The Authorization Leap from Rights to Attributes: Maturation or Chaos?

Prof. Ravi Sandhu
Executive Director and Endowed Chair

SACMAT
June 21, 2012

ravi.sandhu@utsa.edu
www.profsandhu.com
www.ics.utsa.edu

Access Control Status

- Dozens of models proposed and studied. Only three winners (meaningful practical traction)
  - DAC: Discretionary Access Control, 1970
  - MAC: Mandatory Access Control, 1970
  - RBAC: Role-Based Access Control, 1995
- RBAC emerged at an inflection point due to dissatisfaction with the then dominant DAC and MAC
  - We are currently at another inflection point due to dissatisfaction with the now dominant RBAC
  - ABAC (Attribute-Based Access Control) has emerged as the prime candidate to be the next dominant paradigm

ABAC = Final Word?

- NO!! Never!!
- Is ABAC the right word for the moment?
  - Certainly a strong candidate
  - Already too late?
    - ReBAC (relationship-based access control) not ABAC
    - Big Data, Analytics and AI will take care of everything
- ABAC is exponentially more complex than anything that has been an Access Control winner so far (DAC, MAC, RBAC)
  - We need the complexity, but need to manage it
  - If Google can index the web, we can do ABAC!!

Attribute-Based Access Control (ABAC)

- Attributes are name:value pairs
  - possibly chained
- 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

Authorization Leap

Rights to attributes

- Rights
- Labels
- Roles
- Attributes

Maturation ← ?? → **Chaos**

Benefits

- Decentralized
- Dynamic
- Contextual
- Consolidated

Risks

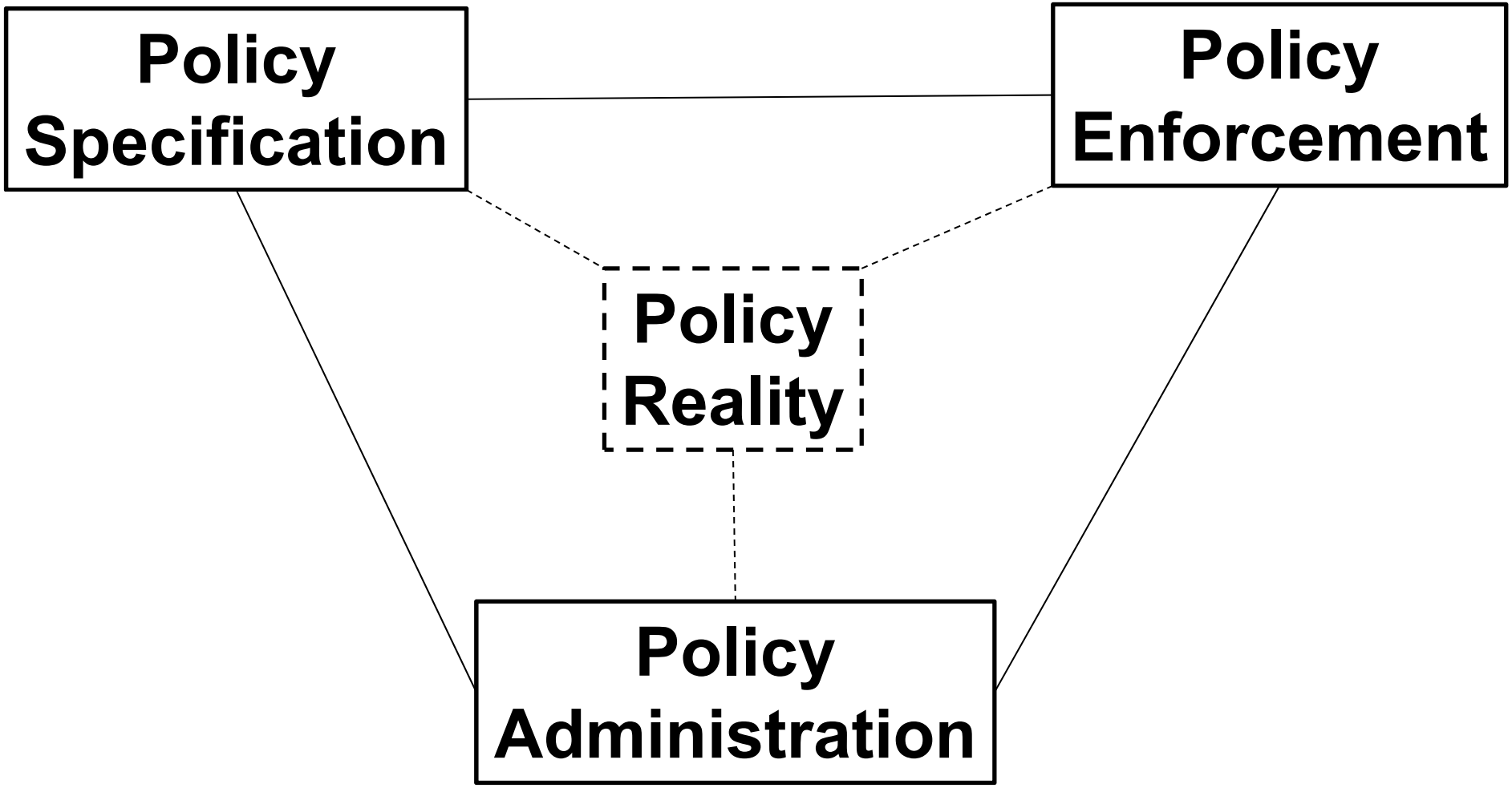
- Complexity
- Confusion
- Attribute trust
- Policy trust

Prognosis

- Cyber technologies and systems trends will drive pervasive adoption of ABAC
  - RBAC is simply not good enough
- ABAC deployment is going to be messy but need not be chaotic

- Researchers can facilitate ABAC adoption and reduce chaos by developing
  - Models
  - Theories
  - Systems

