Distributed Tivoli Identity Manager

- A Topology Enabler for Large Scale TIM deployment

Karthik Satishkumar (karsati@au1.ibm.com), Tivoli Security Principal, IBM SWS

PulseANZ 2010

Meet the people who can help advance your infrastructure

Agenda

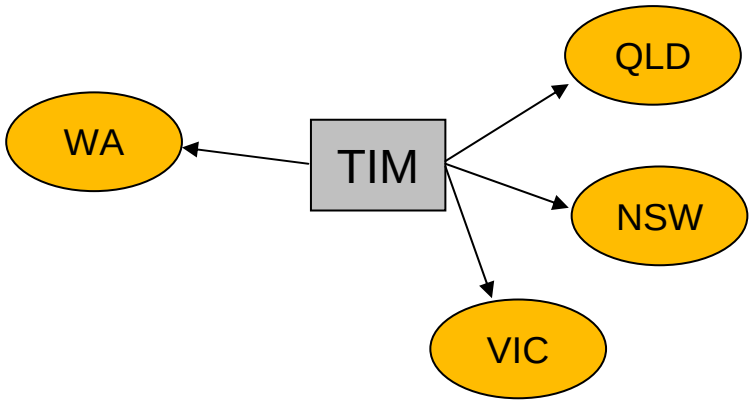
- Large Scale TIM Deployment and Challenges
- What is DTIM?
- DTIM - Benefits
- DTIM architecture
- DTIM Reference UI
- DTIM API
- ITIM Web Services
- Q&A

A Large Scale TIM Deployment...

- Can be an TIM Deployment with Large Userbase
  - Hundreds and thousands of users; sometimes million+ user base
  - Typically observed in HealthCare solution providers, Telecos, etc
  - Mergers and Acquisitions leading to complex consolidation exercises

- Can be an TIM deployment with Large number of services defined

- Can be an TIM deployment spread across a Large Enterprise spanning geographical boundaries



Key Challenges in Large Scale TIM deployments

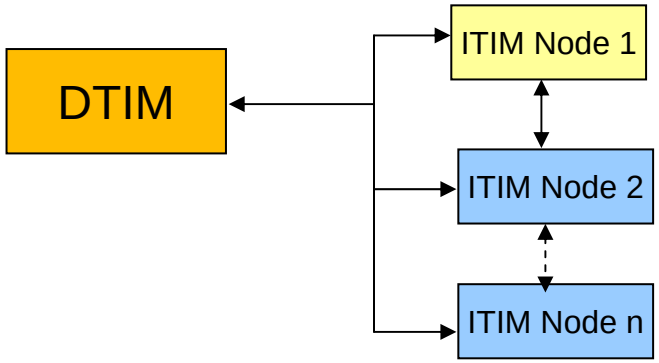
- Performance Implications due to Large Userbase and Large number of policies and services

- Operational Issues with Centralised Administration
  - Especially in Large Enterprises where individual organization units operate as independent entities (e.g: Govt with multiple departments)

- Roll Outs/Downtime affects the entire organisation

What is DTIM?

- A distributed approach for deploying ITIM
- Distributes service workload on different ITIMs (called ITIM nodes).
  - Individual ITIMs host a subset of services.
- A central DTIM node combines information from different ITIMs and presents to end user
  - User requests are routed to apt target ITIM.
- Decentralizes ITIM administration and configuration.



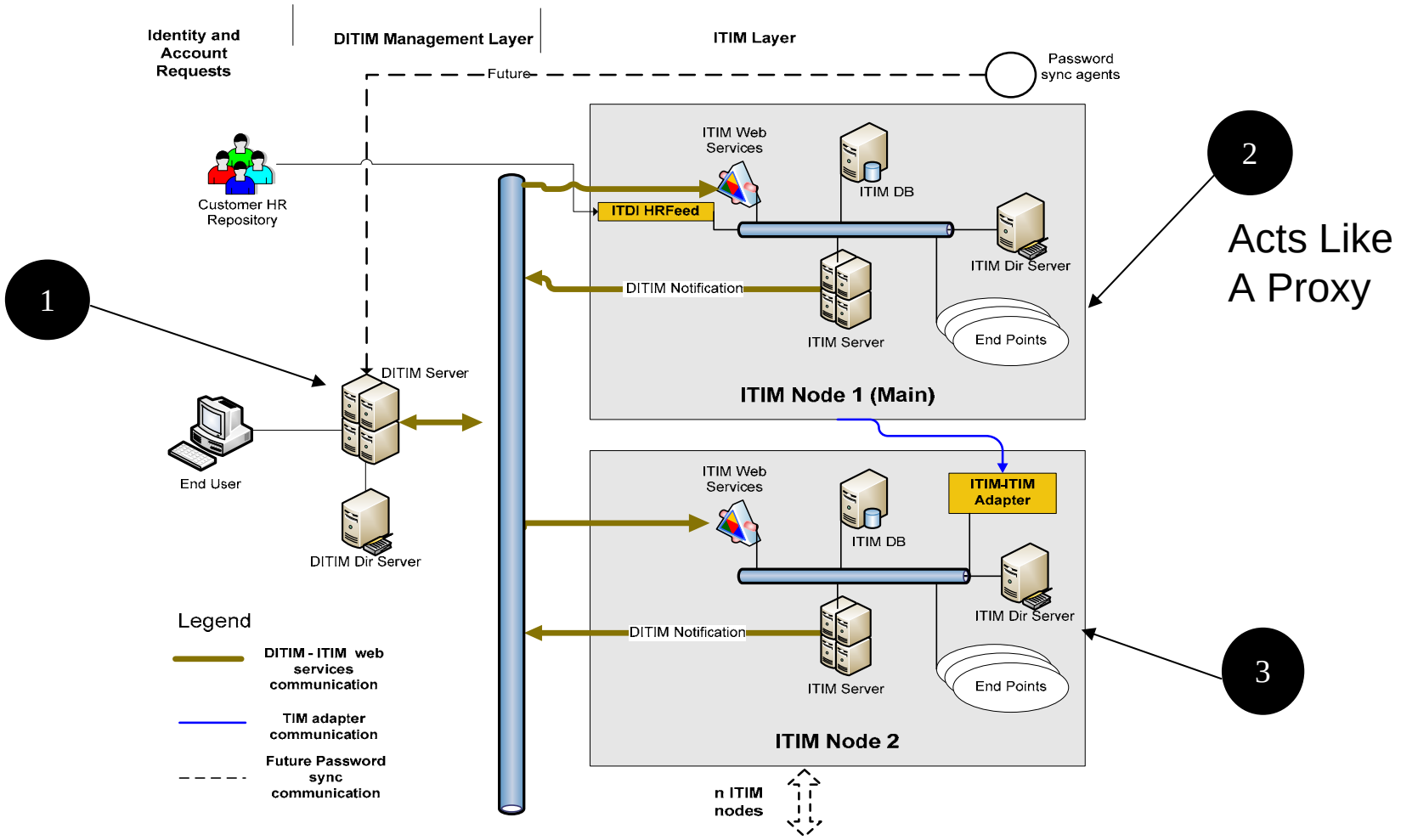
DTIM - Benefits

- Scalability
  - Add ITIM nodes as more services are rolled out.
- Avoid performance bottlenecks
  - Individual ITIMs process their own service requests *and recons.*
- ITIM nodes can be heterogeneous across platforms and ITIM versions.
  - Currently supports ITIM v4.6 and ITIM v5.x
- Allows “rolling” upgrade/maintenance of individual ITIMs.
  - Entire ITIM infrastructure does not shut down.

DTIM - Non Goals & Constraints

- Does not provide administrative and non end user tasks.
- Post Office consolidation will not take place at the DTIM level, only at the ITIM level.
- Constraints
  - Policies can only target multiple services on the same node.
  - Service pre requisites must be on the same node.
  - Custom workflow cannot use operations on entities not on the same node.

