Managing Trust-related Policies:

how did we get here?
what's next for researchers?

(copyrighted images removed)

Marianne Winslett

University of Illinois at Urbana-Champaign

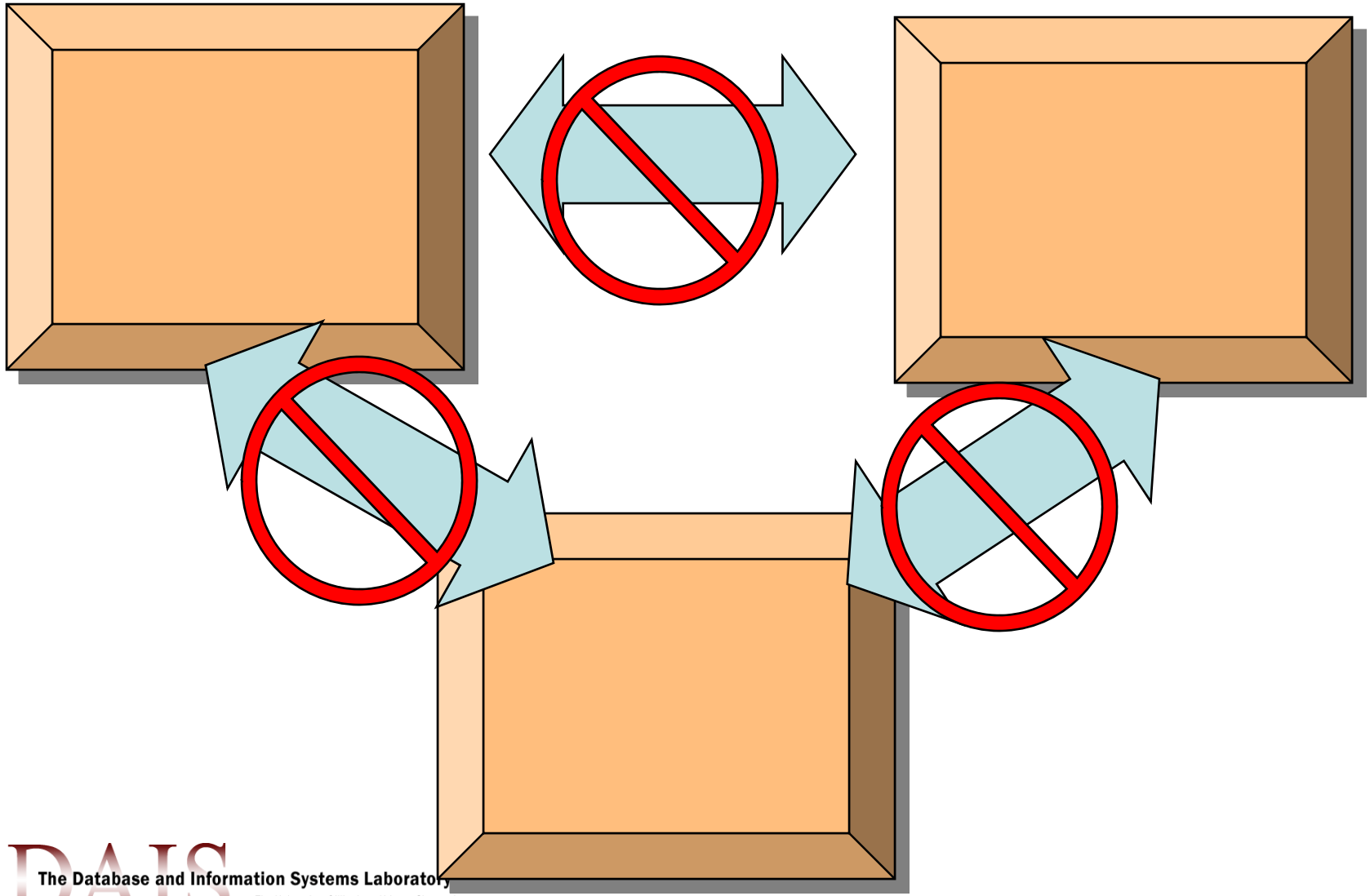
-- *and colleagues* --

PART 1

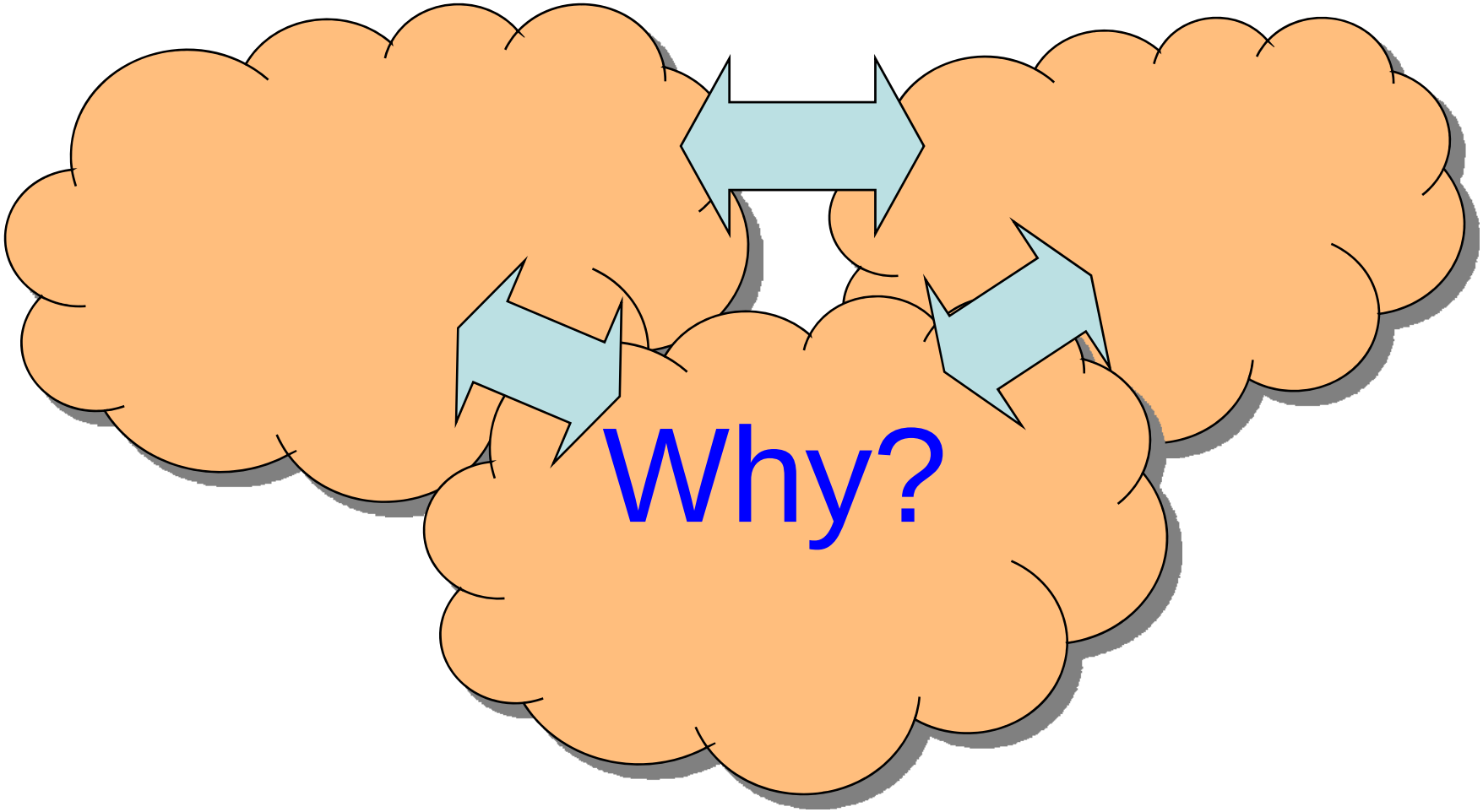
WHY TRUST-RELATED POLICIES ARE GETTING MORE IMPORTANT

A tale of two (example) trends

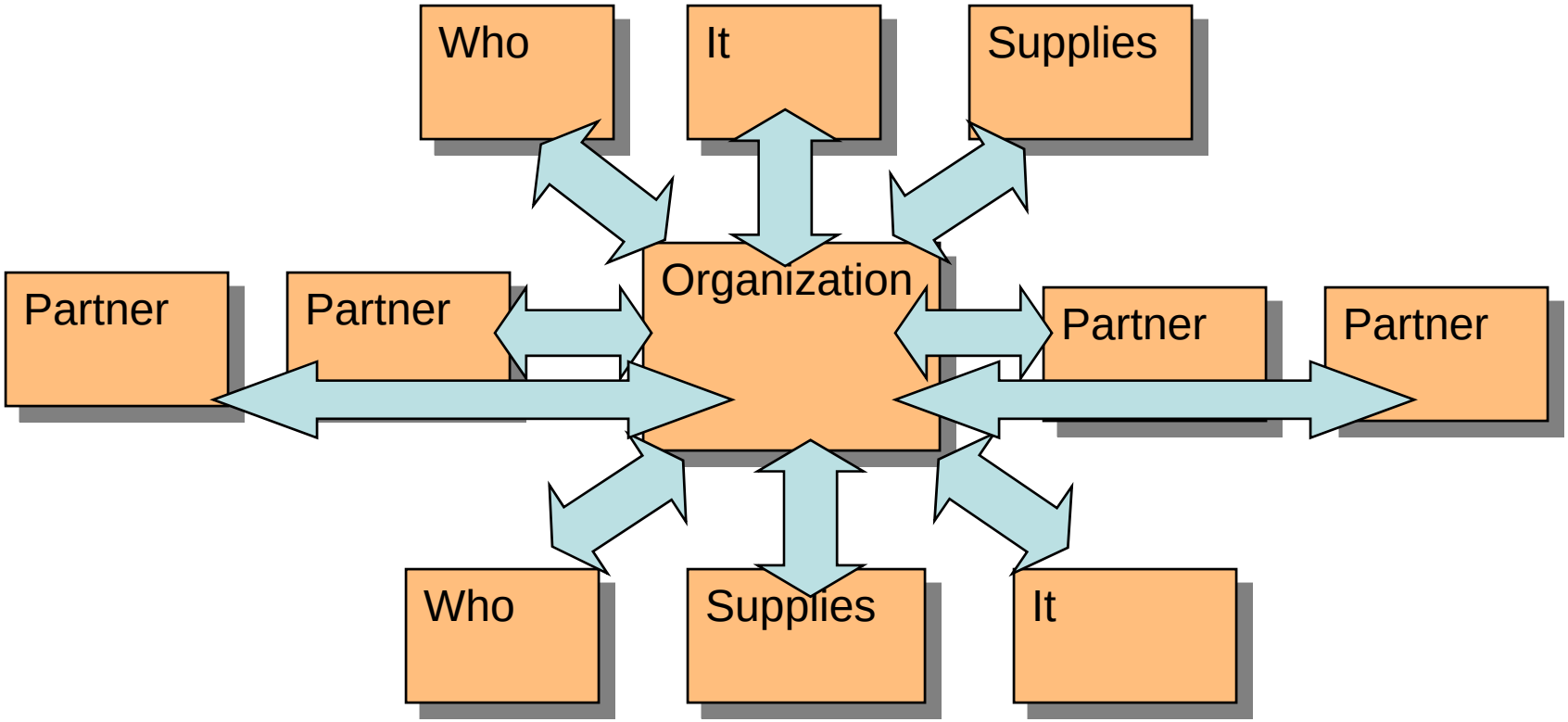
Organizational boundaries used to be solid