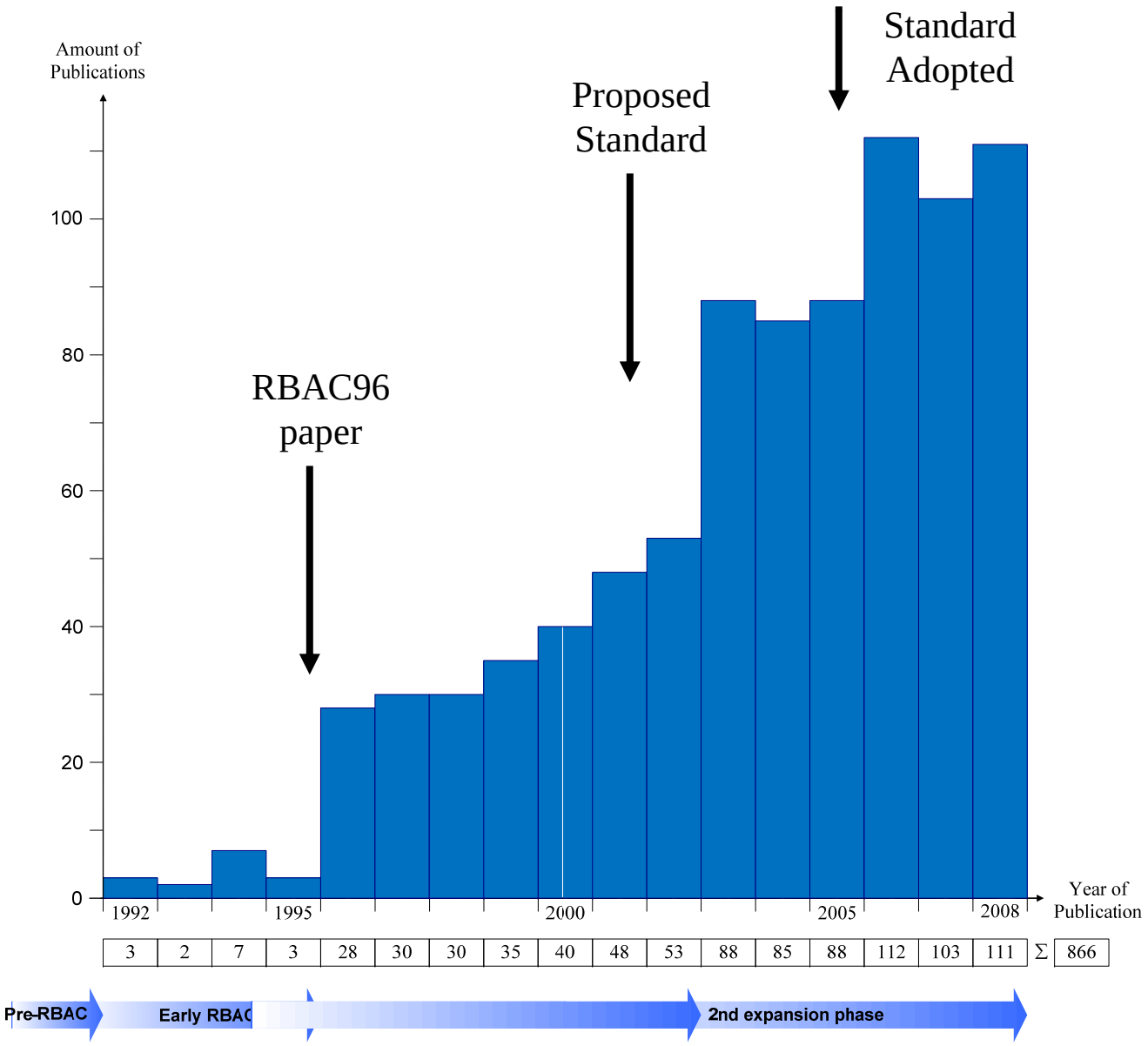
Authorization Challenges



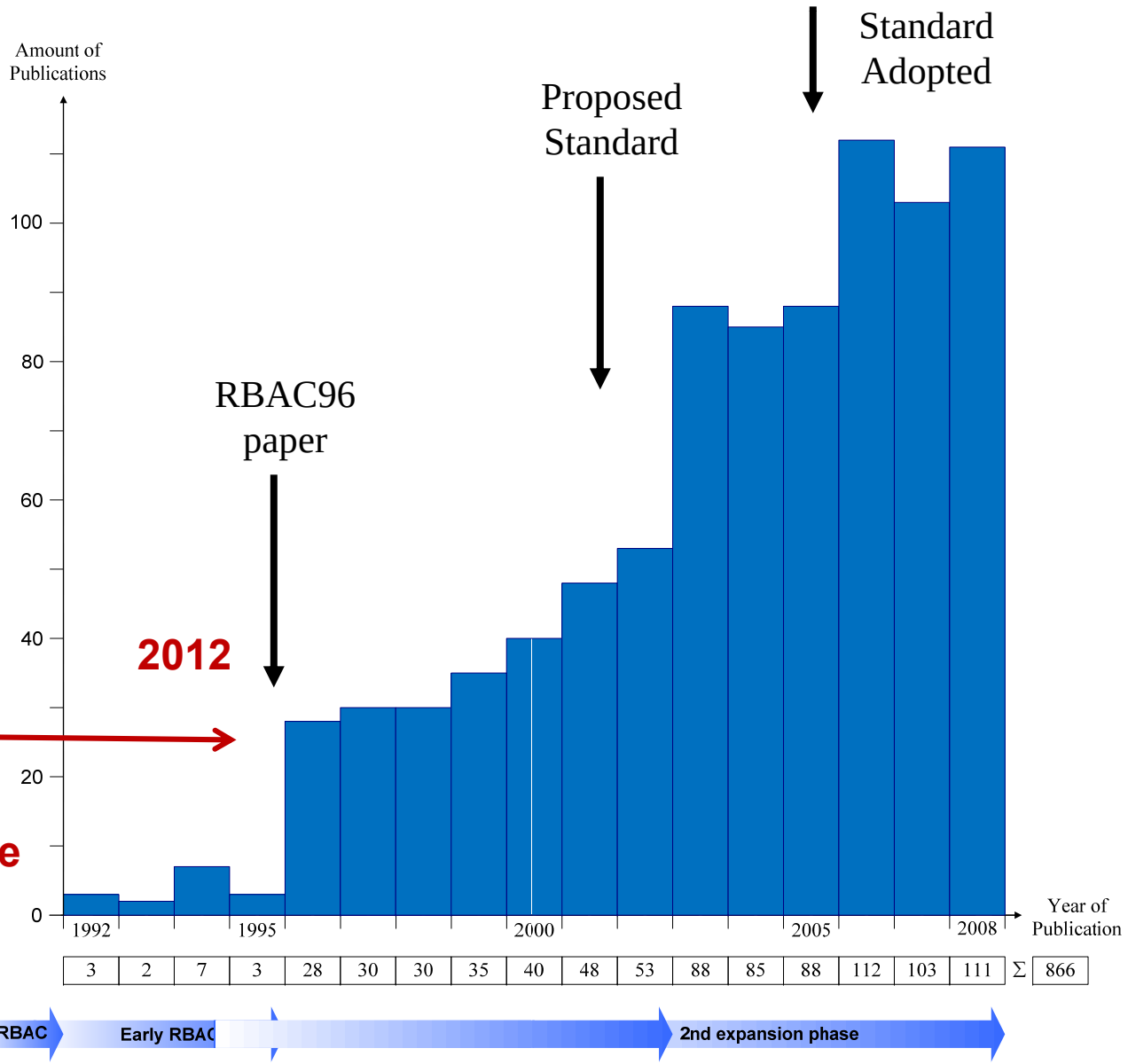
Policy Reality

- Analog Hole
- Inference
- Covert Channels
- Side Channels
- Spoofing
- Attack Asymmetry
- Compatibility
- ….