DTIM Architecture



Architecture

- Architecture is currently based on Master – Child topology.
- Each child ITIM can have its own subset of people as well as services.
- Leverages the custom **ITIM-ITIM adapter** for ITIM to ITIM communication.
- Long term goal is to provide Master – Child as well as Peer to Peer topologies.

Architecture – DTIM server

- Consists of J2EE application and local directory server (LDAP based).
  - Directory server contains stubs of selected objects on ITIM nodes (person, account, service).
  - Stub object points to complete object in target ITIM.
  - LDAP server also stores DTIM configuration.
  - J2EE application provides DTIM services and API implementation.

Architecture – DTIM node (contd.)

- Provides API to combine user information from ITIM nodes.
  - Communicates with ITIM nodes using a web services interface
  - Discovers services in ITIM nodes and maintains service to ITIM node mapping.
  - API provides functionality similar to ITIM self services capability.
  - Utilizes local LDAP server to derive node, person and account relationships.
- Routes user requests to appropriate ITIM over a web services interface.

Architecture – DTIM node (contd.)

- Retrieves person information from the Main ITIM (following the LDAP stub reference).
  - Changes are written back to Main ITIM which sprays to participating ITIMs via ITIM-ITIM adapter.
- Retrieves and updates account objects by following the account stub reference.
- All other objects are retrieved from participating ITIMs in parallel.
  - Time to retrieve objects is as long as the longest time to retrieve from any single ITIM.

Architecture – ITIM nodes

- One node is designated as Main and is used to host identity and optionally services.
- Other nodes host subsets of services.
- ITIM Web Services is installed on each node to enable web services communications.
  - ITIM Web Services is also available separately from DTIM.
  - Enables web services interface to ITIM for end user API.
  - Available as a free download from OPAL.

Architecture – ITIM nodes (contd.)

- Each ITIM node gets an identity feed from the Main ITIM via the ITIM - ITIM adapter.
  - Each child ITIM appears as a managed platform in the Main ITIM.
  - ITIM policies and automatic entitlements can be configured to distribute person objects to the child ITIMs based on person criteria.
  - ITIM-ITIM adapter uses standard adapter framework.

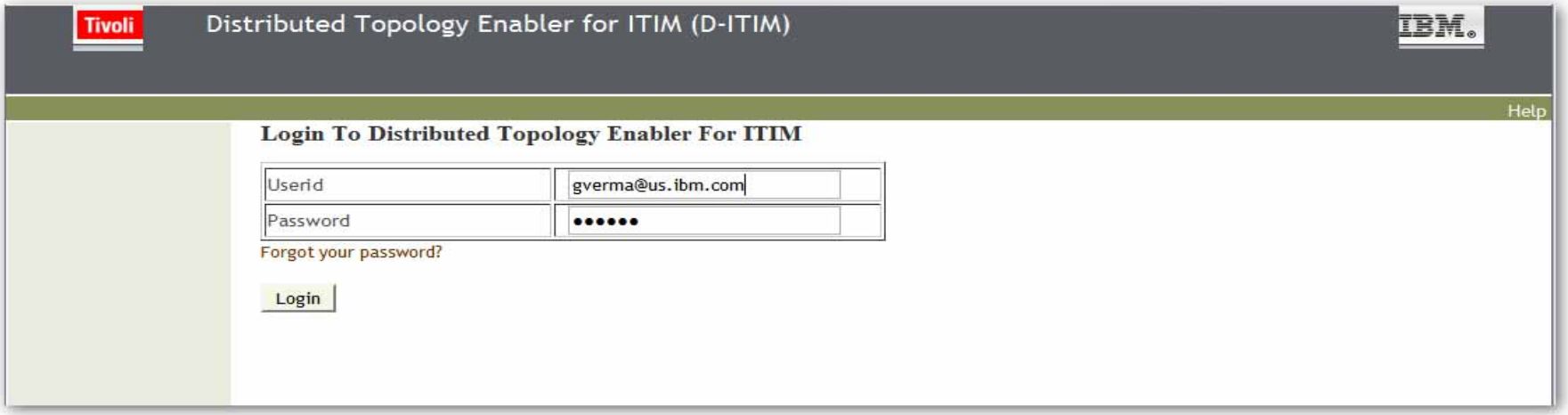
- Each ITIM node notifies DTIM of account and person changes via a Notifier module using a web services interface.
  - Adds and deletes are notified.
  - Attribute changes other than suspend/restore are not notified.

DTIM Reference UI

- DTIM UI is a reference UI that implements the DTIM API.
- Intention is to show DTIM functionality, not satisfy user serviceability requirements.
- Uses Java Server Faces (JSF) for view processing. Source code is included with DTIM.
- Provides self service user functions.
  - In the DTIM architecture, admin tasks are decentralized on local ITIMs.

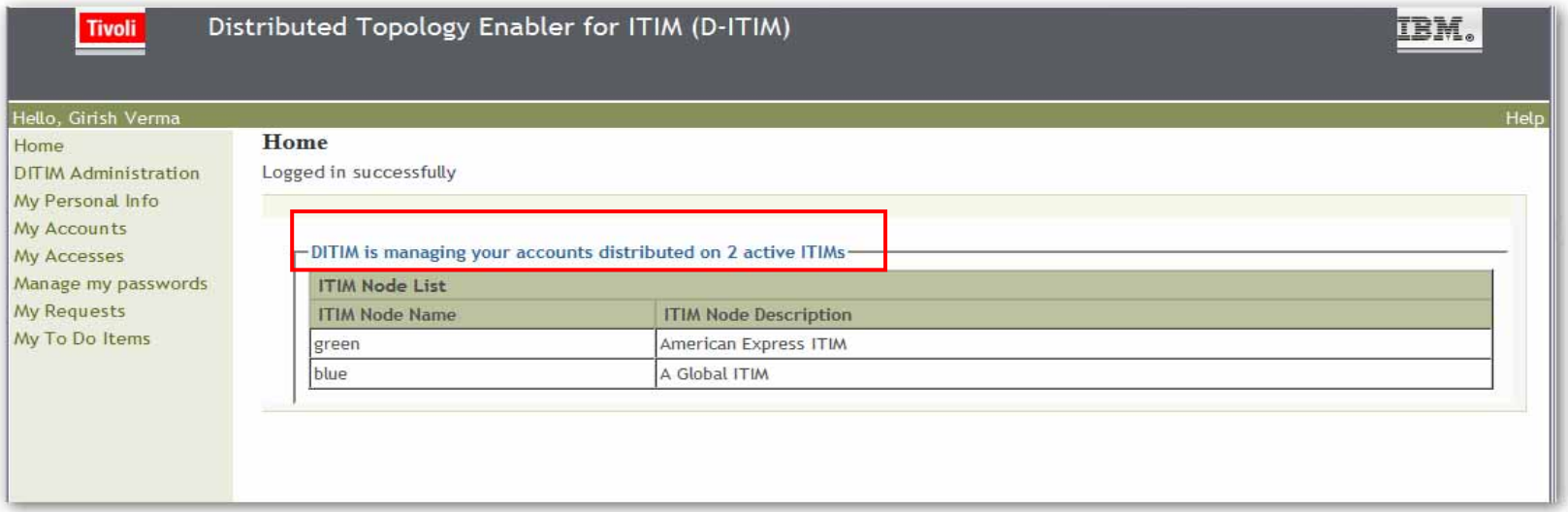
DTIM Reference UI - Login

- Supports simple login or SSO.
- Authenticates against Main ITIM.



