A Policy Framework for Security and Privacy Management

John Karat[1], Clare-Marie Karat[1], Elisa Bertino[2], Ninghui Li[2], Qun Ni[2], Carolyn Brodie[1], Jorge Lobo[1], Seraphin Calo[1], Lorrie Cranor[3], Ponnurangam Kumaraguru[3], and Robert Reeder[3]

IBM TJ Watson Research[1], Purdue University[2], Carnegie Mellon University[3]

Presented by: Monika Akbar

Overview

- Introduction
- Relating Privacy & Security
- Framework for managing privacy & security
- Example
- Conclusion

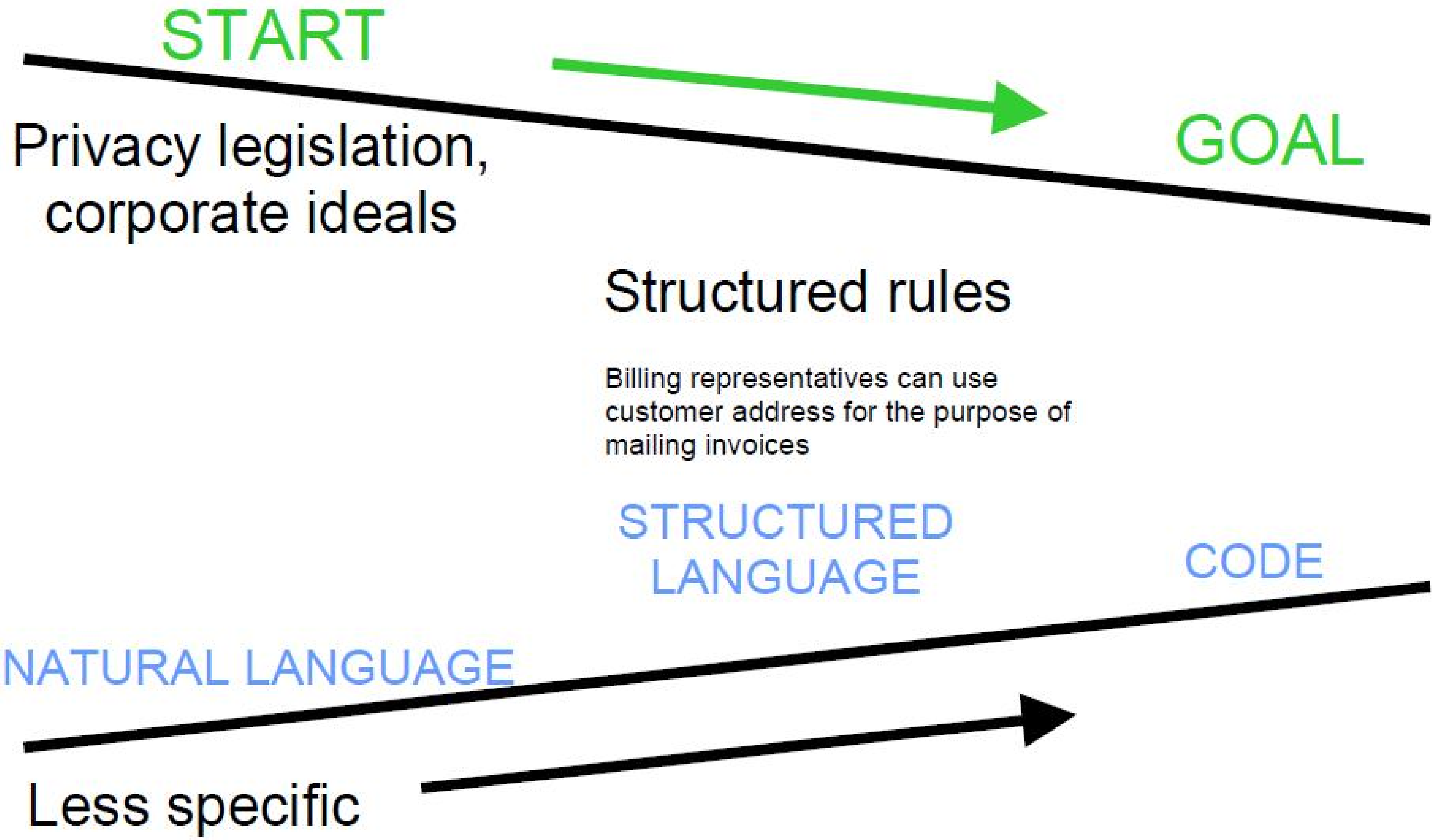
Introduction

- Policy
  - In IT
    - Who can access what – to protect the integrity & confidentiality of information and resources
  - In social systems
    - Proper conduct – to protect the safety of people and effective use of resources

