The OM-AM Framework and Role-Based Access Control

Prof. Ravi Sandhu

George Mason University

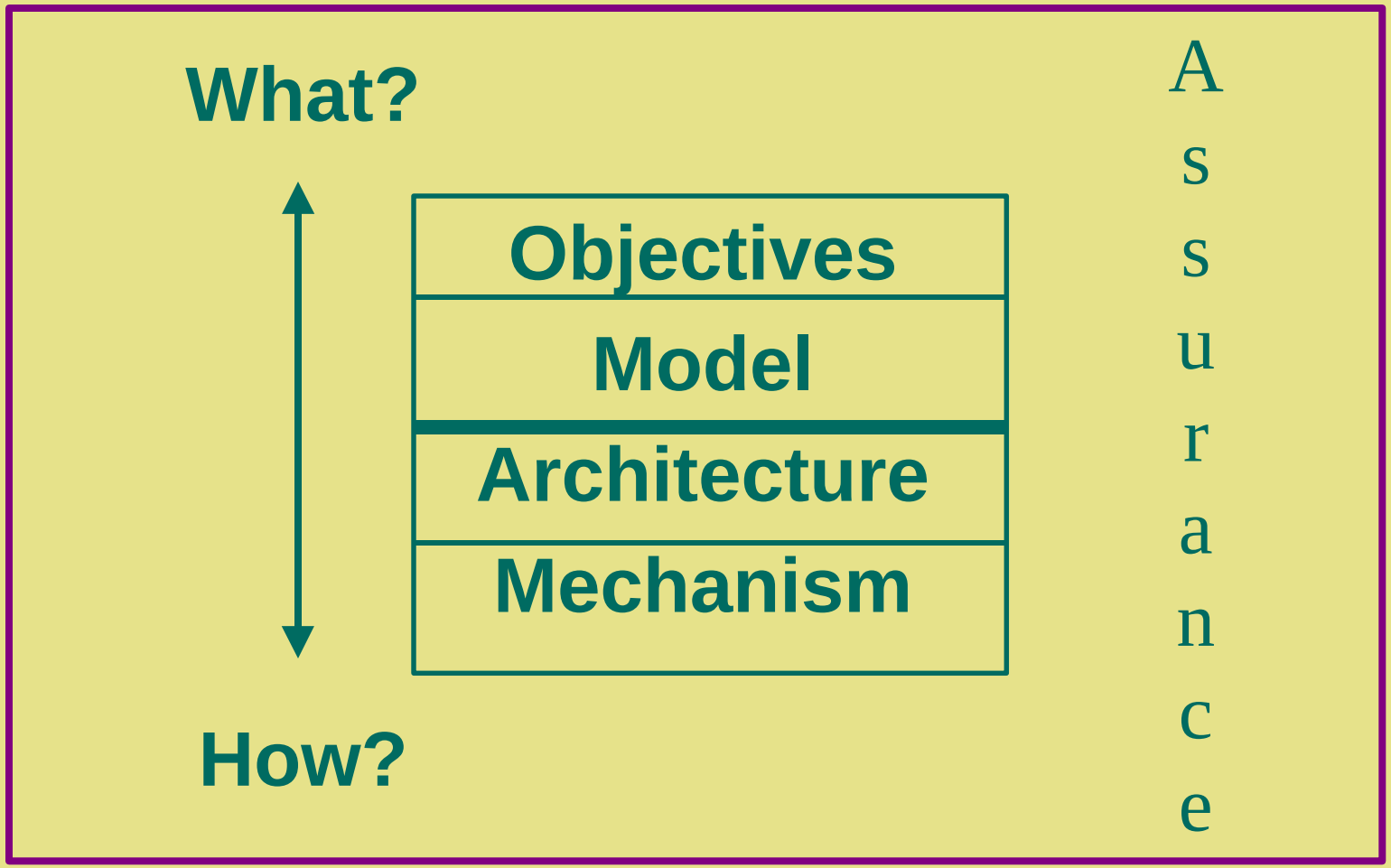
www.list.gmu.edu

AUTHORIZATION, TRUST AND RISK

- **Information security is fundamentally about managing**
  - **authorization and**
  - **trust**

  so as to manage risk

THE OM-AM WAY

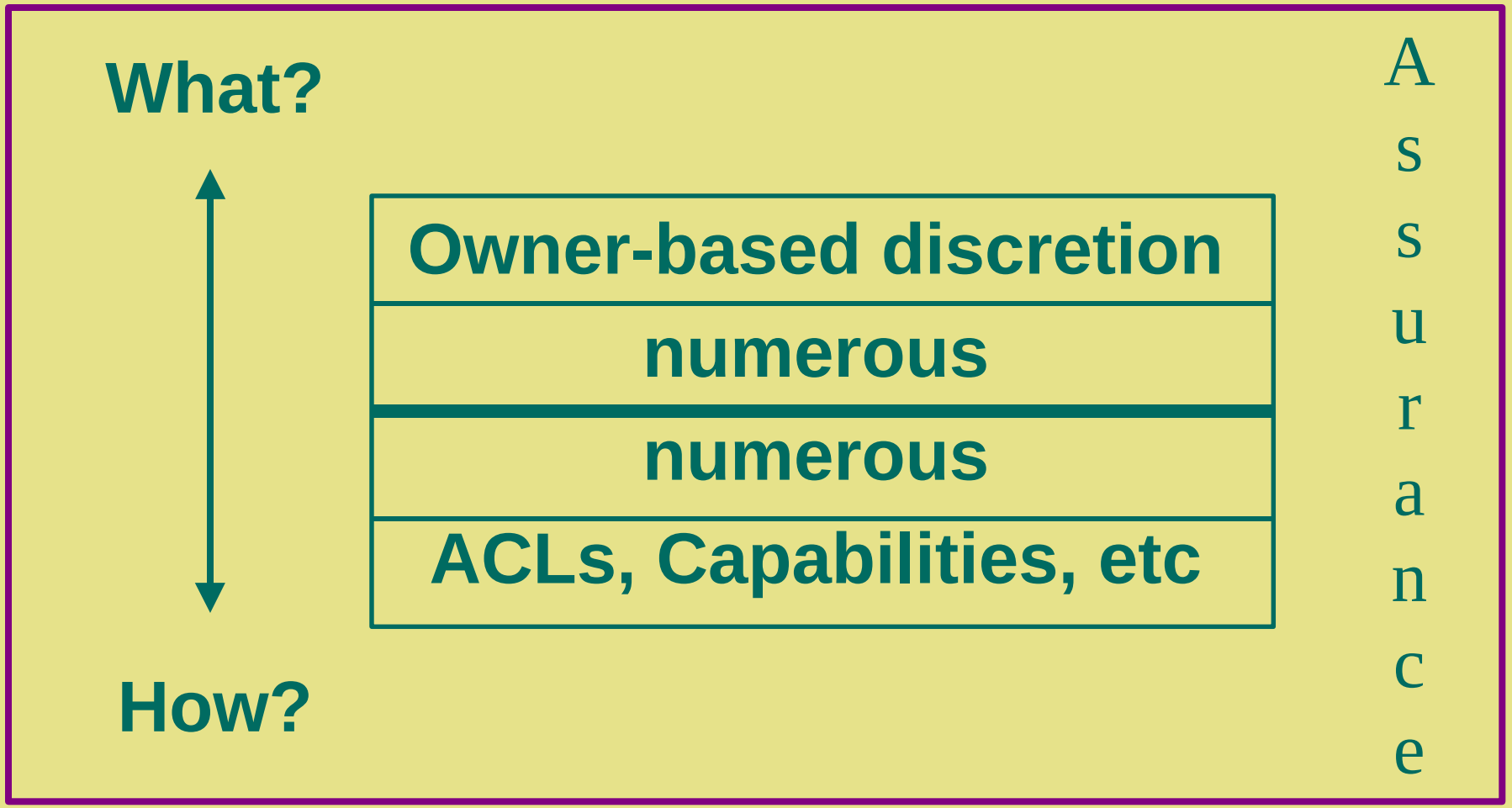


LAYERS AND LAYERS

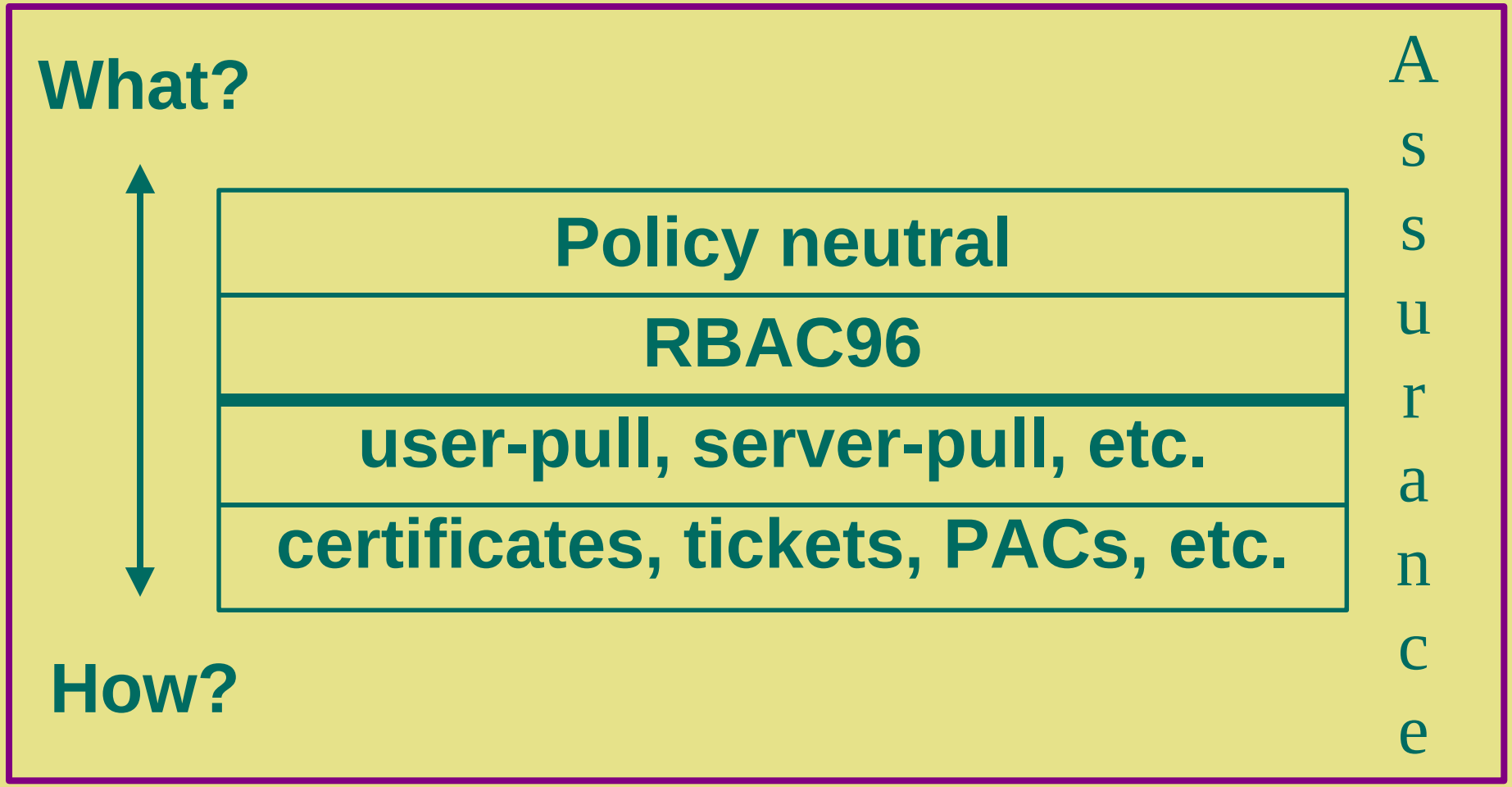
- **Multics rings**
- **Layered abstractions**
- **Waterfall model**
- **Network protocol stacks**
- **OM-AM**

OM-AM AND MANDATORY ACCESS CONTROL (MAC)

OM-AM AND DISCRETIONARY ACCESS CONTROL (DAC)



OM-AM AND ROLE-BASED ACCESS CONTROL (RBAC)



Role-Based Access Control
The RBAC96 Model

ROLE-BASED ACCESS CONTROL (RBAC)

- **A user's permissions are determined by the user's roles**
  - **rather than identity or clearance**
  - **roles can encode arbitrary attributes**
- **multi-faceted**
- **ranges from very simple to very sophisticated**

RBAC SECURITY PRINCIPLES

- **least privilege**
- **separation of duties**
- **separation of administration and access**
- **abstract operations**

