Cross-domain Identity Management System for Cloud Environment

PRESENTED BY:
NAZIA AKHTAR
AISHA SAJID
M. SOHAIB FAROOQI

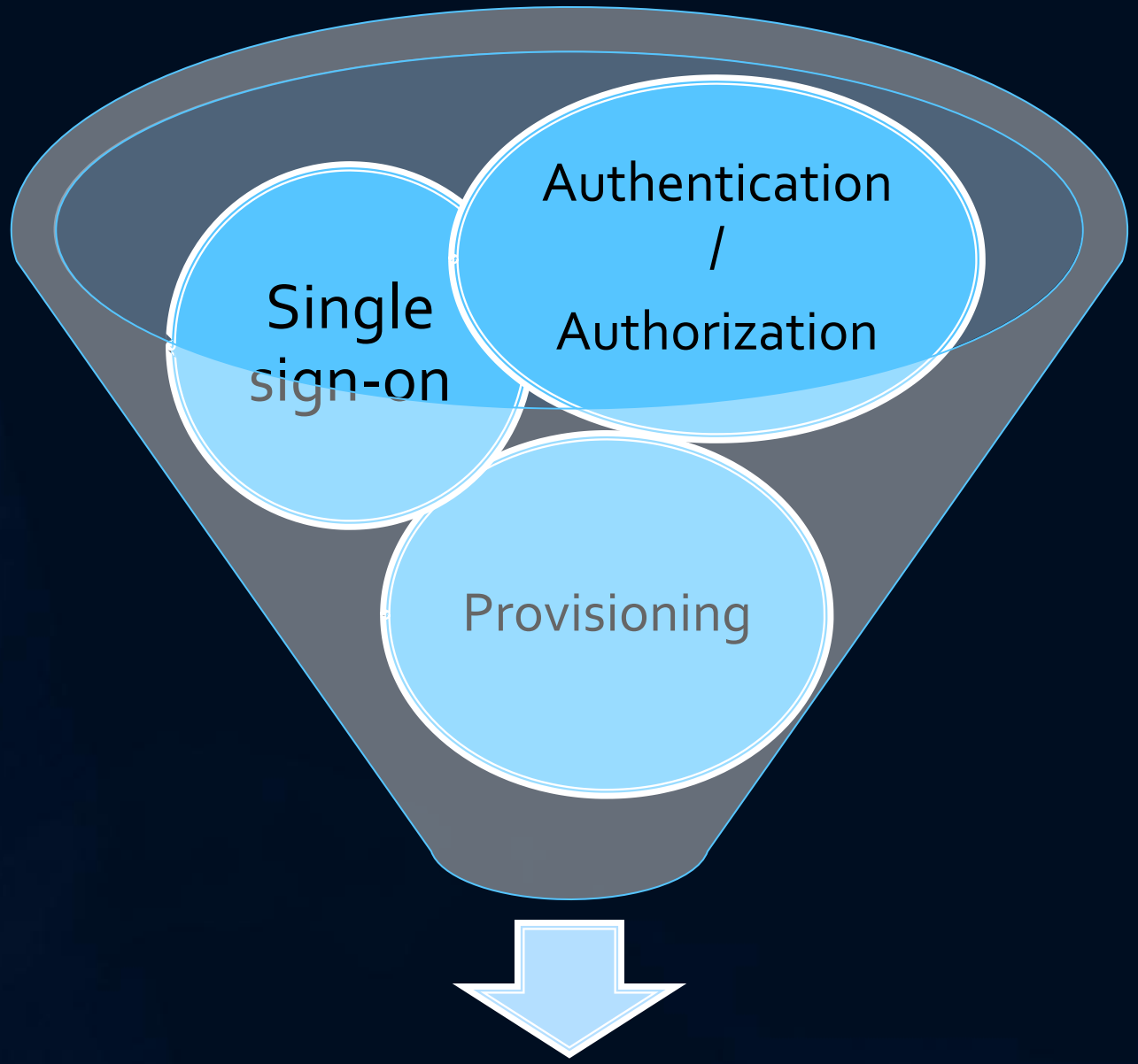
TEAM LEAD: UMME-HABIBA
THESIS SUPERVISOR: DR. M. AWAIS SHIBLI

Overview

- Cross-Domain Identity Management
- SCIM Schema
- UnboundID
- Maven
- Jetty Server
- LDAP
- Demo