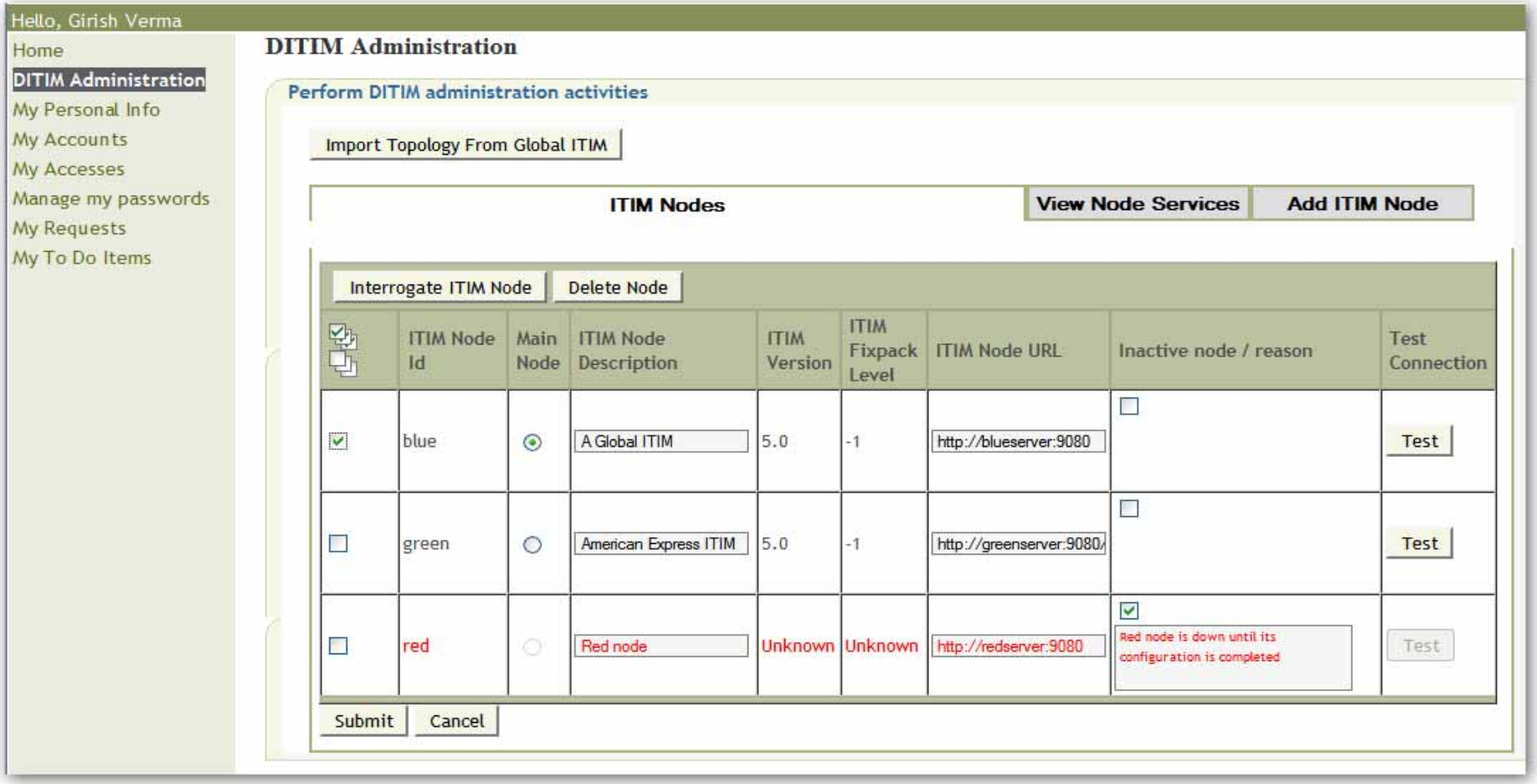
DTIM Reference UI – Home Page

- Lists footprint of logged in user.
  - User may have presence on only some of the ITIMs.
- Configurable to show accounts/pending items.



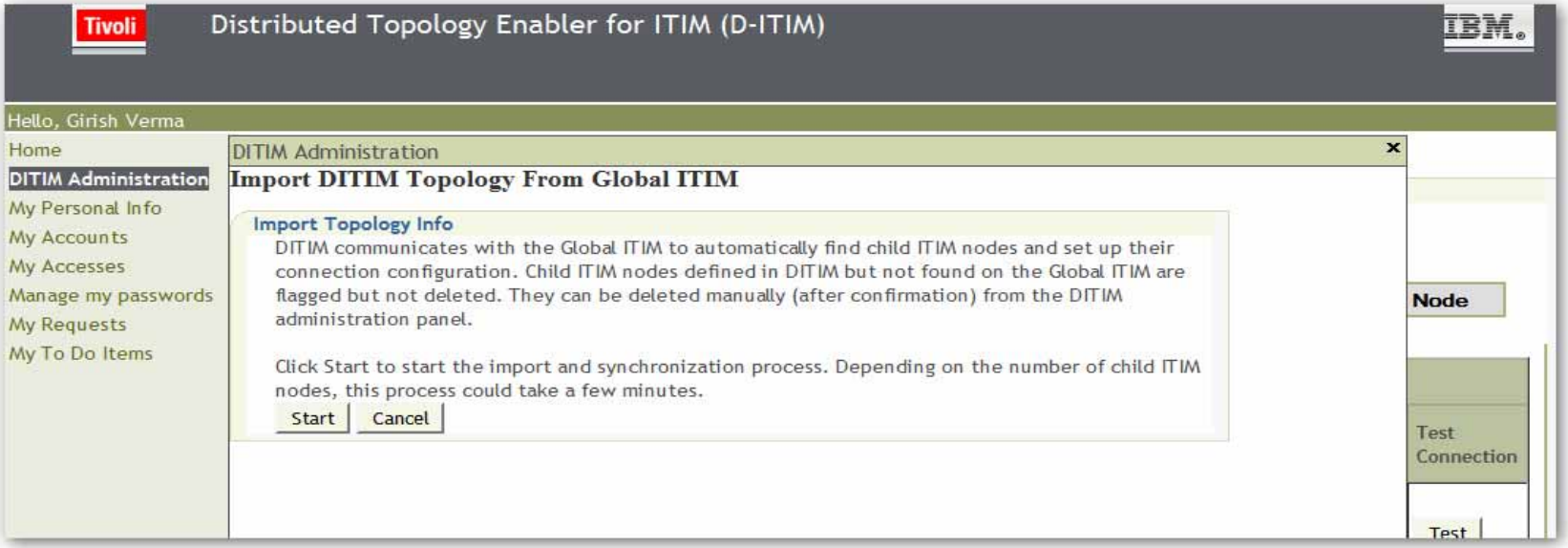
DTIM Reference UI – DTIM Admin

- Allows configuration of list of nodes.
  - Nodes may be inactivated for maintenance.



