactions

- LDAP only guarantees that update to a single object will be completed

- No support for updating multiple objects in one transaction

- Problem: can cause inconsistencies if there is a relationship between objects

- Most real-life data has relationships

- You should try not to store data that requires strict consistency into directory

- Our solution -- transaction server

IP AddressWorks Architecture

SNMP Messages

Management Interface

Dynamic DNS

updates

DNS Server

updates

DHCP Server

Transaction
Server

LDAP
Directory

No Support for Relationships

- Unlike relational DB, no concept of relationships

- In LDAP you maintain relationships by string-pointers from entry to entry

- LDAP searches do not provide a capability to do a SQL-like join operation

- SQL join allows you to compare two fields (columns) on the server

SQL "join" Operation

Users

Username	MacAddress
Bob Smith	01:00:00:01
John Doe	01:00:00:02
Bill Williams	01:00:00:03
Roy Donahue	
Jack Carter	01:00:00:05

Addresses

IP Address	MacAddress
10.0.0.4	01:00:00:01
10.1.1.15	01:00:00:02
10.0.0.148	01:00:00:05

• We need to get a list of all users with associated IP addresses matching by MacAddress

• We can use SQL statement like this:

```
Select Users.Username, Addresses.IPAddress
From Users, Addresses
Where u.MacAddress = a.MacAddress;
```

PROCESS SOFTWARE
CORPORATION

Working Around Lack of "join"

- ## Solution #1:   solve it with code

  Get all user *Names* and their *MacAddresses*
  For each user {
      Get *IPAddress* for a given *MacAddress*
  }

- ## Problem:
  – this does one search for *every* user record
    this can be very expensive.

Working Around Lack of "join"

- ## <u>Solution #2:</u>

  - change your data representation:

    - merge tables -- combine entries User and IPAddress into one or

    - duplicate fields in entries -- store IPAddress attribute in User entries as well as in IPAddress entry

---

- ## <u>Problem #1:</u>

  - this may increase the number of writes to the data.

- ## <u>Problem #2:</u>

  - this may cause inconsistency problems if data is not duplicated correctly.

Lack of Relationships: Updates

- You maintain relationships between entries by way of string pointer from one entry to certain some identifier in another entry

- If identifier changes or entry is deleted you need to update all all entries pointing to it

- When doing updates need to be able to update all related objects

- Our solution: build generic relationship management code layer

PROCESS SOFTWARE
CORPORATION

Performance Implications

- "Typical" LDAP performance:
  - Reads: 100-1000 objects per second
  - Writes: 10-50 objects per second

- DNS Server Max Performance
  - 300-5000+ operations per second

- DHCP Server Max Performance
  - 20-200 operations per second

- <u>Problem:</u> servers are single threaded and can't be delayed by LDAP performance

- <u>Solution:</u> Add Server Managers to manage interaction with LDAP (buffering, etc)

IP AddressWorks Architecture
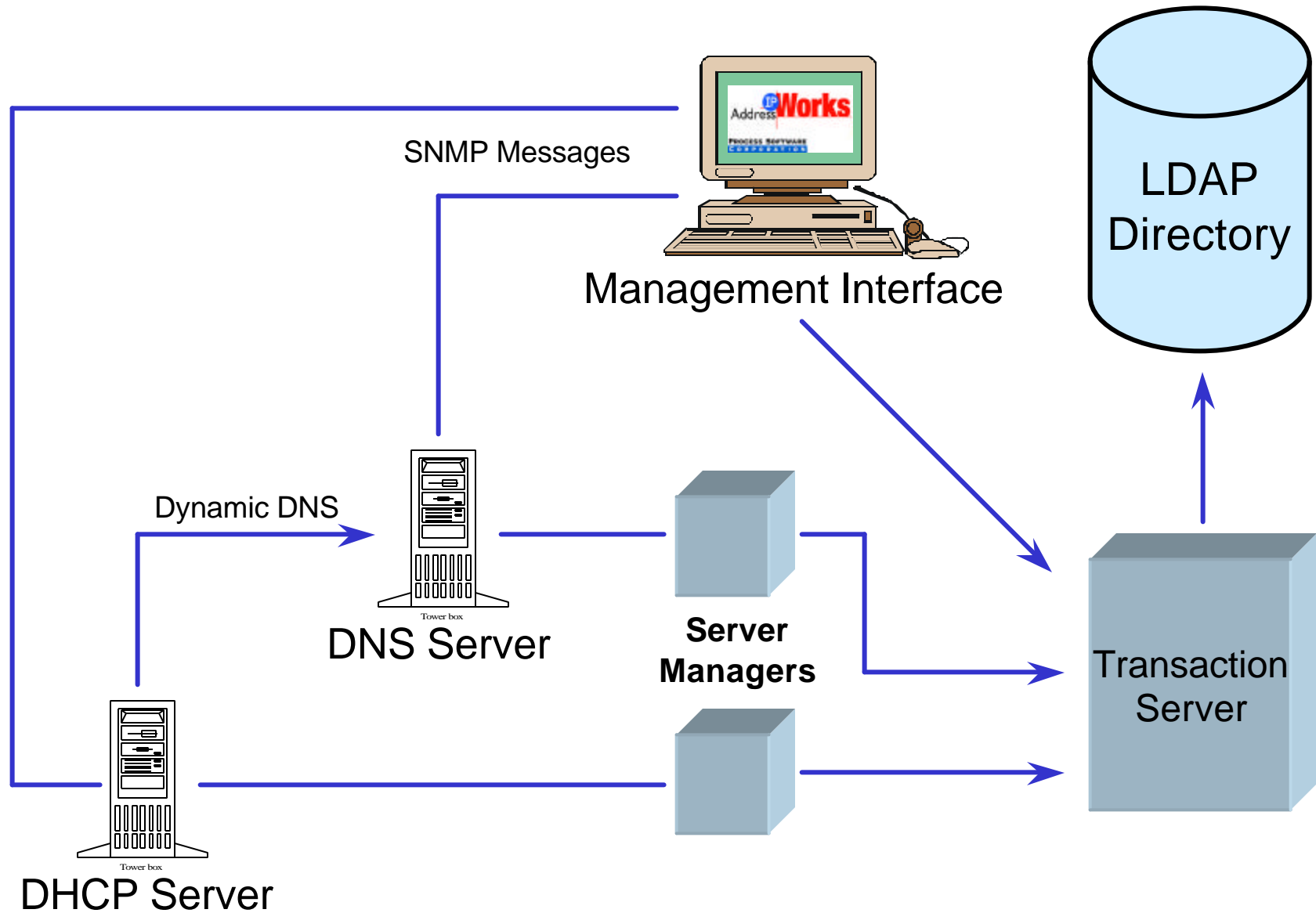
Non-Standard LDAP

- LDAP Extensions
  - persistent search
  - pages results
  - server-side sorting
- Extended server-side operations
- Schema extensibility
- Schema definition & customization
- Indexing of attributed (presence vs. equality)
- Access control

Conclusion

- LDAP directories have an industry momentum

- LDAP is really simple (too simple?)

- Choose carefully what to store in LDAP

- Define you schema diligently (minimize writes)

- Only atomic operations on a single entry are supported

- Need to handle transactions and relationships

- Keep data relationships to minimum

- Be careful about non-standard extensions