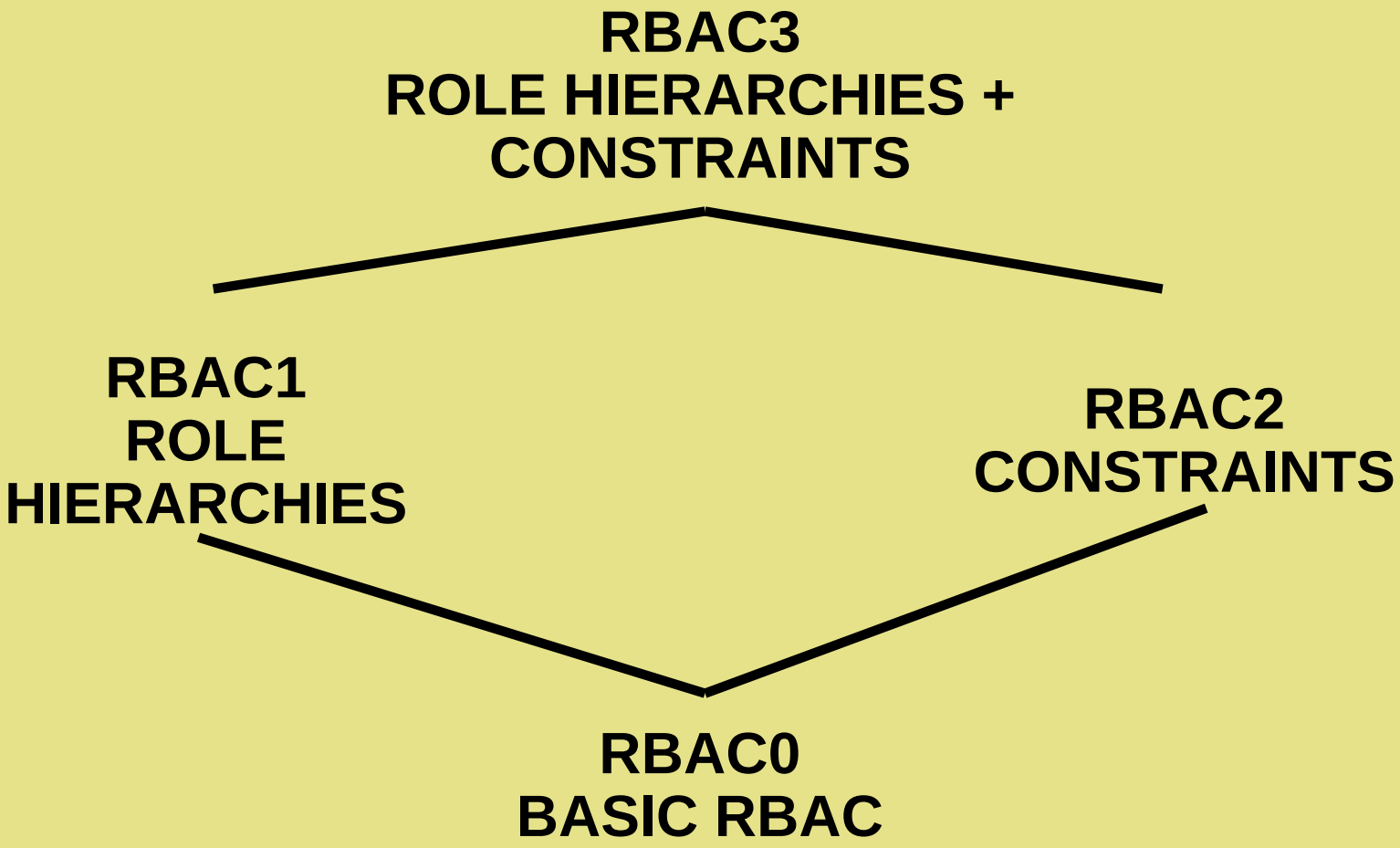
RBAC96
IEEE Computer Feb. 1996

- **Policy neutral**
- **can be configured to do MAC**
  - **roles simulate clearances (ESORICS 96)**
- **can be configured to do DAC**
  - **roles simulate identity (RBAC98)**

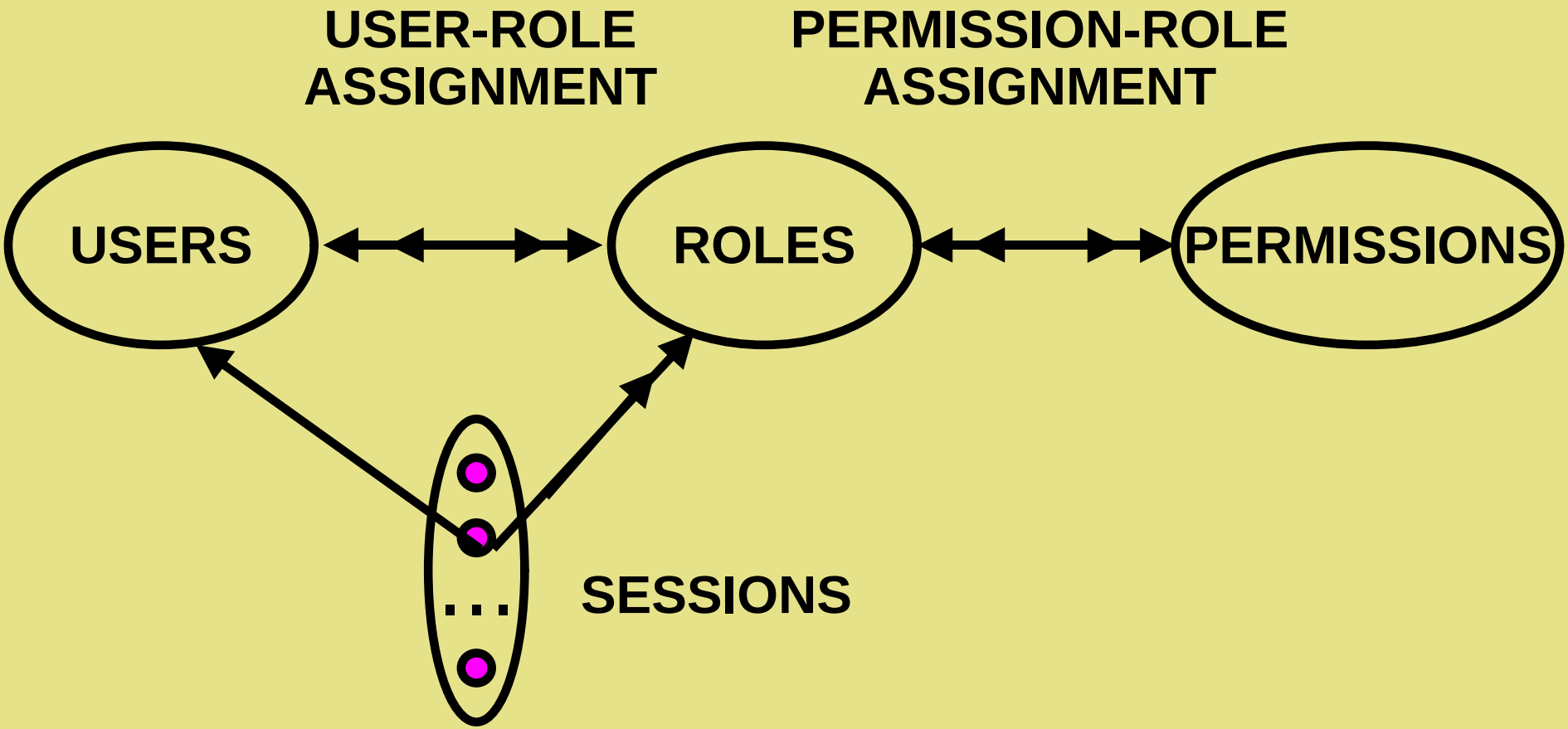
RBAC CONUNDRUM

- **turn on all roles all the time**
- **turn on one role only at a time**
- **turn on a user-specified subset of roles**

RBAC96 FAMILY OF MODELS



RBAC0



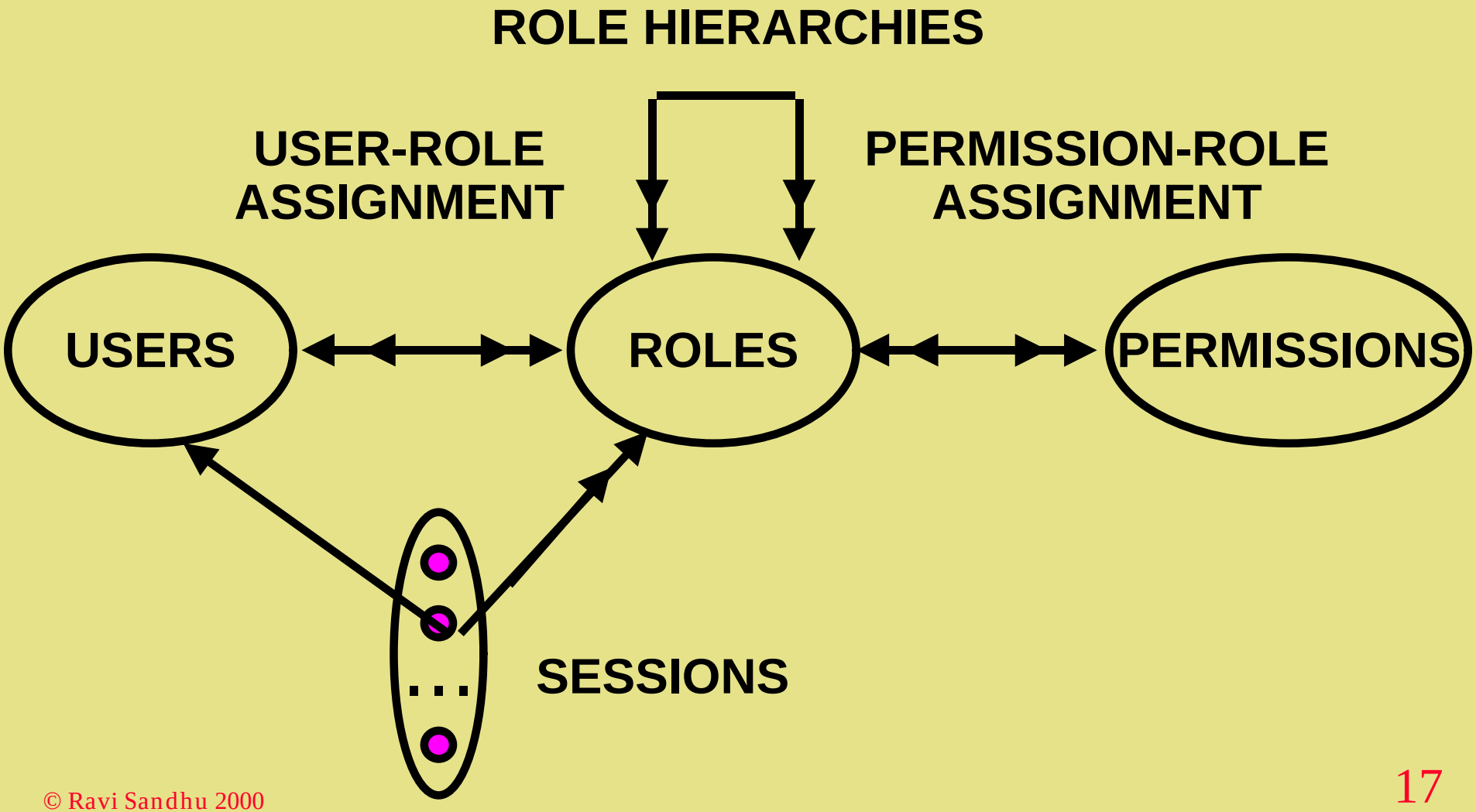
PERMISSIONS

- **Primitive permissions**
  - **read, write, append, execute**
- **Abstract permissions**
  - **credit, debit, inquiry**
- **System permissions**
  - **auditor, operator, back-up operator**

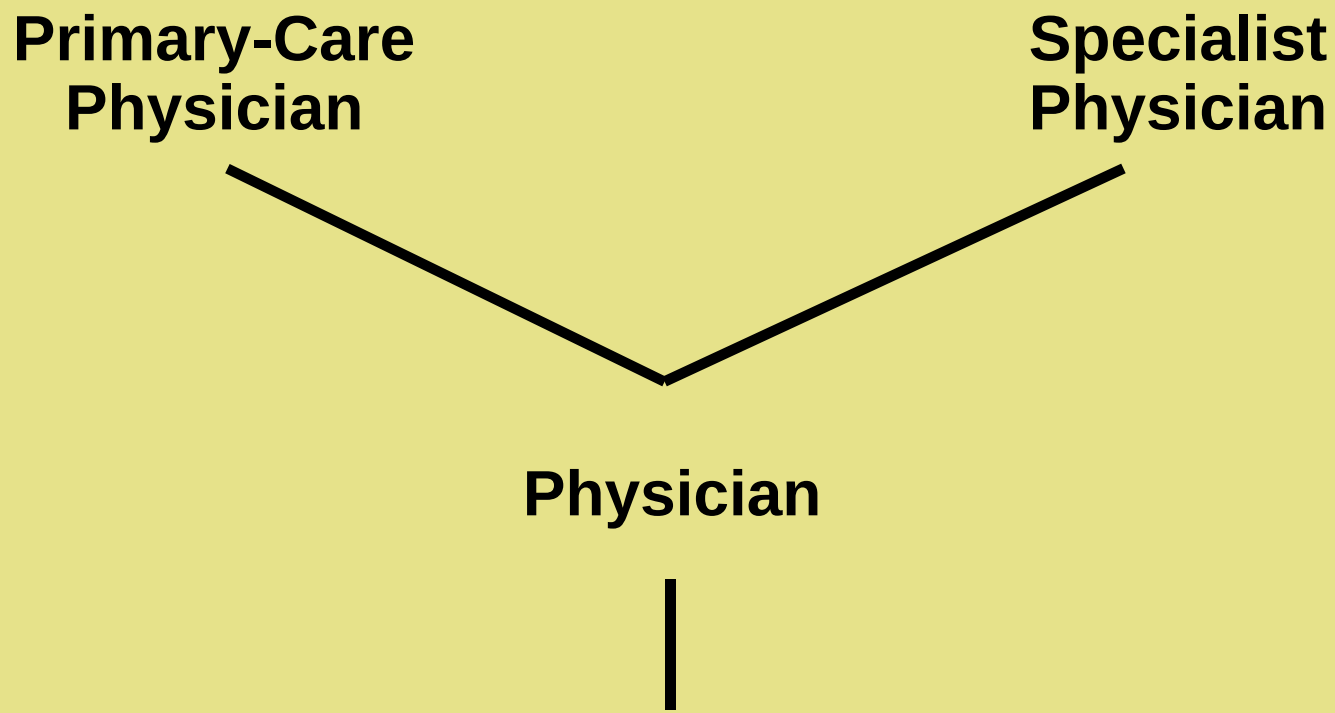
USERS

- **Users are**
  - **human beings or**
  - **other active agents**
- **Each individual should be known as exactly one user**