The RBAC Story



ABAC Status

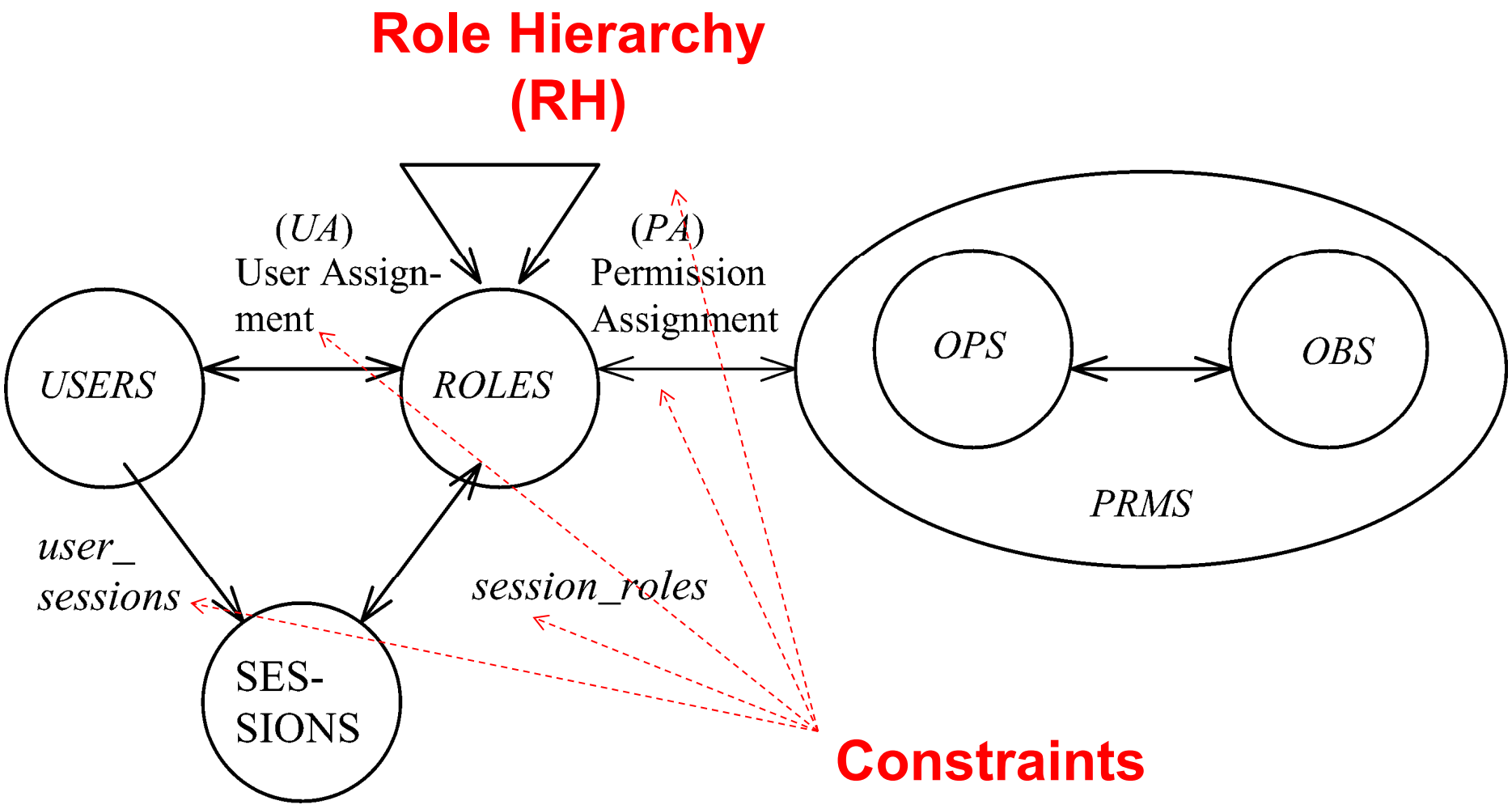


1990?