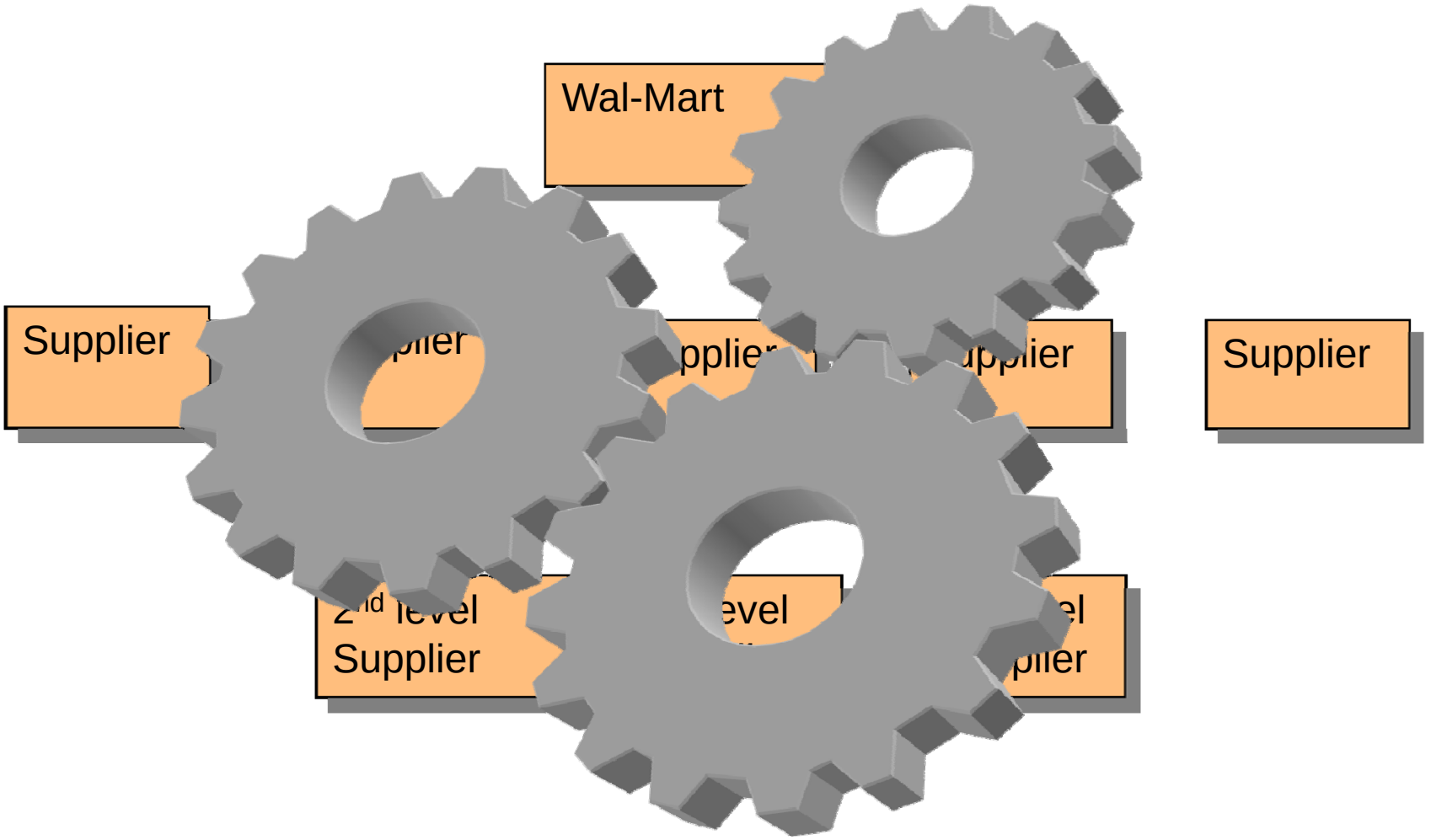
Now boundaries are fuzzy



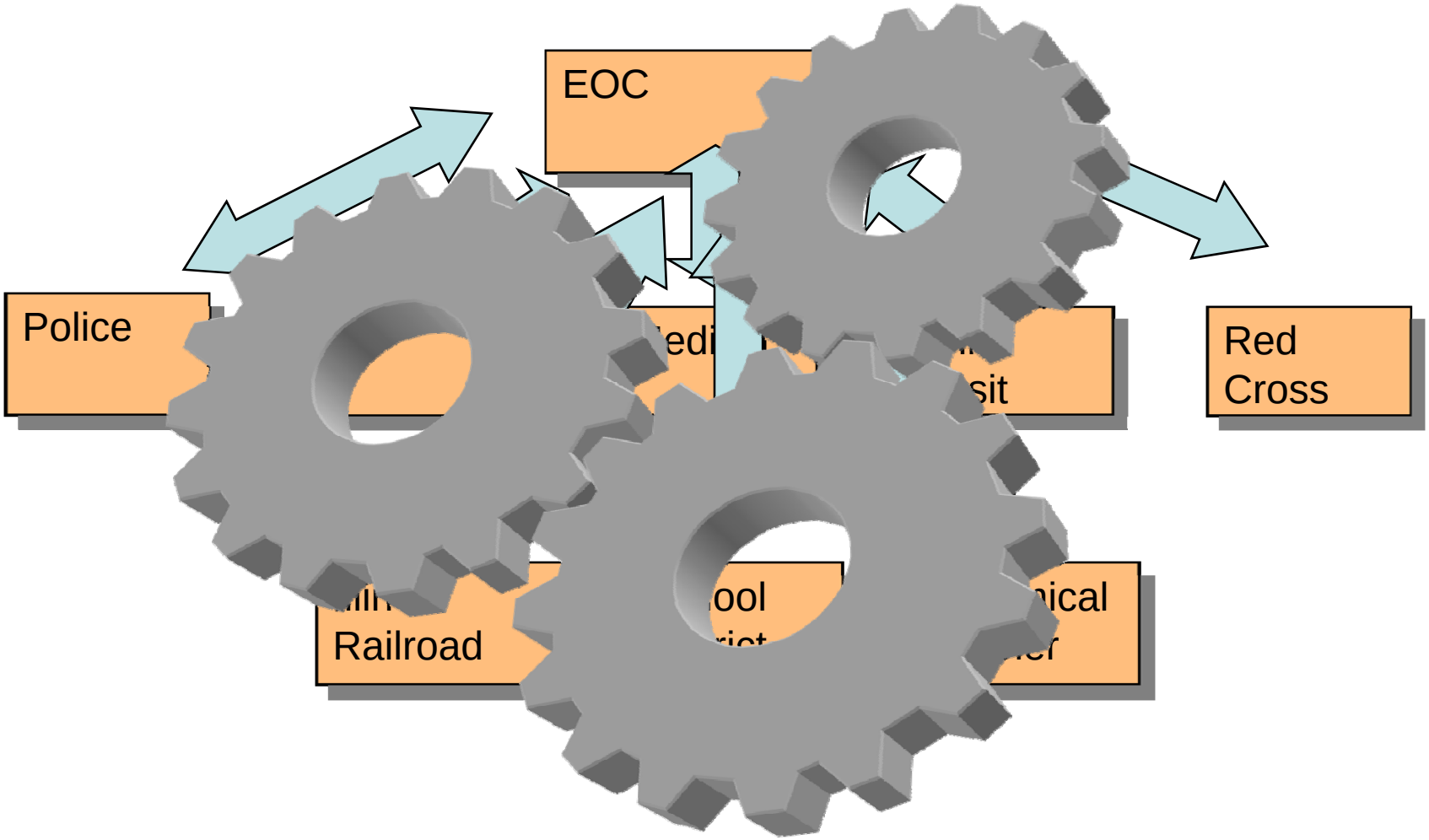
Competitive pressures are dissolving boundaries



