

# The Future of Access Control: Attributes, Automation and Adaptation

Prof. Ravi Sandhu  
Executive Director and Endowed Chair

NSS 2012  
November 21, 2012

[ravi.sandhu@utsa.edu](mailto:ravi.sandhu@utsa.edu)  
[www.profsandhu.com](http://www.profsandhu.com)  
[www.ics.utsa.edu](http://www.ics.utsa.edu)

- Cyberspace will become orders of magnitude more complex and confused very quickly
- Overall this is a very positive development and will enrich human society
- It will be messy but need not be chaotic!
- Cyber security research and practice are loosing ground

- Enable system designers and operators to say:

This system is secure

Not attainable

- There is an infinite supply of low-hanging attacks

- Enable system designers and operators to say:

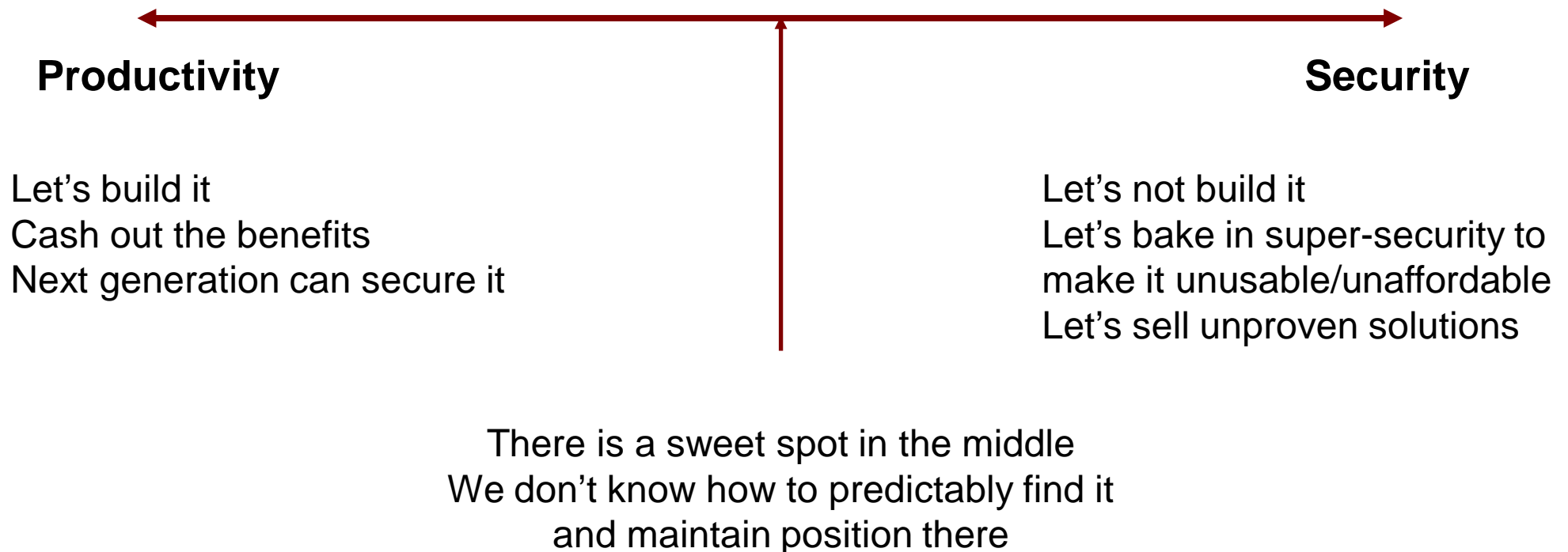
This system is secure enough

Many successful examples

- Mass scale, not very high assurance
  - ❖ ATM network
  - ❖ On-line banking
  - ❖ E-commerce
- One of a kind, extremely high assurance
  - ❖ US President's nuclear football

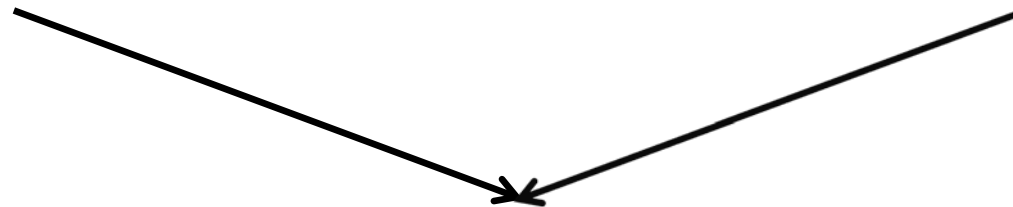
- Our successes are not studied as success stories
- Our successes are not attainable via current cyber security science, engineering, doctrine

- Cyber Security is all about
  - ❖ tradeoffs and adjustments
  - ❖ automation (in future)



**Discretionary Access Control  
(DAC), 1970**

**Mandatory Access Control  
(MAC), 1970**



**Role Based Access Control  
(RBAC), 1995**



**Attribute Based Access Control  
(ABAC), ????**

**Fixed  
policy**



**Discretionary Access Control  
(DAC), 1970**

**Mandatory Access Control  
(MAC), 1970**

**Role Based Access Control  
(RBAC), 1995**

**Attribute Based Access Control  
(ABAC), ????**

**Flexible  
policy**



**Human  
Driven**



**Discretionary Access Control  
(DAC), 1970**

**Mandatory Access Control  
(MAC), 1970**

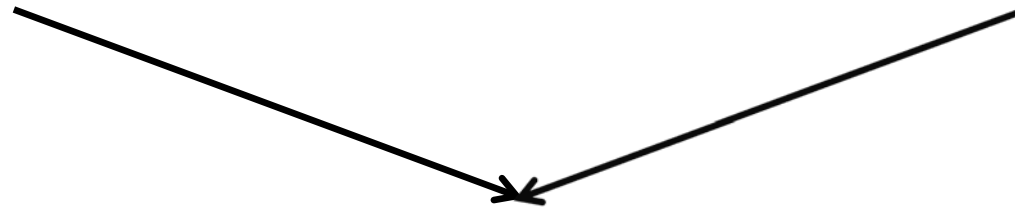
**Role Based Access Control  
(RBAC), 1995**