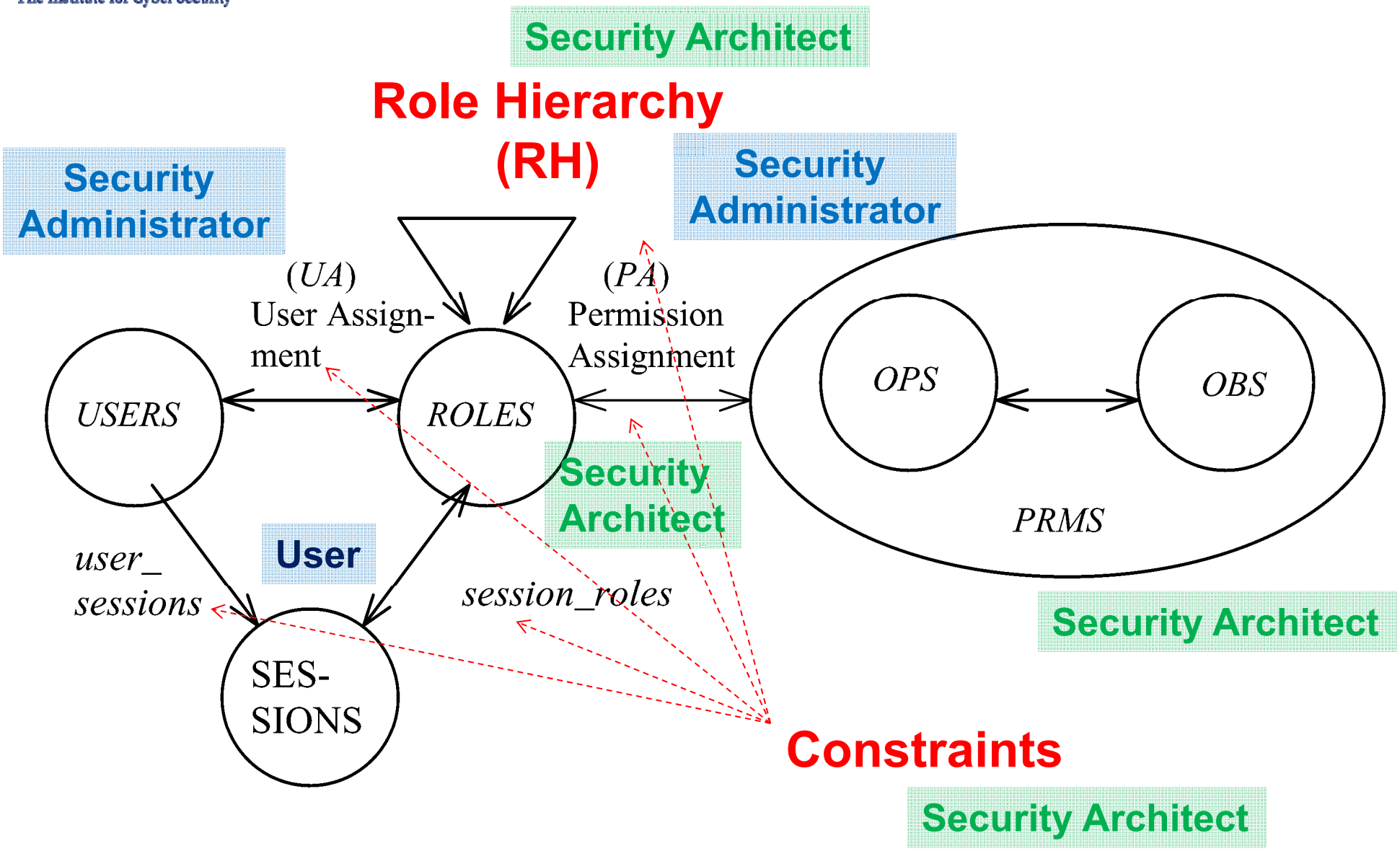
2012

ABAC still in pre/early phase

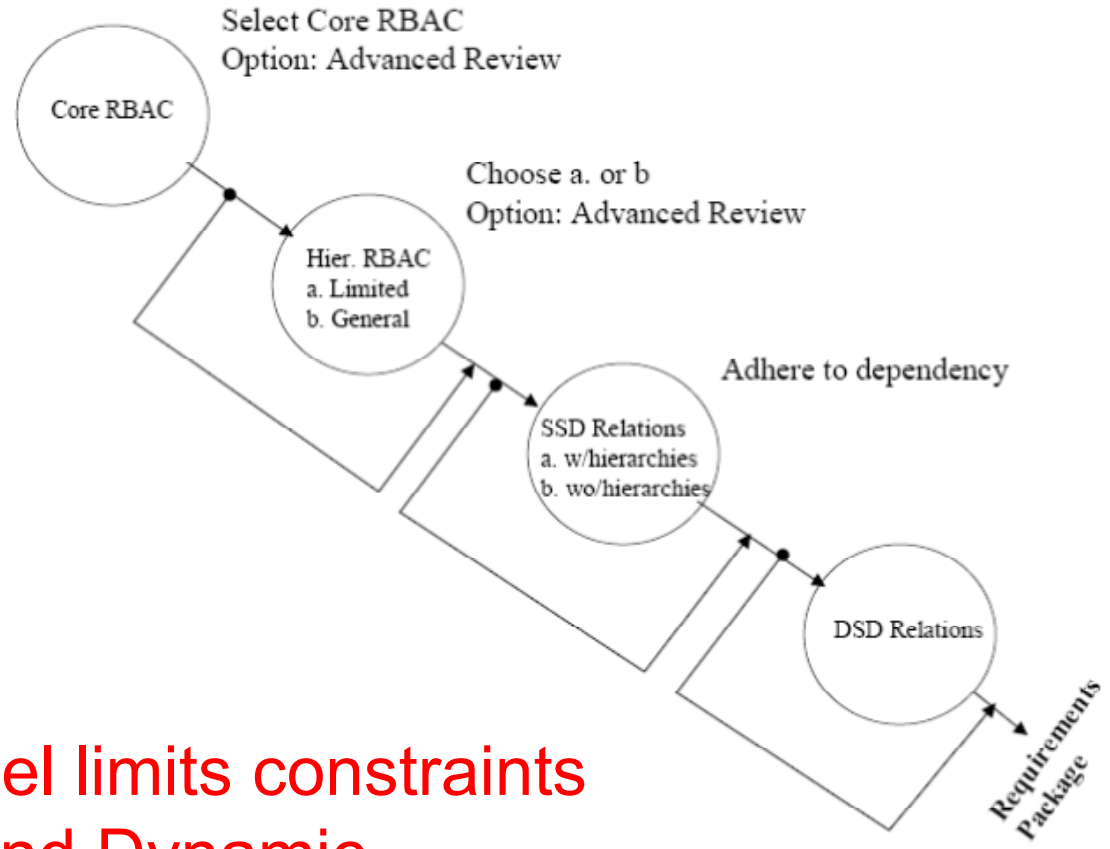
RBAC Policy Configuration Points



RBAC Policy Configuration Points

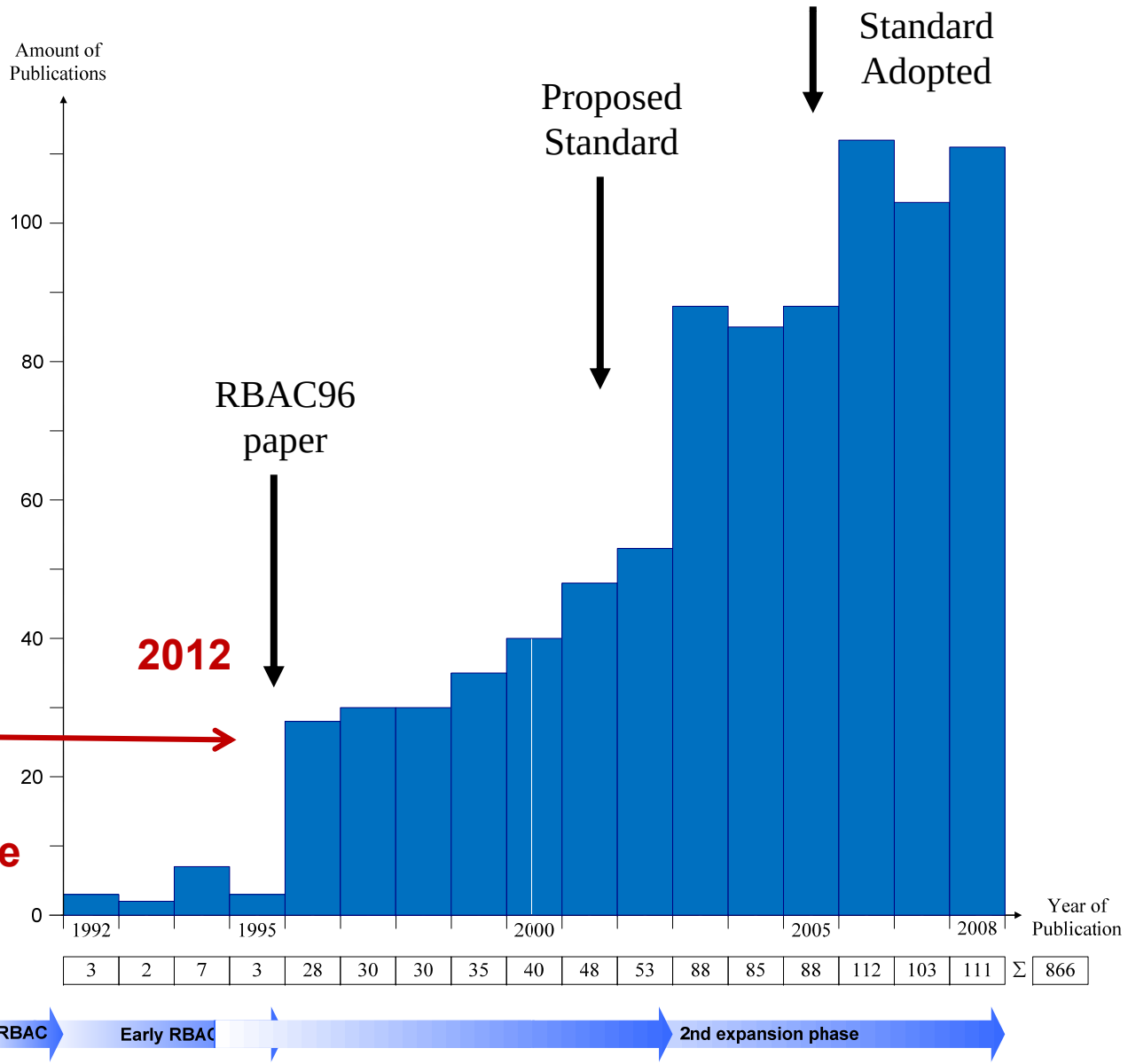


RBAC Policy Configuration Points



NIST model limits constraints to Static and Dynamic Separation of Duties

ABAC Status



1990? **2012**

ABAC still in pre/early phase