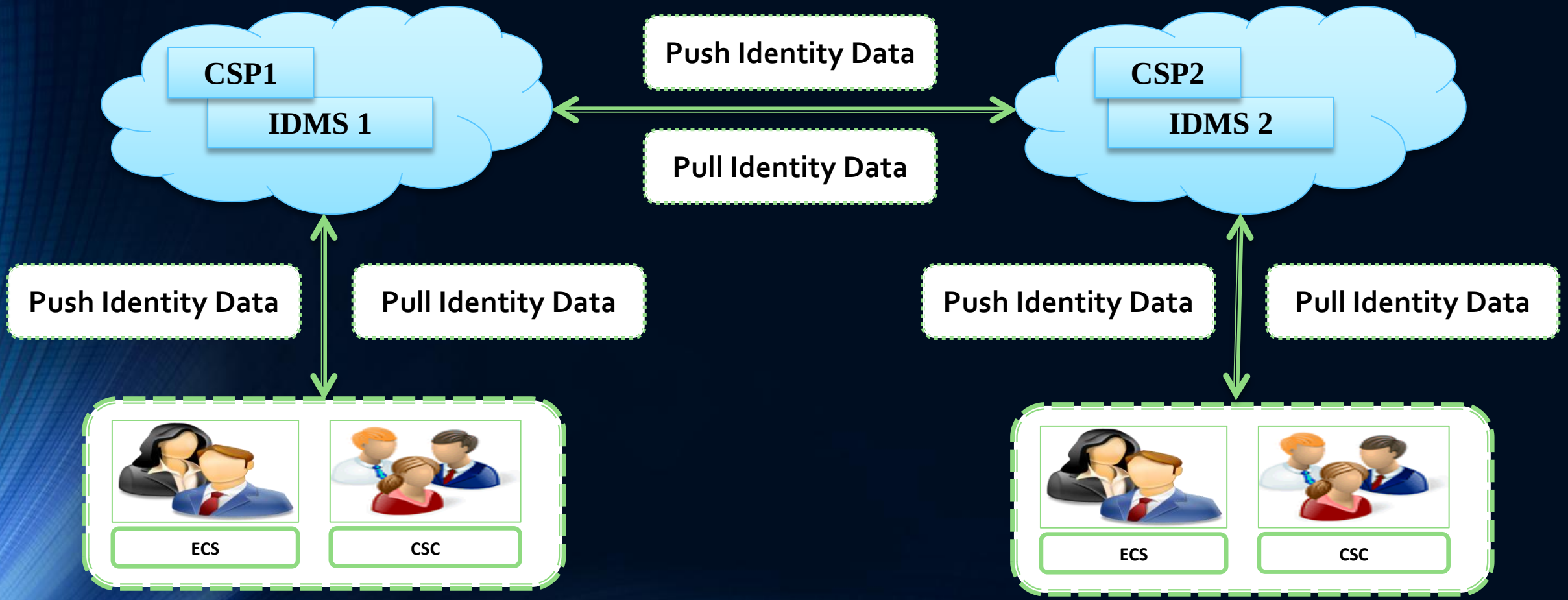
Identity Management

- Management of individual principles and privileges across the system.
- Their **Authentication** and **Authorization**.
- **GOAL:**
  - Increasing security and productivity
  - Decreasing cost and repetition of tasks