Relating Security and Privacy

- Security
  - Protect from unauthorized use
  - Main focus – Access to information
- Privacy
  - Storage of personal information
  - Appropriate use of personal information
- To protect the privacy, we need security

End-to-end Policy Management



START

Privacy legislation,
corporate ideals

Structured rules

Billing representatives can use customer address for the purpose of mailing invoices

GOAL

NATURAL LANGUAGE

STRUCTURED
LANGUAGE

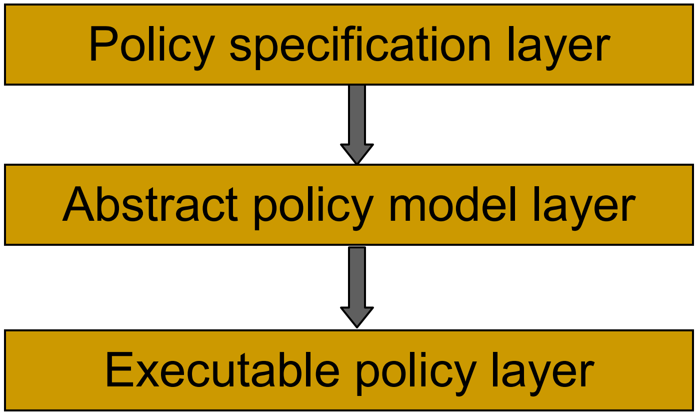
CODE

Less specific

Policy Management Framework

- Three levels of abstraction
- Transformation between them
- Issues discussed here:
  - Brief details of each level
  - Policy Analysis and Ratification
  - End User issues