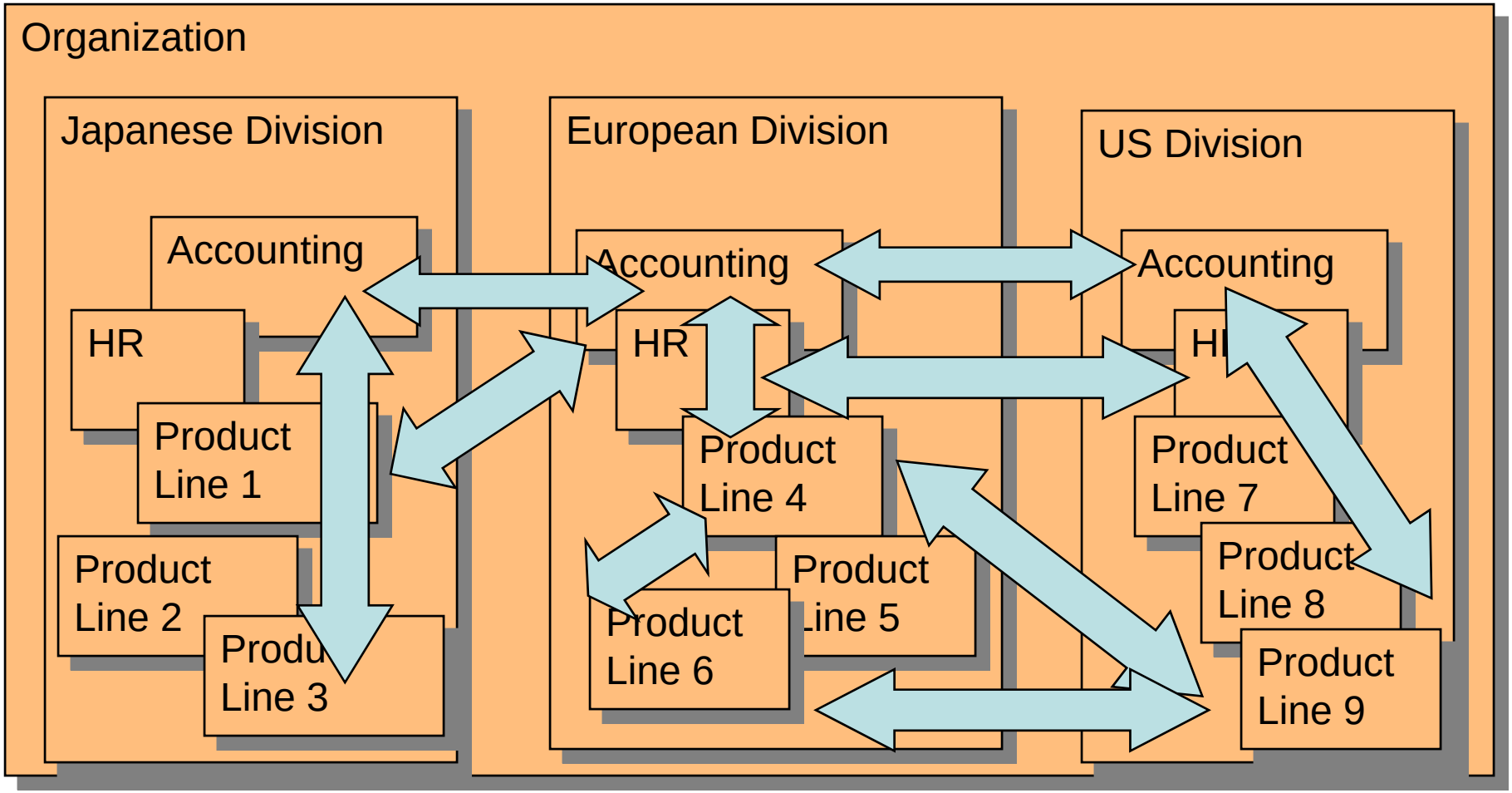
Example: supply chains



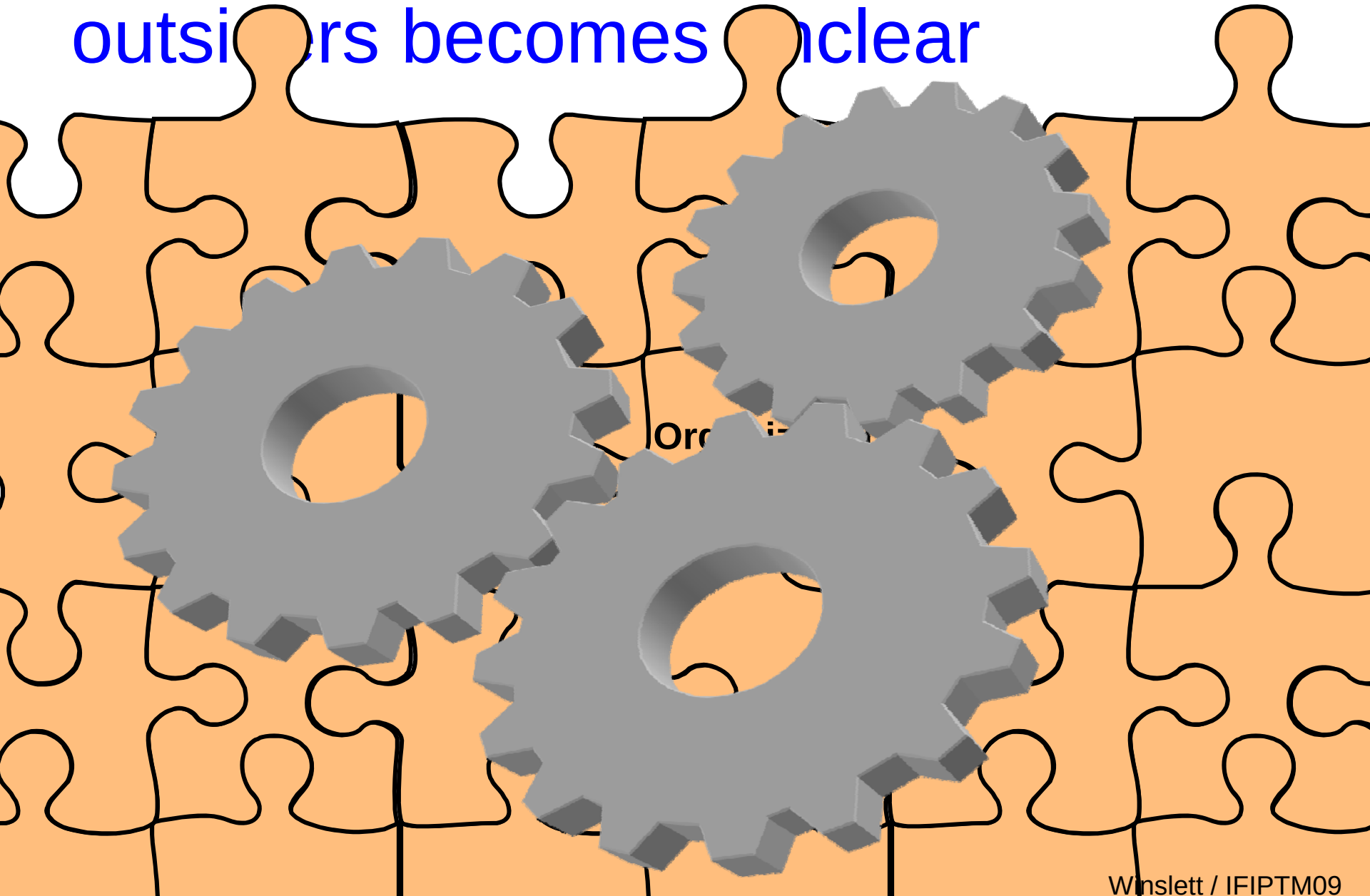
Example: first responders



Example: any large enterprise

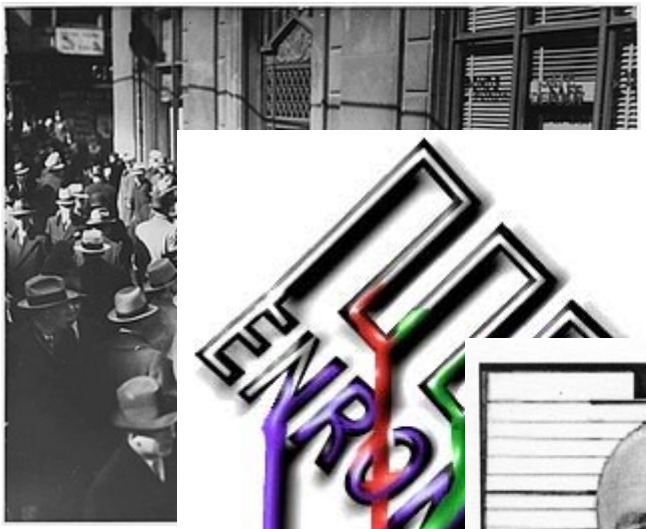


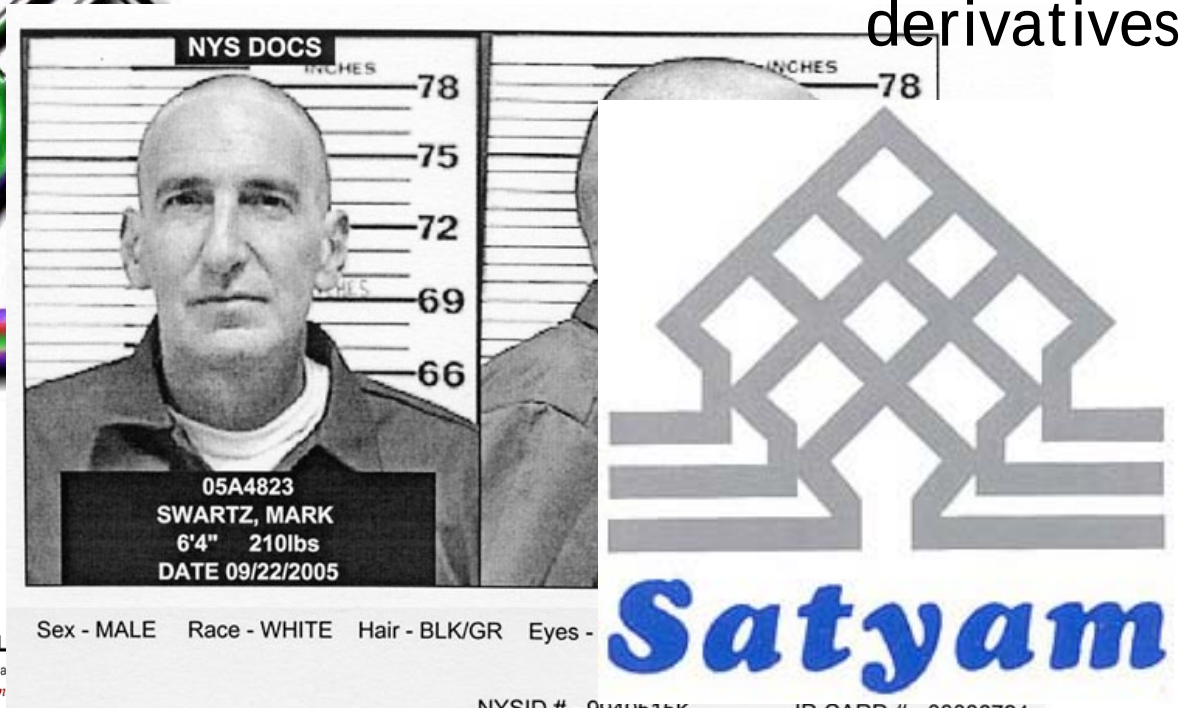
Distinction between insiders and outsiders becomes unclear



Organizations are also facing new pressures for accountability

When something really bad happens, the government likes to quickly take action to restore society's trust in its institutions