**Attribute Based Access Control  
(ABAC), ????**

**Automated  
Adaptive**

**Discretionary Access Control  
(DAC), 1970**

**Mandatory Access Control  
(MAC), 1970**

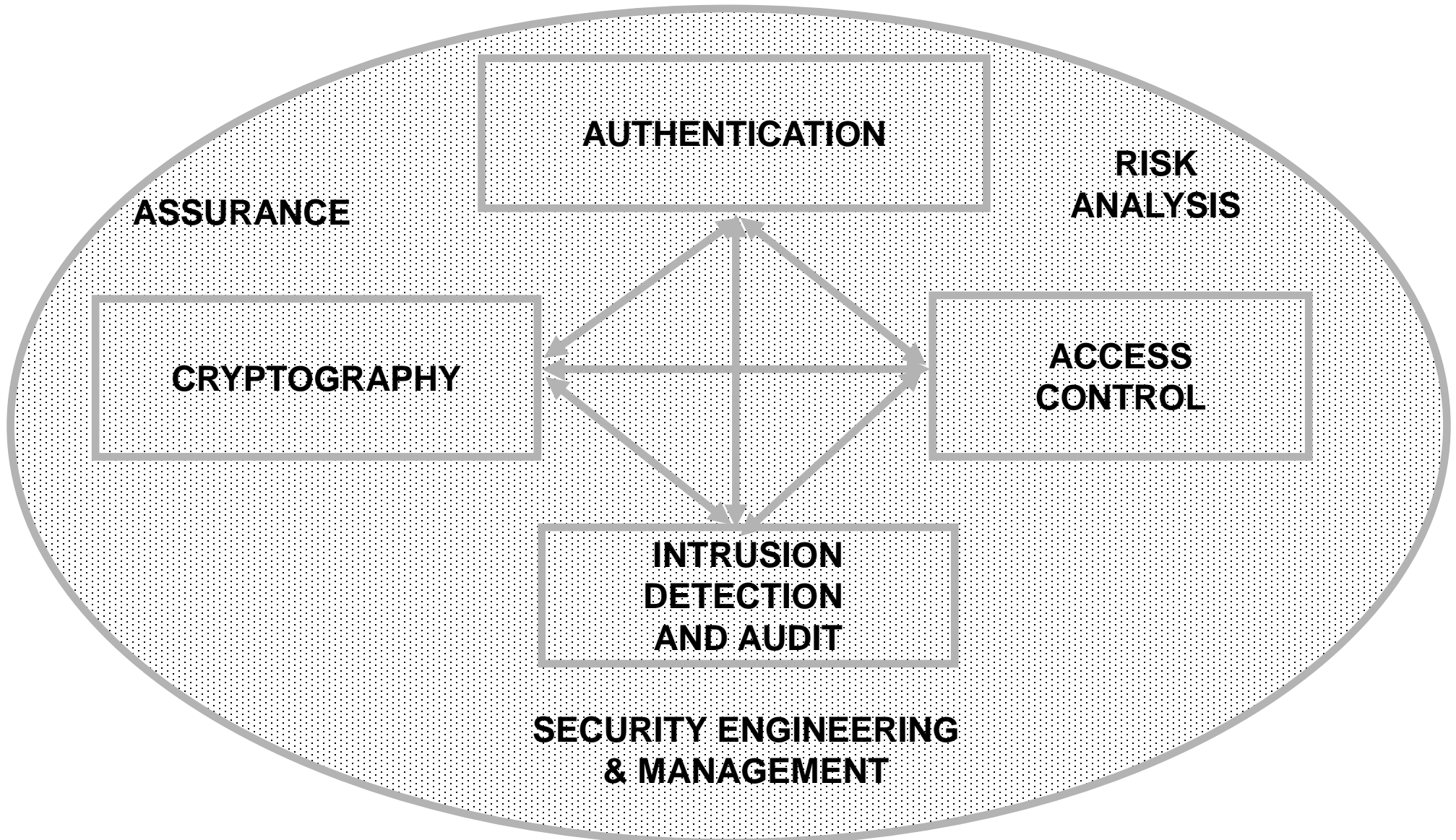


**Role Based Access Control  
(RBAC), 1995**

Messy or  
Chaotic?



**Attribute Based Access Control  
(ABAC), ????**



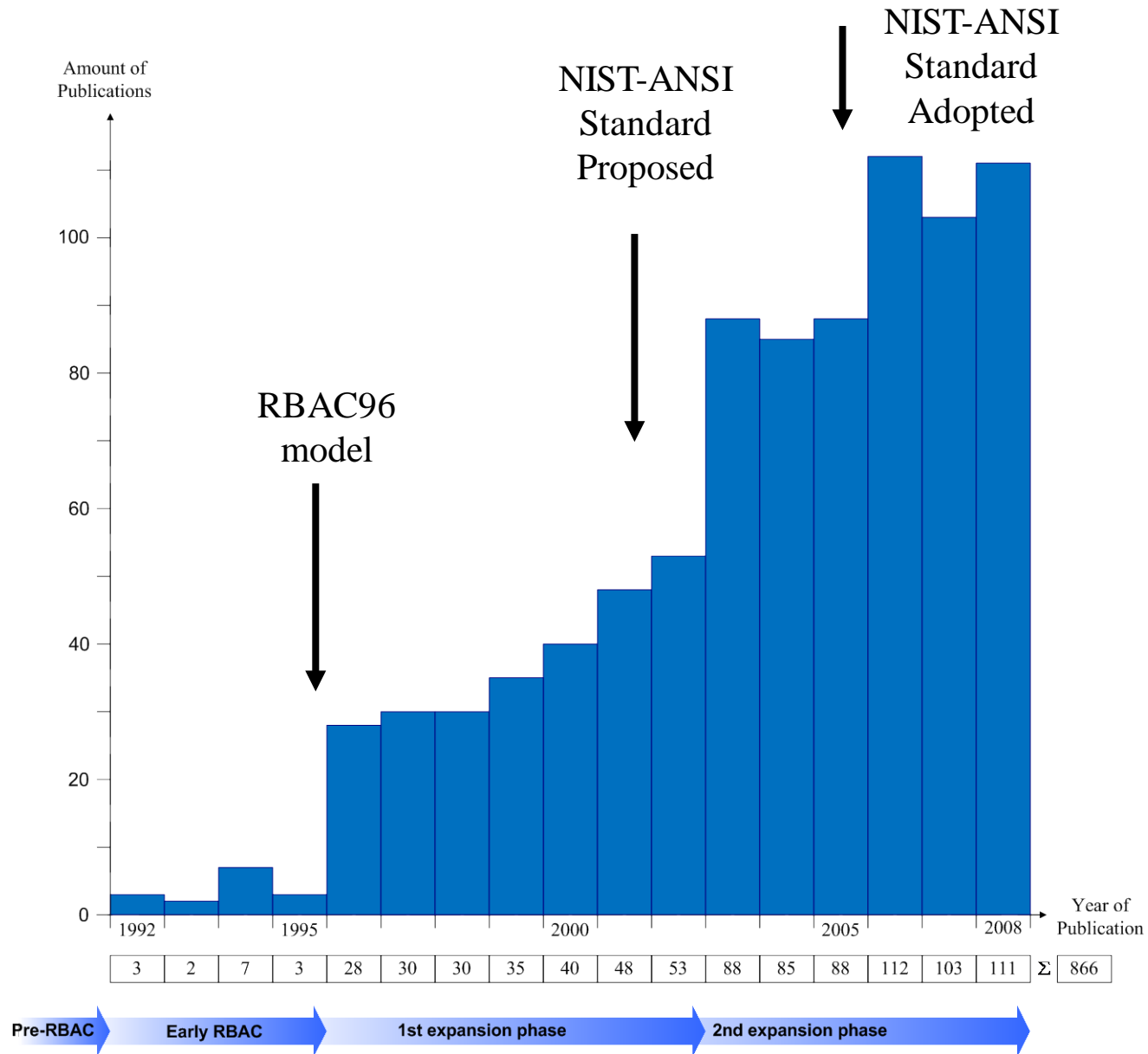
- Analog Hole
- Inference
- Covert Channels
- Side Channels
- Phishing
- Safety
- Usability
- Privacy
- Attack Asymmetry
- Compatibility
- Federation
- ....

