Business Process Driven Framework for defining an Access Control Service based on Roles and Rules

NISSC - October 2000

R. Chandramouli (Mouli)

(Security Division ITL - NIST)

Business Process Driven Framework for defining An Application-level Access Control Service (BPD-ACS) - *Outline*

- Building Blocks
- Drawbacks in Existing Approaches
- BPD-ACS Framework applied to a Hospital-based Laboratory Information System (HLIS).
- Other Potential Applications

Building Blocks for defining an Application-level Access Control Service

- Identify application-level operations (ACS-T1).
- Identify constraints on the exercise of those operations based on enterprise security policy requirements.
  Also Define User base and Profiles (ACS-T2)
- Model User-Operation association using an Access Control Model (ACS -T3).
- Implement mechanisms to enforce User-Operation constraints identified in T2 using the model (ACS -T4).

Drawbacks in Existing Approaches for Enforcing User-Operation Constraints

- Enforce User-Operation constraints through application logic. - MAINTABILITY BECOMES AN ISSUE

- Through a trigger procedure - CAN BE DONE ONLY IN LIMITED ENVIRONMENTS LIKE A DBMS.

- Parameterized Groups or Roles - MAKES ROLE DEFINITIONS AND ASSOCIATED PRIVILEGES TIGHTLY COUPLED.

Using BPD-ACS Framework for defining an Access Control Service for a Hospital Laboratory Information System (HLIS)

- Identify application-level operations **(BPD_ACS-T1).**
- Determine protection requirements for operations based on the Enterprise Security Policy **(BPD_ACS-T2).**
- Develop the RBAC Model for the application **(BPD_ACS -T3)**
- Formulating & Processing Access Decision Rules and associating them with Roles. **(BPD_ACS-T4).**

Identifying Application-level operations for HLIS using business-process analysis (BPD_ACS-T1)

<u>LIST OF BUSINESS PROCESSES SUPPORTED</u>

a. **Lab Order Entry**

b. Lab Test Scheduling

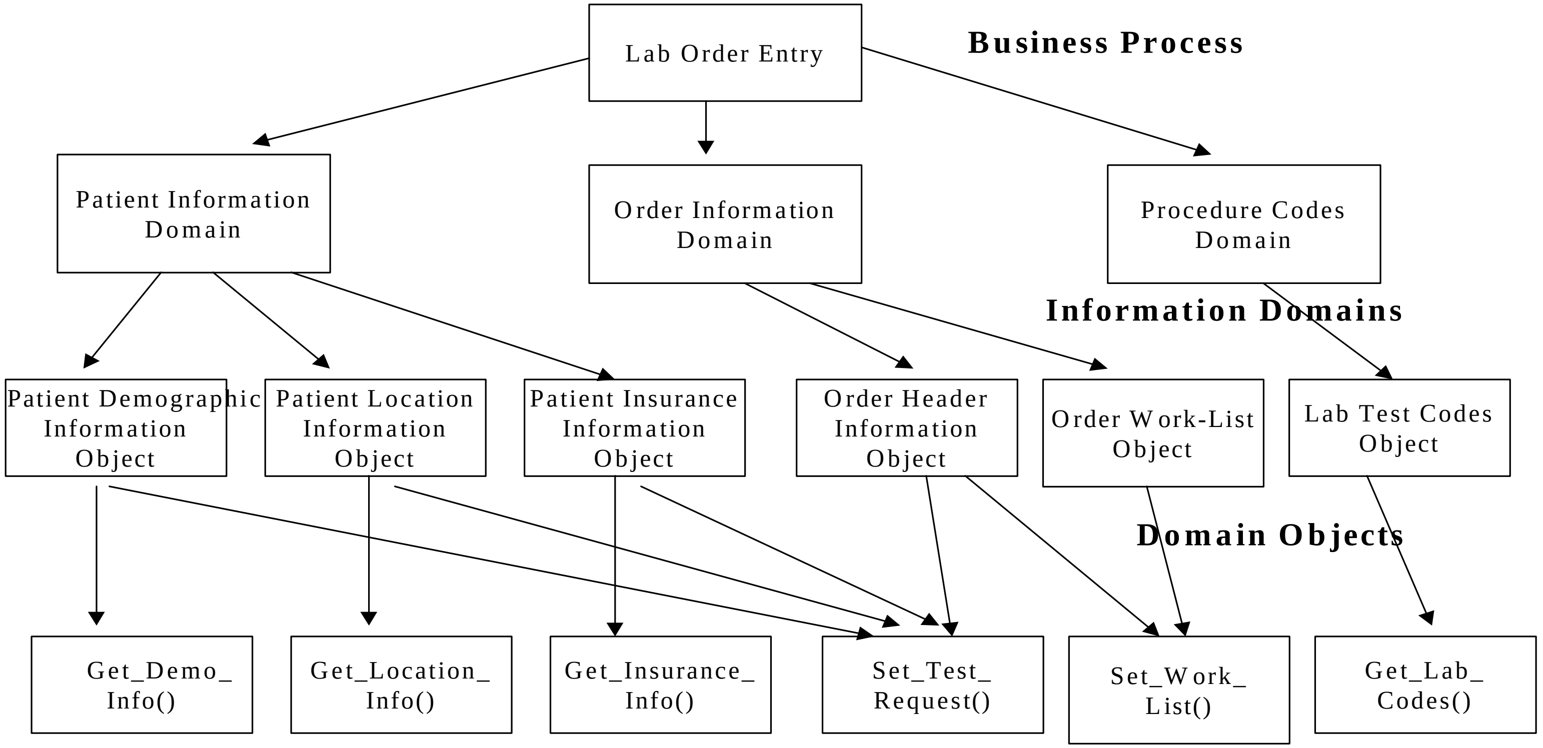
c. Capture and Recording of Test Results

