Remarks on Authorization and Accountability in DIG Projects

Joan Feigenbaum

http://www.cs.yale.edu/homes/jf

Madison, WI; July 29, 2009
Standard Authorization

user

\((r, \pi)\) Y/N

reasoner

P

resource

justification

resource

ID Loc. Size ID Age Sal Dept.

Department Employee
Authorization Research

• Technical challenges
  – Policy languages and query languages
  – Logic for and reasoning about compliance
  – Human-readable justifications
  – Evidence-based policy revision

• This is a **preventive** approach to policy compliance: Actions that cannot be **authorized before the fact** should be prevented.
Prevention is Inadequate

• Examples
  – Emergency medical
  – Battlefield
  – Counter-terrorism and law enforcement
  – Retail banking
  – Web crawling

• Reasons
  – Complex, hard-to-formulate policies
  – Inaccessible proofs of compliance
  – Computationally expensive decision procedures
Examples in DIG Projects

- Logging, analysis, and revision of policies and queries
  - Policy assurance in PIR
  - Data exchange in Fusion Centers
- Flagging but not stopping non-compliant actions
  - Policy-aware mashups
  - License validation in Creative Commons
  - Social-web privacy
- Similar experiences with policy compliance in earlier DIG projects: TAMI (NSF), PAW (NSF), and E2ESA (IARPA)
Multistage Authorization

ACCEPT       ADJUDICATE       REJECT

\[ (r_i, \pi_i) \]

\[ (r_1, \pi_1) \]

\[ v_o : \text{START} \]
Two Properties of “Accountable Systems”?

- Finite number of steps to a decision:
  For all requests \((r_0, \pi_0)\) and all policies \(P\), all execution paths are finite and end at a terminal node.

- Best effort to authorize:
  For all \((r_i, \pi_i)\), all policies \(P\), and all non-terminal nodes \(v_i\), if there is a path to the ACCEPT node, then \([(r_{i+1}, \pi_{i+1}), v_{i+1}]\) must be a next hop on one such path.