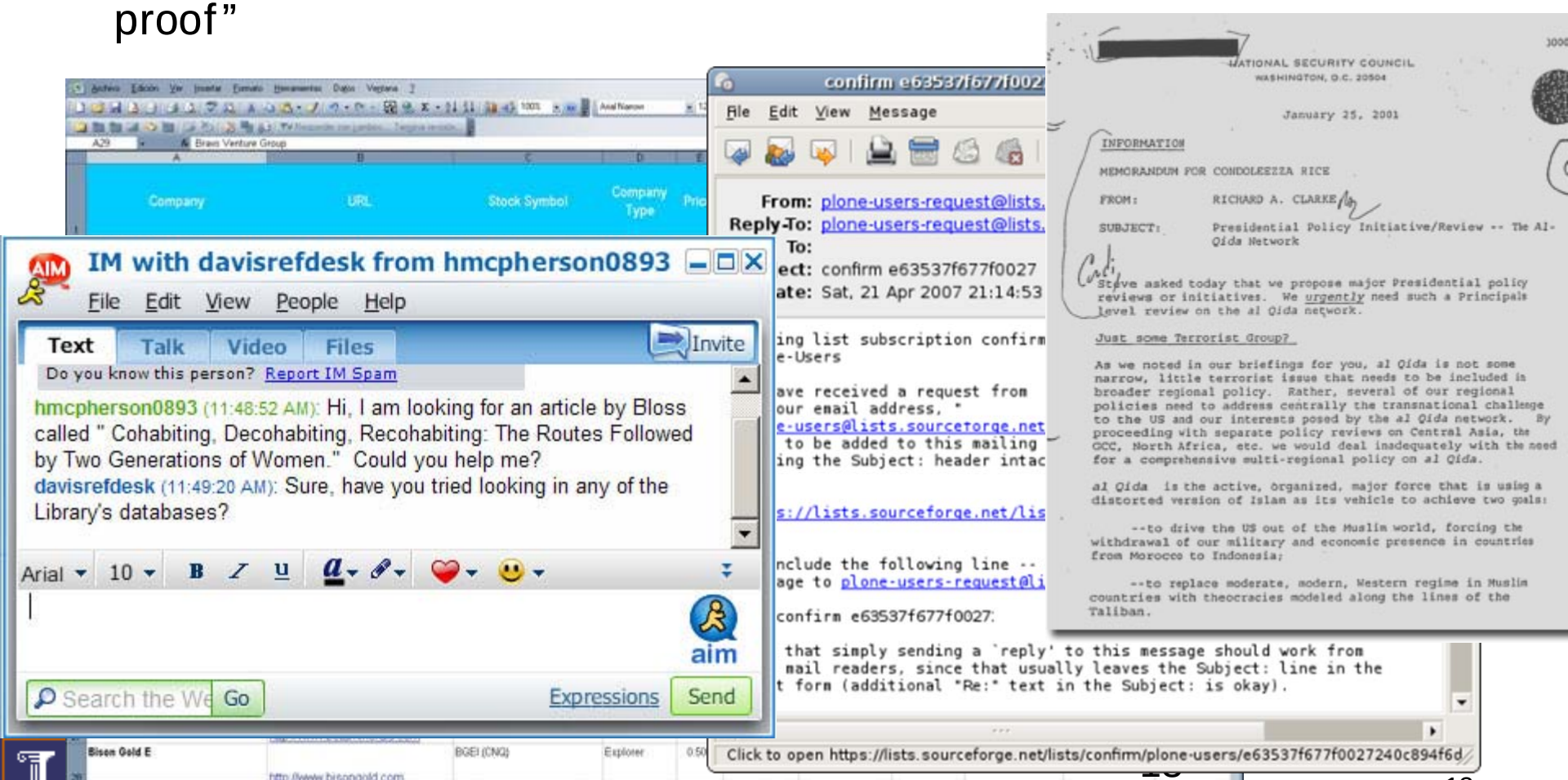




- FDIC
- Sarbanes-Oxley Act
- Regulating derivatives

SOX had major repercussions for corporate IT

- Top execs have to sign off on financial reports
- Retain routine business documents for (typically) 7 years, "tamper-proof"



Compliance regulations have teeth: periodic audits, fines, jail terms

SEC Rule 17a-4:

- $1.65M each
- Deutsche Bank
- Goldman Sachs
- Morgan Stanley
- Solomon Smith Barney
- U.S. Bancorp

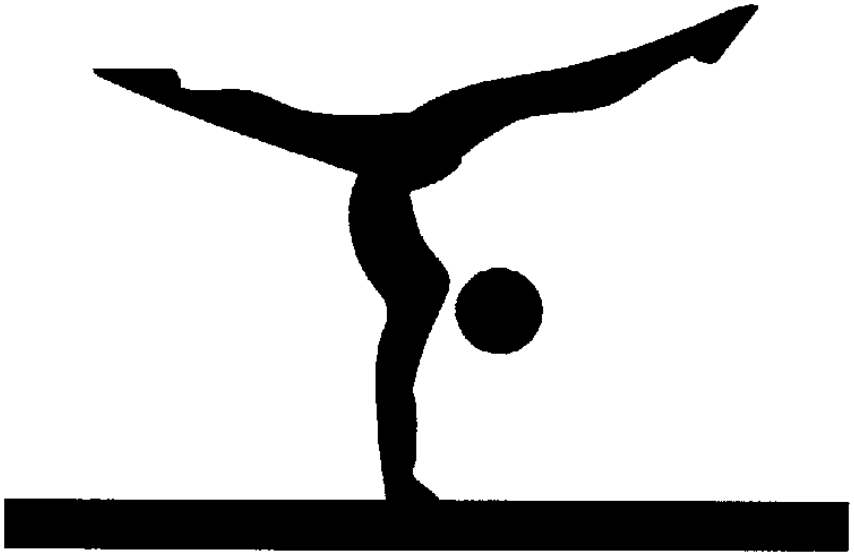
SOX:

- Rica Foods CEO $25K
- Deloitte $1M poor audit

The government likes to step in for non-corporate scandals as well.

- Video Privacy Protection Act of 1988
- Gramm-Leach-Bliley Act's Financial Privacy Rule
- Health Insurance Portability and Accountability Act (HIPAA)

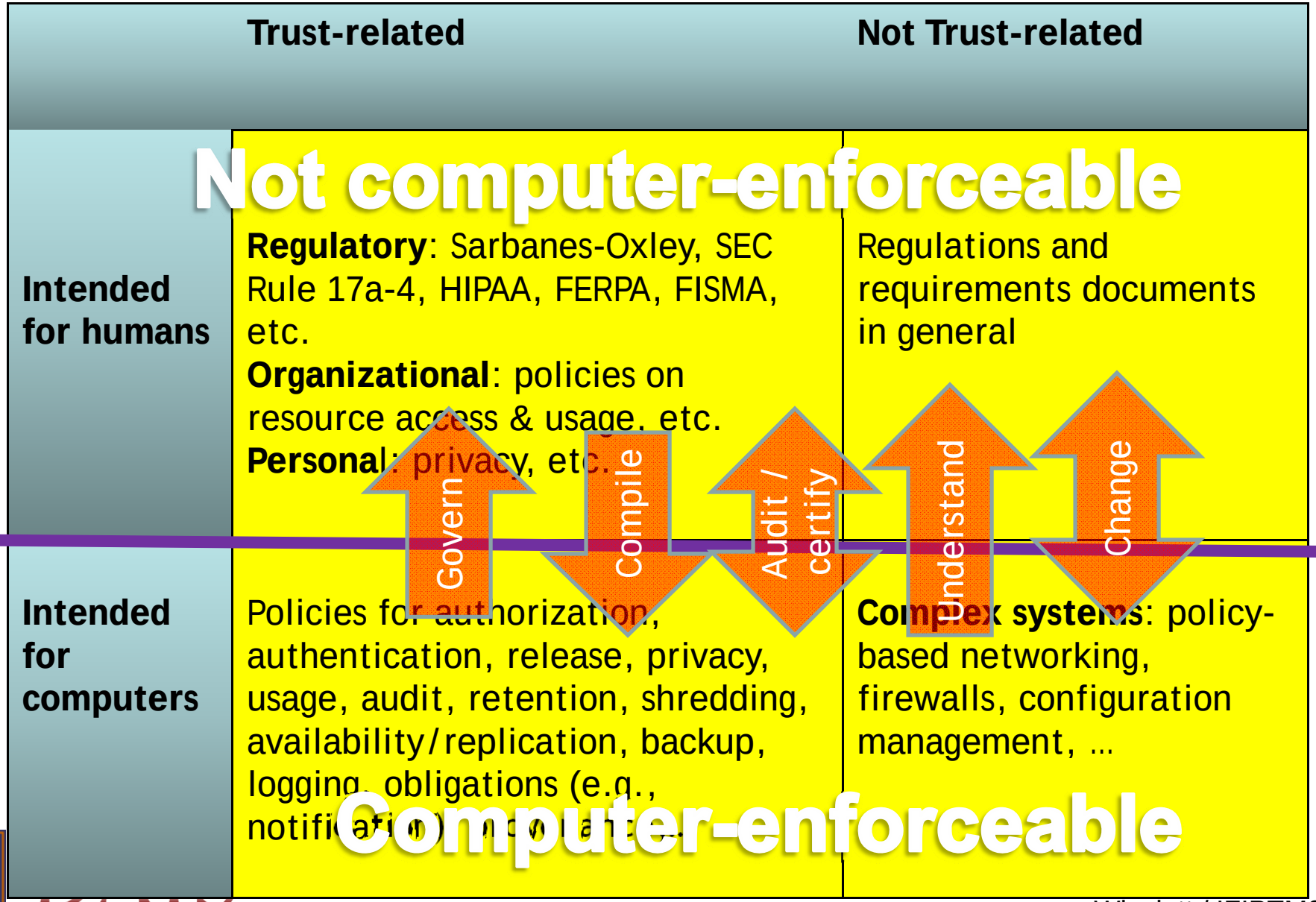
E-government records are also at risk for falsification.



How old is your athlete?