Abstract Framework



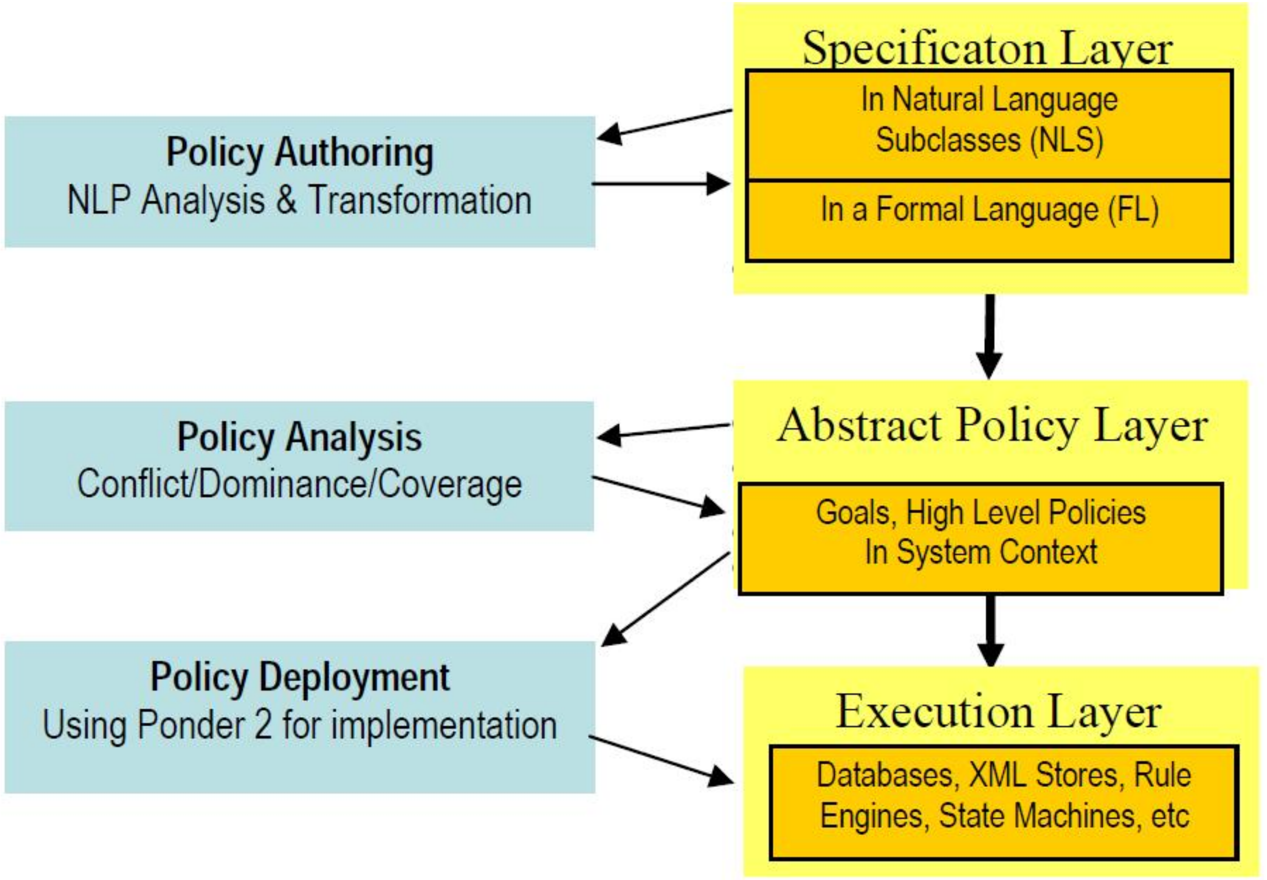
Policy specification layer

Abstract policy model layer

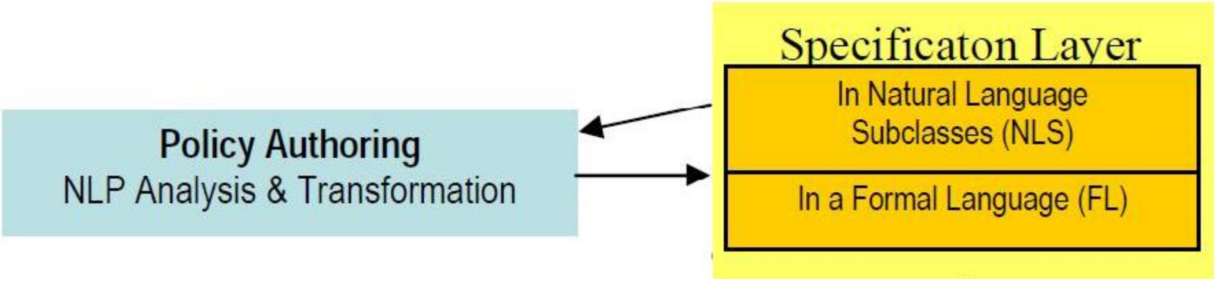
Executable policy layer

- Objective
  - Identify characteristics of each layer
    - Function, input, output
  - Specify elements of refinement process