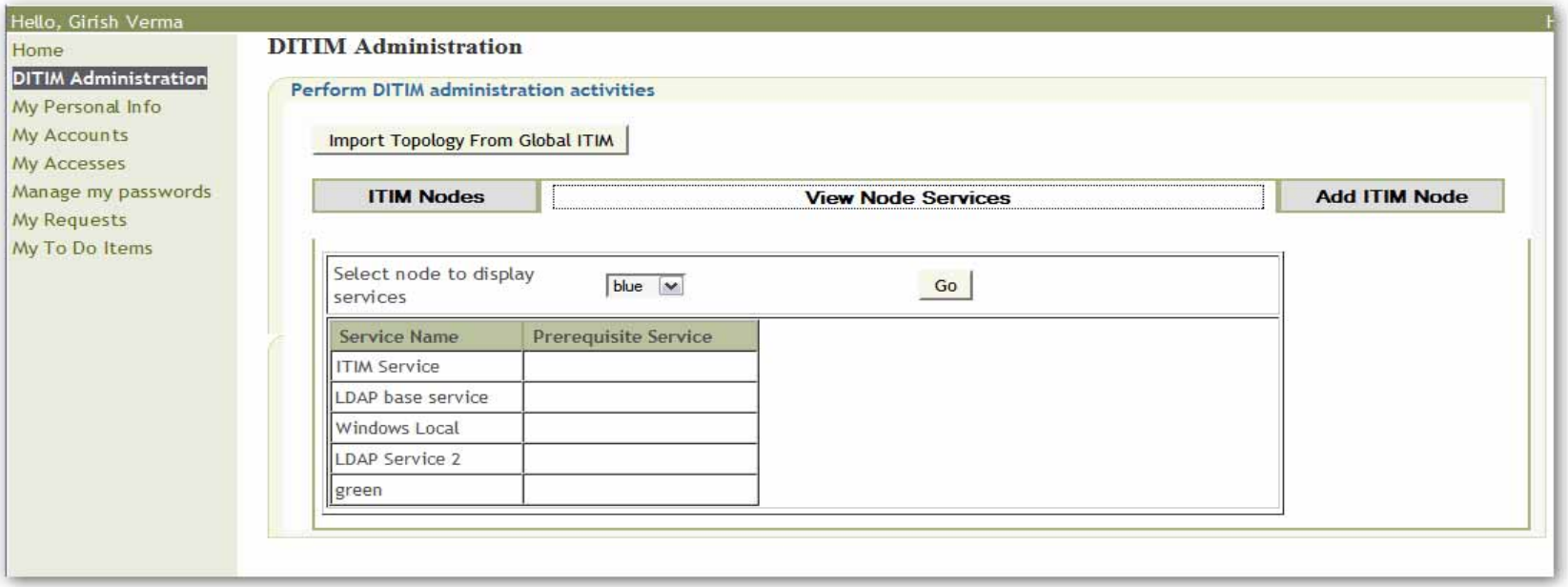
DTIM Reference UI – DTIM Admin

- Import DTIM configuration automatically from the Main ITIM.



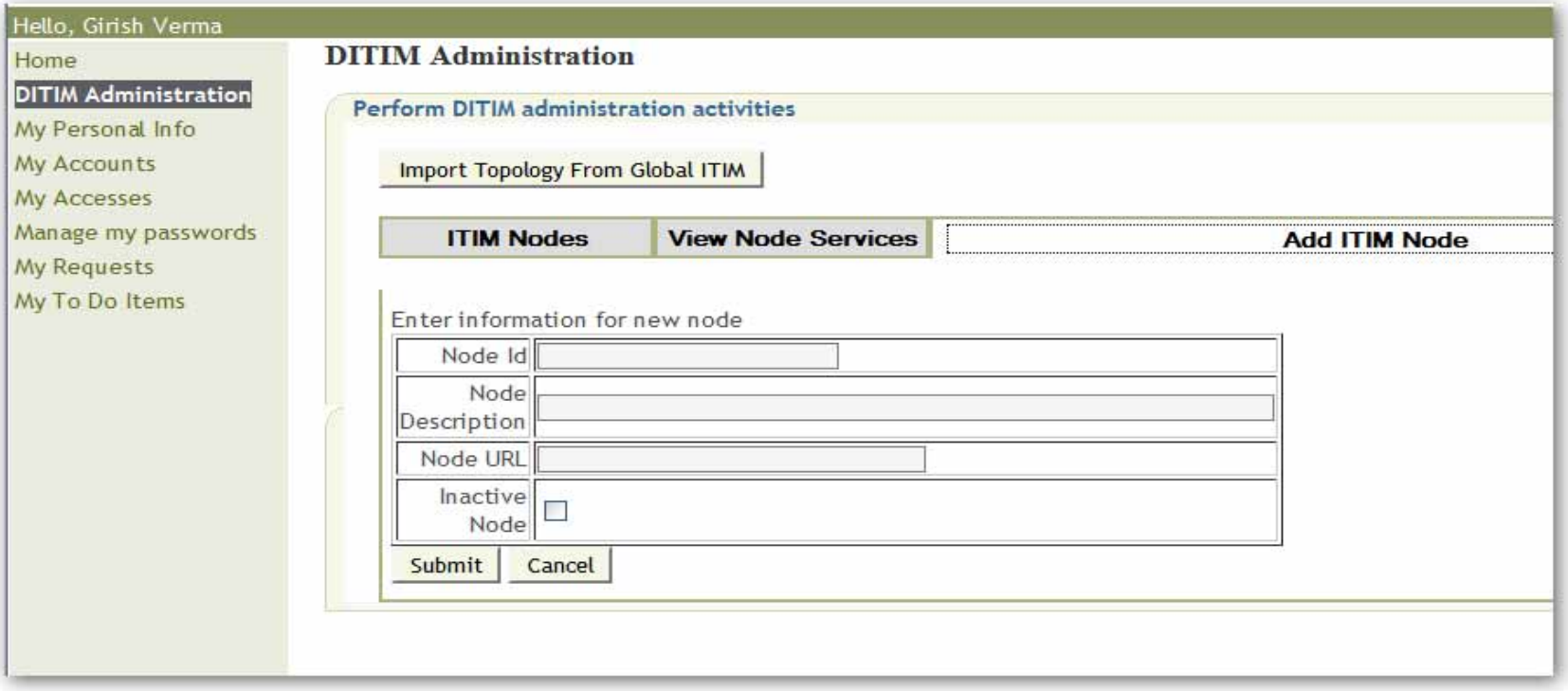
DTIM Reference UI – DTIM Admin

- View Services on an ITIM node
  - Services are imported by interrogating an ITIM node



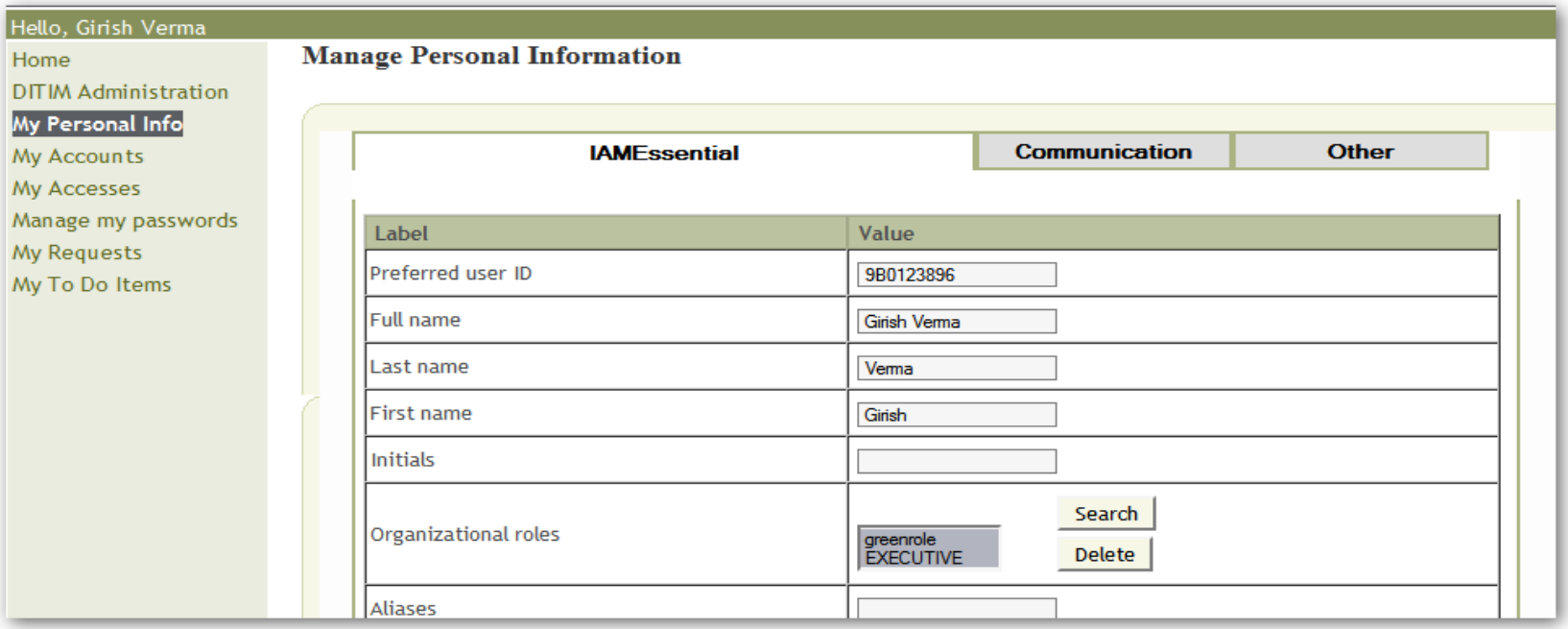
DTIM Reference UI – DTIM Admin.