Accountability includes knowing who can/did do what to your data when

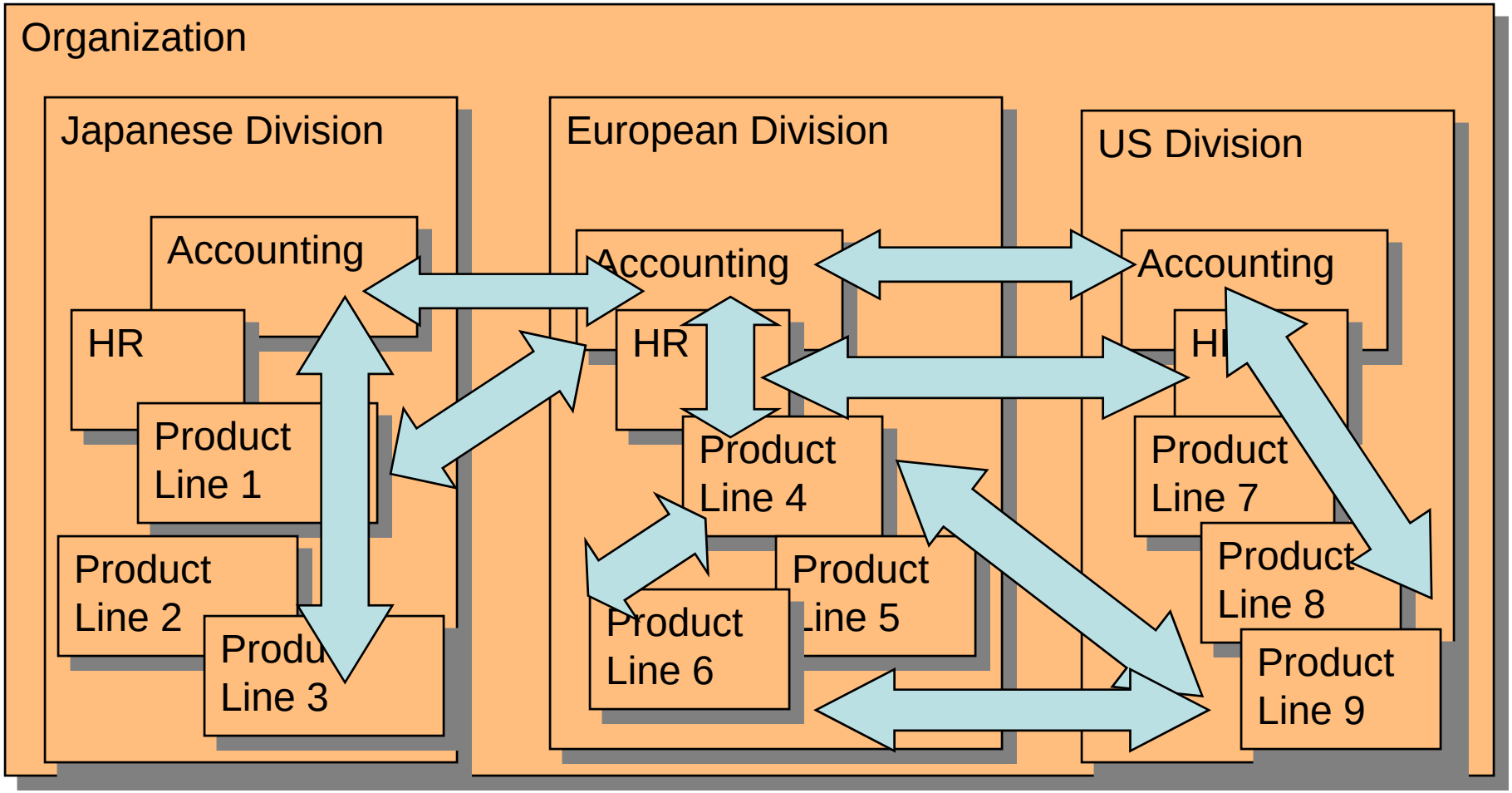
We use policies when our intent is too hard to specify, implement & manage directly.



PART 2

THE RISE OF POLICY-BASED SYSTEMS

Example: any large enterprise

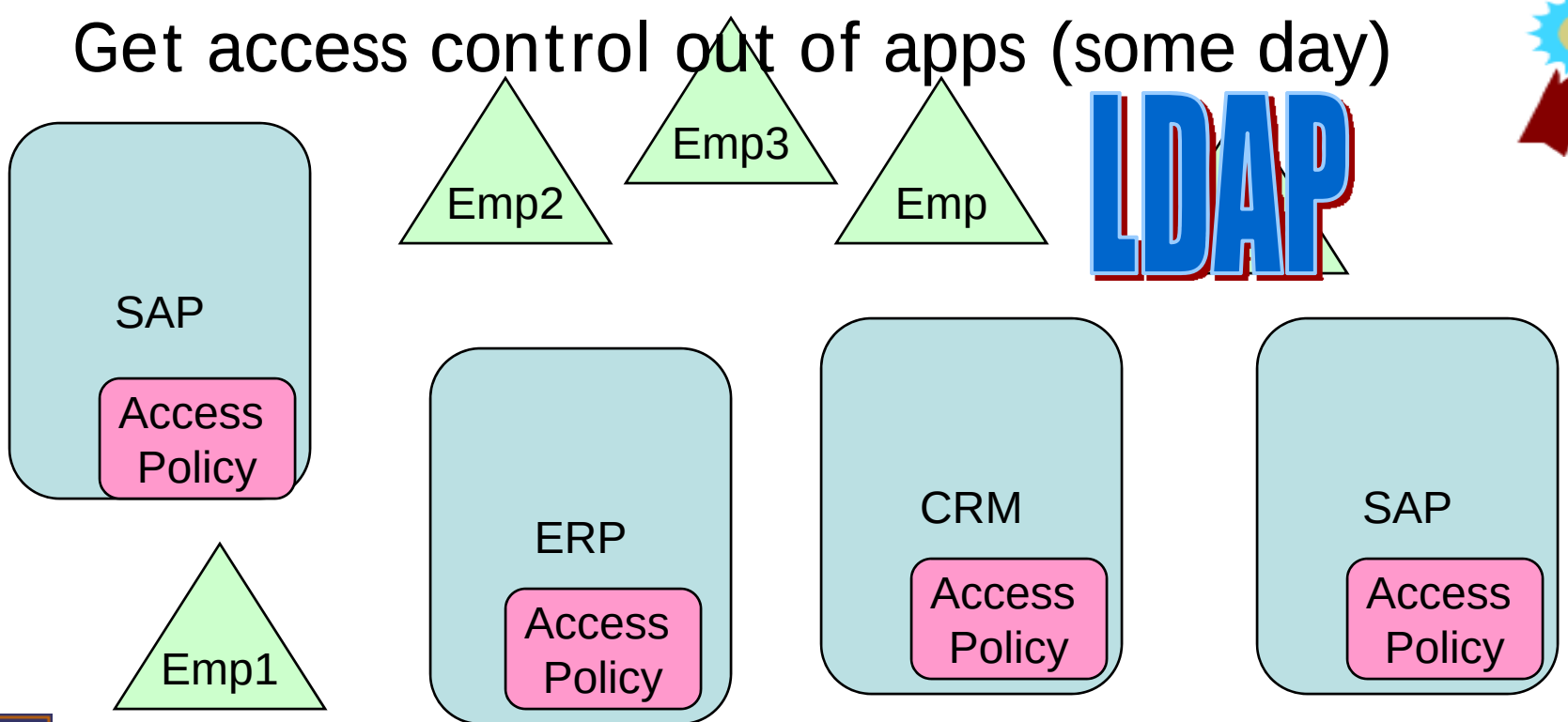


Industry is taking several steps to meet these needs

Strong authentication (X.509)

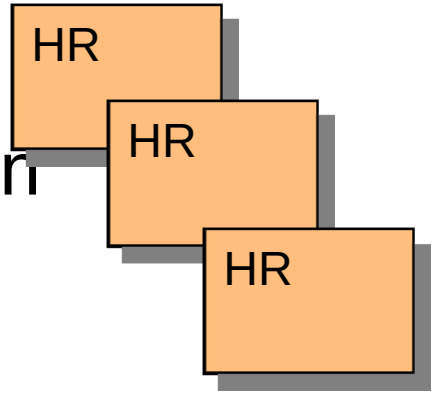
Centralize role definitions, base on attributes

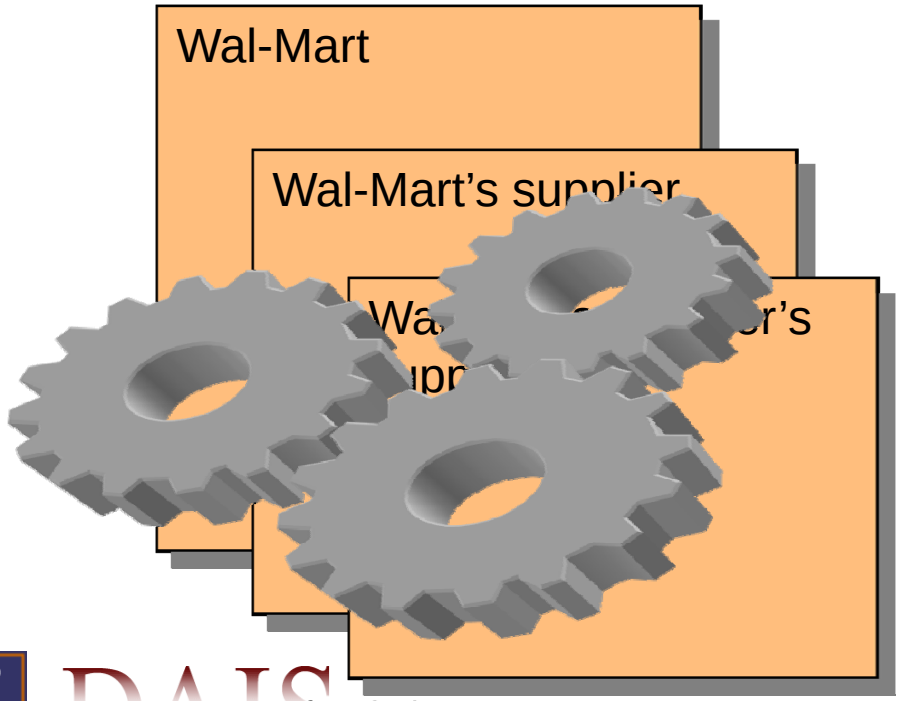
Get access control out of apps (some day)



So enterprises are moving toward attribute-based access control

- Based off centralized LDAP + X.509
- Avoids inconsistency due to distribution
- Easier to maintain, compared to ACLs





Less insider threat

I claim that policies are becoming more important in other trust-related areas, too.

Firewalls

Routing

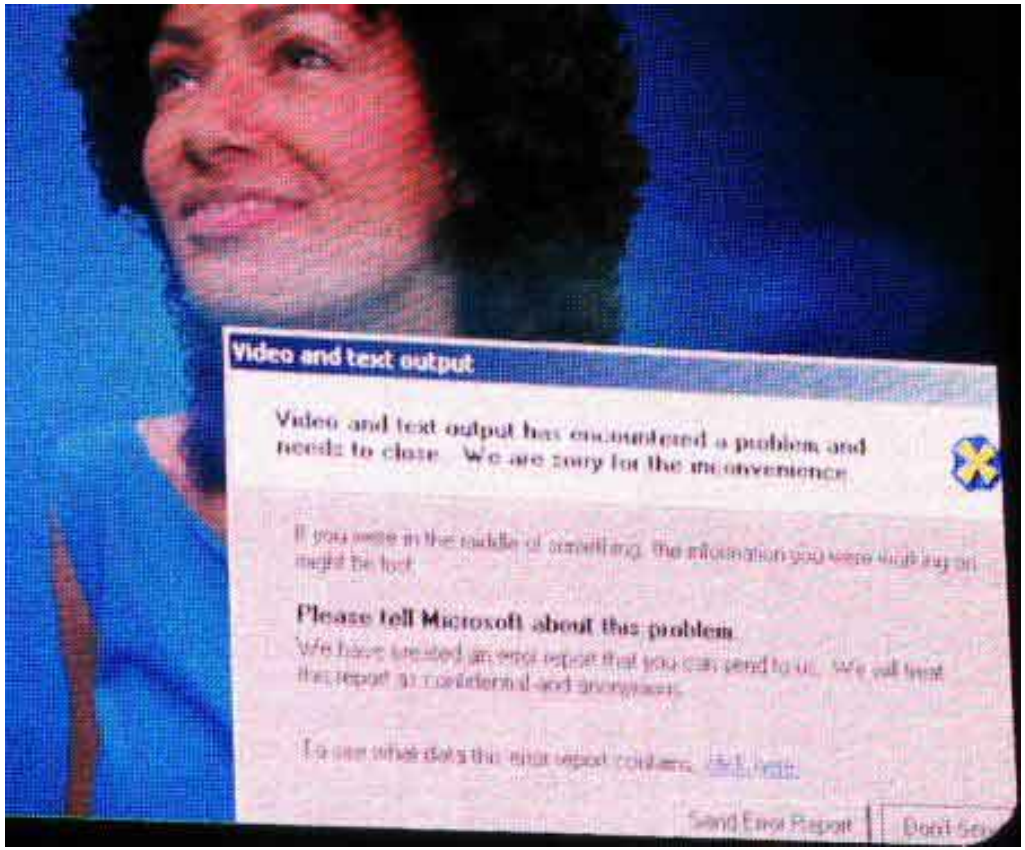
How long must we retain this address tuple?