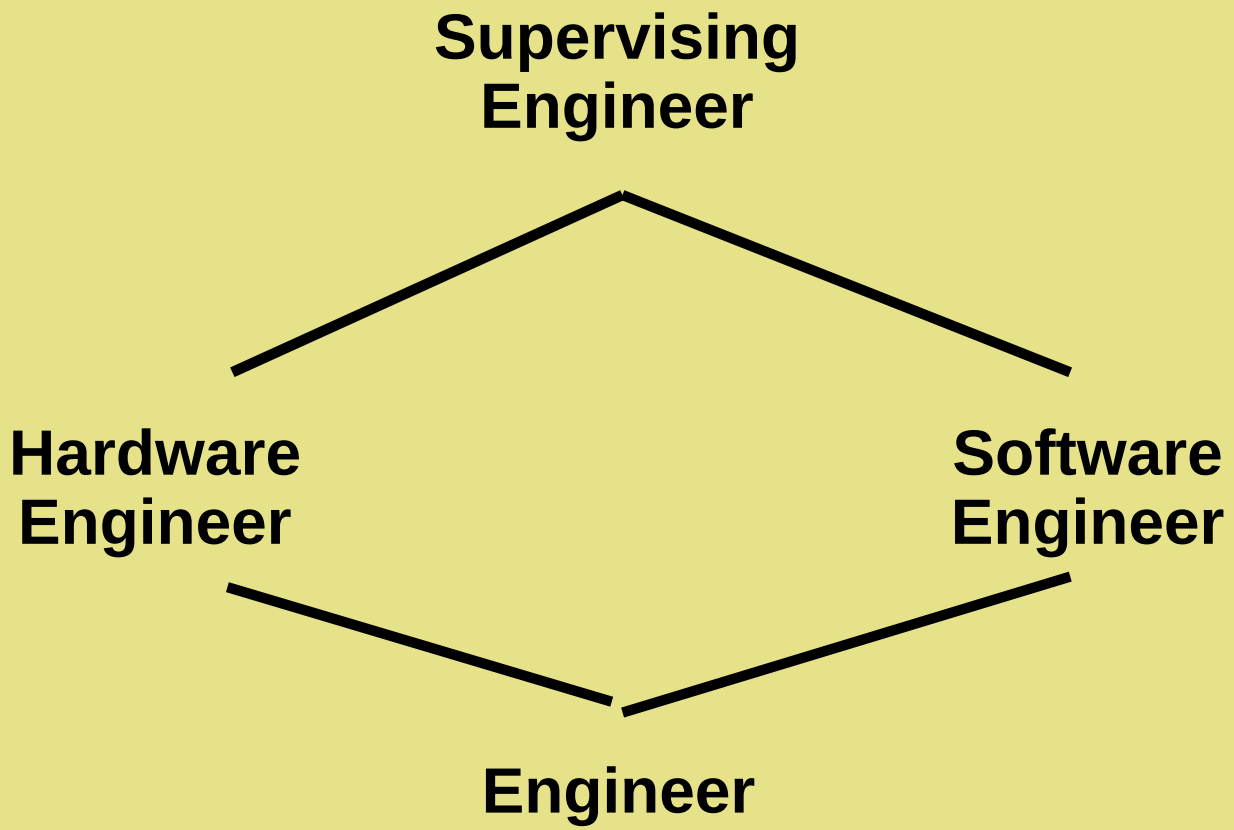
RBAC1



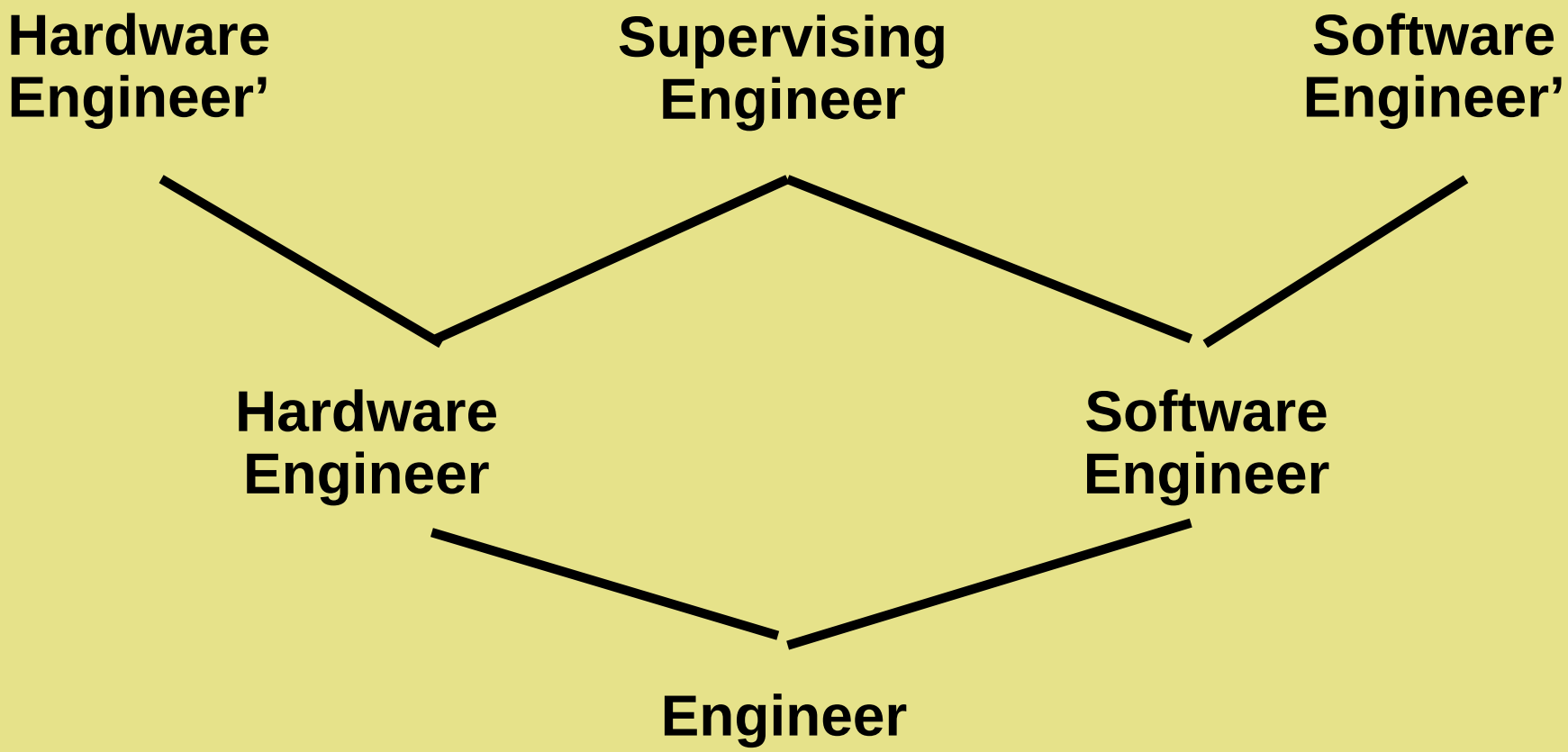
HIERARCHICAL ROLES



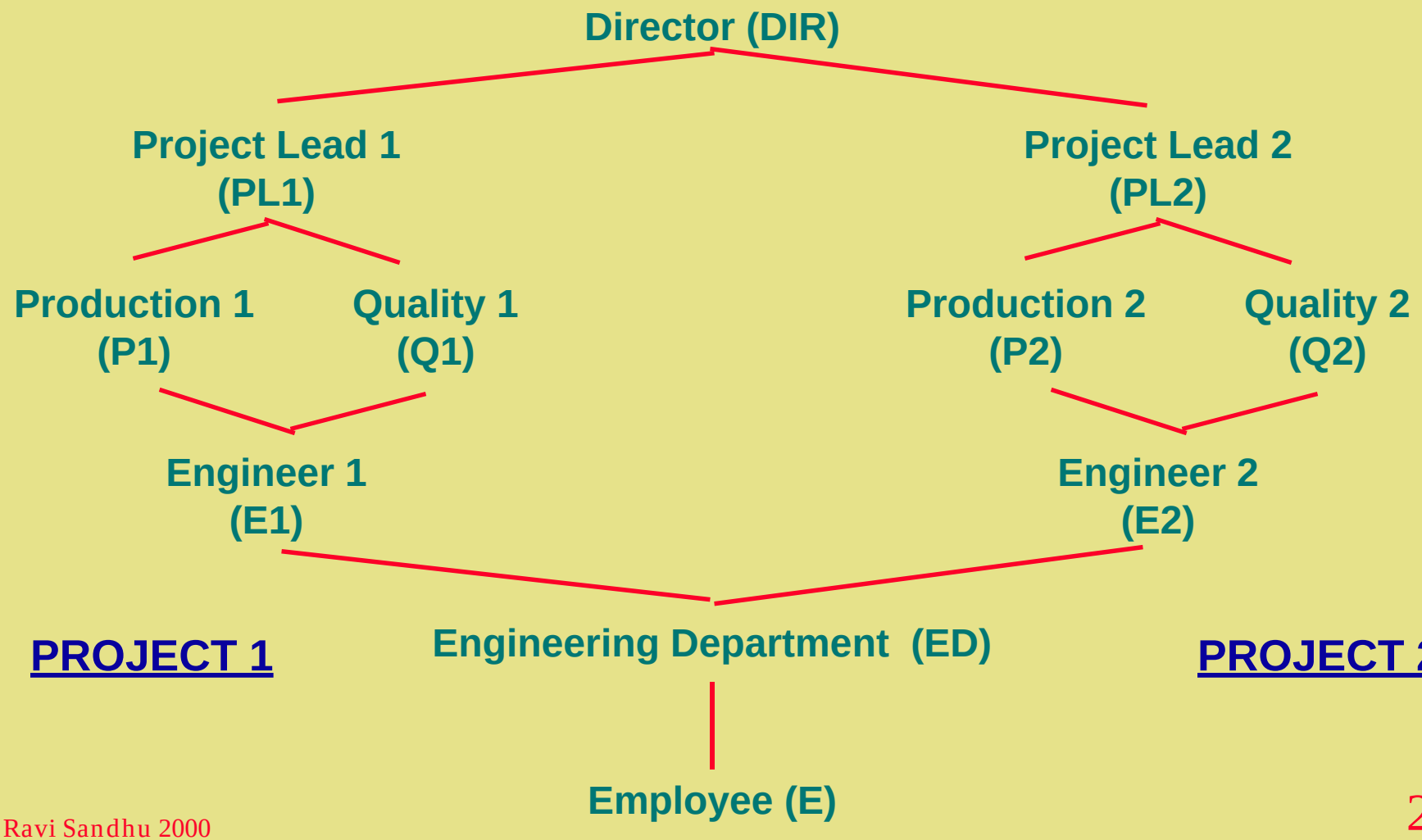
HIERARCHICAL ROLES



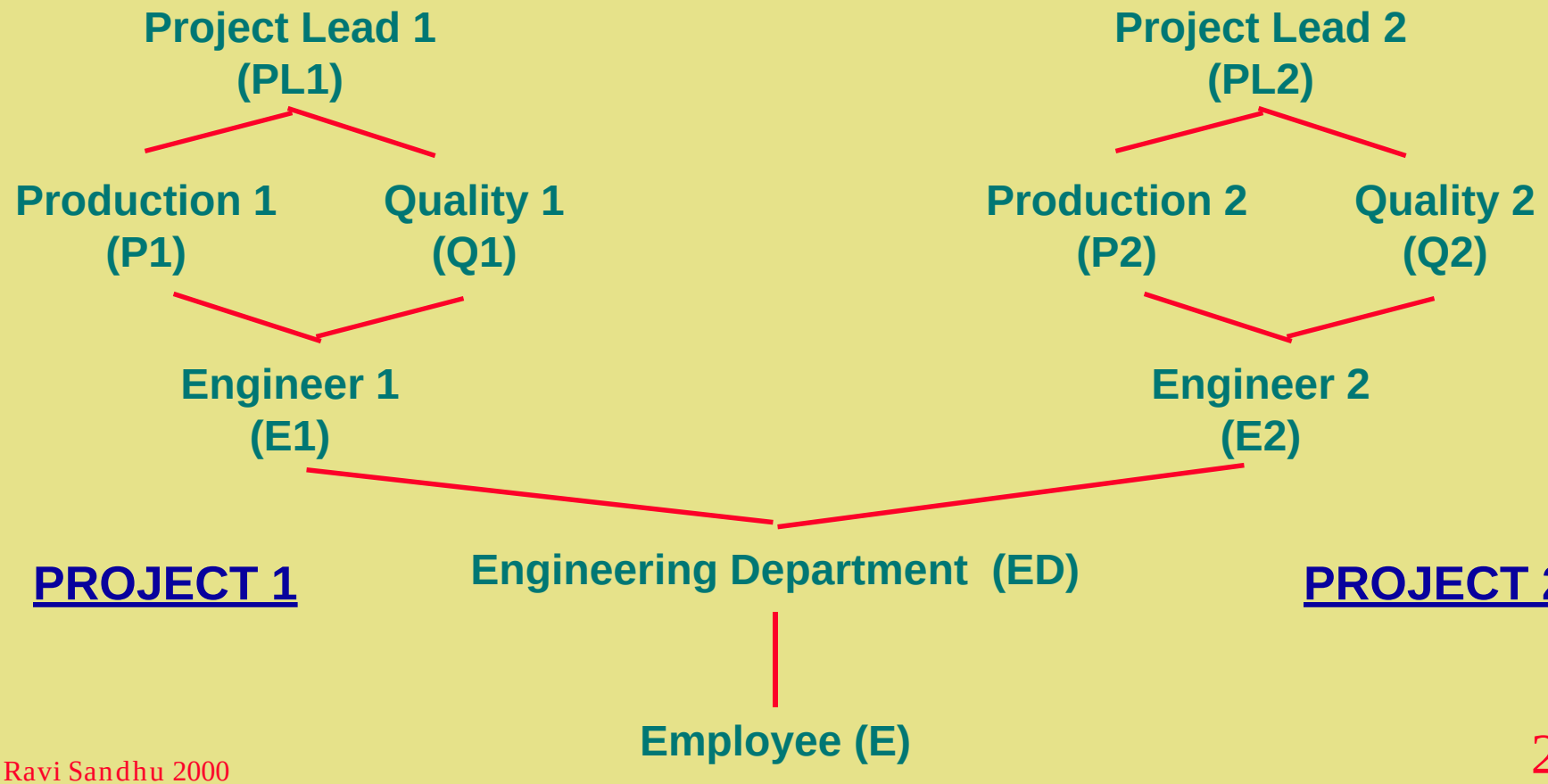
PRIVATE ROLES



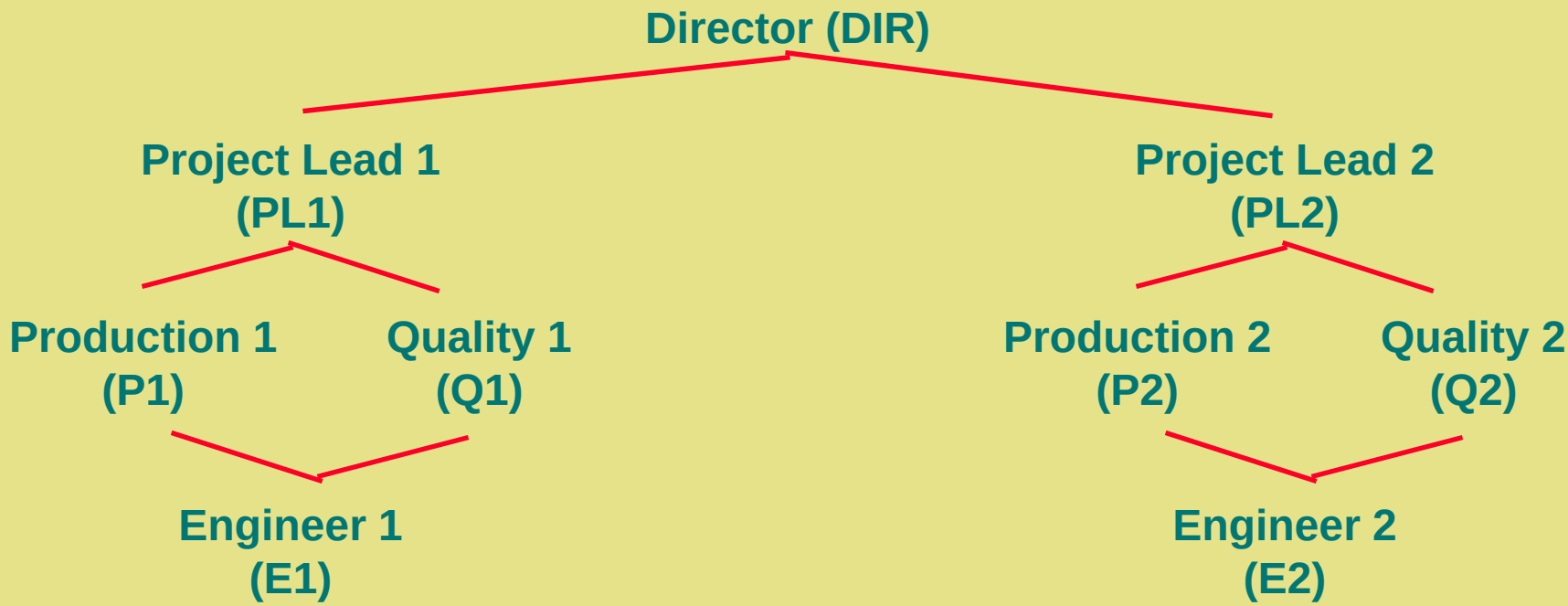
EXAMPLE ROLE HIERARCHY



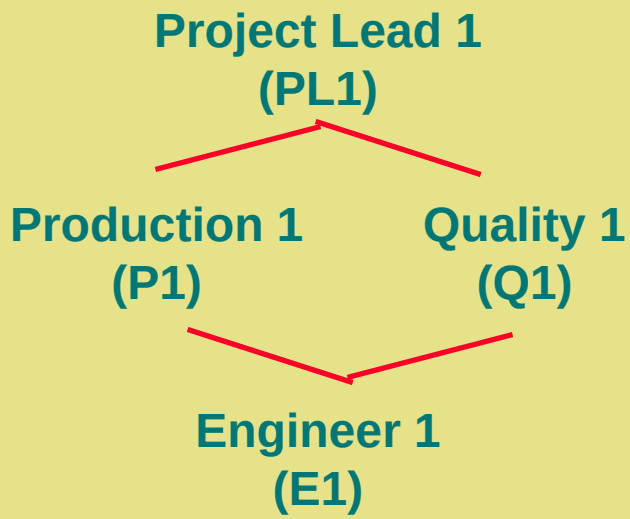
EXAMPLE ROLE HIERARCHY



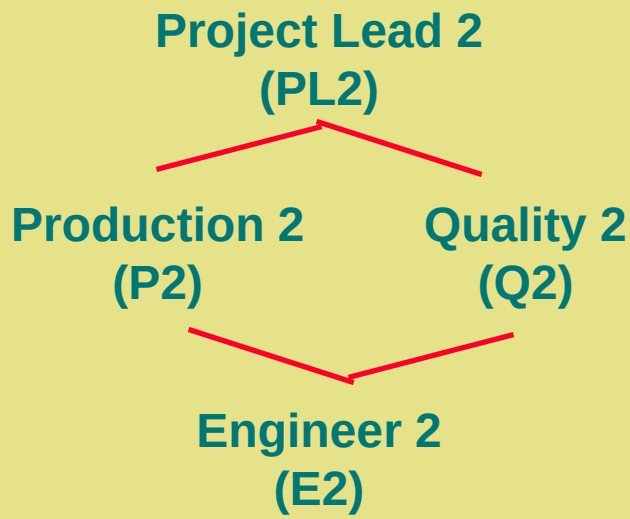
EXAMPLE ROLE HIERARCHY



EXAMPLE ROLE HIERARCHY

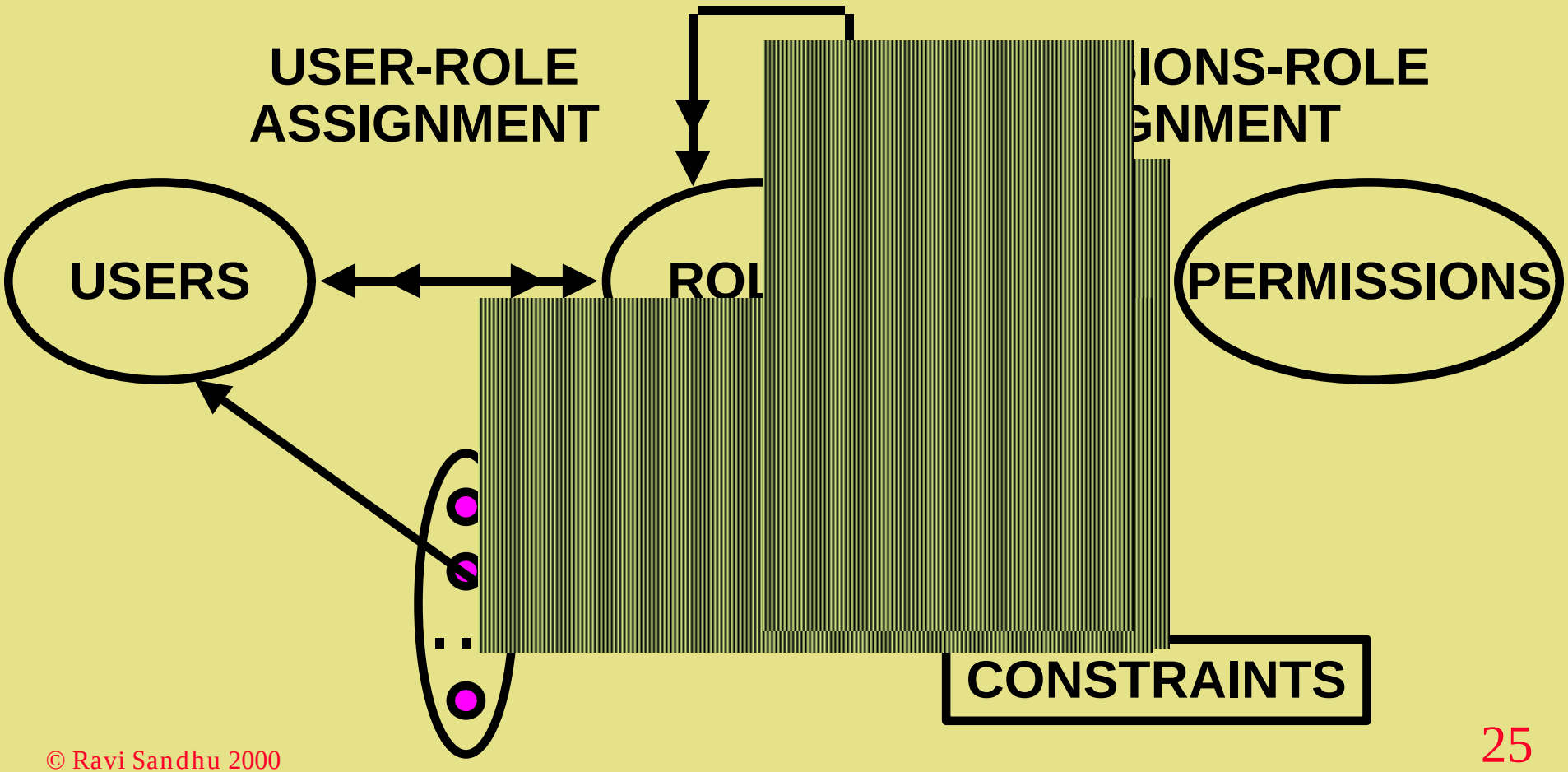


PROJECT 1



PROJECT 2

RBAC3



CONSTRAINTS

- **Mutually Exclusive Roles**
  - **Static Exclusion: The same individual can never hold both roles**
  - **Dynamic Exclusion: The same individual can never hold both roles in the same context**