- Add new ITIM node (manually)
  - Also needs installation of DTIM node artifacts.



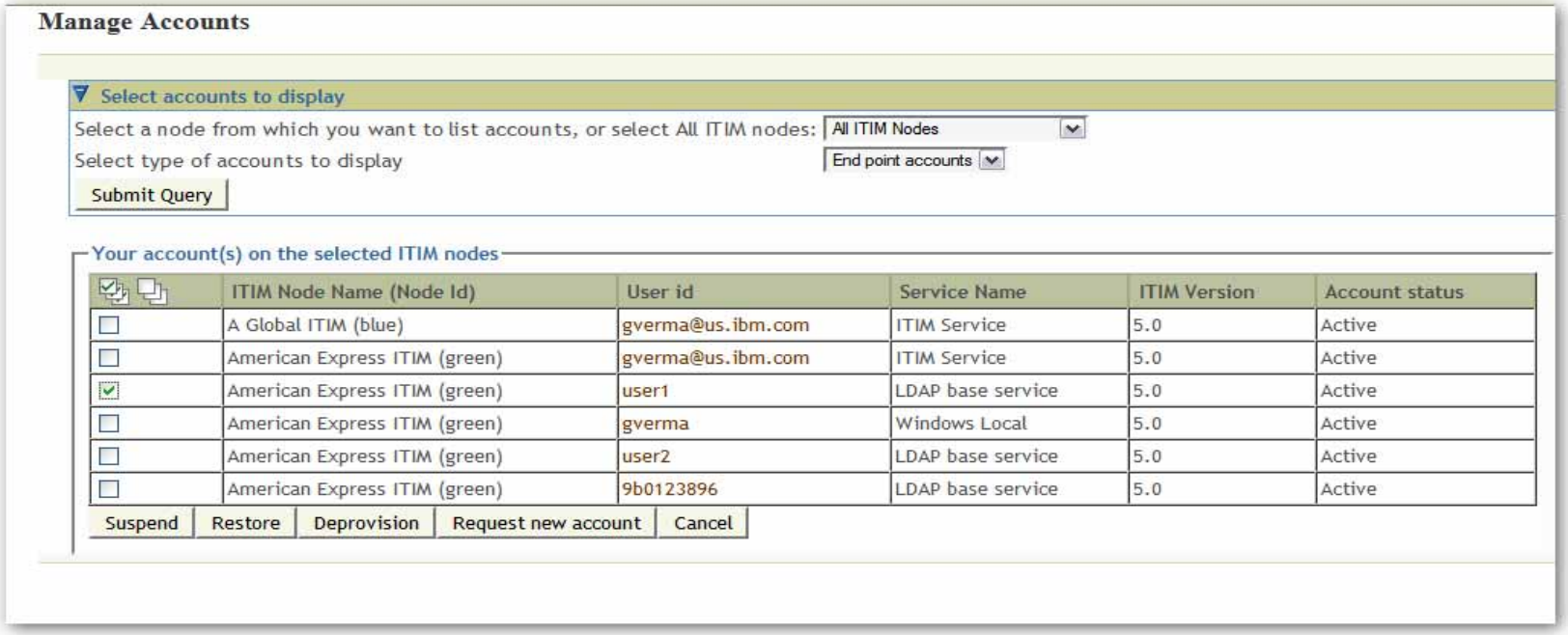
DTIM Reference UI – My Personal Info

- Retrieved from Main ITIM
  - Rendered using the ITIM form definition.
  - ITIM form changes are reflected in real time.