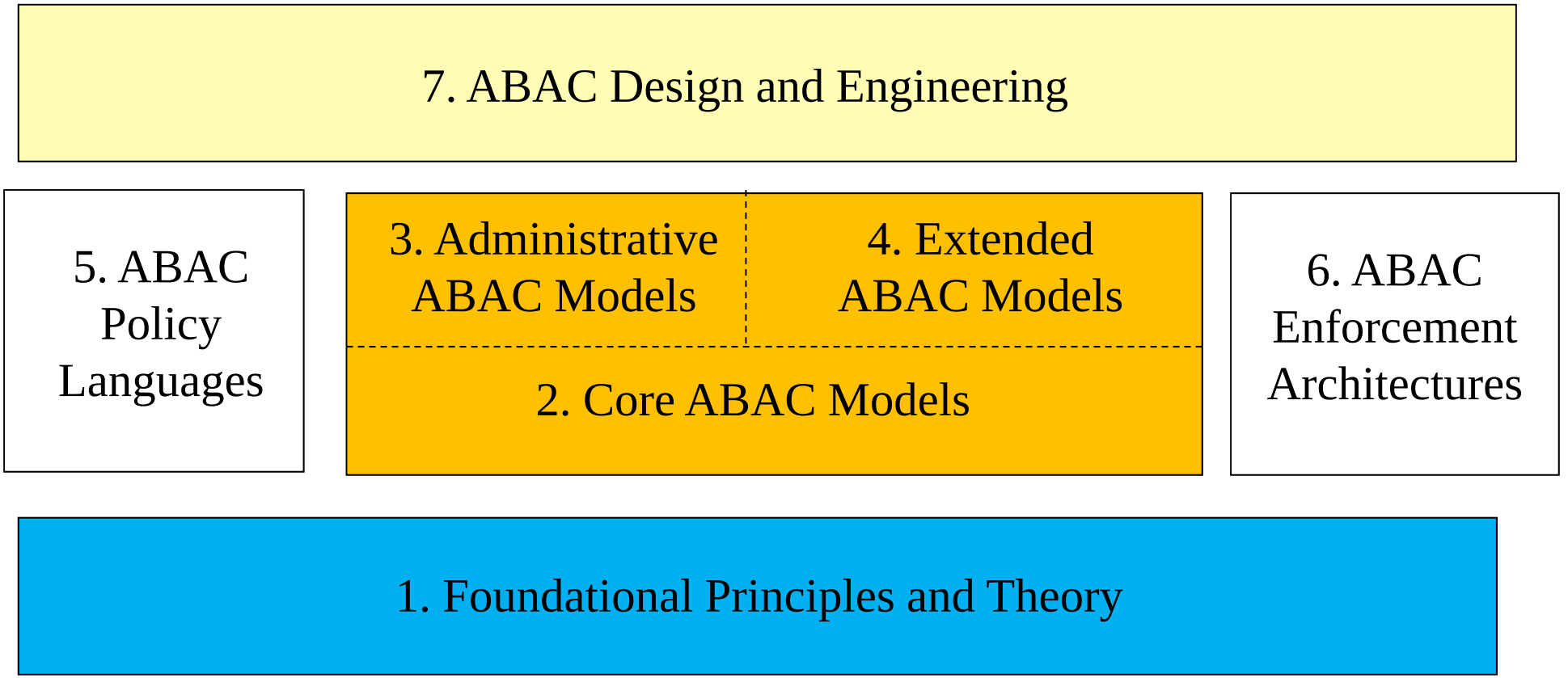
ABAC Prior Work Includes

- 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

RBAC Overall Assessment

- 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?

ABAC Research Agenda



ABAC Research Agenda: RBAC Inspiration

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)

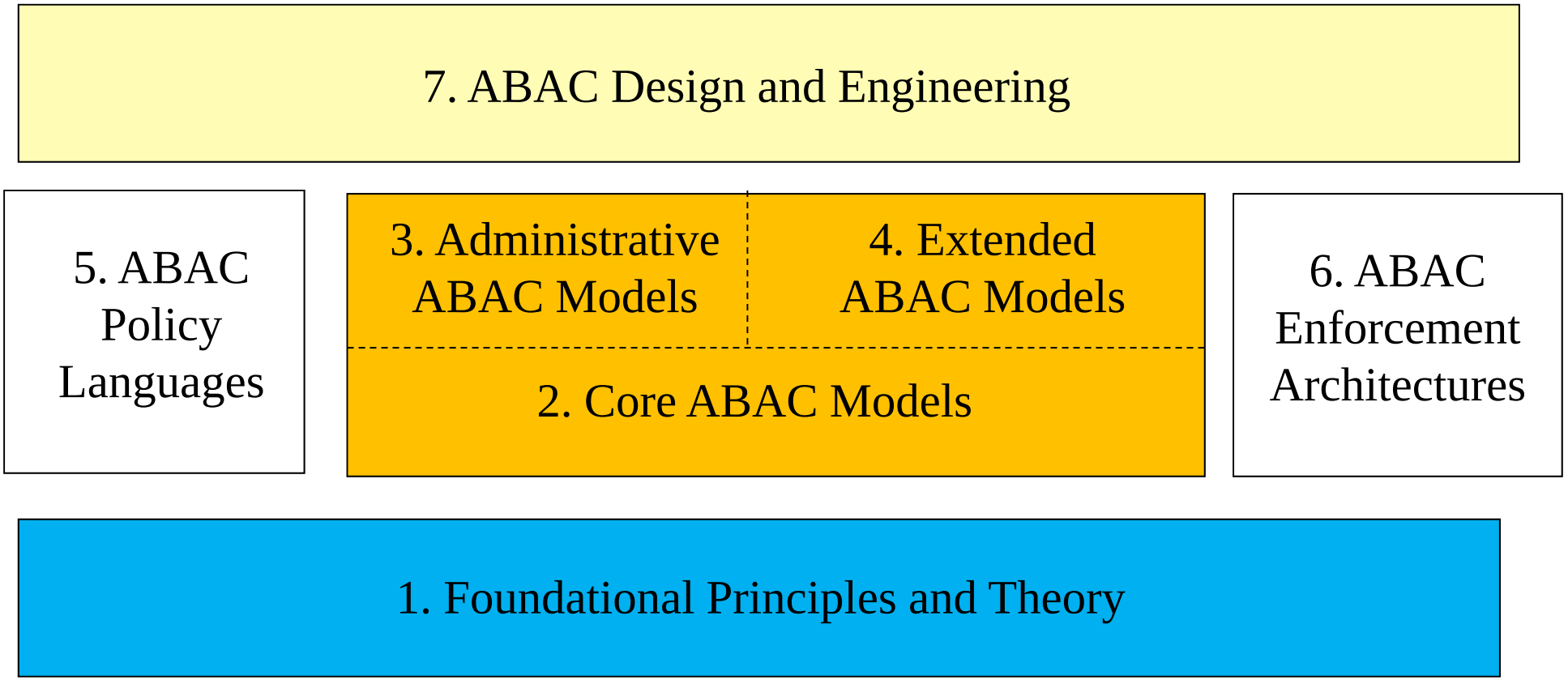
2. Core Models: RBAC96 (1996), ANSI-NIST Standard (2000, 2004)