CONSTRAINTS

- **Mutually Exclusive Permissions**
  - **Static Exclusion: The same role should never be assigned both permissions**
  - **Dynamic Exclusion: The same role can never hold both permissions in the same context**

CONSTRAINTS

- **Cardinality Constraints on User-Role Assignment**
  - **At most k users can belong to the role**
  - **At least k users must belong to the role**
  - **Exactly k users must belong to the role**

CONSTRAINTS

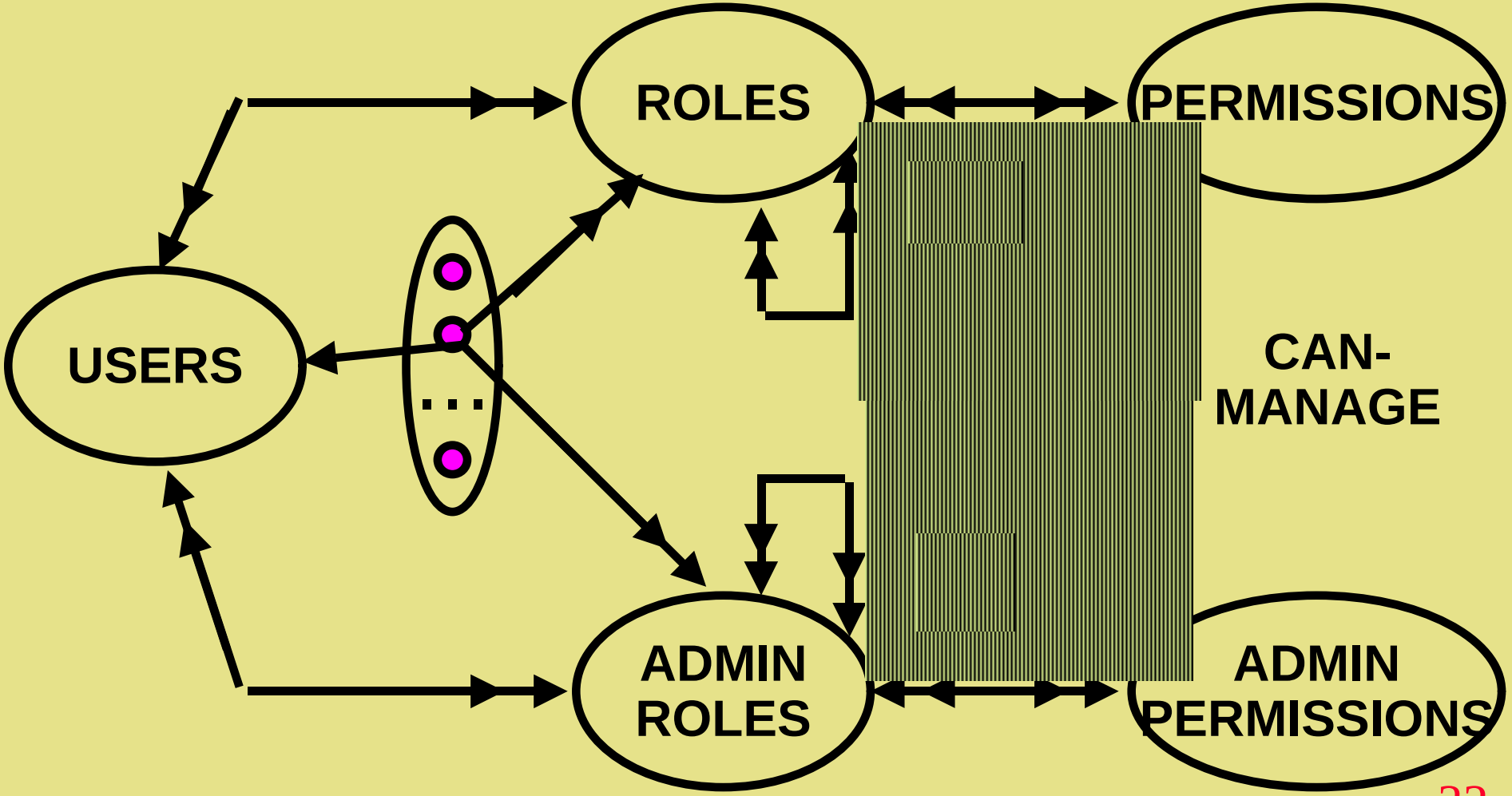
- **Cardinality Constraints on Permissions-Role Assignment**
  - **At most k roles can get the permission**
  - **At least k roles must get the permission**
  - **Exactly k roles must get the permission**