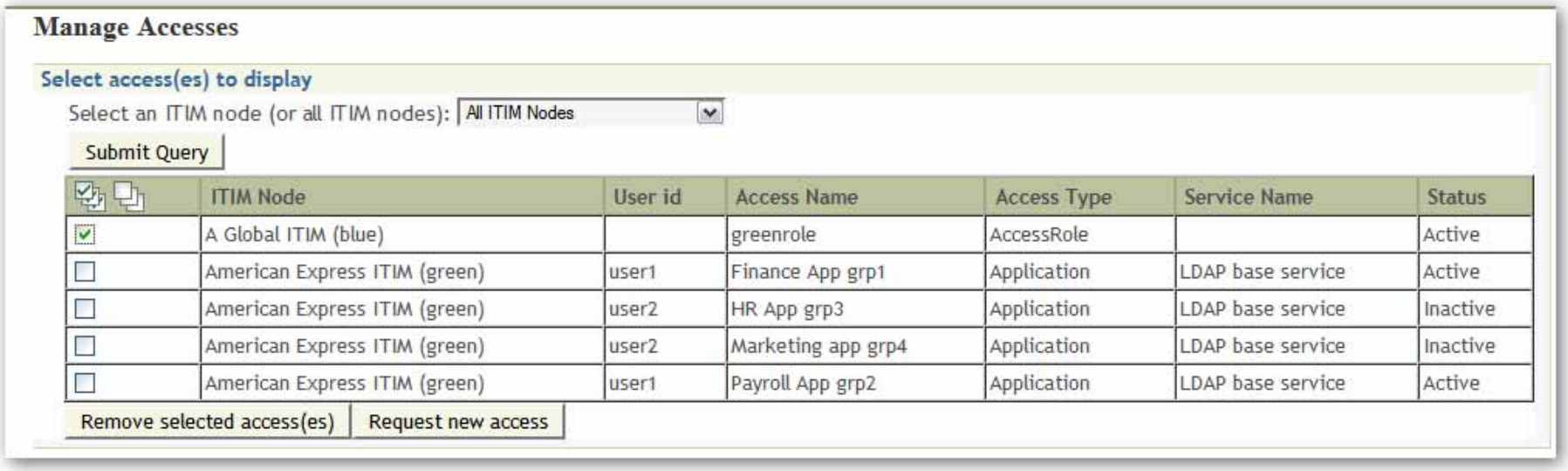
DTIM Reference UI – My Accounts

- Accounts can be selected from one or all participating ITIM nodes



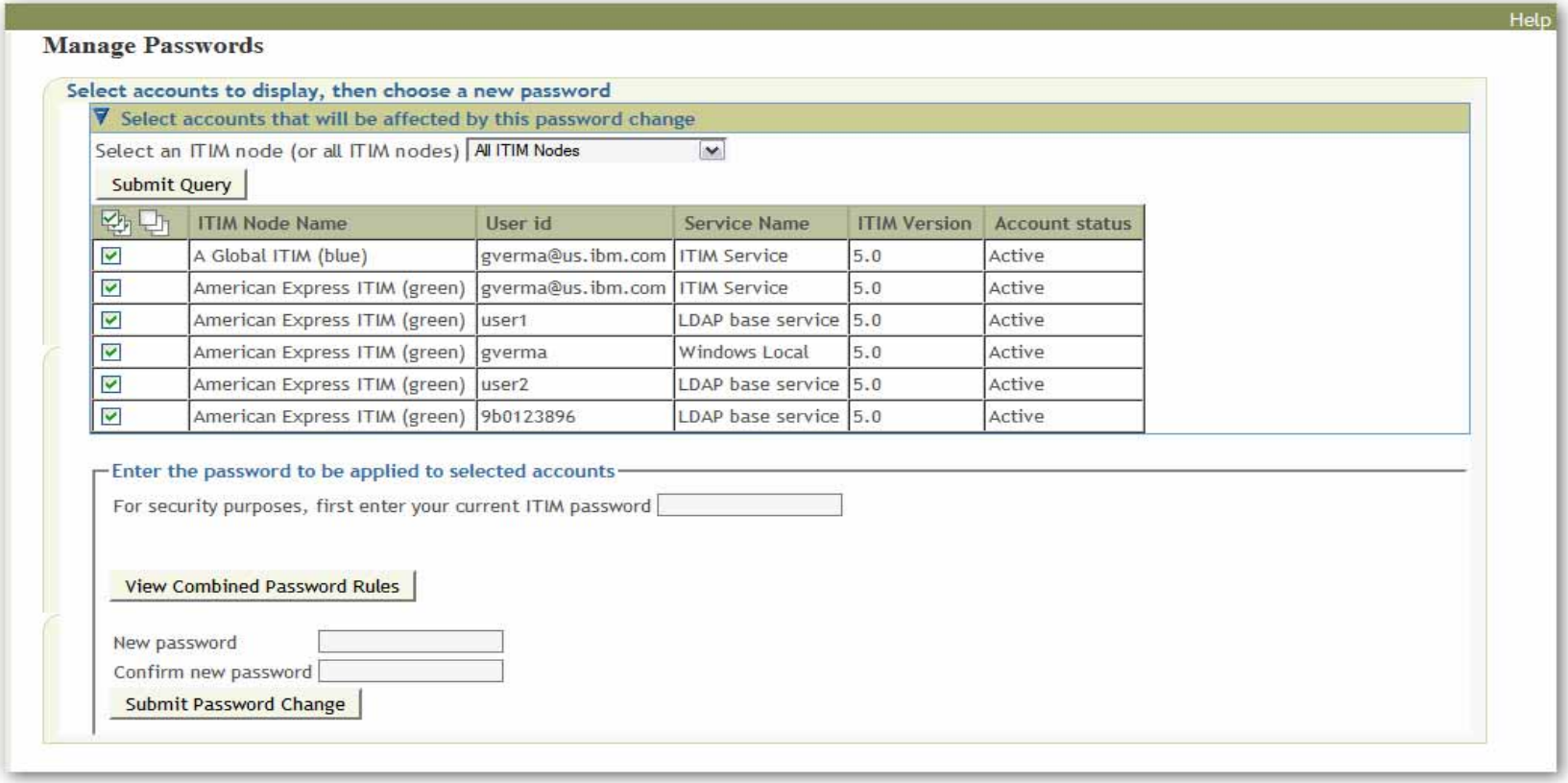
DTIM Reference UI – My Accesses

- Retrieved only from ITIM 5.x nodes
  - Supports provisioning of new accesses.



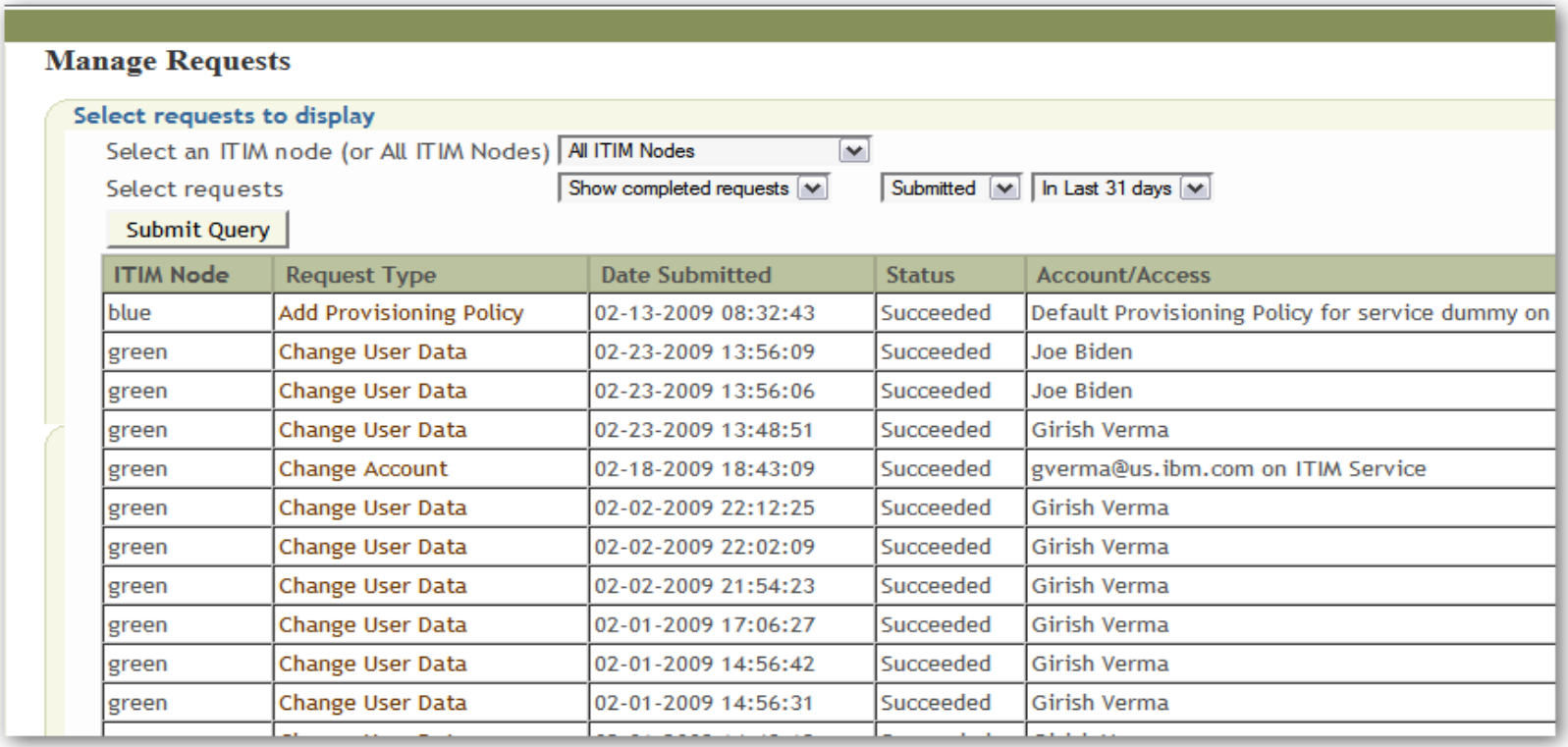
DTIM Reference UI – Manage Passwords

- Submits password changes across ITIM nodes.
- Merges password rules across ITIM nodes.



DTIM Reference UI – My Requests

- Retrieves Completed or Pending Requests.
- Pending requests can be aborted across ITIM nodes.