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)

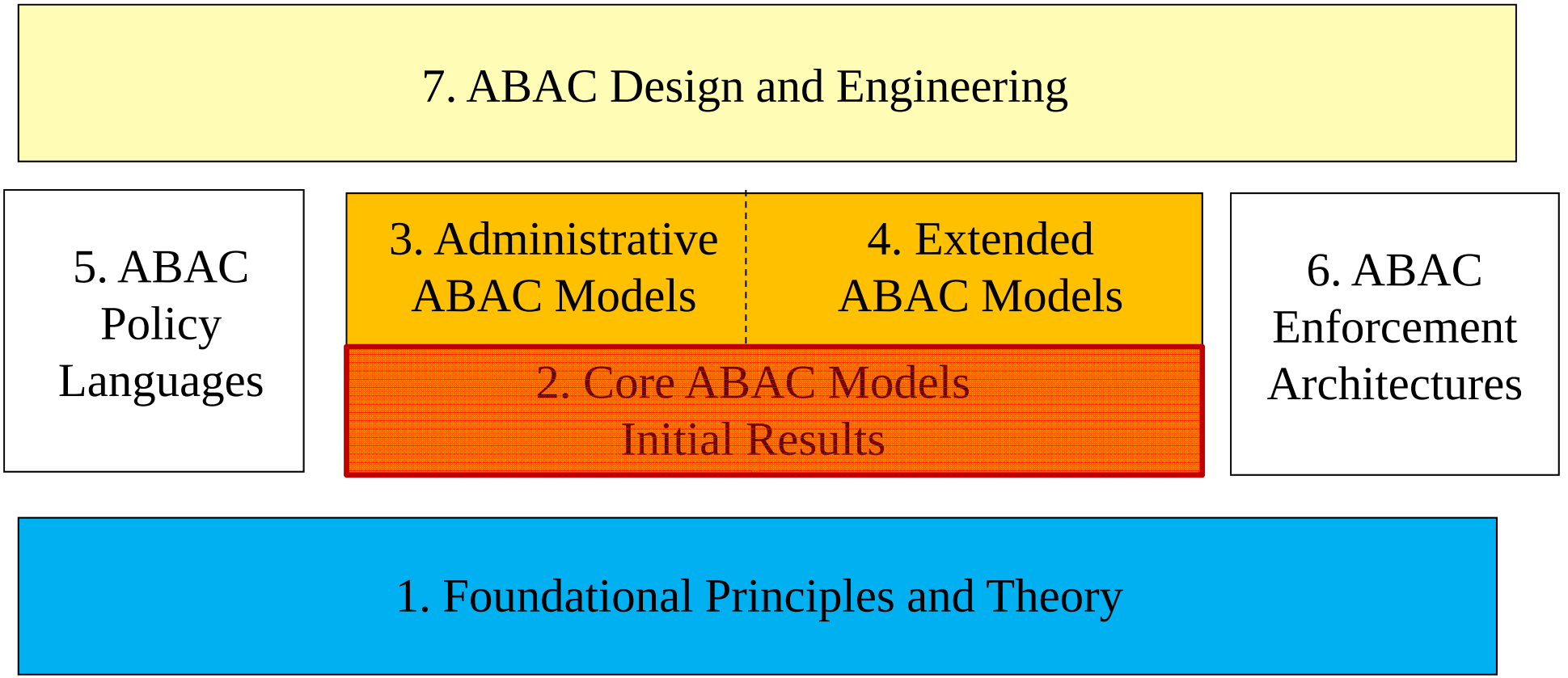
1. Foundational Principles and Theory
Principles: RBAC96 (1996), OM-AM (2000), NIST Standard (2000, 2004), PEI (2006), ASCAA (2008)
Theory: ATAM Simulation (1999), LBAC-DAC Simulations (2000), Li-Tripunitara (2006), Stoller et al (2006, 2007), Jha et al (2008)

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

ABAC Research Agenda



ABAC Research Agenda



ABAC Core Models

- Approach this challenge from several perspectives
- Initial results on a bottom-up approach