Administrative RBAC
ARBAC97

SCALE AND RATE OF CHANGE

- roles: 100s or 1000s
- users: 1000s or 10,000s or more
- Frequent changes to
  - user-role assignment
  - permission-role assignment
- Less frequent changes for
  - role hierarchy

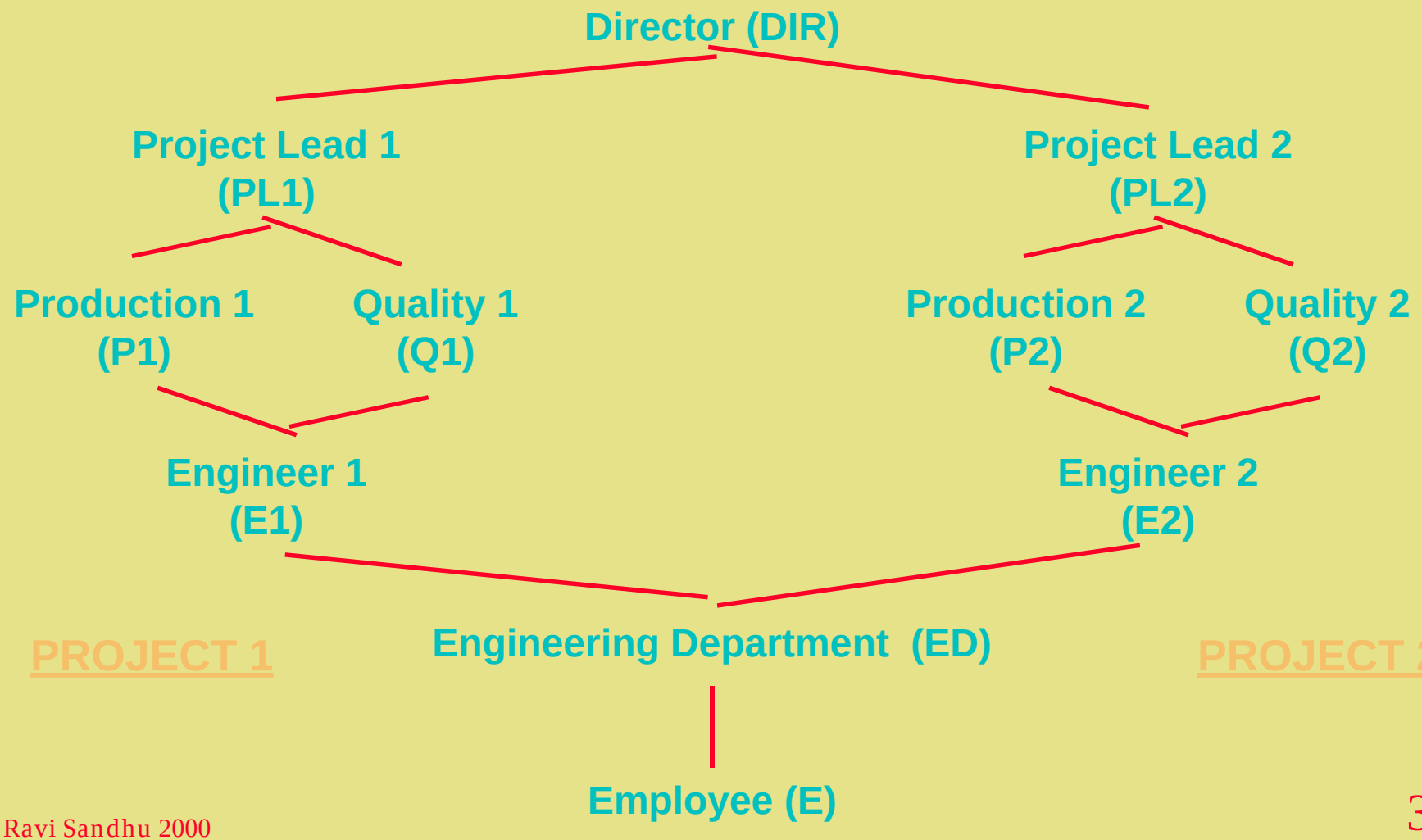
ADMINISTRATIVE RBAC



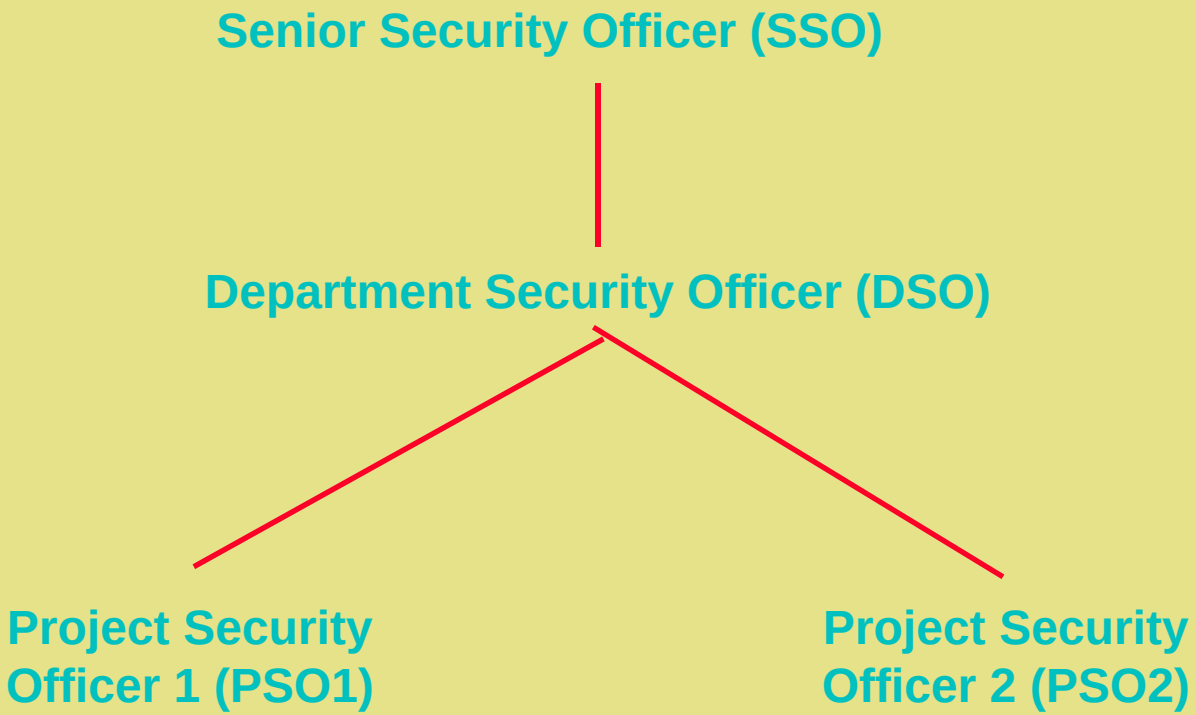
ARBAC97 DECENTRALIZES

- user-role assignment (URA97)
- permission-role assignment (PRA97)
- role-role hierarchy
  - groups or user-only roles (extend URA97)
  - abilities or permission-only roles (extend PRA97)
  - UP-roles or user-and-permission roles (RRA97)

EXAMPLE ROLE HIERARCHY



EXAMPLE ADMINISTRATIVE ROLE HIERARCHY



URA97 GRANT MODEL: can-assign

ARole	Prereq Role	Role Range
PSO1	ED	[E1,PL1)
PSO2	ED	[E2,PL2)
DSO	ED	(ED,DIR)
SSO	E	[ED,ED]
SSO	ED	(ED,DIR]

URA97 GRANT MODEL : can-assign

ARole	Prereq Cond	Role Range
PSO1	ED	[E1,E1]
PSO1	ED & ¬ P1	[Q1,Q1]
PSO1	ED & ¬ Q1	[P1,P1]
PSO2	ED	[E2,E2]
PSO2	ED & ¬ P2	[Q2,Q2]
PSO2	ED & ¬ Q2	[P2,P2]

URA97 REVOKE MODEL : can-revoke

ARole	Role Range
PSO1	[E1,PL1)
PSO2	[E2,PL2)
DSO	(ED,DIR)
SSO	[ED,DIR]

URA97 REVOKE MODEL

- **WEAK REVOCATION**
  - **revokes explicit membership in a role**
  - **independent of who did the assignment**
- **STRONG REVOCATION**
  - **revokes explicit membership in a role and its seniors**
  - **authorized only if corresponding weak revokes are authorized**

PERMISSION-ROLE ASSIGNMENT

- dual of user-role assignment
- can-assign-permission

  can-revoke-permission
- weak revoke

  strong revoke (propagates down)

PERMISSION-ROLE ASSIGNMENT CAN-ASSIGN-PERMISSION

ARole	Prereq Cond	Role Range
PSO1	PL1	[E1,PL1)
PSO2	PL2	[E2,PL2)
DSO	E1 $\vee$ E2	[ED,ED]
SSO	PL1 $\vee$ PL2	[ED,ED]
SSO	ED	[E,E]

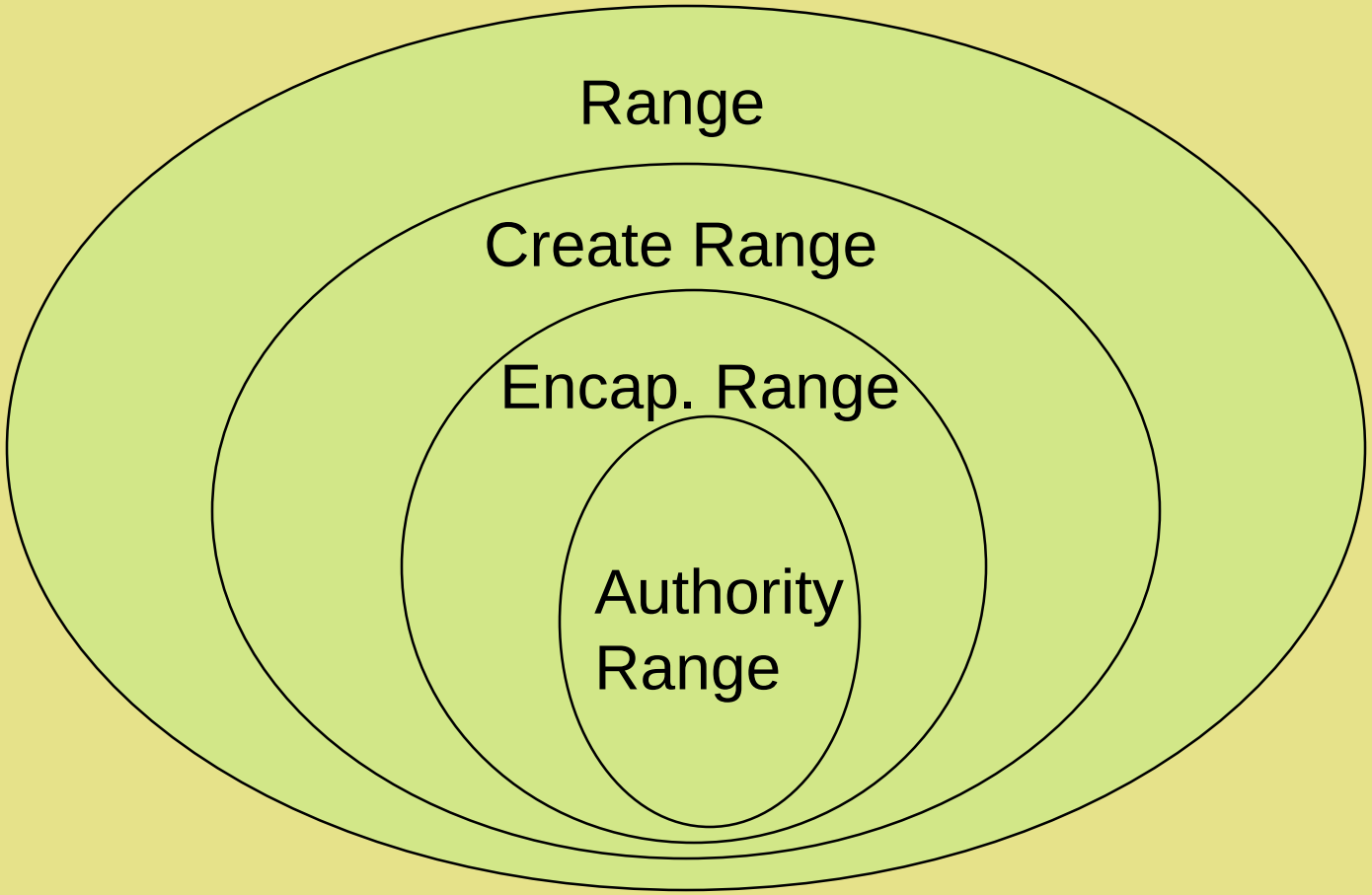
PERMISSION-ROLE ASSIGNMENT CAN-REVOKE-PERMISSION

ARole	Role Range
PSO1	[E1,PL1]
PSO2	[E2,PL2]
DSO	(ED,DIR)
SSO	[ED,DIR]