Identity Management

Proposed Architecture



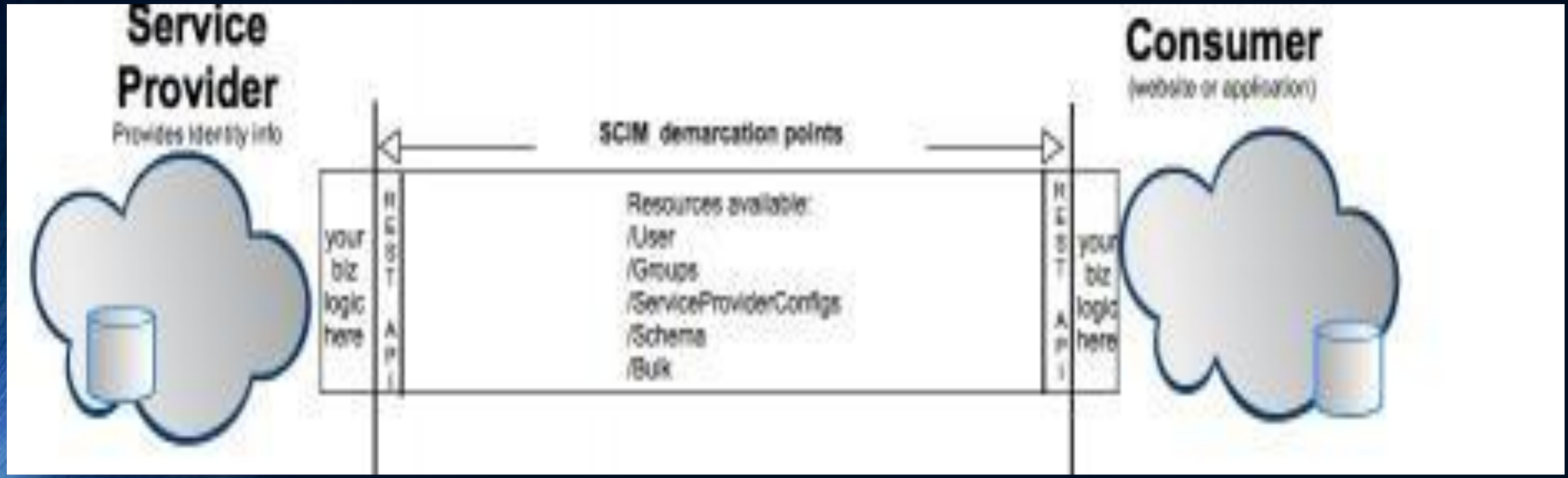
Problem?

In order to solve the interoperability issues in cross-domain Cloud environment, we are proposing an identity management system that will ensure seamless integration and utilization of identity credentials. In addition to basic identity management features, we intend to provide advanced security features including access right delegation, synchronization and user centricity in Cloud computing scenarios.

SCIM- System for Cross-domain Identity Management

- Designed for easier user identity management in cloud-based applications and services.
- Existing schemas and deployments
- Simplicity in applying existing authentication, authorization and privacy models
- Intend to reduce the cost and complexity of user management operations
- Provides a common user schema and extension model for exchange, using standard protocols

SCIM- Protocol



REST

Roy Fielding described REST as an architecture style which attempts “to minimize latency and network communication, while at the same time maximizing the independence and scalability of component implementations"

Roy T Fielding. REST Dissertation at http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

REST - Not a Standard

- REST is not a standard
- REST is just a **design pattern**
- REST does prescribe the **use** of standards:
  - HTTP
  - URL
  - XML/HTML/GIF/JPEG/*etc.* **(Resource Representations)**
  - text/xml, text/html, image/gif, image/jpeg, *etc.* (Resource Types, MIME Types)

What is REST?

- Uniform Interface
- Stateless
- Cacheable
- Client-Server
- Layered System
- Code on Demand (Optional)

REST- Operations

- **GET:** Retrieves a complete or partial resource.
- **POST:** Creates a new resource or bulk modifies resources.
- **PUT:** Modifies a resource with a complete, consumer-specified resource (replace).
- **PATCH:** Modifies a resource with a set of consumer-specified changes (partial update).
- **DELETE:** Deletes a resource.

UnboundID

- Dynamically store, manage, protect and share customer identity data in real time.
- The key editors and contributors for SCIM API
- First open source reference implementation for both SCIM client and server-side components.

Companies Working On SCIM

Maven

- **Maven** is a build automation tool used primarily for Java projects
- Maven uses an XML file to describe the software project
  - Dependencies
  - Build order
  - Directories and plug-ins.
- Uses the Project Object Model (POM)
- Maven dynamically downloads Java libraries and Maven plug-ins from one or more repositories such as the Maven 2 Central Repository, and stores them in a local cache.
- In NetBeans IDE 6.7 and newer, Maven support is included in the IDE