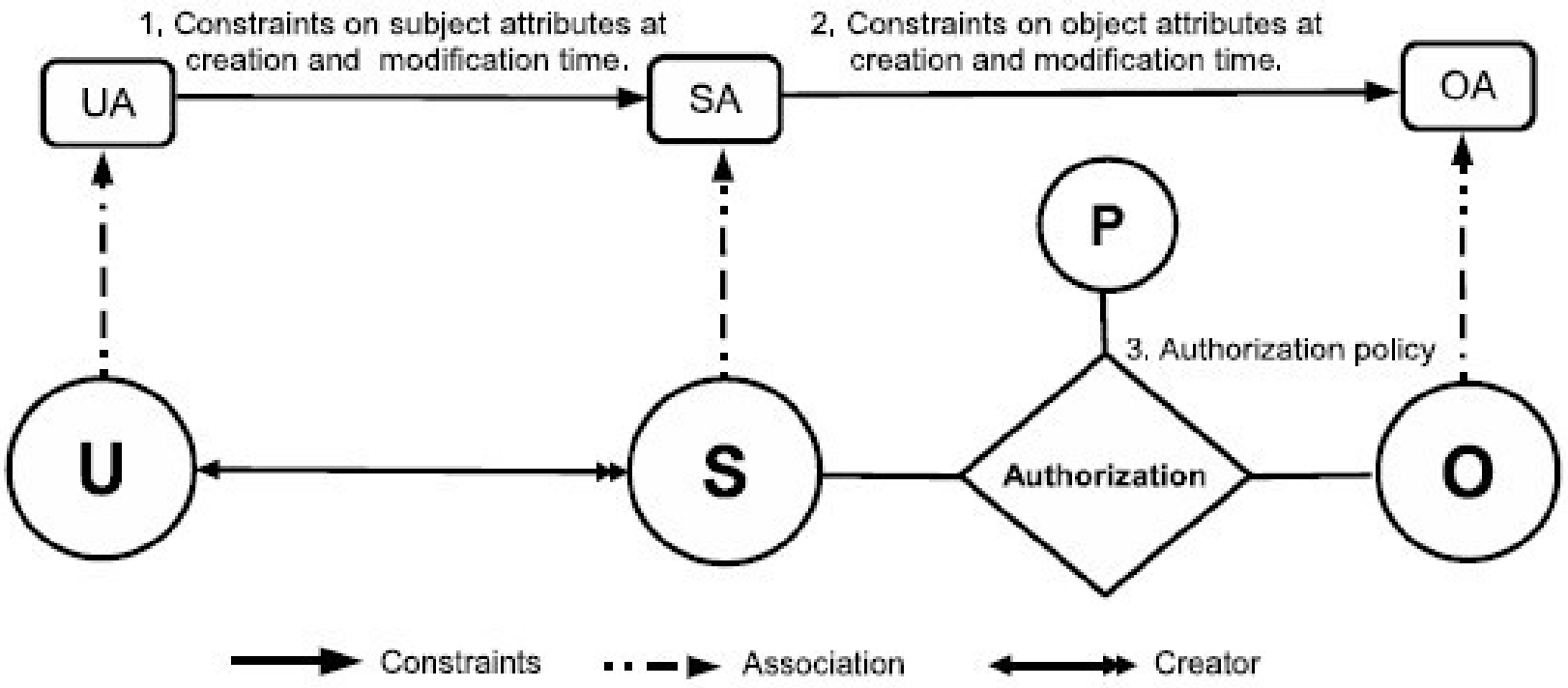
- ABACα model (DBSEC 2012)
  - Just sufficient to cover the core of DAC, MAC and RBAC
  - No extraneous features (however attractive and desirable)

- ABACβ model (in progress)
  - Grow ABACα to accommodate additional models, including numerous RBAC extensions and RBAC-related models (e.g. RT)

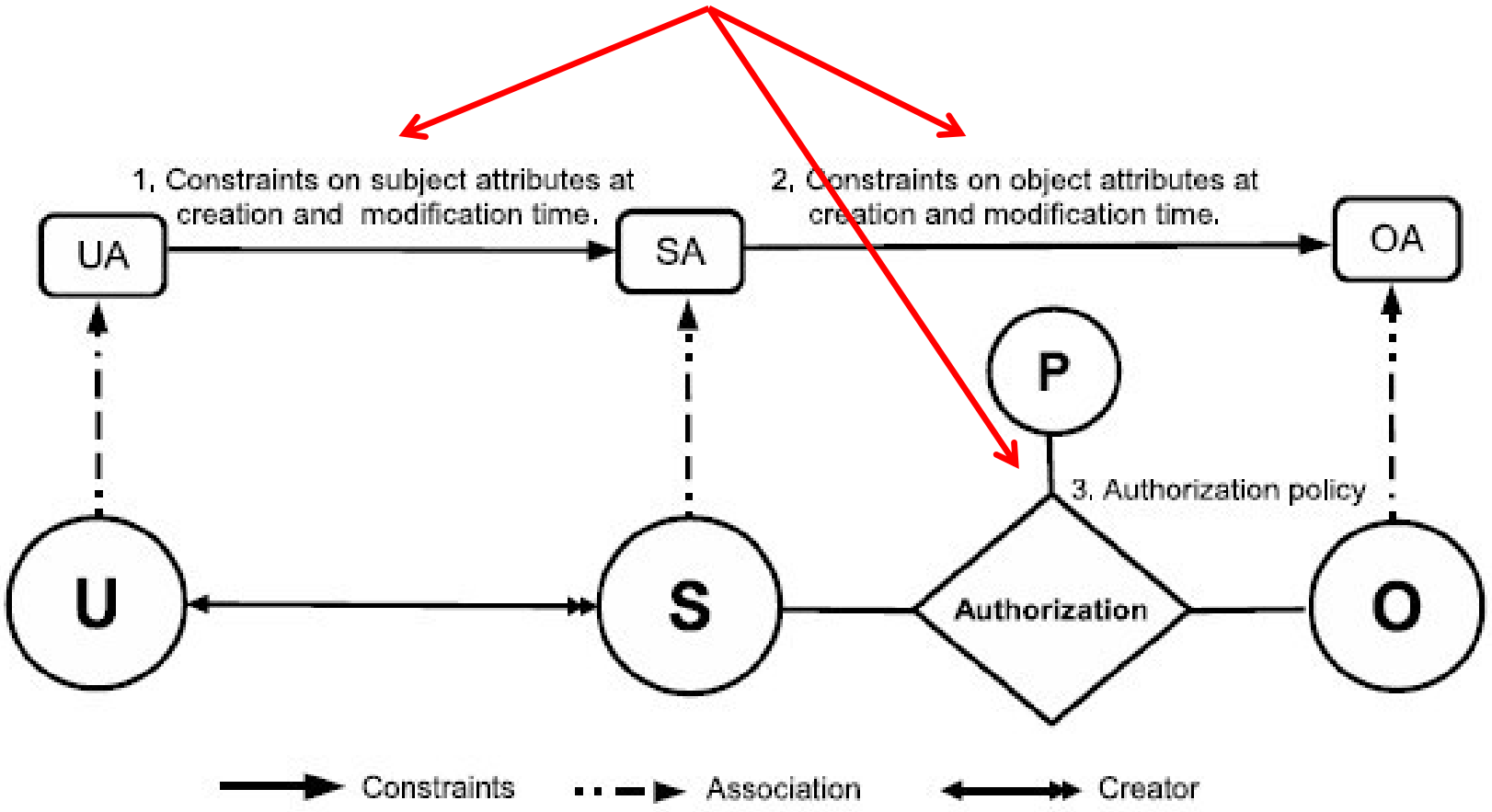
ABACα Requirements

	Subject attribute values constrained by creating user?	Object attribute values constrained by creating subject?	Attribute range ordered?	Attribute functions return set value?	Object attributes modification?	Subject attribute modification by creating user?
DAC	YES	YES	NO	YES	YES	NO
MAC	YES	YES	YES	NO	NO	NO
$RBAC_0$	YES	NA	NO	YES	NA	YES
$RBAC_1$	YES	NA	YES	YES	NA	YES
$ABAC_\alpha$	YES	YES	YES	YES	YES	YES

ABACα Model Structure



ABACα Model Structure



ABACα Basic Sets and Functions

U, S and O represent finite sets of *existing* users, subjects and objects respectively.

UA, SA and OA represent finite sets of user, subject and object attribute functions respectively. (Henceforth referred to as simply attributes.)

P represents a finite set of permissions.