d. Quality Control checks on Test Results

e. Generation of Summary Reports (if needed).

f. Retrieve/Access Test Results.

Identifying Application-level operations [LAB ORDER ENTRY] (BPD_ACS-T1 ..contd..)

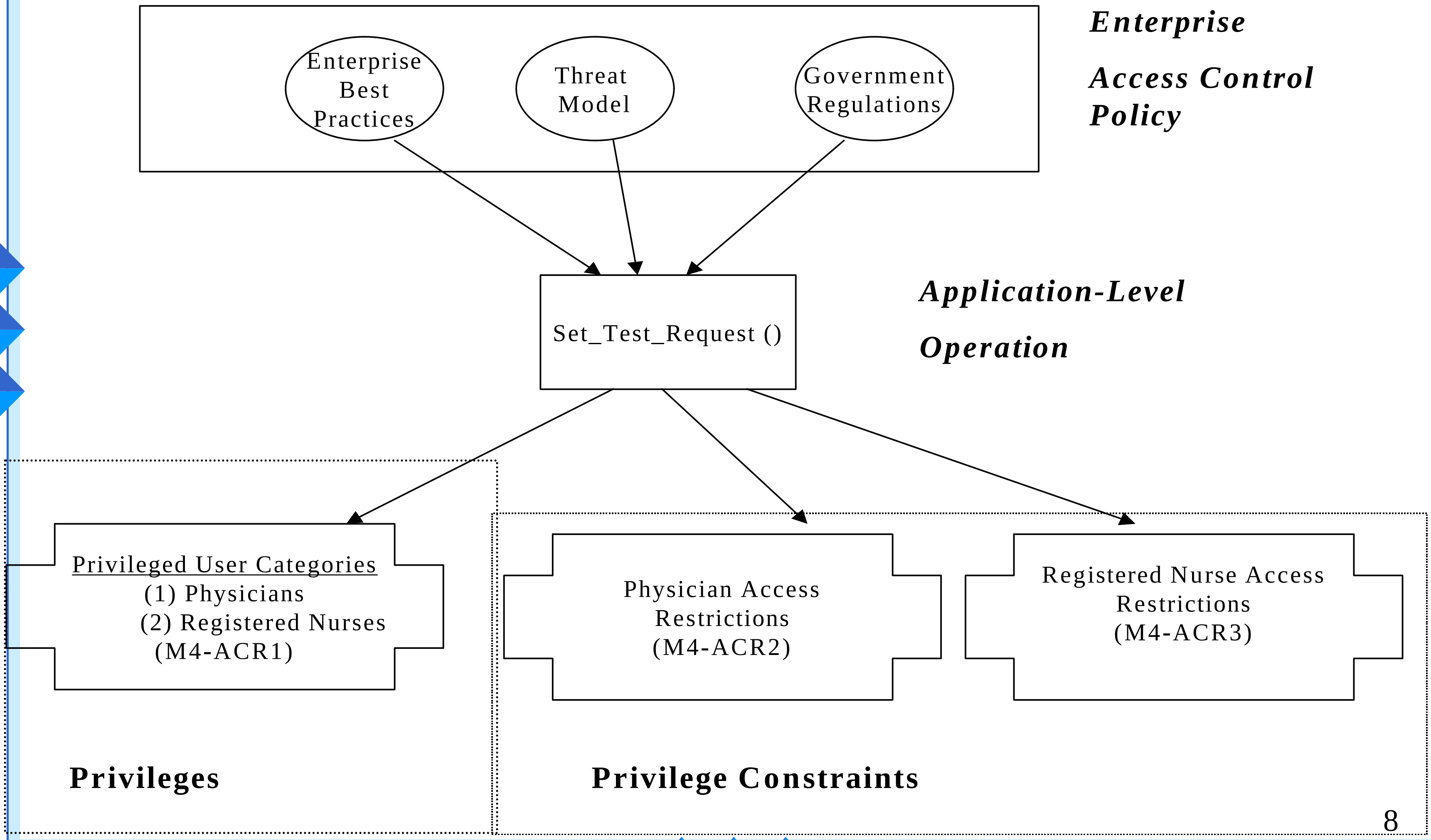


Application-Level

Operations (Methods)

Determine Protection Requirements

[SET_TEST_REQUEST] (BPD_ACS-T2)



Enterprise Best Practices

Threat Model

Government Regulations

Enterprise

Access Control Policy

Set_Test_Request ()

Application-Level

Operation

<u>Privileged User Categories</u>
(1) Physicians
(2) Registered Nurses