…

Aren't policy-based trust-related systems a good thing?

Why this scares me:

Automated exploitation of policy errors & loopholes



Why this scares me:

Centralized policy-based services can be attacked

Why this scares me:

Understanding policies

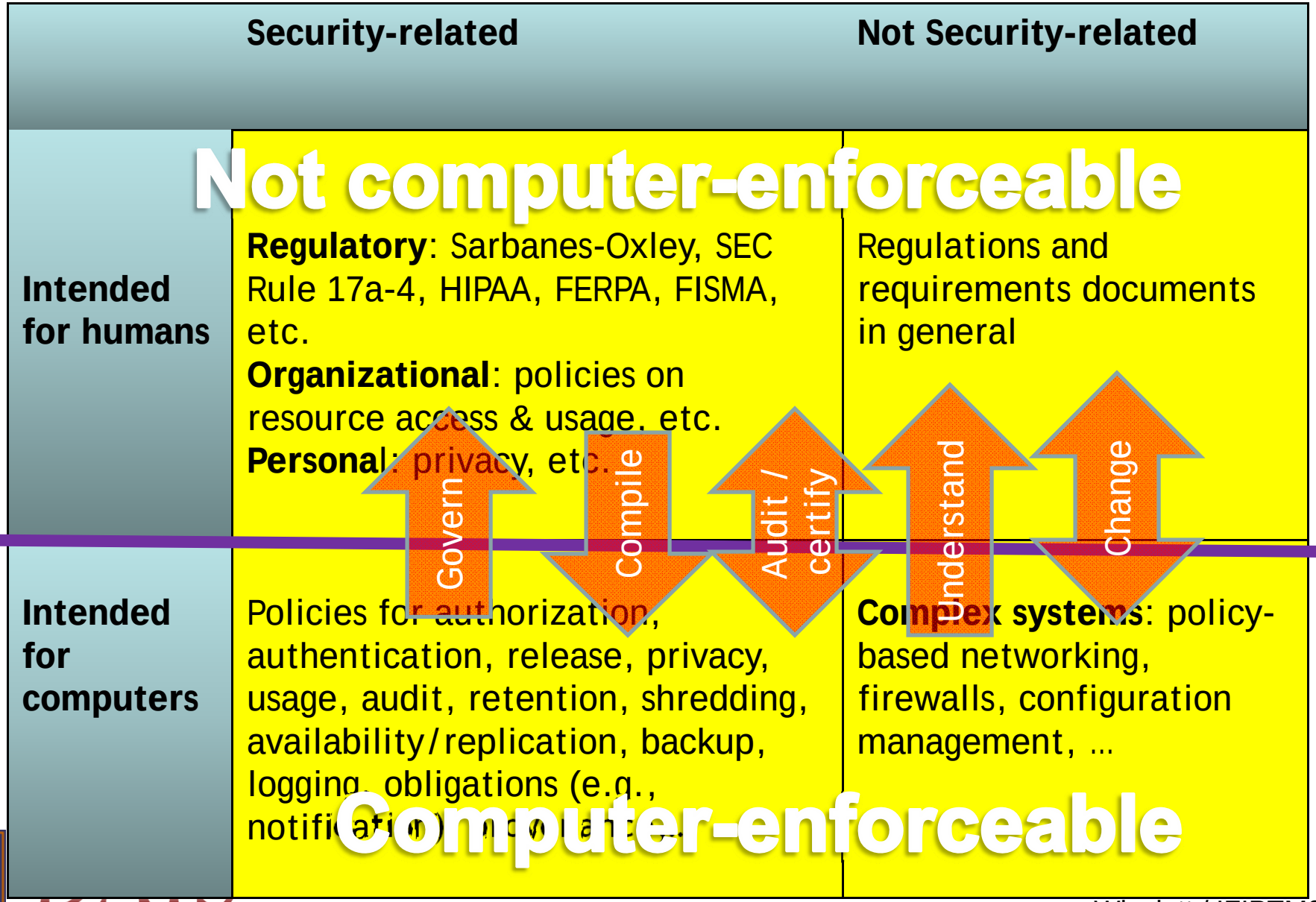
Industrial policy languages were not intended for rigorous analysis or user-friendliness

Analysis tools

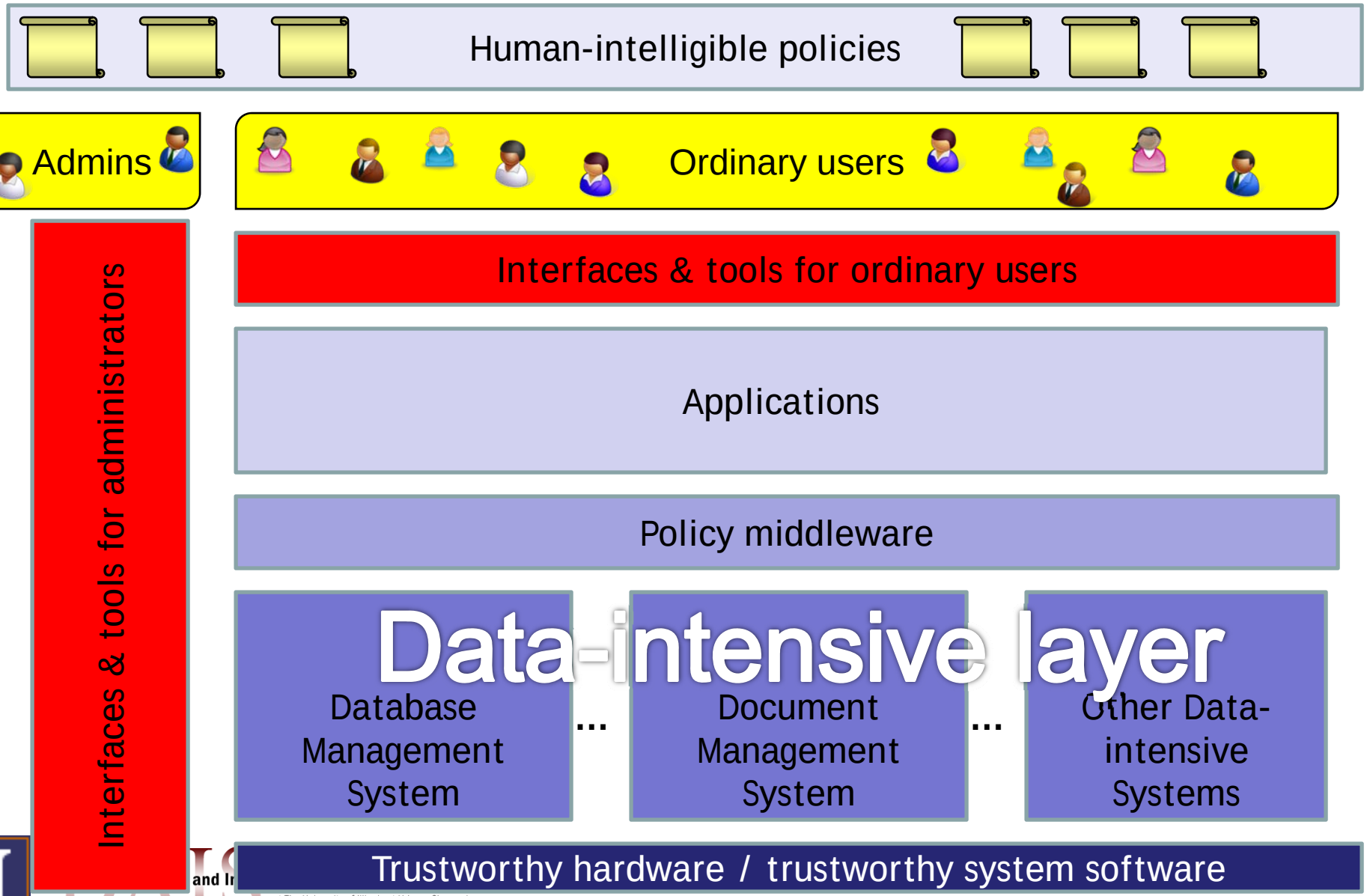
PART 3

WHAT CAN WE DO TO HELP?

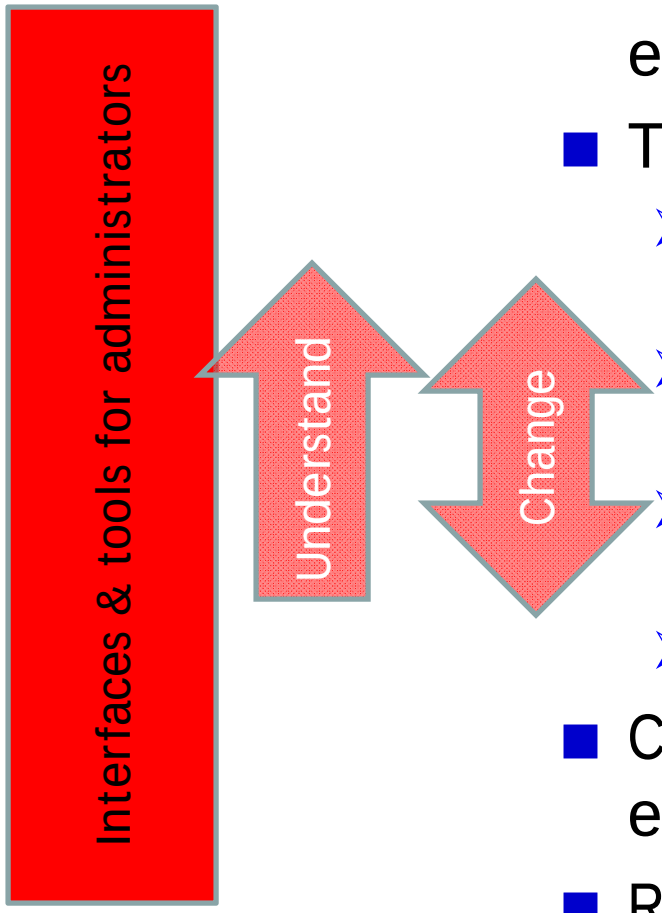
We use policies when our intent is too hard to specify, implement & manage directly.