ARBAC97 DECENTRALIZES

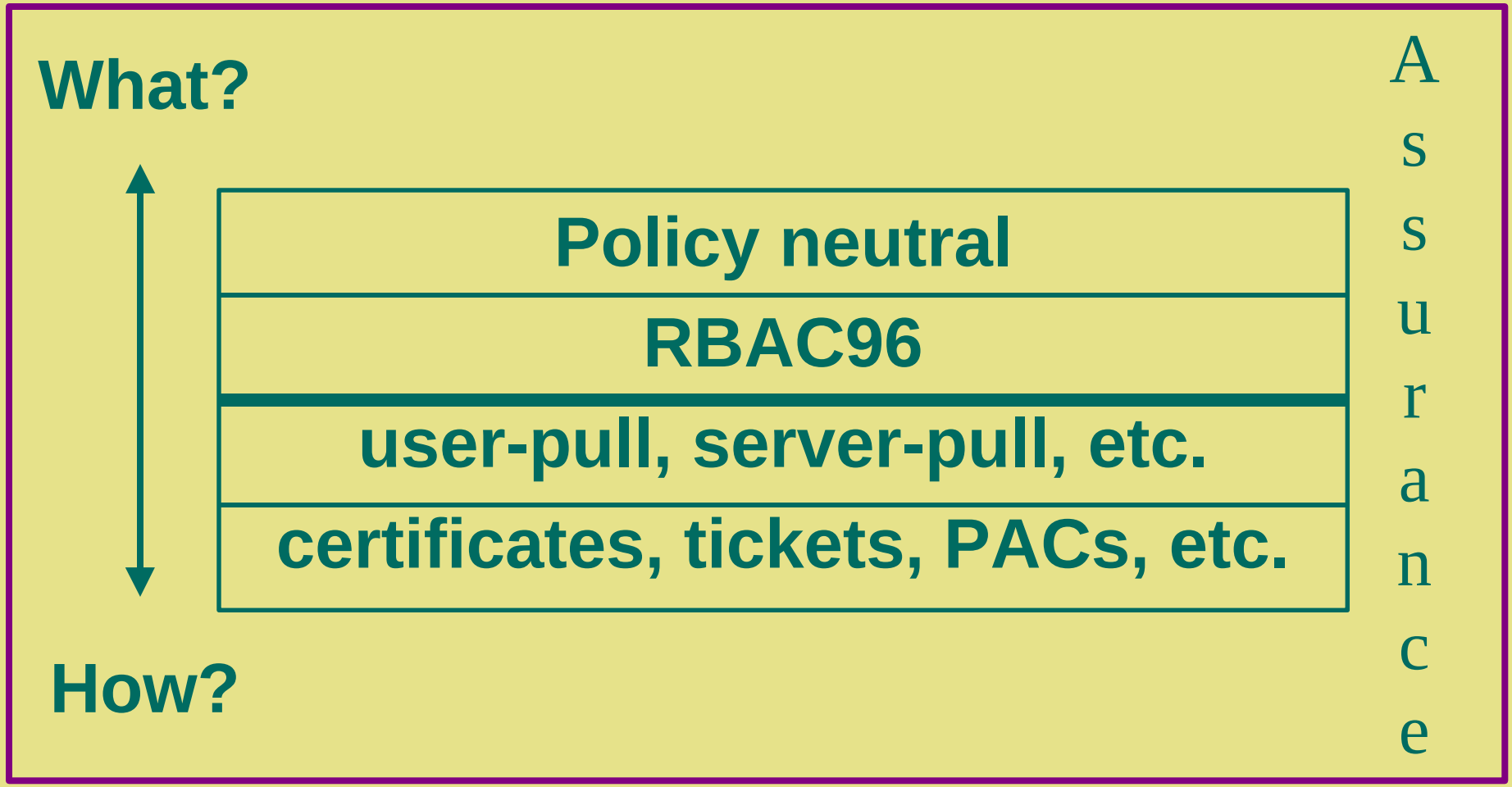
- user-role assignment (URA97)
- permission-role assignment (PRA97)
- role-role hierarchy
  - groups or user-only roles (extend URA97)
  - abilities or permission-only roles (extend PRA97)
  - UP-roles or user-and-permission roles (RRA97)

Range Definitions

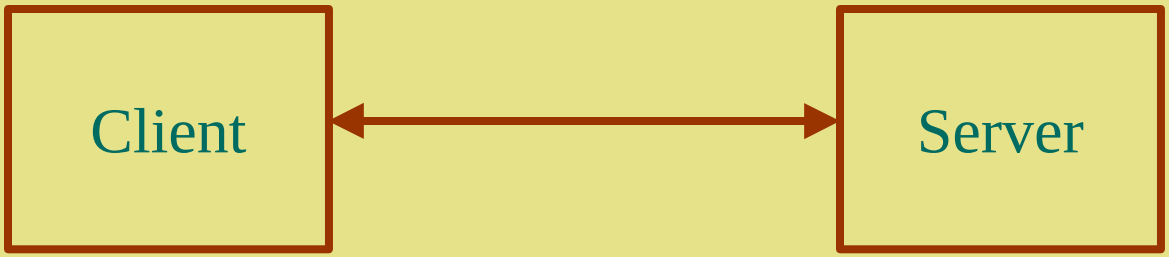


RBAC ARCHITECTURES

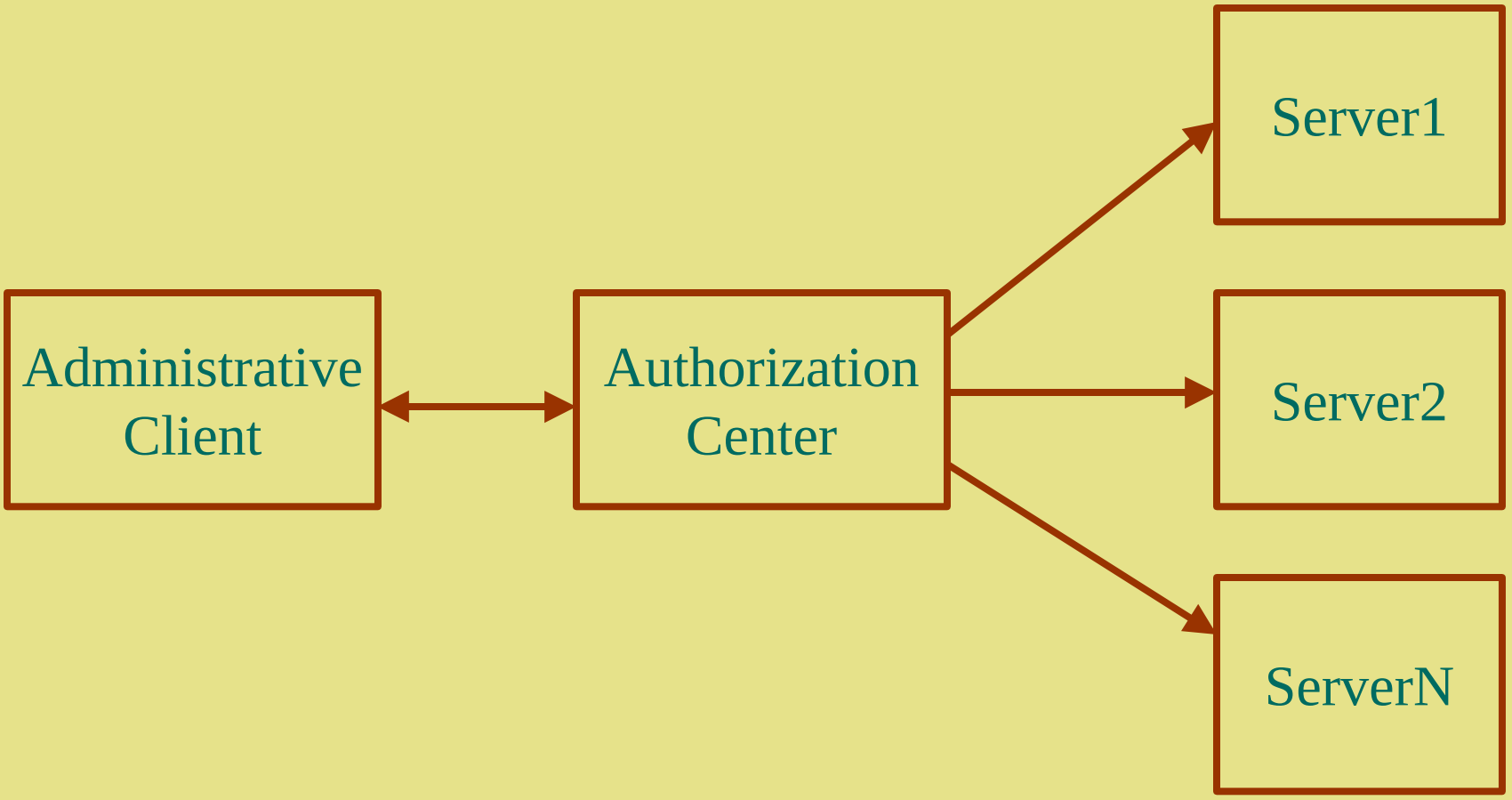
OM-AM AND ROLE-BASED ACCESS CONTROL (RBAC)



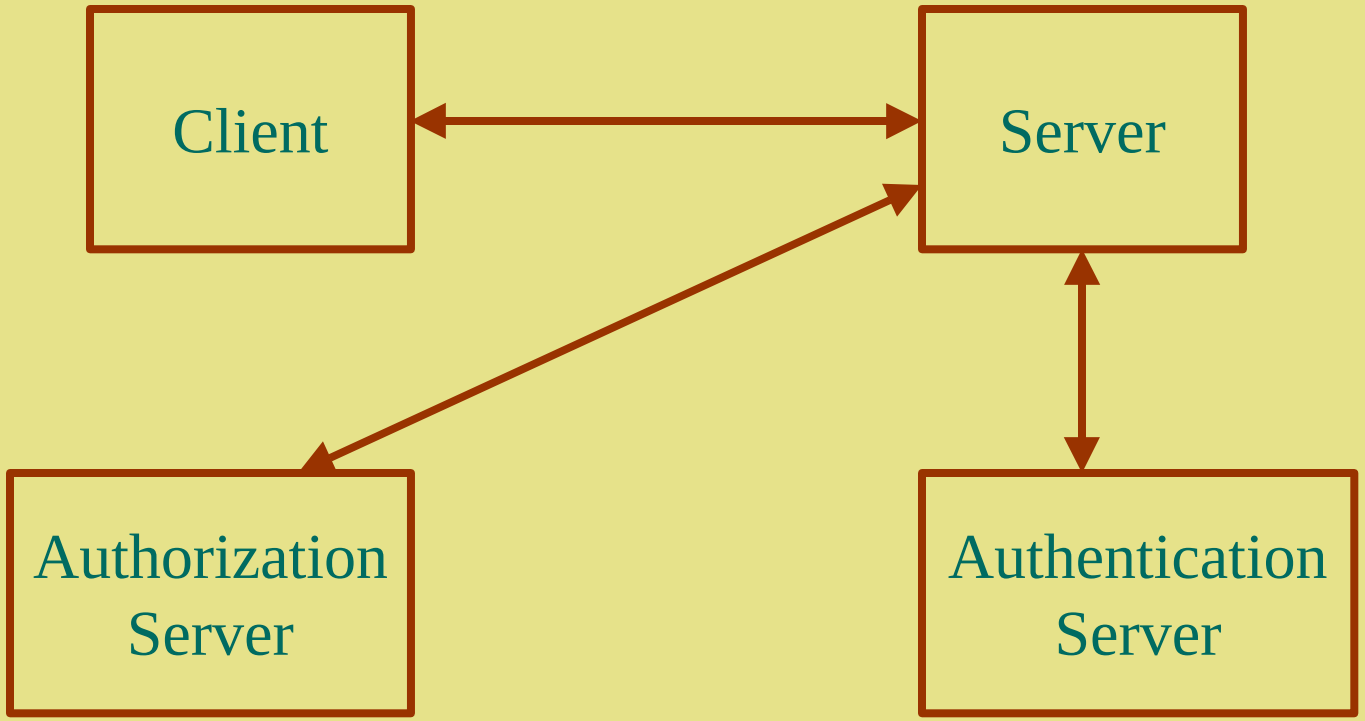
CLASS I SYSTEMS
ENFORCEMENT ARCHITECTURE



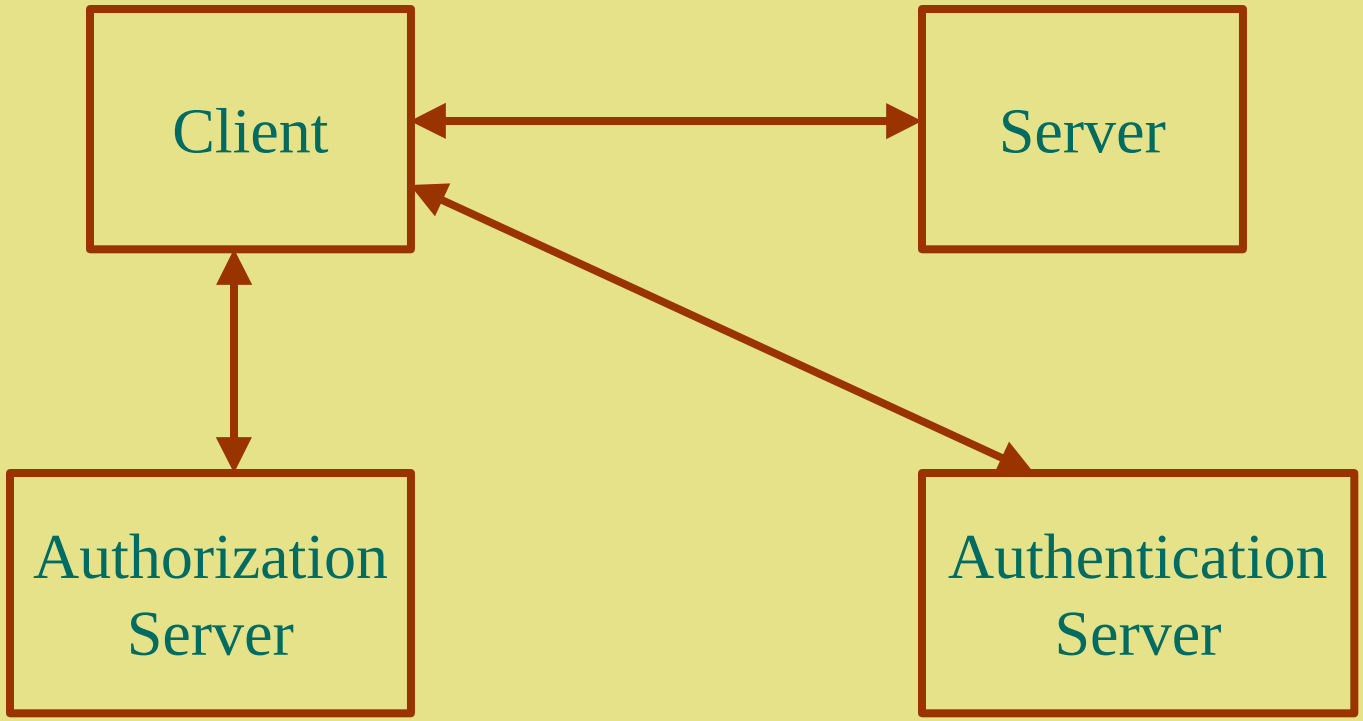
CLASS I SYSTEMS
ADMINISTRATION ARCHITECTURE