(M4-ACR1)

Physician Access Restrictions (M4-ACR2)

Registered Nurse Access Restrictions (M4-ACR3)

Privileges

Privilege Constraints

Developing the RBAC Model for modeling User-Operation Association in HLIS (BPD-ACS-T3)

Justification for using RBAC as the model

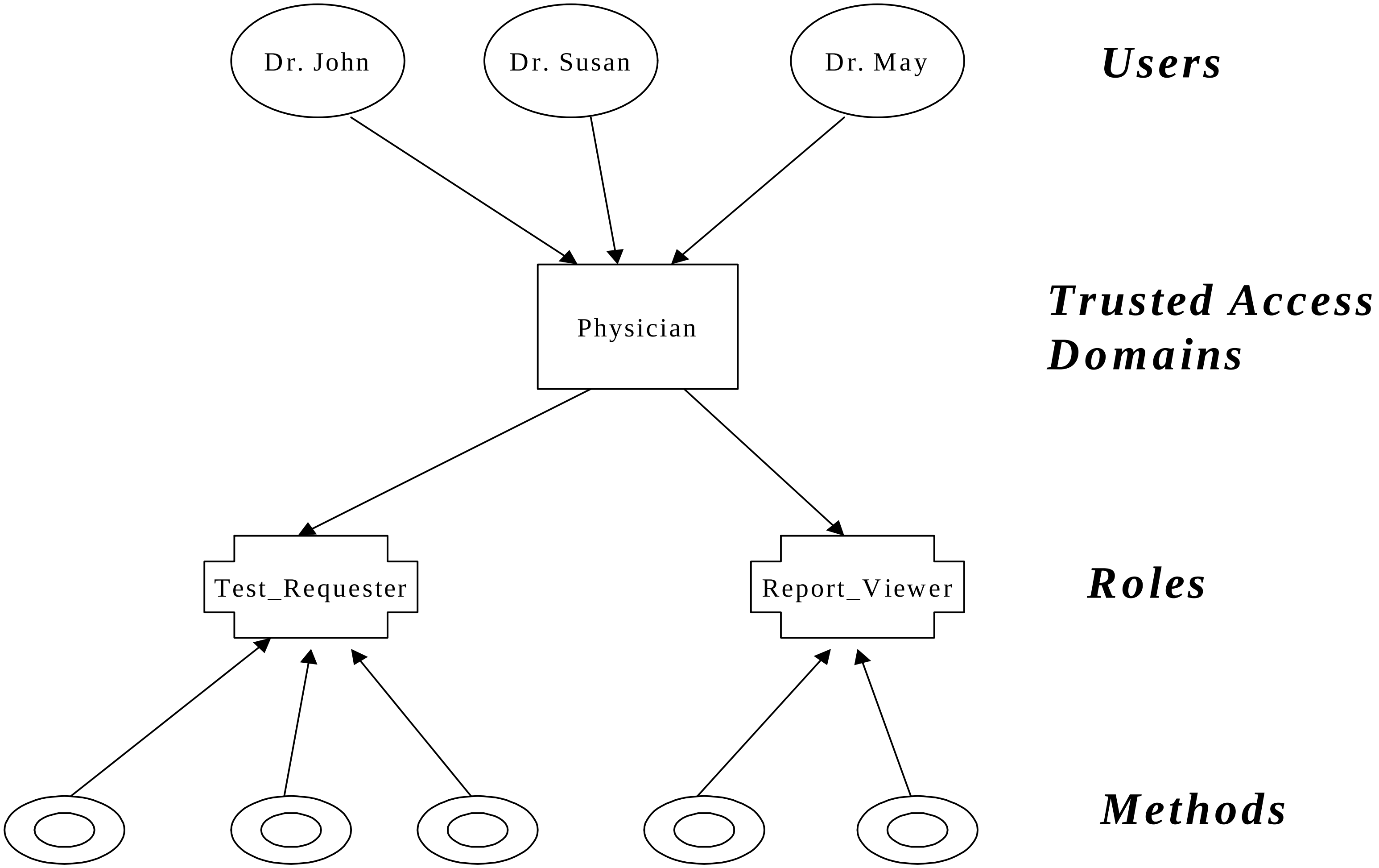
- Encapsulation mechanism for grouping privileges associated with a business process.
- _Simplified Privilege Management due to hierarchical relationships among roles.
- Availability on a number of platforms - DBMS,O/S..
- Taxonomy of Models with varying complexity