Security and Privacy Policy Framework

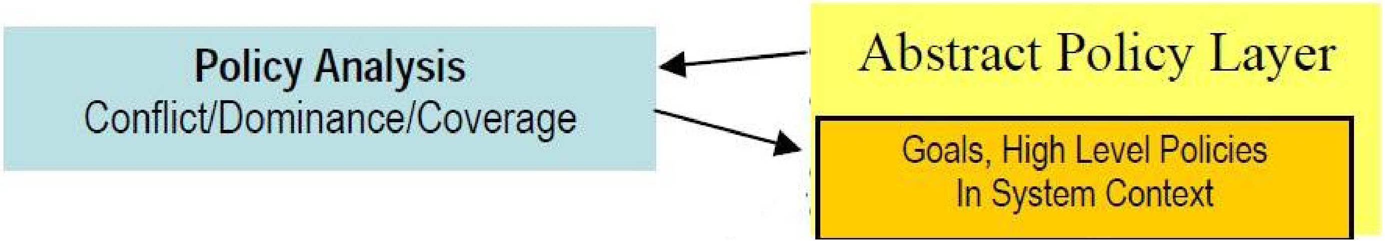

Policy Authoring
NLP Analysis & Transformation
Specificaton Layer
In Natural Language Subclasses (NLS)
In a Formal Language (FL)
Policy Analysis
Conflict/Dominance/Coverage
Abstract Policy Layer
Goals, High Level Policies In System Context
Policy Deployment
Using Ponder 2 for implementation
Execution Layer
Databases, XML Stores, Rule Engines, State Machines, etc

Specification Layer


Specificaton Layer
In Natural Language Subclasses (NLS)
In a Formal Language (FL)
Policy Authoring
NLP Analysis & Transformation

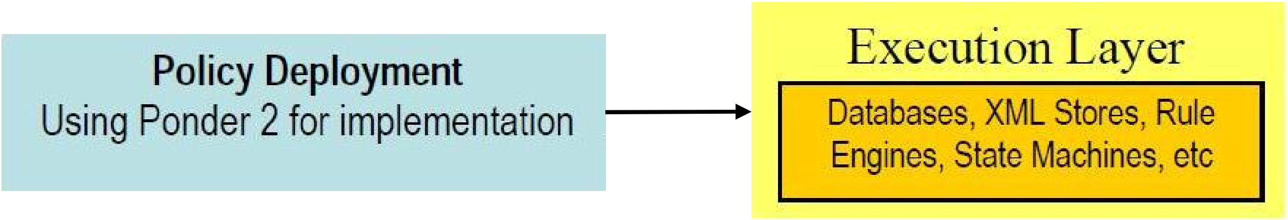
- Authoring policy
  - Capture the structure & syntax in a formal manner
- Input – policy specification from user
- Output – automatic transformation to formal language.
- Some existing techniques include
  - Item selection from structured list
  - Graphical rule selection methods
  - Constrained natural language authoring

Abstract Policy Layer



- Goal and high level objectives of the system
- Policy analysis
  - Conflict, dominance, coverage
  - Suggestions for resolving conflicts
- Policy transformation

Execution Layer

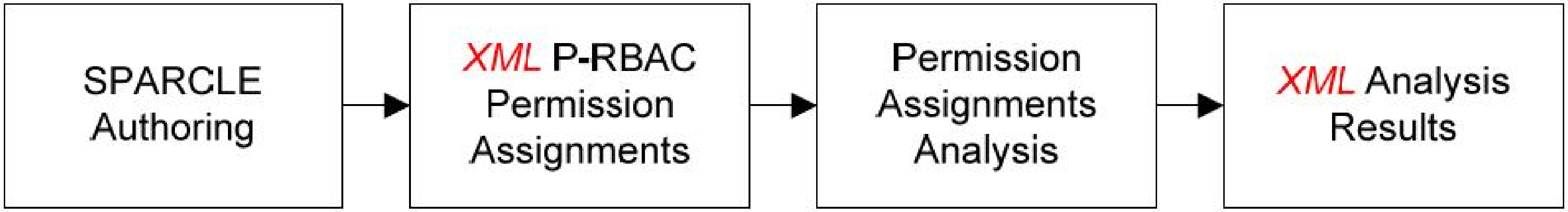

Policy Deployment
Using Ponder 2 for implementation
Execution Layer
Databases, XML Stores, Rule Engines, State Machines, etc

- Constraints on resources
  - To ensure security
- Policy in machine executable format
- Policy deployment and execution layer
  - Logs, monitoring, auditing

Relationships between Levels

- Policies are defined: *Specification layer*
- Transformation into a more structured format: *Abstract policy model*
  - Further analysis to interpret them in context of the system
- Transformation into concrete policy: *Executable policy model*
- Policy transformation
  - Must be transparent and consistent within the system
- Policy synchronization
  - Track the relationships between policies at each level.