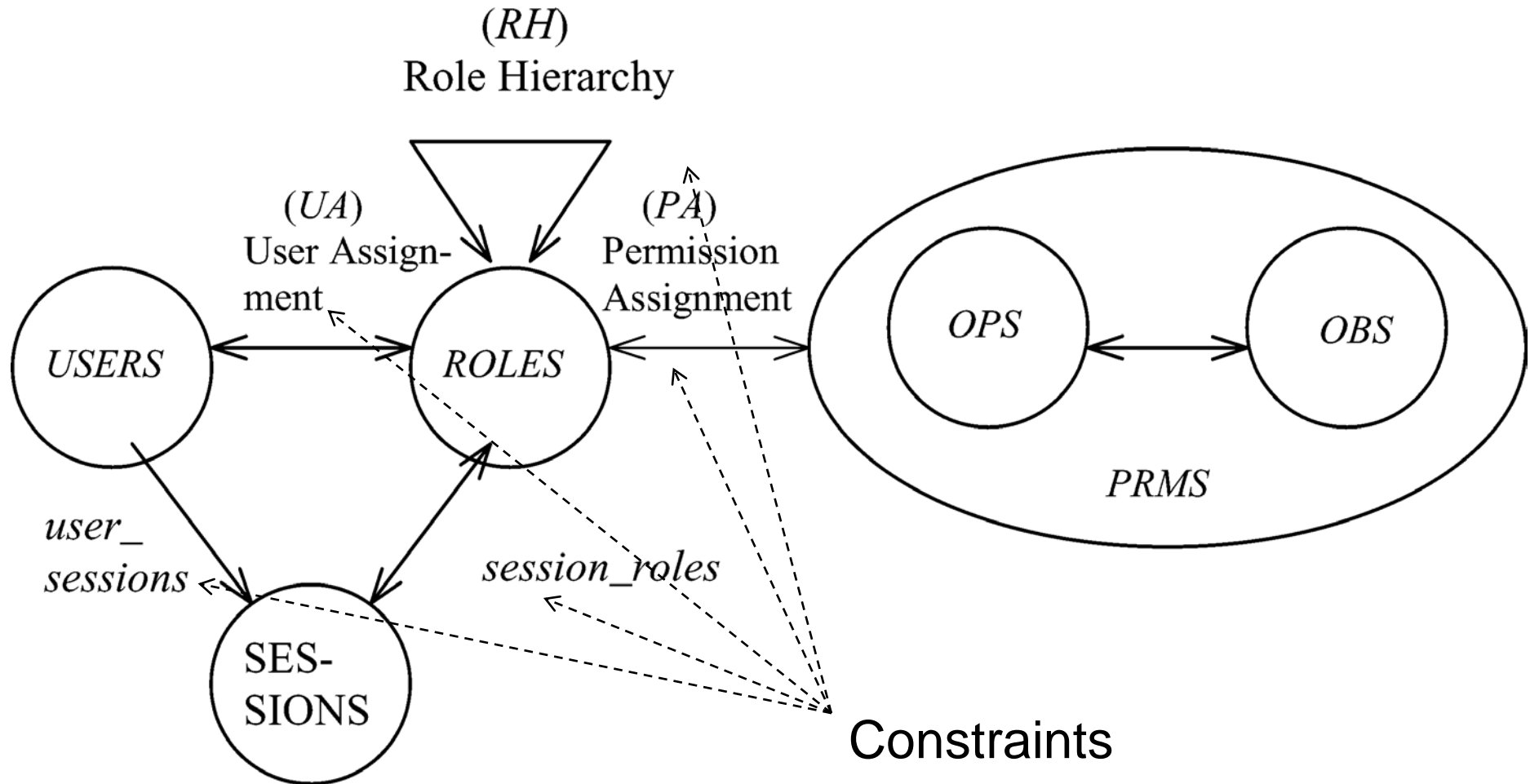
- Analog Hole
- Inference
- Covert Channels
- Side Channels
- Phishing
- Safety
- Usability
- Privacy
- Attack Asymmetry
- Compatibility
- Federation
- ....