Manage Requests

Select requests to display

Select an ITIM node (or All ITIM Nodes) [All ITIM Nodes]

Select requests [Show completed requests] [Submitted] [In Last 31 days]

Submit Query

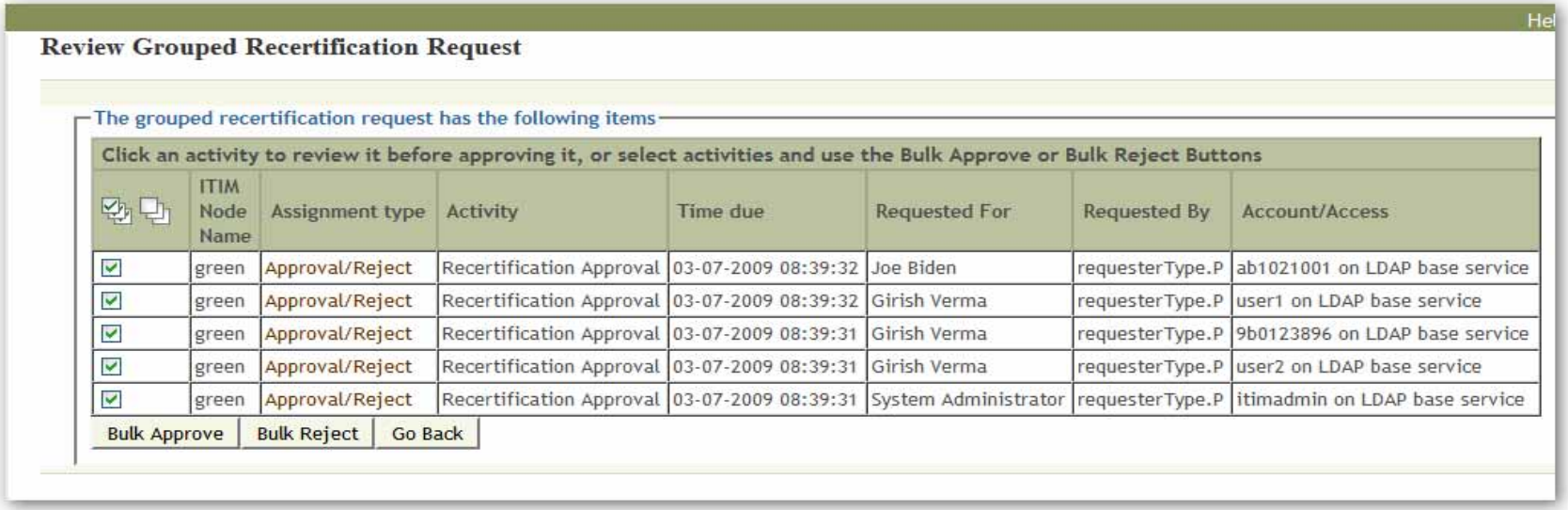
ITIM Node	Request Type	Date Submitted	Status	Account/Access
blue	Add Provisioning Policy	02-13-2009 08:32:43	Succeeded	Default Provisioning Policy for service dummy on
green	Change User Data	02-23-2009 13:56:09	Succeeded	Joe Biden
green	Change User Data	02-23-2009 13:56:06	Succeeded	Joe Biden
green	Change User Data	02-23-2009 13:48:51	Succeeded	Girish Verma
green	Change Account	02-18-2009 18:43:09	Succeeded	gverma@us.ibm.com on ITIM Service
green	Change User Data	02-02-2009 22:12:25	Succeeded	Girish Verma
green	Change User Data	02-02-2009 22:02:09	Succeeded	Girish Verma
green	Change User Data	02-02-2009 21:54:23	Succeeded	Girish Verma
green	Change User Data	02-01-2009 17:06:27	Succeeded	Girish Verma
green	Change User Data	02-01-2009 14:56:42	Succeeded	Girish Verma
green	Change User Data	02-01-2009 14:56:31	Succeeded	Girish Verma

DTIM Reference UI – My To Do Items

- Provides individual as well as grouped items.
- Provides Bulk Approve / Reject across ITIM nodes.

DTIM Reference UI – My To Do Items

- Example of Bulk Approve / Reject



Review Grouped Recertification Request

The grouped recertification request has the following items

Click an activity to review it before approving it, or select activities and use the Bulk Approve or Bulk Reject Buttons

	ITIM Node Name	Assignment type	Activity	Time due	Requested For	Requested By	Account/Access
☑	green	Approval/Reject	Recertification Approval	03-07-2009 08:39:32	Joe Biden	requesterType.P	ab1021001 on LDAP base service
☑	green	Approval/Reject	Recertification Approval	03-07-2009 08:39:32	Girish Verma	requesterType.P	user1 on LDAP base service
☑	green	Approval/Reject	Recertification Approval	03-07-2009 08:39:31	Girish Verma	requesterType.P	9b0123896 on LDAP base service
☑	green	Approval/Reject	Recertification Approval	03-07-2009 08:39:31	Girish Verma	requesterType.P	user2 on LDAP base service
☑	green	Approval/Reject	Recertification Approval	03-07-2009 08:39:31	System Administrator	requesterType.P	itimadmin on LDAP base service

Bulk Approve | Bulk Reject | Go Back

DTIM API

- API is Java based.
- Provides a topology transparent API.
  - Client does not need to be aware of the ITIM node distribution.
- Provides self service functions.
- Can be used by a custom UI or a Java client application to communicate with DTIM.
- The Reference UI is an example of an application using the DTIM API.

DTIM API - Categories

- DTIM API categories
  - AccountServiceProvider
  - AuthenticationProvider
  - FormProvider
  - OrgContainerServiceProvider
  - PersonServiceProvider
  - RequestServiceProvider
  - RoleServiceProvider
  - ServiceServiceProvider
  - SystemUserProvider
  - ToDoProvider

DTIM API

- Selected examples of using the DTIM API.
- The authentication API provides an AuthInfo object that is used in subsequent API calls.
  - Example – Log in using DTIM API.
    - Client does not need to be aware of DTIM topology or configuration.

```
ITIMCredential itimCred = new
        ITIMCredential("jdoe","abc");
AuthInfo authInfo = new AuthInfo(itimCred);
try {
        authInfo.authenticate();
        // authInfo now contains a valid session unless
        // a login exception was thrown.
```

DTIM API (Continued)

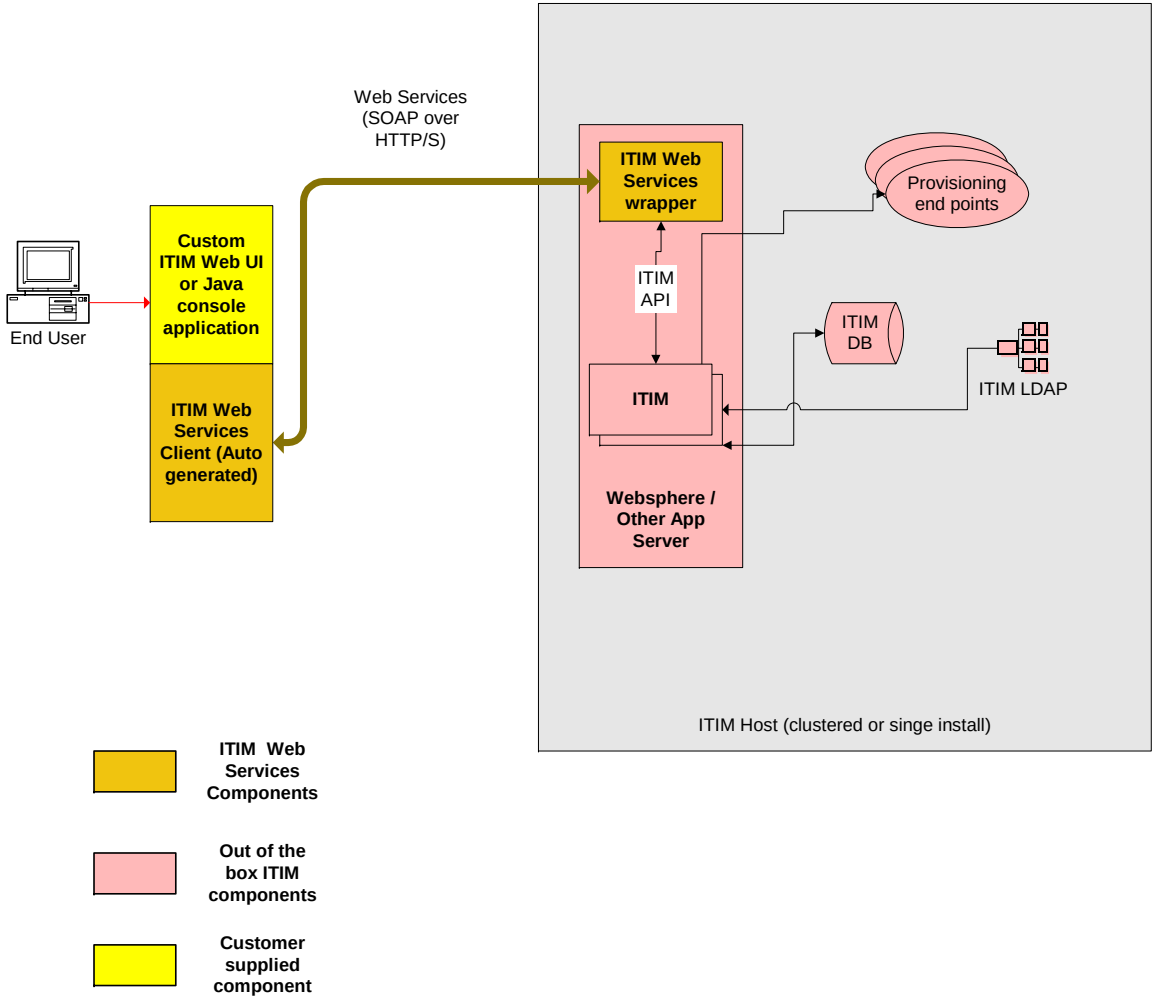
- Example – Get a list of To Do items from all participating ITIM nodes

```
// authInfo contains authenticated AuthInfo object.
ToDoProvider toDoProvider = new ToDoProvider(authInfo);
String selectedNode = “*”;  // Select all ITIM nodes
Collection assignments = toDoProvider.getToDoList(selectedNode);
```