Relationships between Levels - Example

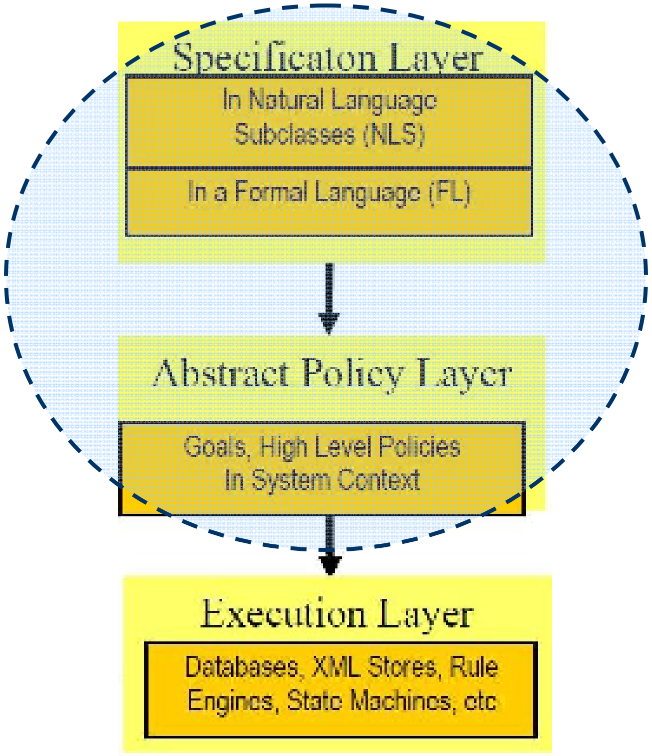


SPARCLE Authoring

XML P-RBAC Permission Assignments

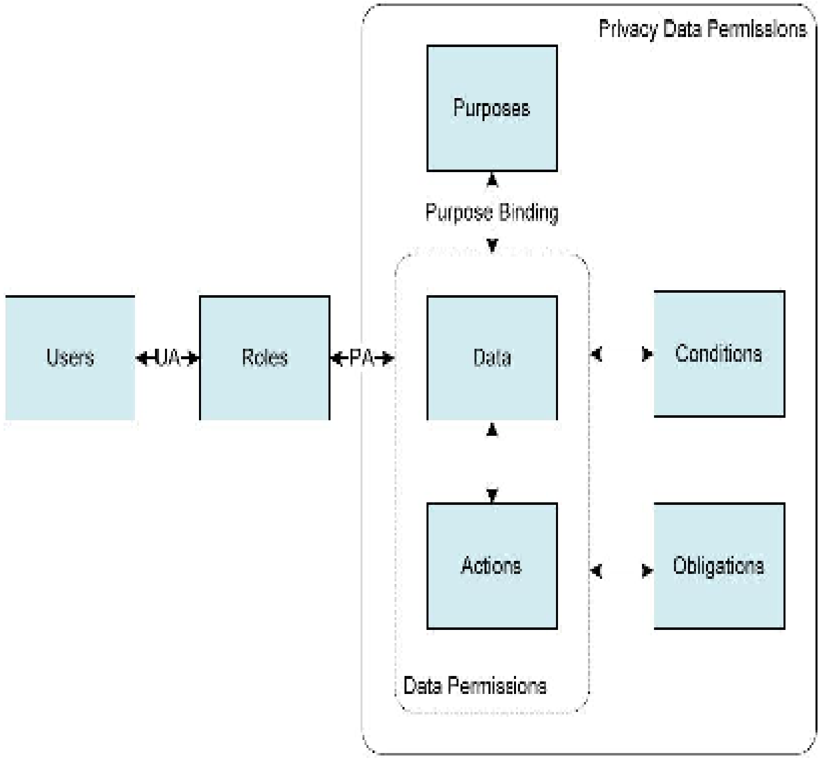
Permission Assignments Analysis

XML Analysis Results


Specificaton Layer
In Natural Language Subclasses (NLS)
In a Formal Language (FL)
Abstract Policy Layer
Goals, High Level Policies In System Context
Execution Layer
Databases, XML Stores, Rule Engines, State Machines, etc

- SPARCLE
  - From natural language to P-RBAC permission.
- Next – P-RBAC

Core P-RBAC – Abstract Policy Model



- Privacy-aware permission
- User
- Roles
- Data
  - Purpose
  - Condition
- Actions
  - Obligation
- Next – Policy analysis
  - To confirm validity, correctness and consistency.