Can manage  
Cannot eliminate

- Discretionary Access Control (DAC), 1970
  - ❖ Owner controls access
  - ❖ But only to the original, not to copies
  - ❖ Grounded in pre-computer policies of researchers
- Mandatory Access Control (MAC), 1970
  - ❖ Synonymous to Lattice-Based Access Control (LBAC)
  - ❖ Access based on security labels
  - ❖ Labels propagate to copies
  - ❖ Grounded in pre-computer military and national security policies
- Role-Based Access Control (RBAC), 1995
  - ❖ Access based on roles
  - ❖ Can be configured to do DAC or MAC
  - ❖ Grounded in pre-computer enterprise policies

**Numerous other models but only 3 successes: SO FAR**





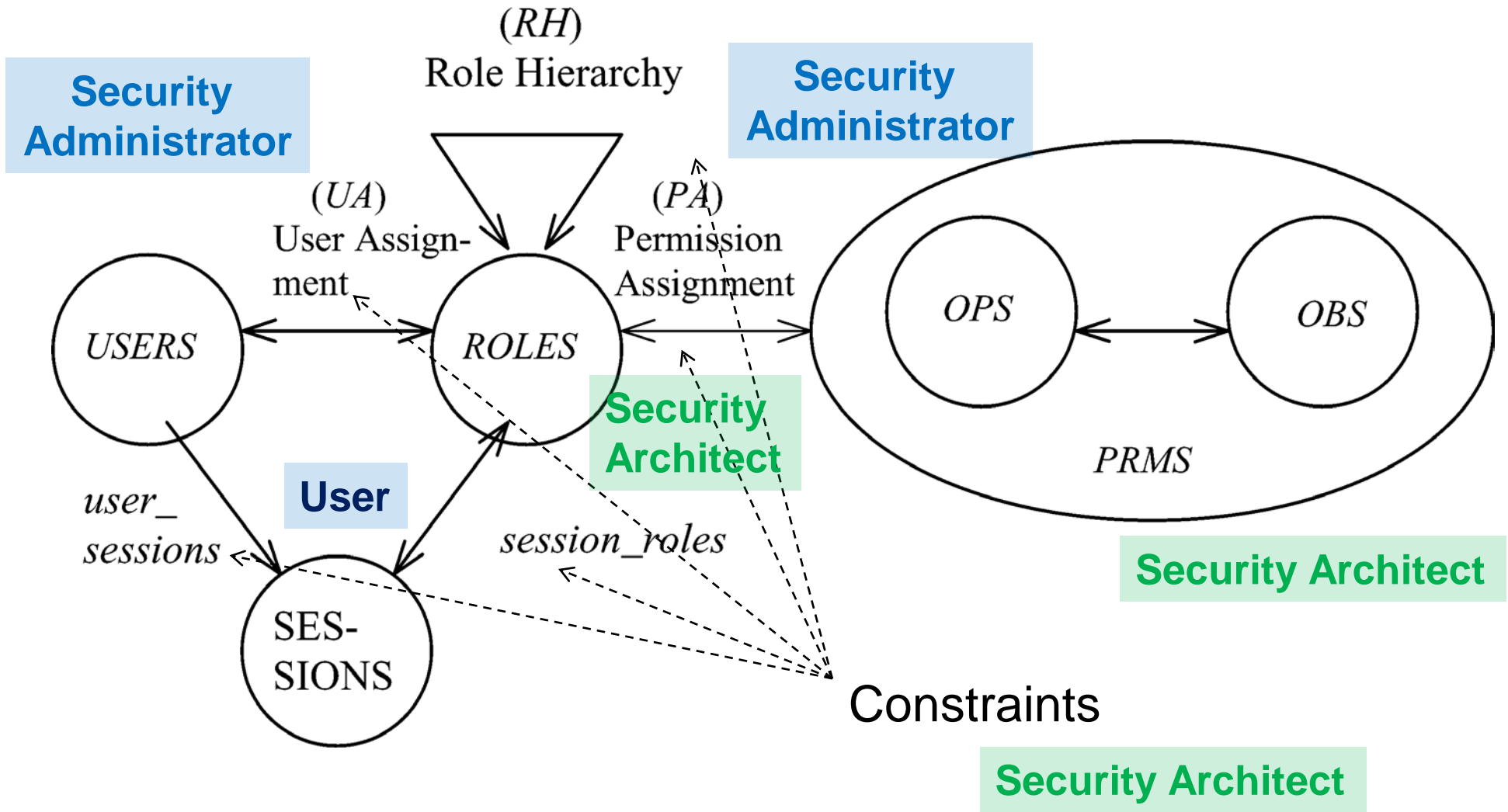


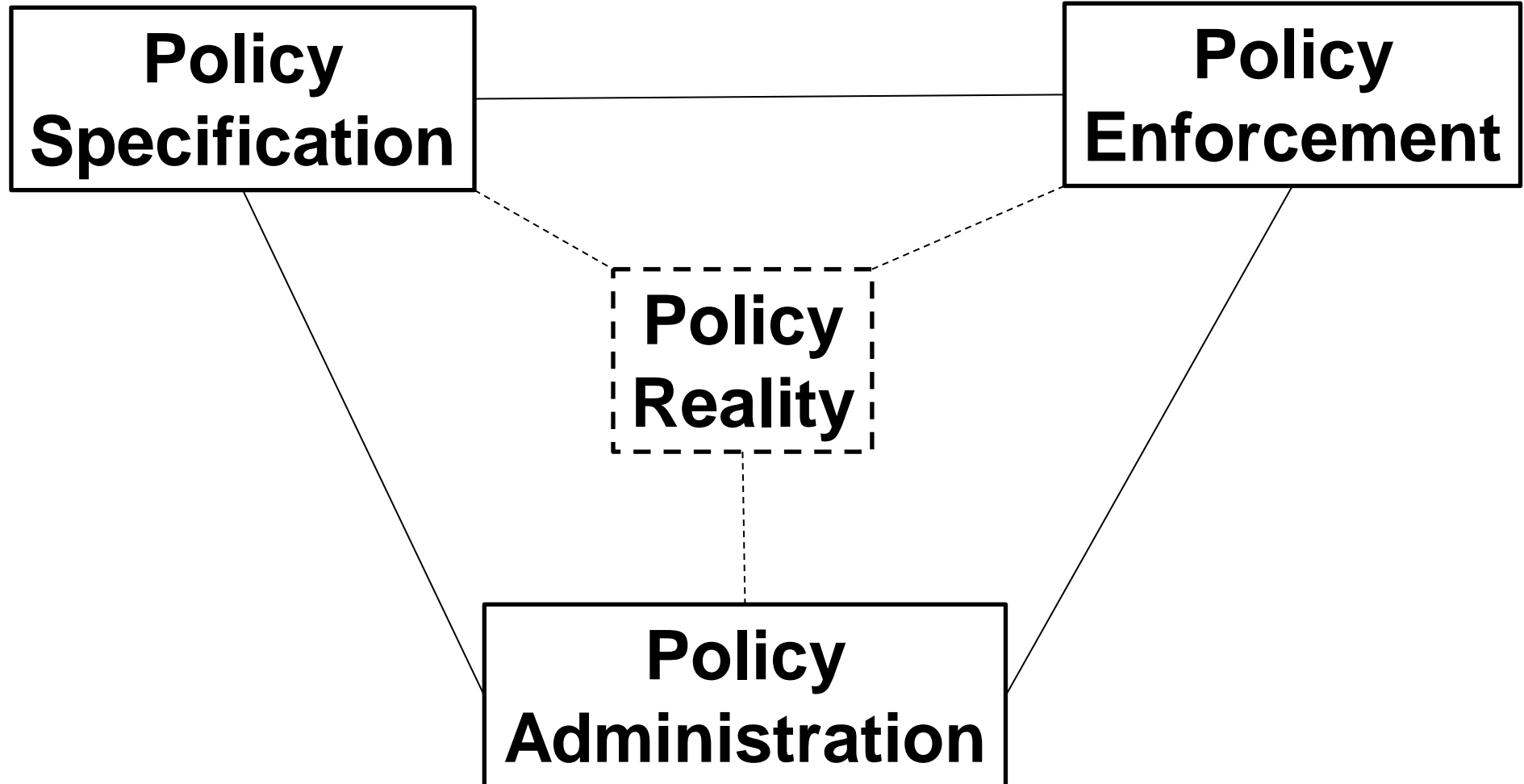
- RBAC can be configured to do MAC
- RBAC can be configured to do DAC
- RBAC is policy neutral

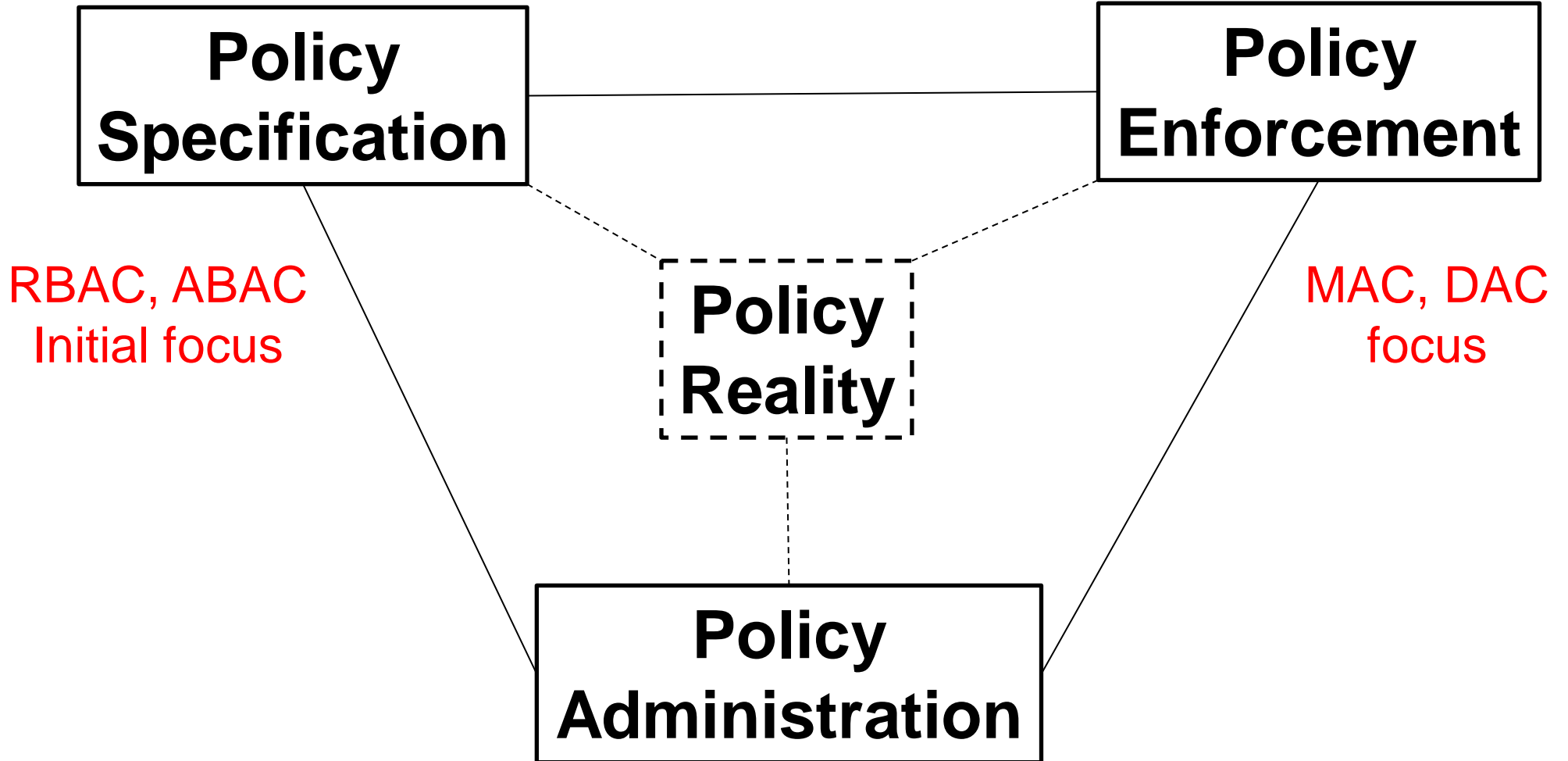
**RBAC is neither MAC nor DAC!**

- Role granularity is not adequate leading to role explosion
  - ❖ Researchers have suggested several extensions such as parameterized privileges, role templates, parameterized roles (1997-)
- Role design and engineering is difficult and expensive
  - ❖ Substantial research on role engineering top down or bottom up (1996-), and on role mining (2003-)
- Assignment of users/permissions to roles is cumbersome
  - ❖ Researchers have investigated decentralized administration (1997-), attribute-based implicit user-role assignment (2002-), role-delegation (2000-), role-based trust management (2003-), attribute-based implicit permission-role assignment (2012-)
- Adjustment based on local/global situational factors is difficult
  - ❖ Temporal (2001-) and spatial (2005-) extensions to RBAC proposed
- **RBAC does not offer an extension framework**
  - ❖ **Every shortcoming seems to need a custom extension**
  - ❖ **Can ABAC unify these extensions in a common open-ended framework?**