ITIM Web Services Overview

- Lightweight SOAP protocol over HTTP/S to talk to ITIM.
  - Eliminates the need for ITIM or Websphere jars on the client.
- Non Java clients can talk to ITIM.
  - .NET clients can talk to ITIM.
- ITIM API complexity is abstracted by the web services functional API.
- Runs as a web application co-located with ITIM.

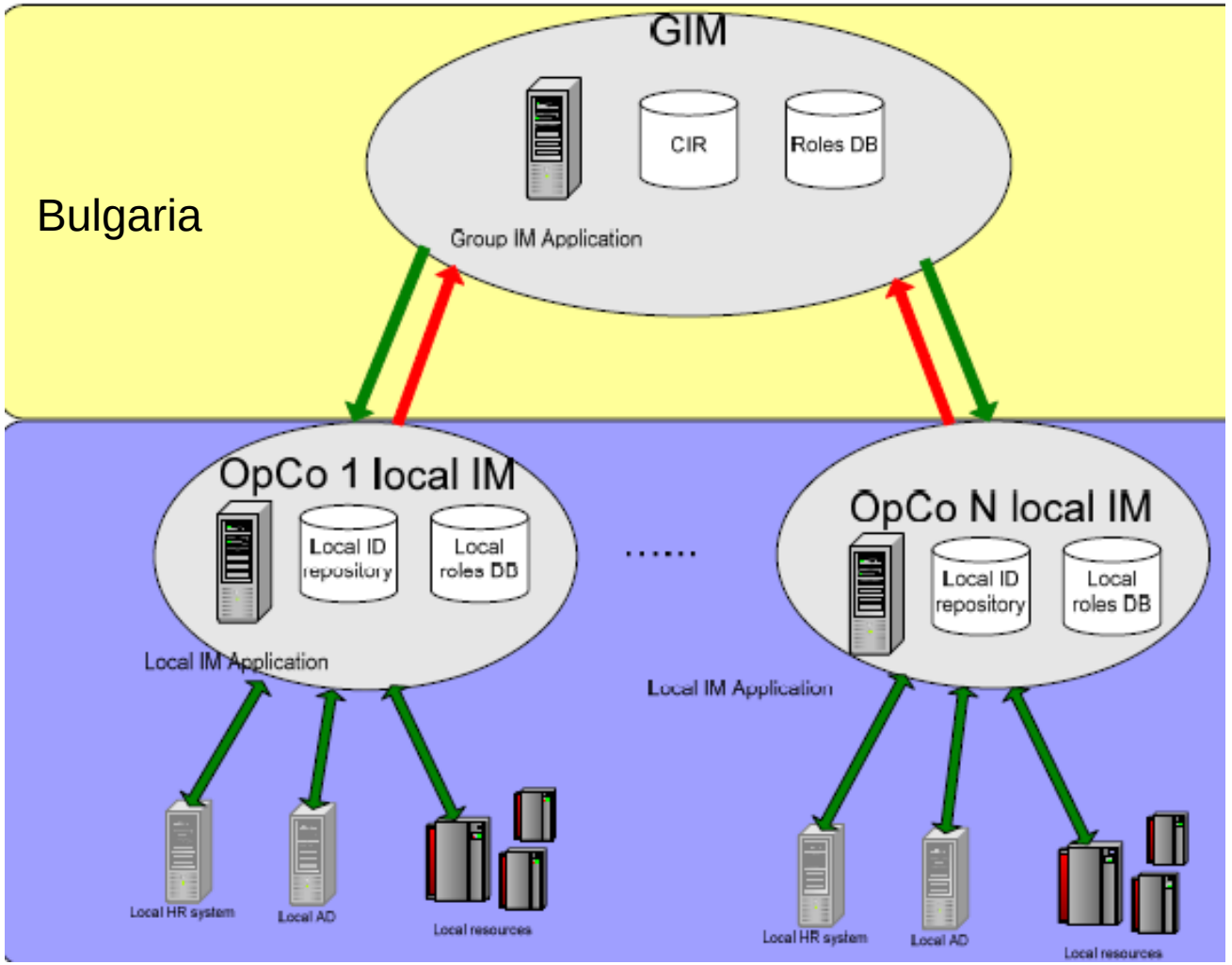
ITIM Web Service Architecture



ITIM Web Services Suite

- The web services suite is made of functionally separate services
  - Session Service
  - Account Service
  - Form Service
  - Organizational Container Service
  - Password Service
  - Request Service
  - Role Service
  - Service Service
  - Search Data Service
  - System User Service
  - To Do Service
  - Access Service (ITIM 5.x Only)
- DTIM Solution Leverages WebServices Interfaces

Example Implementation – Telco based out of Bulgaria



Austria

Slovenia

Questions?
Thank You

Trademarks and disclaimers

Why DTIM?

- Why is it needed?
  - To distribute workload in implementations with hundreds or thousands of services.
  - To provide progressive scaling using a distributed services model.
  - To provide enterprises with a rolling upgrade & maintenance window without bringing down ITIM completely.
  - To allow localized administration and centralized identity / account operations.
    - ITIM nodes can be geographically separate.
  - Can provide one off provisioning for a subset of population (say an isolated Dept or a lab).

ITIM Web Services example

- Example of using the web services to authenticate to ITIM and get user's person object.

```
ITIMWebServiceFactory factory = new ITIMWebServiceFactory("http://host/ITIMWebServices");
WSSession session = factory.getWSSessionService().login(userid, password);
WSPersonService personService = factory.getWSPersonService();
WSPerson myPerson = personService.getPrincipalPerson(session);
```

ITIM Web Services Overview

- Runs as a web application co-located with ITIM.
- Simple client and data model
  - WSDL files can be used to generate a web services client and data model.
    - *WSDL can be used to generate non Java clients.*
  - A pre-generated Java client is included.
- Provides a threaded conversation by establishing a session id.

ITIM Web Service Factory

- The pre-generated web service client provides a web service factory (class).
  - The web service factory “publishes” each web service and eliminates the client’s need to determine the service address.
  - It can provide an instance of any of the ITIM Web Services. Obtaining a web service is functionally simple.
    - Example: webServiceFactory.getWSPaswordService().

DTIM Architecture