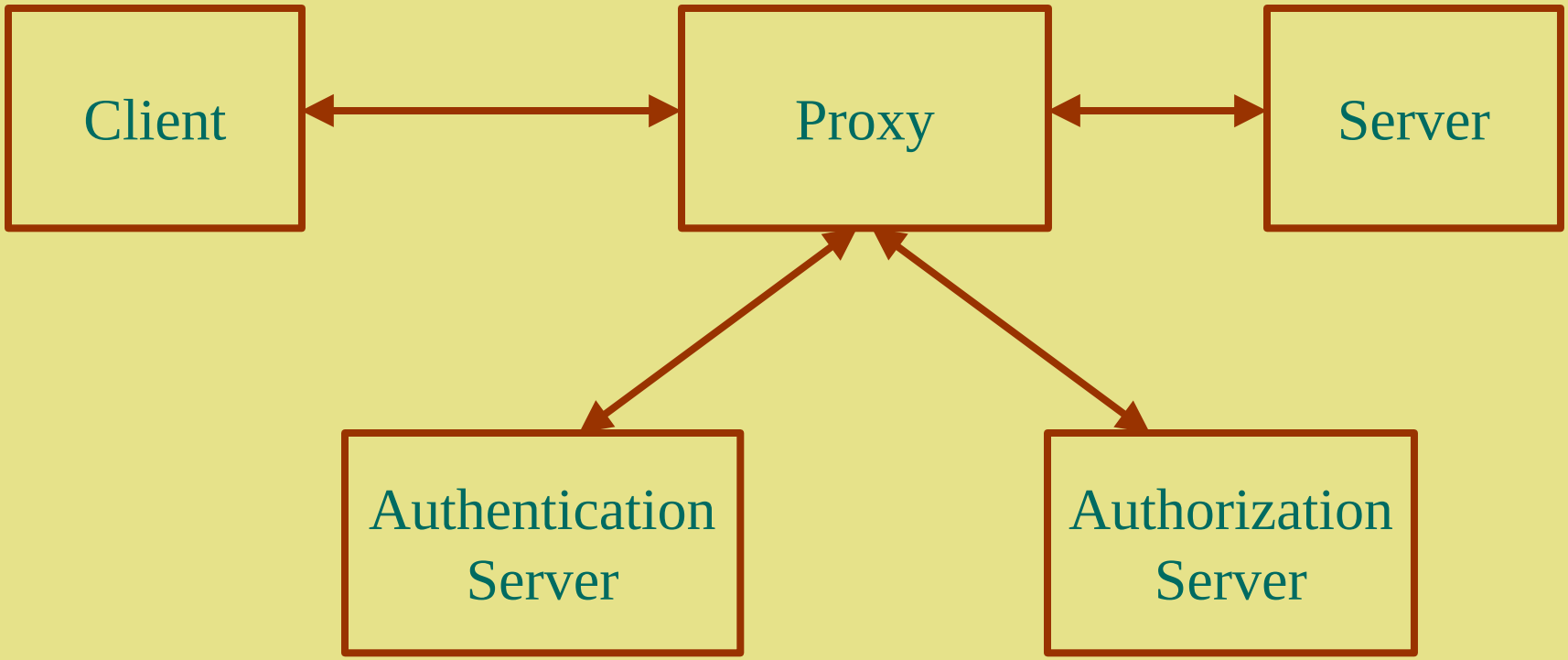
CLASS II SYSTEMS SERVER-PULL



CLASS II SYSTEMS USER-PULL



CLASS II SYSTEMS
PROXY-BASED SYSTEMS



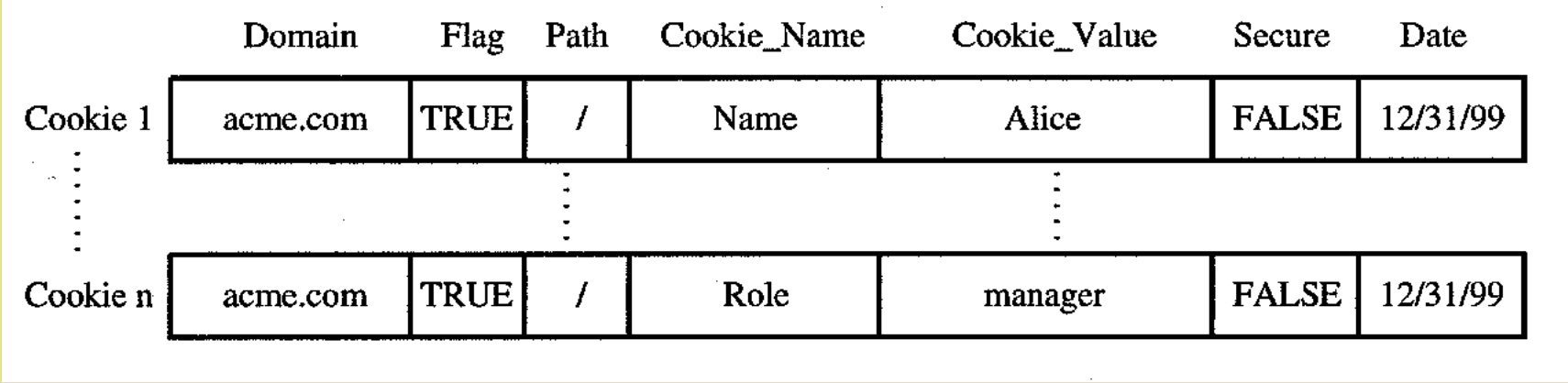
RBAC MECHANISMS

- **These architectures can be supported by means of**
  - **X.509 certificates**
  - **Secure cookies**
  - **Etc.**
- **Different links can be protected by different means**

Related Technologies

- **Cookies**
  - **in widespread current use for maintaining state of HTTP**
  - **becoming standard**
  - **not secure**
- **Public-Key Certificates (X.509)**
  - **support security on the Web based on PKI**
  - **standard**
  - **simply, bind users to keys**
  - **have the ability to be extended**

Cookies

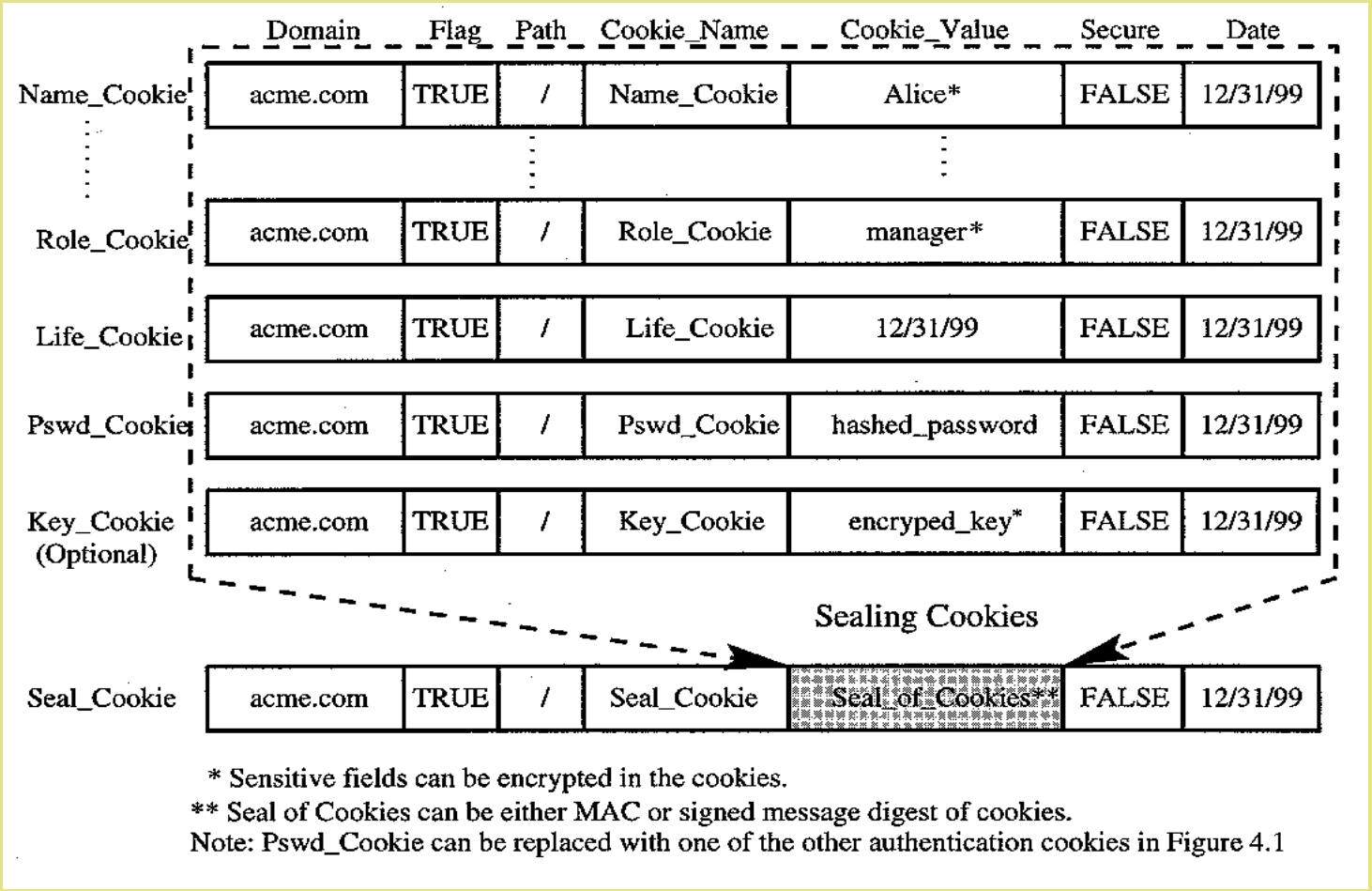


	Domain	Flag	Path	Cookie_Name	Cookie_Value	Secure	Date
Cookie 1	acme.com	TRUE	/	Name	Alice	FALSE	12/31/99
Cookie n	acme.com	TRUE	/	Role	manager	FALSE	12/31/99

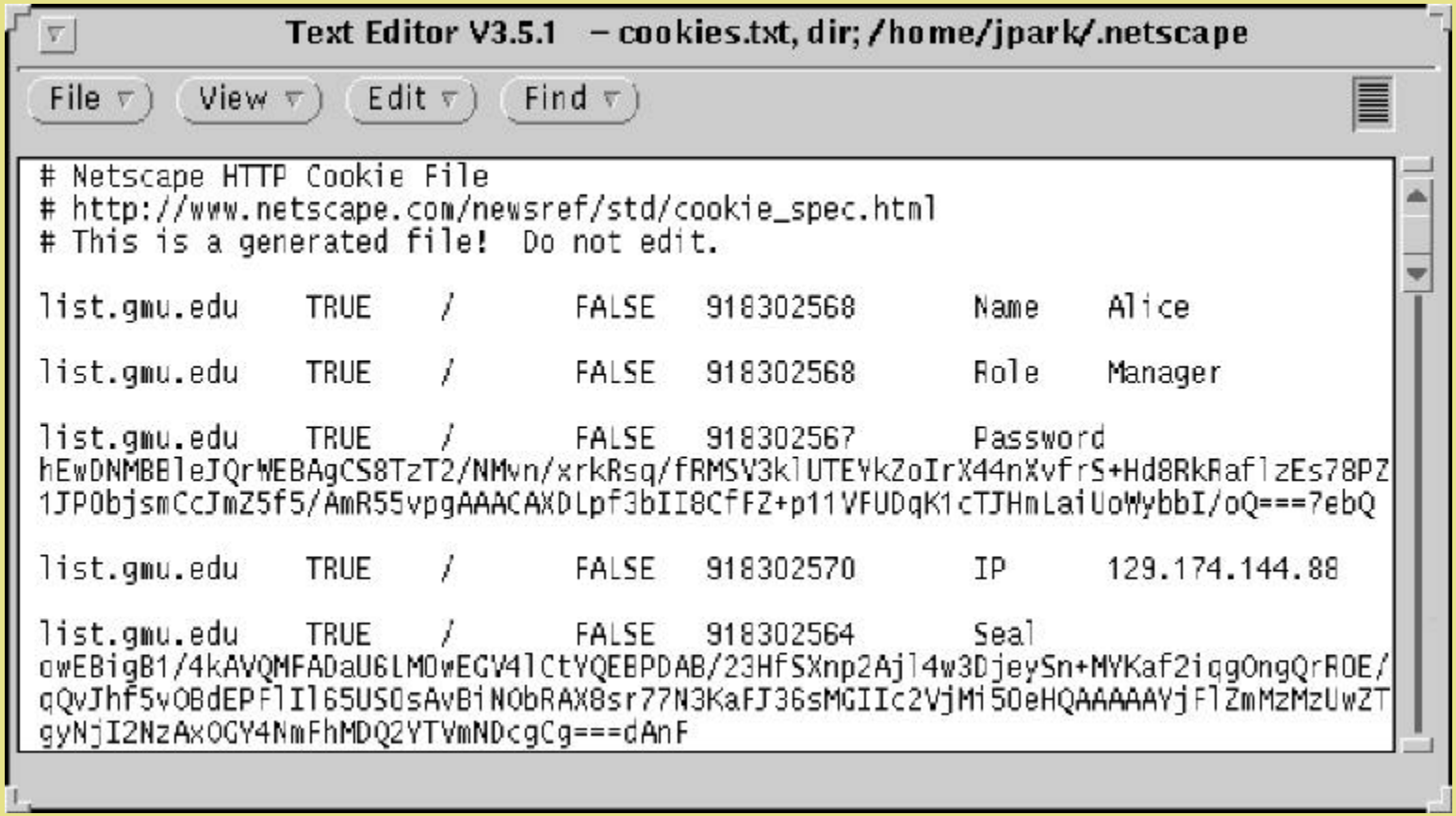
Security Threats to Cookies

- **Cookies are not secure**
  - **No authentication**
  - **No integrity**
  - **No confidentiality**
- **can be easily attacked by**
  - **Network Security Threats**
  - **End-System Threats**
  - **Cookie Harvesting Threats**