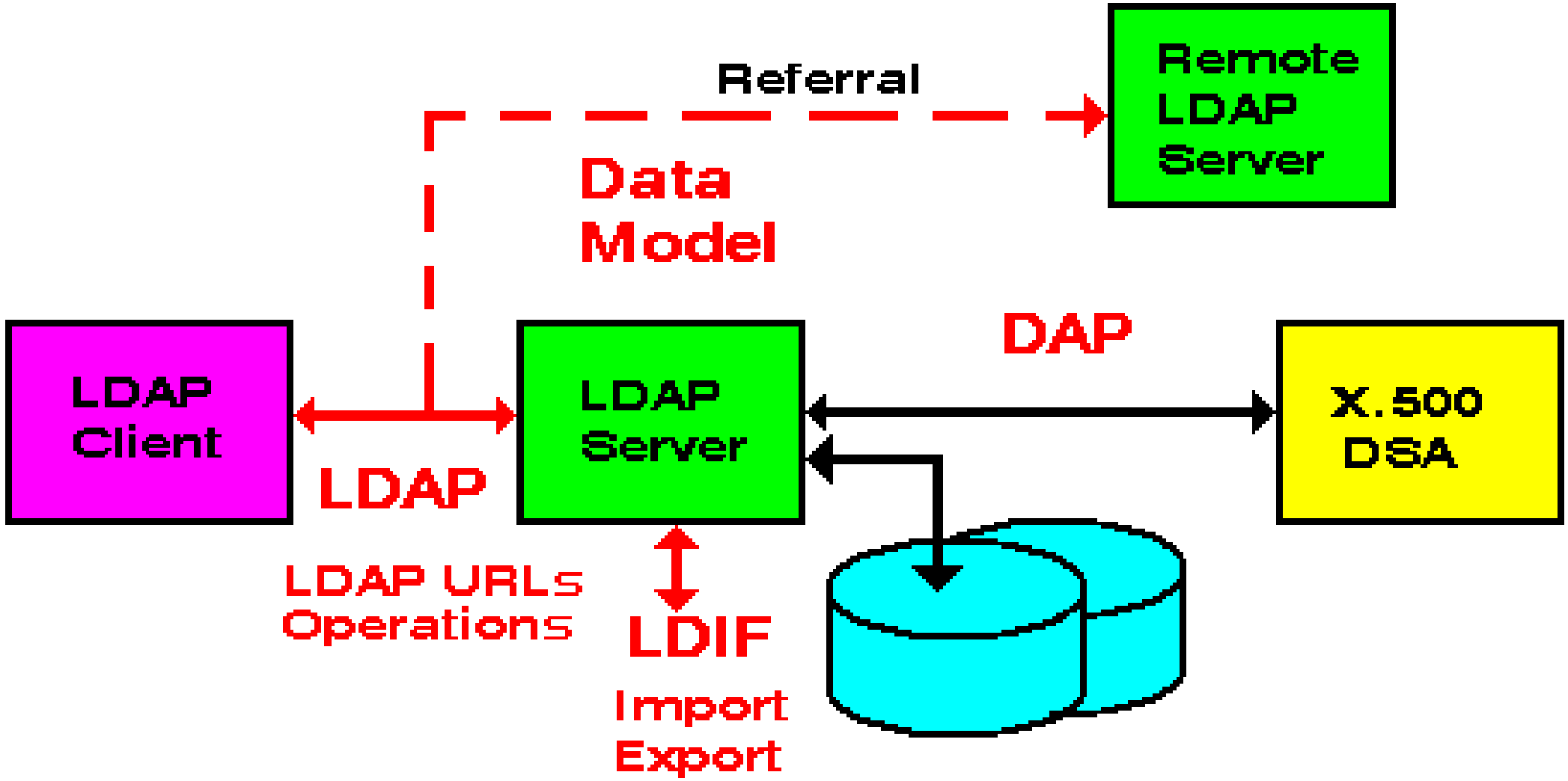
Jetty Server

- Jetty is an open source servlet container
- It serves Java-based web content such as servlets and JSPs.
- Jetty is written in Java and its API is available as a set of JARs.
- Developers can instantiate a Jetty container as an object, instantly adding network and web connectivity to a stand-alone Java app.

Lightweight Directory Access Protocol

- An application protocol for accessing and maintaining distributed directory information services over an Internet Protocol
- By directory services are organized set of records
- For example, email directory or phone directory

LDAP Architecture



Operations Supported for LDAP client

- **StartTLS** — use the LDAP TLS extension for a secure connection
- **Bind** — authenticate and specify LDAP protocol version
- **Search** — search for and/or retrieve directory entries
- **Compare** — test if a named entry contains a given attribute value
- **Add** , **Delete** or **Modify** an entry
- **Modify Distinguished Name (DN)** — move or rename an entry
- **Abandon** — abort a previous request
- **Unbind** — close the connection (not the inverse of Bind)

THANK YOU!

Let’s proceed to the demonstration…