For each *att* in UA ∪ SA ∪ OA, Range(*att*) represents the attribute's range, a finite set of *atomic* values.

SubCreator: $S \rightarrow U$. For each subject SubCreator gives its creator.

attType: UA ∪ SA ∪ OA → {set, atomic}. Specifies attributes as set or atomic valued.

Each attribute function maps elements in U, S and O to atomic or set values.

$$\forall ua \in \text{UA}.\, ua : \text{U} \rightarrow \begin{cases} \text{Range(ua)} & \text{if attType}(ua) = \text{atomic} \\ 2^{\text{Range(ua)}} & \text{if attType}(ua) = \text{set} \end{cases}$$

$$\forall sa \in \text{SA}.\, sa : \text{S} \rightarrow \begin{cases} \text{Range(sa)} & \text{if attType}(sa) = \text{atomic} \\ 2^{\text{Range(sa)}} & \text{if attType}(sa) = \text{set} \end{cases}$$

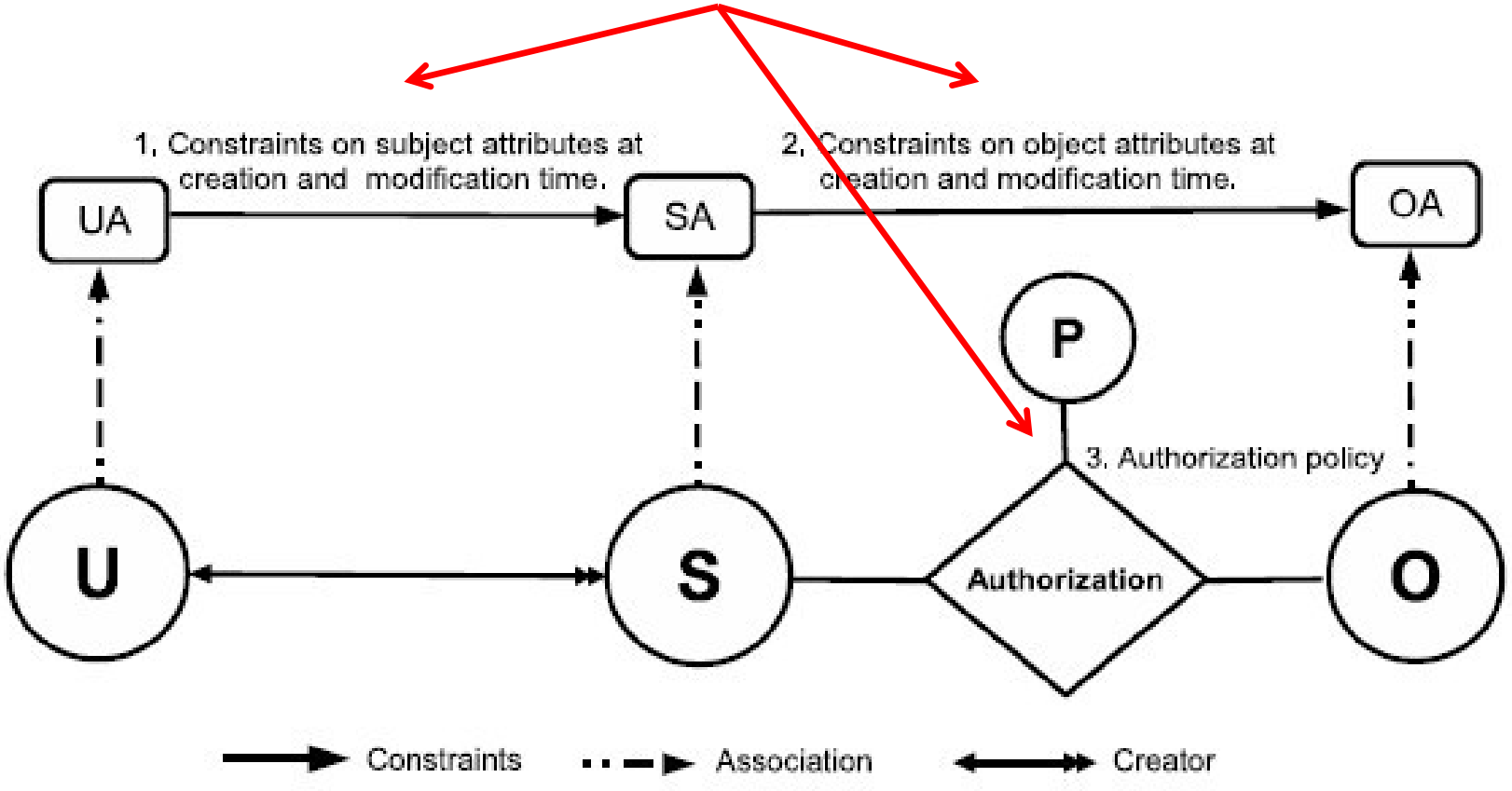
$$\forall oa \in \text{OA}.\, oa : \text{O} \rightarrow \begin{cases} \text{Range(oa)} & \text{if attType}(oa) = \text{atomic} \\ 2^{\text{Range(oa)}} & \text{if attType}(oa) = \text{set} \end{cases}$$

ABACα Additional Components

- Administrative Functions
  - AddUser (u:NAME,uaset:UASET)
  - DeleteUser (u:NAME)
  - ModifyUserAtt (u:NAME,uaset:UASET)
- System Functions
  - CreateSubject (u; s:NAME,saset:SASET)
  - DeleteSubject (u; s:NAME)
  - ModifySubjectAtt (u; s:NAME,saset:SASET)
- Review Functions
  - UserAttributes (u:NAME)
  - UserOperationsOnObject (u,o: NAME)
  - AssignedUser(ua: NAME, value: Range(ua))
  - UserPermissions(u: NAME)
  - SubjectPermissions(s: NAME)
- Policy Configuration Languages

ABACβ Model Structure

Policy Configuration Points
Same as ABACα



Enrich other ABACα Components

ABACβ Examples

Model	Description	Extension-Specific	Extension-General
Attribute based User-Role Assignment [1]	Roles are computed from user attribute based on predefined rules	Subject attribute constrained by user attribute	Extensions to languages for specifying subject attribute constraints.
Role based Trust Management [2]	Each entity has role. Role is managed by the owned entity.	Chained attribute, distinguished attribute intentional vs extensional representation.	Extensions to nature of attribute.
Role and Organization based Access Control [3]	User is assigned to role and organization pair.	Attribute should be able to represent pair-value (org, role).	Extensions to nature of attribute.

[1]. MA Al-Kahtani and R. Sandhu. A model for attribute-based user-role assignment. ACSAC, 2002.
[2]. Ninghui Li, John C. Mitchell, and William H. Winsborough. Design of a role-based trust management framework. IEEE S&P 2002 .
[3]. Zhixiong Zhang, Xinwen Zhang and Ravi Sandhu, ROBAC: Scalable Role and Organization Based Access Control Models, TrustCol 2006

ABAC Research Agenda