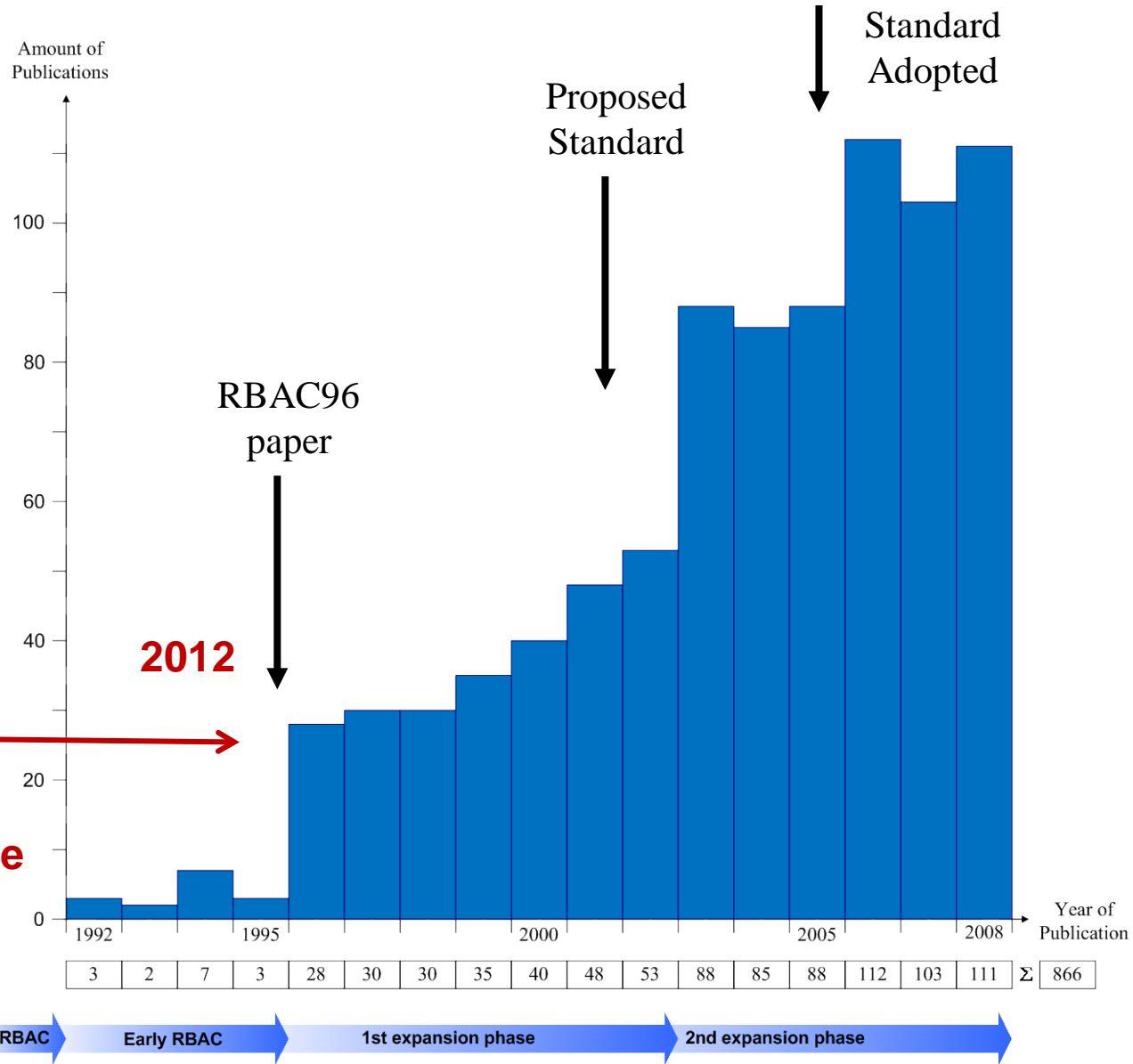
**Security Architect**







- Attributes are name:value pairs
  - ❖ possibly chained
  - ❖ values can be complex data structures
- Associated with
  - ❖ users
  - ❖ subjects
  - ❖ objects
  - ❖ contexts
    - device, connection, location, environment, system ...
- Converted by policies into rights just in time
  - ❖ policies specified by security architects
  - ❖ attributes maintained by security administrators
  - ❖ ordinary users morph into architects and administrators
- **Inherently extensible**