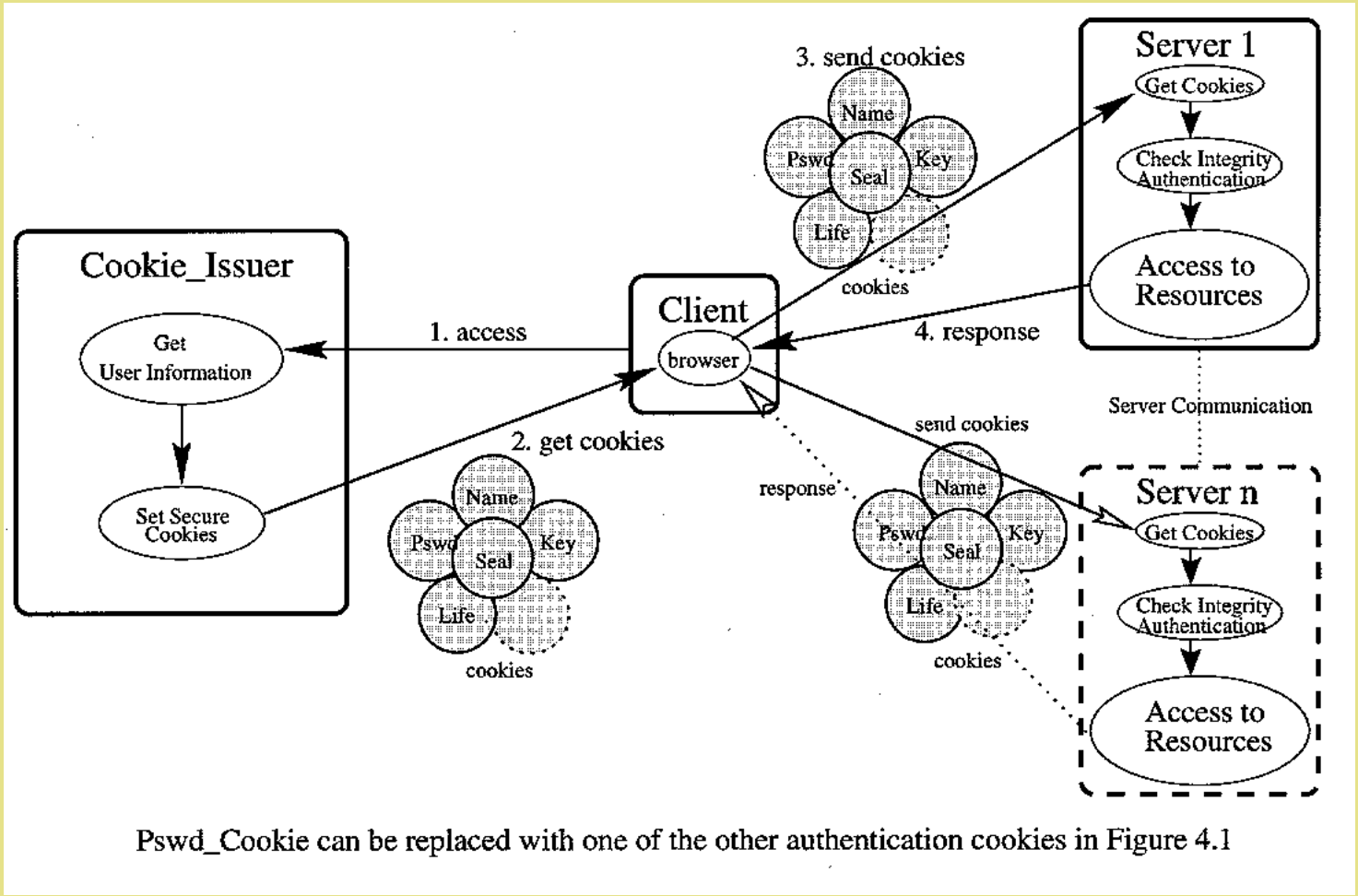
Secure Cookies on the Web



A Set of Secure Cookies



How to Use Secure Cookies



Pswd_Cookie can be replaced with one of the other authentication cookies in Figure 4.1

X.509 Certificate

- Digitally signed by a certificate authority
  - to confirm the information in the certificate belongs to the holder of the corresponding private key
- Contents
  - version, serial number, subject, validity period, issuer, optional fields (v2)
  - subject's public key and algorithm info.
  - extension fields (v3)
  - digital signature of CA
- Binding users to keys
- Certificate Revocation List (CRL)

X.509 Certificate

Certificate Content:

```
Certificate:
    Data:
        Version: v3 (0x2)
        Serial Number: 5 (0x5)
        Signature Algorithm: PKCS #1 MD5 With RSA Encryption
        Issuer: CN=data.list.gmu.edu, OU=LIST, O=GMU, C=US
        Validity:
            Not Before: Tue Feb 09 03:10:38 1999
            Not  After: Wed Feb 09 03:10:38 2000
        Subject: CN=admin.list.gmu.edu, OU=LIST, O=GMU, C=US
        Subject Public Key Info:
            Algorithm: PKCS #1 RSA Encryption
            Public Key:
                Modulus:
                    00:bc:d7:fc:4f:29:a4:29:a5:21:be:69:47:4d:55:db:37:50:
                    18:2b:6e:3e:b0:85:3e:0f:86:0f:be:58:2b:c9:d3:dc:bc:03:
                    bc:86:44:c4:f4:18:94:51:96:c6:f9:c5:db:b8:9d:88:5b:53:
                    b7:08:2f:86:64:cb:c2:7b:60:36:87
                Public Exponent: 65537 (0x10001)
        Extensions:
            Identifier: Certificate Type
                Critical: no
                Certified Usage:
                    SSL Client
            Identifier: Authority Key Identifier
                Critical: no
                Key Identifier:
                    a5:d7:08:bc:ff:07:bd:5a:d4:8d:d4:68:53:87:4b:af:81:90:
                    f0:4d
    Signature:
        Algorithm: PKCS #1 MD5 With RSA Encryption
        Signature:
            11:ca:b1:94:14:fb:67:a2:ad:90:f1:ee:88:24:a8:d3:fd:5c:75:34:fc:
            c1:68:23:e6:12:19:3a:5c:45:62:af:51:a0:2f:44:96:f8:2e:1f:75:9a:
            4b:9c:ed:2a:45:2e:db:c8:9c:56:1a:e1:75:0a:8e:bf:f8:44:b8:84:31:
            d8
```

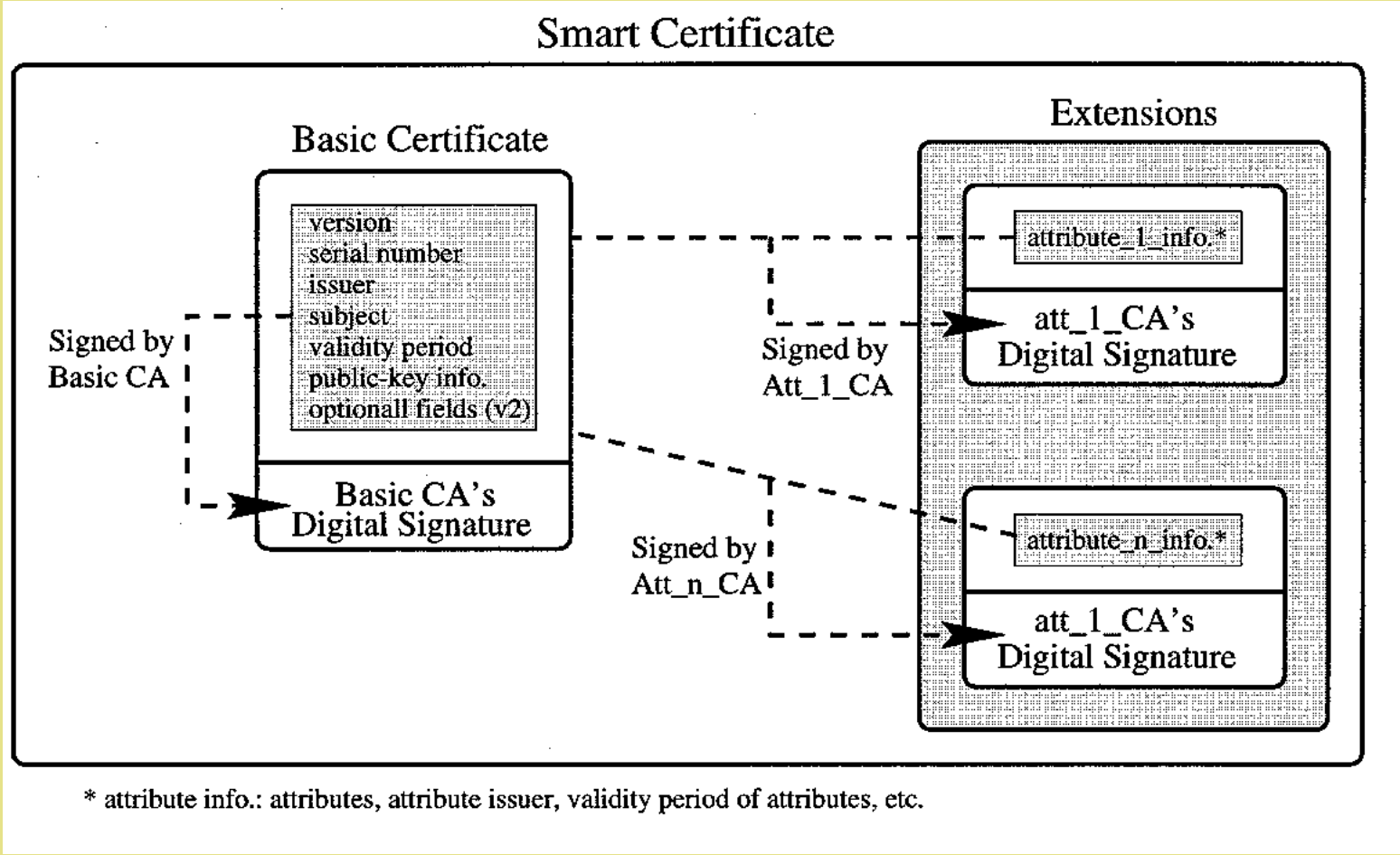
Smart Certificates

- **Short-Lived Lifetime**
  - **More secure**
    - **typical validity period for X.509 is months (years)**
    - **users may leave copies of the corresponding keys behind**
    - **the longer-lived certificates have a higher probability of being attacked**
  - **No Certificate Revocation List (CRL)**
    - **simple and less expensive PKI**

Smart Certificates

- **Containing Attributes Securely**
  - **Web servers can use secure attributes for their purposes**
  - **Each authority has independent control on the corresponding information**
    - **basic certificate (containing identity information)**
    - **each attribute can be added, changed, revoked, or re-issued by the appropriate authority**
      - e.g., role, credit card number, clearance, etc.
  - **Short-lived certificate can remove CRLs**

Separate CAs in a Certificate



Smart Certificates

- **Postdated Certificates**
  - **The certificate becomes valid at some time in the future**
  - **possible to make a smart certificate valid for a set of duration**
  - **supports convenience**
- **Confidentiality**
  - **Sensitive information can be**
    - **encrypted in smart certificates**
      - e.g. passwords, credit card numbers, etc.

A Smart Certificate

Certificate Content:

```
Certificate:
    Data:
        Version: v3 (0x2)
        Serial Number: 26 (0x1a)
        Signature Algorithm: PKCS #1 MD5 With RSA Encryption
        Issuer: CN=data.list.gmu.edu, OU=LIST, O=GMU, C=US
        Validity:
            Not Before: Sun May 02 17:25:31 1999
            Not  After: Mon May 03 01:25:31 1999
        Subject: CN=Alice List, UID=alice, OU=LIST, O=GMU, C=US
        Subject Public Key Info:
            Algorithm: PKCS #1 RSA Encryption
            Public Key:
                Modulus:
                    00:9d:31:41:cf:45:d3:25:10:41:b3:ca:23:f6:09:91:ad:3d:
                    2d:c0:62:e1:ff:24:43:fe:39:90:c0:13:03:11:b5:77:ec:79:
                    17:b8:63:be:aa:36:4e:29:08:9b:76:64:b7:97:94:19:06:a7:
                    7a:b2:8b:31:f3:b5:72:3f:04:8f:17
                Public Exponent: 65537 (0x10001)
        Extensions:
            Identifier: Certificate Type
                Critical: no
                Certified Usage:
                    SSL Client
                    Secure E-mail
            Identifier: role
                Critical: no
                Value: hEwDNMBB1eJQrWEBAgCS8TzT2/NMvn/xrkRsq/fRMSV3klUTEYkZoI
            Identifier: Authority Key Identifier
                Critical: no
                Key Identifier:
                    a5:d7:08:bc:ff:07:bd:5a:d4:8d:d4:68:53:87:4b:af:81:90:
                    f0:4d
    Signature:
        Algorithm: PKCS #1 MD5 With RSA Encryption
        Signature:
            c7:39:f7:b8:59:19:52:1c:fc:08:7c:11:f6:6e:5a:07:5b:55:80:a5:d8:
            65:a4:40:dc:06:5e:e4:ff:96:ad:71:9b:21:7a:4b:be:50:48:c2:f1:a6:
            7c:16:12:61:c7:bf:57:07:6d:c5:f4:f8:c2:e1:62:27:f6:d6:ae:09:77:
            46
```

Applications of Smart Certificates

- **On-Duty Control**
- **Compatible with X.509**
- **User Authentication**
- **Electronic Transaction**
- **Eliminating Single-Point Failure**
- **Pay-per-Access**
- **Attribute-based Access Control**

OM-AM AND ROLE-BASED ACCESS CONTROL (RBAC)