Developing the RBAC Model for HLIS (BPD-ACS-T3) .. contd

Mapping User Domains to Application Roles

Hospital Trusted Access Domains (TADs)	***HLIS Application Roles***
General Physician	Test_Requester, Report_Viewer
Speciality Physician	Test_Requester, Report_Viewer
Lab Supervisor	Test_Scheduler,Results_QC
Lab Technician	Test_Results_Generator
Registered Nurse	Test_Requester, Report_Viewer

Developing the RBAC Model (BPD-ACS-T3) .. contd



Defining Access Decision Rules

[Allow_Set_Test_Request] (BPD_ACS - T4)

Rule Name

Allow_Set_Test_Request

Access Request Attributes

PatientId: string

PhysicianId: string

AccessorId: string

Environmental Attributes

Accessor_Domain: string

Temporal Business Association Database Attributes

Table_Name: ATTENDING_CLINICIAN

Field_Names:

Patient_Identifier: string;

Physician_Identifier: string;

Auth_Nurse_Identifier: string;

Rule Predicate

PatientId == **:Patient_Identifier** &
((Accessor_Domain = “Physician” & PhysicianId == **:Physician_Identifier**) |
(Accessor_Domain = “Nurse” & AccessorId == **:Auth_Nurse_Identifier**))

Instantiating Access Decision Rules [Allow_Set_Test_Request] (BPD_ACS - T4) .. Contd..

Entries in Temporal Business Association Database

Patient_ Identifier	**Physician_ Identifier**	**Auth_Nurse _Identifier**
P102068	MD23456	RN8967

Truth Values for Rule Predicates are evaluated by instantiating these predicates by retrieving matching entries from Temporal Business Association Database.

Associating Rules with Roles (BPD_ACS-T4) .. Contd ..