Policy Management Framework

- Three levels of abstraction
- Transformation between them
- Issues discussed here:
  - Brief details of each level
  - Policy Analysis and Ratification
  - End User issues

Policy Analysis and Ratification

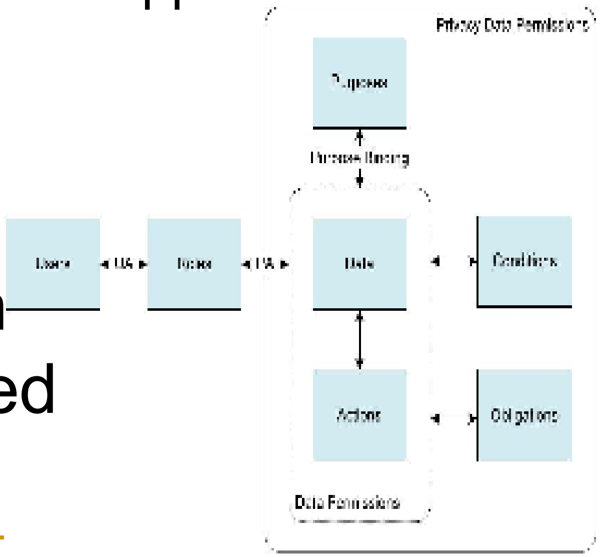
- Analysis
  - Policy validation – system can implement it
    - Mapping with mechanisms which are supported or not.
  - Policy ratification – certify the appropriateness
  - Policy run-time analysis – monitor, audit etc.
- Ratification
  - Conflict detection – cannot be executed simultaneously
  - Dominance – dominated policy will not change behavior
  - Coverage – determine if all cases are covered by policy set
  - Application dependent properties
    - Conflict of duty
    - Conflict of interest

End User Issues

- Policy presentation
  - Language to represent the policy
    - Natural (SPARCLE)
      - Ambiguous, inconsistent
    - Formal (P3P)
      - Not ambiguous
      - Consistent presentation of different policies
      - Allow comparison between policies
  - A mean to present the policy to user
    - machine readable ←→human understandable format
    - High level view (drill down)
- Policy Explanation
- Policy Technologies

Example: Healthcare scenario

- **Policy Specification Layer**
  - Privacy Policy Rules
    - Healthcare staff can forward patient medical information for the purpose of national medical research if the information is anonymized.
  - Security Policy Rules
    - Healthcare staff can access test results databases.
    - Healthcare staff can access upload and email applications
- Users =Healthcare staff
- Actions = can forward
- Data = medical information
- Purpose = national medical research
- Condition = information is anonymized



Example: Healthcare scenario (cont.)

- **Abstract Policy Model Layer**
  - Privacy Policy Rules
    - Healthcare staff (user group A) can upload (upload application) patient test results (DB table patient info, column results) to the NIH DB (NIH DB Study Results) if patient identity is not disclosed (Do not use DB table patient info, column name).
  - Security Policy Rules
    - Healthcare staff (user group A) can access (read/write/modify) test results databases (DB table patient info, column results).

Example: Healthcare scenario (cont.)

- **Executable Policy Layer**
  - Privacy Policy Rules
    - If request(transmit(destination_address,Type)) && (Type =/= testData OR NOT(member(destination_address,RegisterUniversityList))
    - then deny(transmit(destination_address,Type))
  - Security Policy Rules
    - If user(member group A) && Read(PatientDB) then allow.
    - If user(member group A) && Access(App1) then allow.
    - If user (member group A) && Access (App2) then allow.

Summary

- Presented three layer framework for discussing policy
- Other issues:
  - Context
  - Trust and Risk
- Research continues
  - Models to support management of policies
  - Suitable abstraction for relating security & privacy

Conclusion

- Sound framework
- No practical deployment result.
- No comparison between any standards or frameworks
- No indication of how the abstraction from high to low level might take place
- The overhead of modifying existing policy is not clear.

- Thank you.