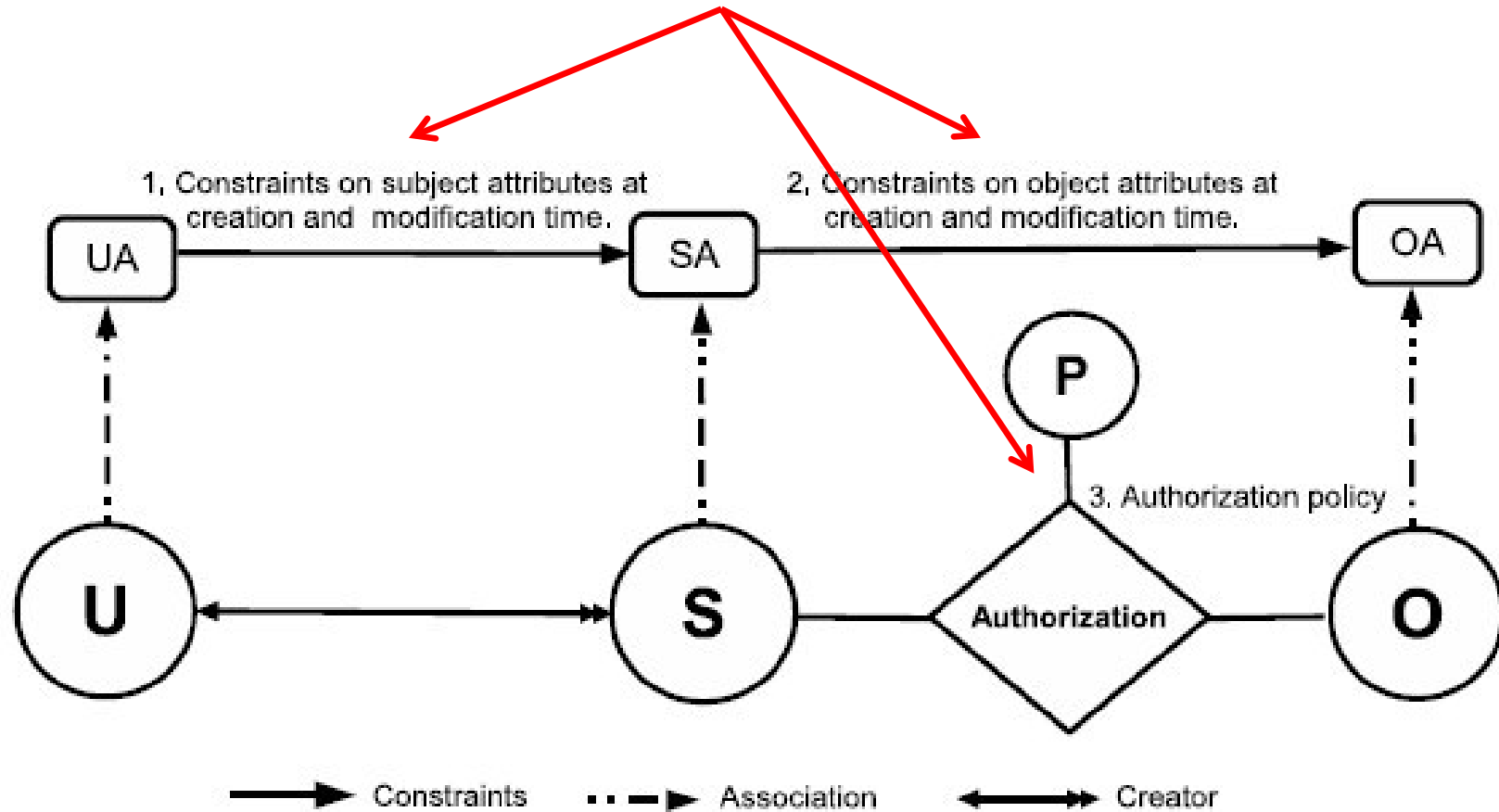
- X.509, SPKI Attribute Certificates (1999 onwards)
  - ❖ IETF RFCs and drafts
  - ❖ Tightly coupled with PKI (Public-Key Infrastructure)
- XACML (2003 onwards)
  - ❖ OASIS standard
  - ❖ Narrowly focused on particular policy combination issues
  - ❖ Fails to accommodate the ANSI-NIST RBAC standard model
  - ❖ Fails to address user subject mapping
- Usage Control or UCON (Park-Sandhu 2004)
  - ❖ Fails to address user subject mapping
  - ❖ Focus is on extended features
    - Mutable attributes
    - Continuous enforcement
    - Obligations
    - Conditions
- Several others .....



- An ABAC model requires
  - ❖ identification of policy configuration points (PCPs)
  - ❖ languages and formalisms for each PCP
- A core set of PCPs can be discovered by building the ABAC $\alpha$  model to unify DAC, MAC and RBAC
- Additional ABAC models can then be developed by
  - ❖ increasing the sophistication of the ABAC $\alpha$  PCPs
  - ❖ discovering additional PCPs driven by requirements beyond DAC, MAC and RBAC

A small but crucial step

## Policy Configuration Points



## ❖ DAC

$Authorization_{read}(s, o) \equiv SubCreator(s) \in reader(o)$

$Authorization_{write}(s, o) \equiv SubCreator(s) \in writer(o)$

## ❖ MAC

$Authorization_{read}(s, o) \equiv sensitivity(o) \leq sclearance(s)$

Liberal star :  $Aauthorization_{write}(s, o) \equiv sclearance(s) \leq sensitivity(o)$

Strict star :  $Aauthorization_{write}(s, o) \equiv sensitivity(o) = sclearance(s)$

## ❖ RBAC0

$Authorization_{read}(s, o) \equiv \exists r \in srole(s). r \in rrole(o)$

## ❖ RBAC1

$Authorization_{read}(s, o) \equiv \exists r1 \in srole(s). \exists r2 \in rrole(o). r2 \leq r1$

❖ MAC  $ConstrSub(u, s, \{(sclearance, value)\}) \equiv value \leq uclearance(u)$

❖ RBAC0  $ConstrSub(u, s, \{srole, value\}) \equiv value \subseteq urole(u)$

❖ RBAC1  $ConstrSub(u, s, \{srole, value\}) \equiv \forall r1 \in value. \exists r2 \in urole(u). r1 \leq r2$