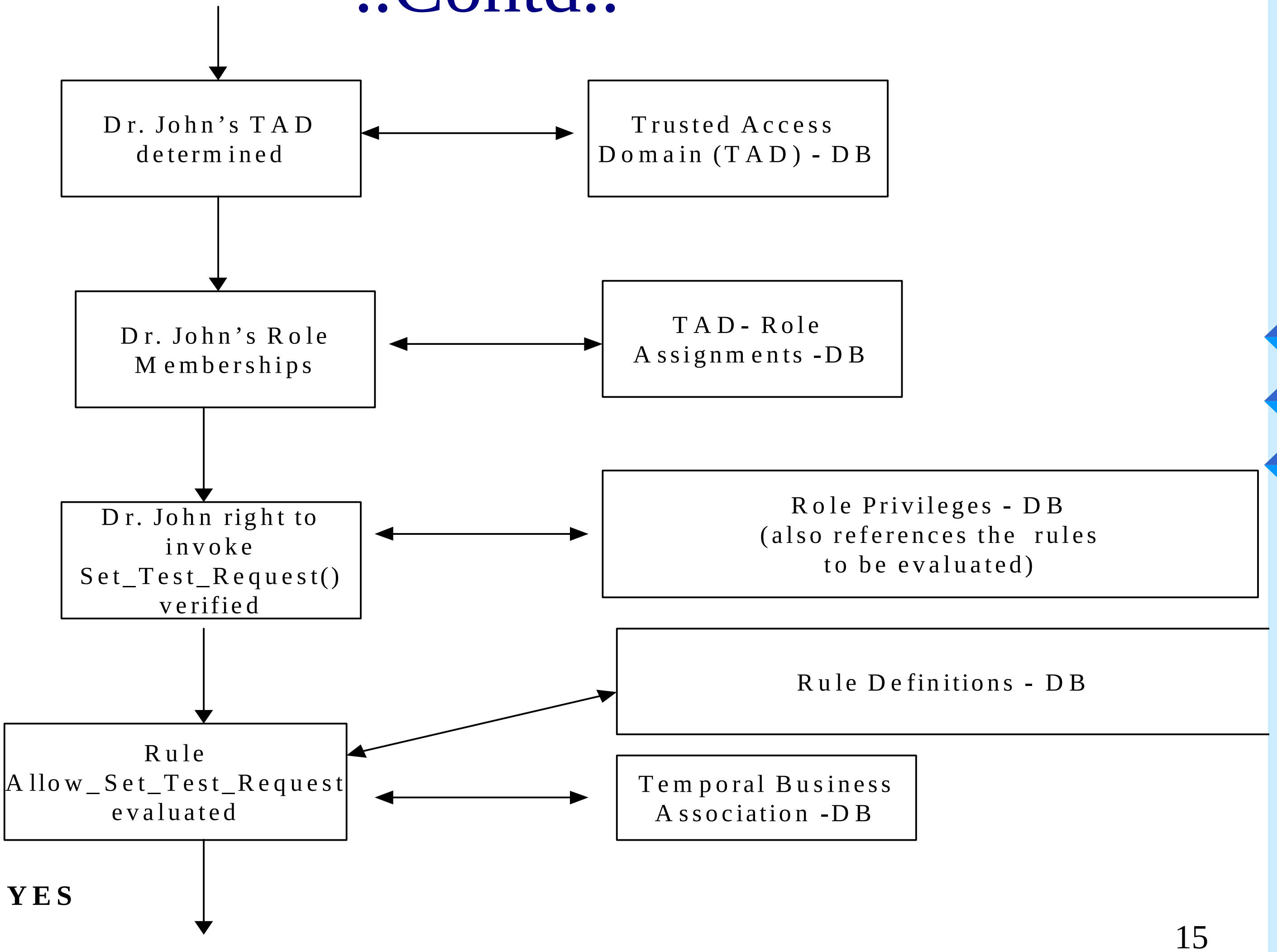
Role Name = “Test_Requester”
Role Memberships = <none> /* Here memberships means other roles – not users */
Privileges:

Privilege Name = Get_Demo_Info(PatientId,AccessorId)
Privilege Rules:
    Rule Name: Allow_Get_Demo_info
…………
Privilege Name = *Set_Test_Request (PatientId,PhysicianId,AccessorId)*
Privilege Rules:
    Rule Name: *Allow_Set_Test_Request*
…………
…………..

Access Decision Logic (BPD_ACS-T4) ..Contd..

John Logs in with the request

Set_Test_Req (DavidId, JohnId, JohnId)



Dr. John's TAD determined

Trusted Access Domain (TAD) - DB

Dr. John's Role Memberships

TAD- Role Assignments -DB

Dr. John right to invoke Set_Test_Request() verified

Role Privileges - DB (also references the rules to be evaluated)

Rule Definitions - DB

Rule Allow_Set_Test_Request evaluated

Temporal Business Association -DB

Allow Access = YES

Other Potential Applications

Where ever rights of Interacting Parties are determined based on occurrence of events and current state of relationships

- Extranet applications with relatively short period of business association/relationship.
- Web-based auction and bidding application