## Constraints at creation: LConstrObj

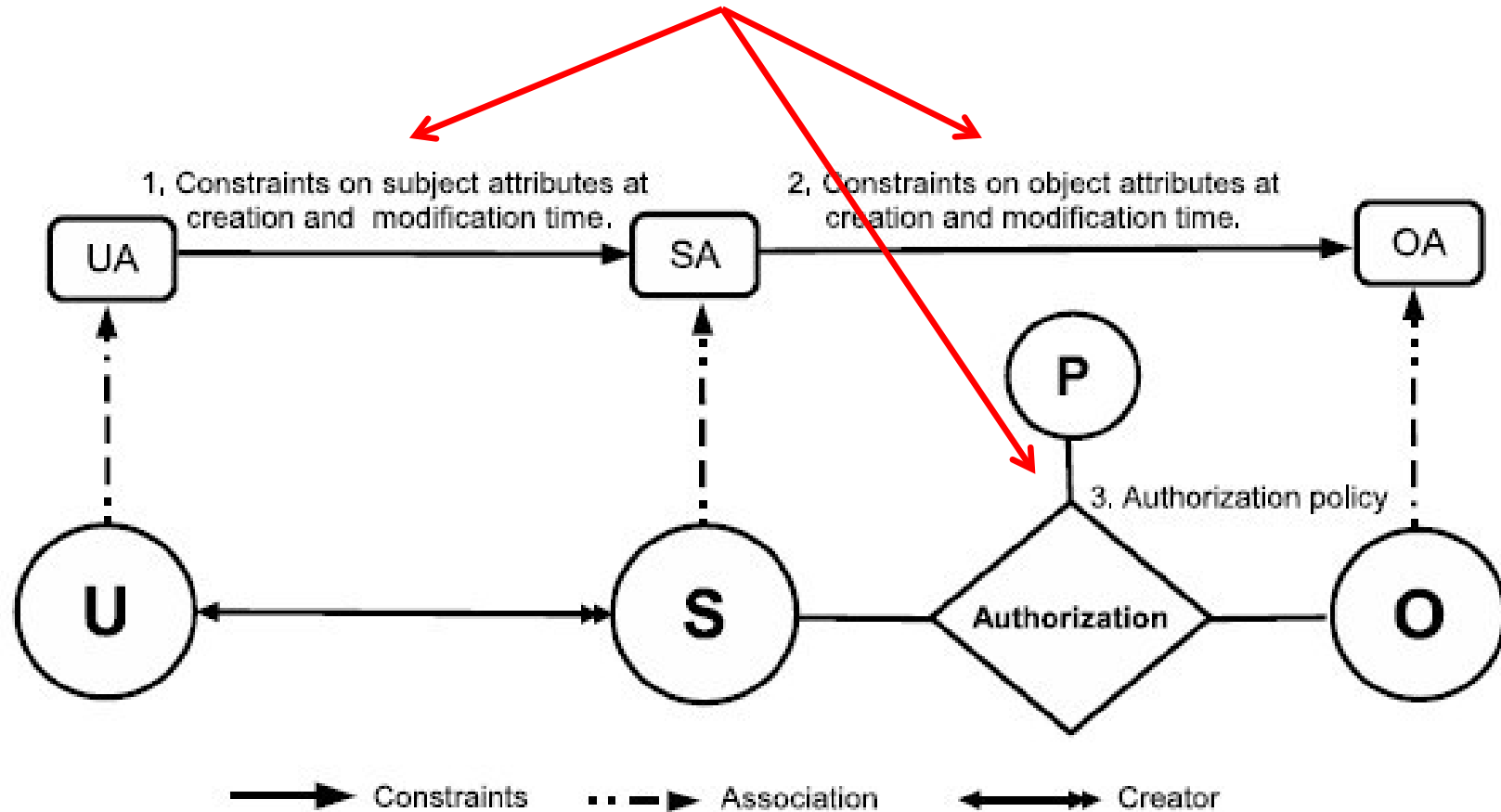
❖ DAC       $ConstrObj(s, o, \{(reader, val1), (writer, val2), (createdby, val3)\}) \equiv val3 = SubCreator(s)$

❖ MAC       $ConstrObj(s, o, \{sensitivity, value\}) \equiv sclearance(s) \leq value$

## Constraints at modification: LConstrObjMod

❖ DAC       $ConstrObj(s, o, \{(reader, val1), (writer, val2), (createdby, val3)\}) \equiv createdby(o) = SubCreator(s)$

## Policy Configuration Points



## Future work

- ❖ increasing the sophistication of the ABAC $\alpha$  PCPs
- ❖ discovering additional PCPs

7. ABAC Design and Engineering

5. ABAC  
Policy  
Languages

3. Administrative  
ABAC Models

4. Extended  
ABAC Models

6. ABAC  
Enforcement  
Architectures

2. Core ABAC Models

1. Foundational Principles and Theory

7. ABAC Design and Engineering

5. ABAC  
Policy  
Languages

3. Administrative  
ABAC Models

4. Extended  
ABAC Models

6. ABAC  
Enforcement  
Architectures

2. Core ABAC Models  
Initial Results

1. Foundational Principles and Theory



**7. Design and Engineering:**

**Role engineering:** Coyne (1996), Thomsen et al (1999), Epstein-Sandhu (2001), Strembeck (2005)

**Role mining:** Kuhlmann-Schimpf (2003), RoleMiner (2006, 2007), Minimal Perturbation (2008)

**5. Policy Languages**

**Constraints:** RCL (2000), Jaeger-Tidswell (2001), Crampton (2003), ROWLBAC (2008)

**User-role assignment:** RB-RBAC (2002), RT (2003)

**3. Administrative Models:** ARBAC97 (1997), RBDM (2000), RDM (2000), RB-RBAC (2002), ARBAC02 (2002), PBDM (2003) ARBAC07 (2007), SARBAC (2003, 2007)

**4. Extended Models:** TMAC (1997) Workflow (1999), T-RBAC (2000), OrBAC (2003), TRBAC (2001), RT (2003), GTRBAC (2005), GEO-RBAC (2005), P-RBAC (2007)

**6. Enforcement Architectures:** Ferraiolo et al (1999), OM-AM (2000), Park et al (2001), xoRBAC (2001), RCC (2003), RB-GACA (2005), XACML Profiles (2004, 2005, 2006)

**2. Core Models:** RBAC96 (1996), ANSI-NIST Standard (2000, 2004)

**1. Foundational Principles and Theory**

**Principles:** RBAC96 (1996), OM-AM (2000), NIST Standard (2000, 2004), PEI (2006), ASCAA (2008)

**Theory:** ATAM Simulation (1999), LBAC-DAC Simulations (2000), Li-Tripunitara (2006), Stoller et al (2006, 2007), Jha et al (2008)

**NOTE: Only a small sampling of the RBAC literature is cited in this diagram**

## Rights to attributes

- ❖ Rights
- ❖ Labels
- ❖ Roles
- ❖ Attributes

**Messy** ← **??** → **Chaotic**

## Benefits

- ❖ Decentralized
- ❖ Dynamic
- ❖ Contextual
- ❖ Consolidated

## Risks

- ❖ Complexity
- ❖ Confusion
- ❖ Attribute trust
- ❖ Policy trust

- Attributes
- Automated
- Adaptive
  
- Managed but not solved