Advances are needed at & between *all levels* of the system.



We need ***easy-to-use*** tools for policy admins.



- To help them visualize & understand enormous policies
- To analyze large policies
  - Safety and availability questions: Can this user take this action under these conditions?
  - What-if analysis, regression testing for proposed policy changes
  - Explanation of why particular actions were taken
  - Conflict identification & resolution
- Compile policies into actionable enforcement (discussed later)
- Rewrite policies to equivalent form to make them faster, simpler, or meet other goals

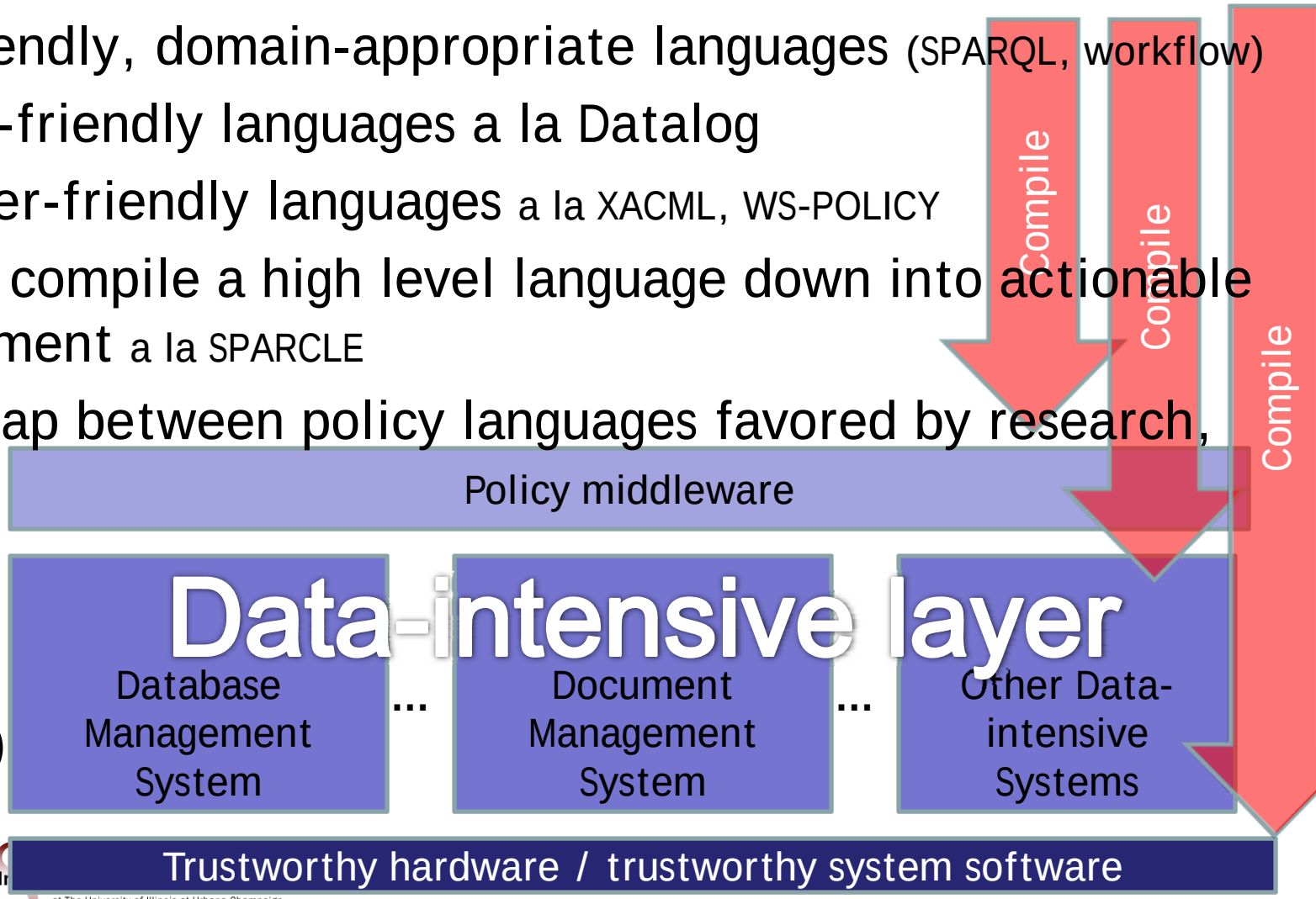
We need ***easy-to-use*** tools for ordinary users.

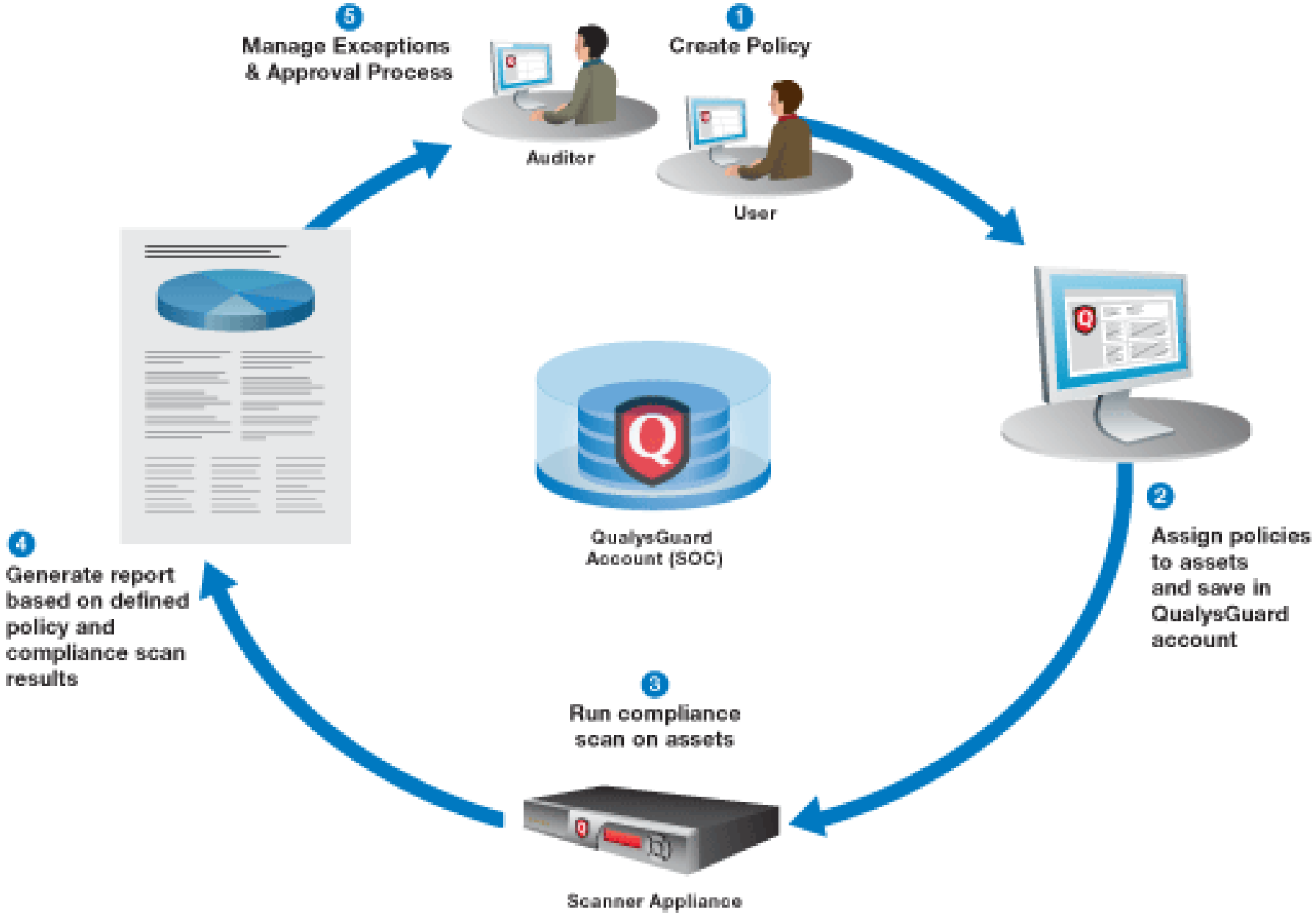
Interfaces & tools for ordinary users

- To manage their own policies: all the tools that system administrators need, but with an interface suitable for them
- For real-time discovery of a system's policies that are relevant to them or to their software agents
- To understand why a particular policy-based action was taken (e.g., their access request was denied), and actionable steps they can take to change that outcome

We need policy languages, compilation techniques for every situation.

- User-friendly, domain-appropriate languages (SPARQL, workflow)
- Analysis-friendly languages a la Datalog
- Computer-friendly languages a la XACML, WS-POLICY
- Ways to compile a high level language down into actionable enforcement a la SPARCLE
- Bridge gap between policy languages favored by research, industry (e.g., XACML vs. Datalog)





From the QualysGuard literature on the web

SPARCLE is a research prototype of a policy management workbench. SPARCLE allows the polic

IBM's SPARCLE policy workbench

Policy Name : My Bank Policy **Domain :** Finance **Created Date:** 2005-07-27

Policy Description : Template for privacy policies in American divisons. **Last Modified Date :** 2005-07-27

Example Rule Guide:

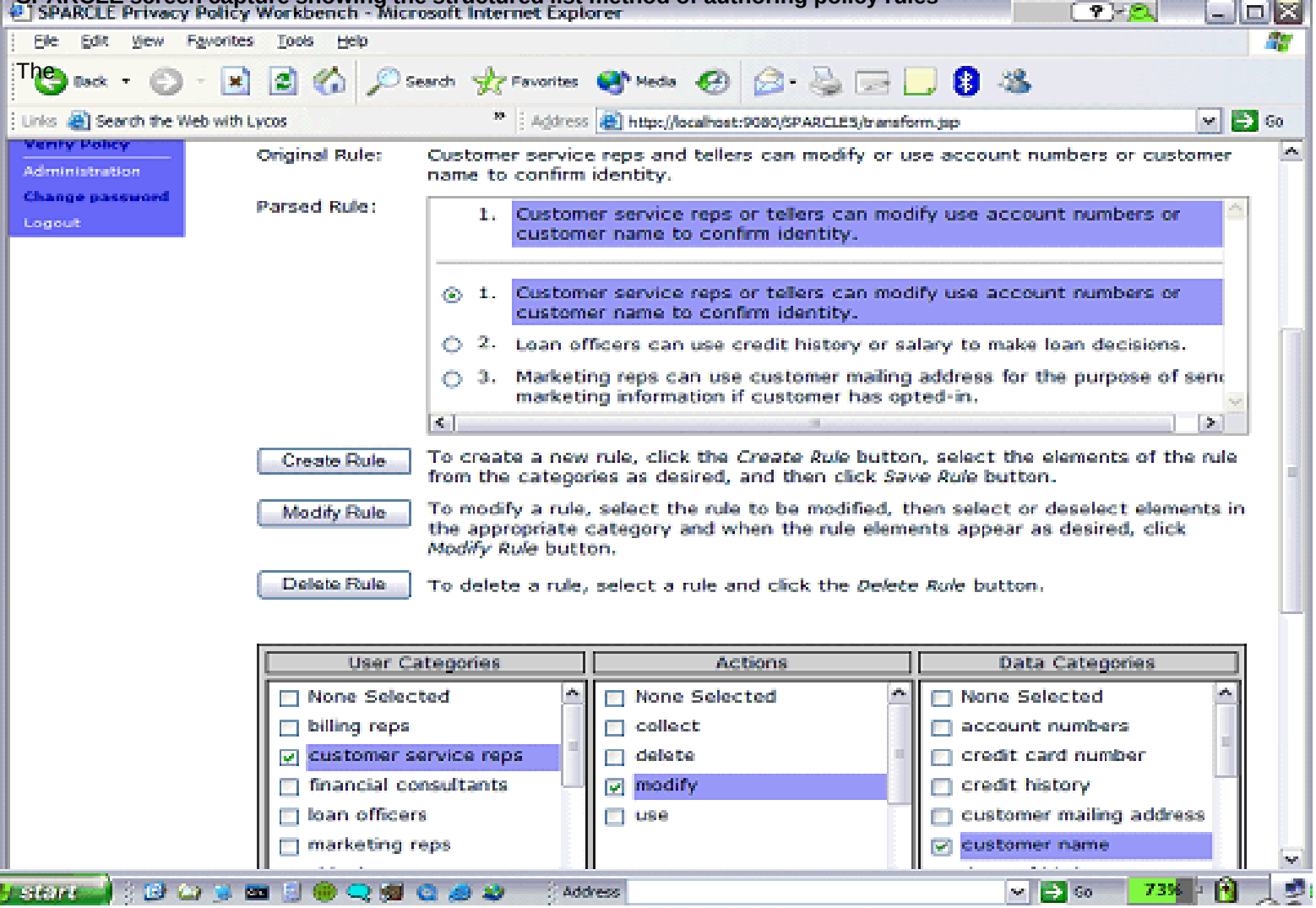
[User Category(ies)] can **[Action(s)]** **[Data Category(ies)]** for the purpose(s) of **[Purpose(s)]** **Condition(s)]** with **[(optional) Obligation(s)]**.

```
1. Customer service reps and tellers can modify or use account numbers or
customer name to confirm identity.

2. Loan officers can use credit history or salary to make loan decisions.

3. Marketing reps can use customer mailing address for the purpose of send
marketing information if customer has opted-in.
```

The



"In coordination with IBM Research, IBM Global Business Services (GBS) has used the SPARCLE policy management workbench to help clients with a variety of policy-related issues, ranging from policy definition to policy templates for mandated compliance requirements such as HIPAA and SOX. In addition, GBS clients have used SPARCLE to assist in policy gap analysis, policy conflict resolution and streamlining, and verification of policy consistency."

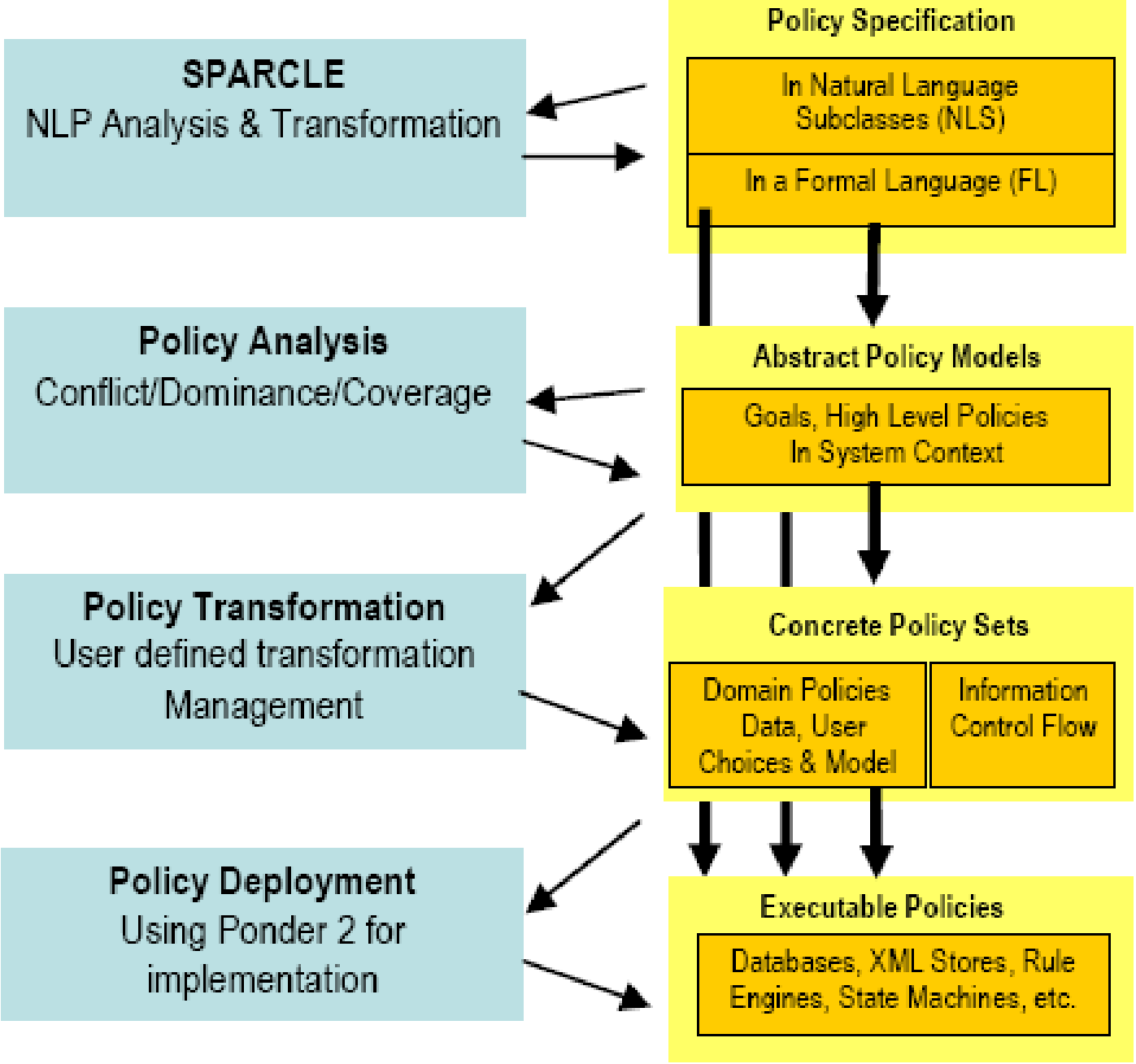


Figure 1. The Components Demonstrated through the Coalition Policy Management Portal.

From the demo by Brodie et al. in POLICY 2008

We need advances in runtime facilities for policy-based systems

- **Usability**: clean ways to involve the human in the loop as needed, & make their task easy (no 16-digit passwords)
- **Scalability**
  - Fast policy compliance checking at runtime
  - Fast run-time automated resolution of policy conflicts, multiple-choice situations
  - Fast provenance collection, interpretation
- **Sticky policies**: how to ensure enforcement, esp. across organizational boundaries?

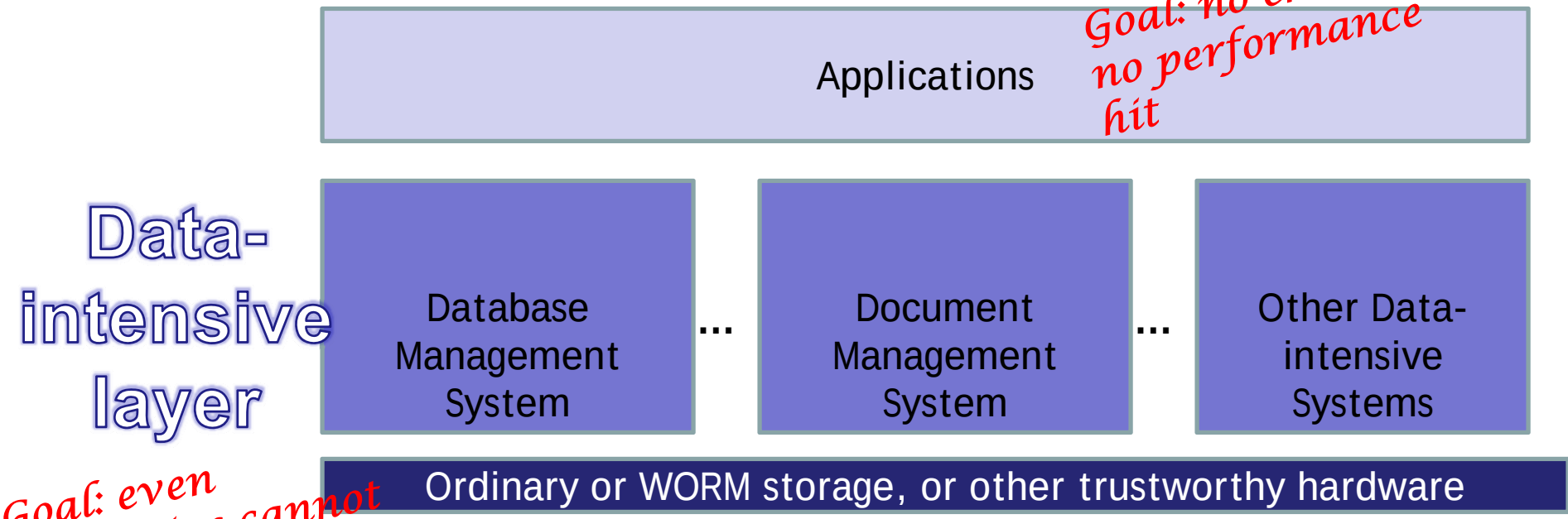
We need user-friendly approaches to help with compliance and audit

$250B/year losses due to insiders: how to track/undo what they did?

- Prevent non-compliance, when possible
- Automate audit of activity (self-auditing)
- Validate actionable policies against specification
- Evaluate effectiveness of policies against intended high-level goals
- Forensic analysis to identify instances of non-compliance, determine/undo their effects as appropriate (self-healing)

Concentrate on prevention for long-term, widely deployed policies (e.g., SOX)

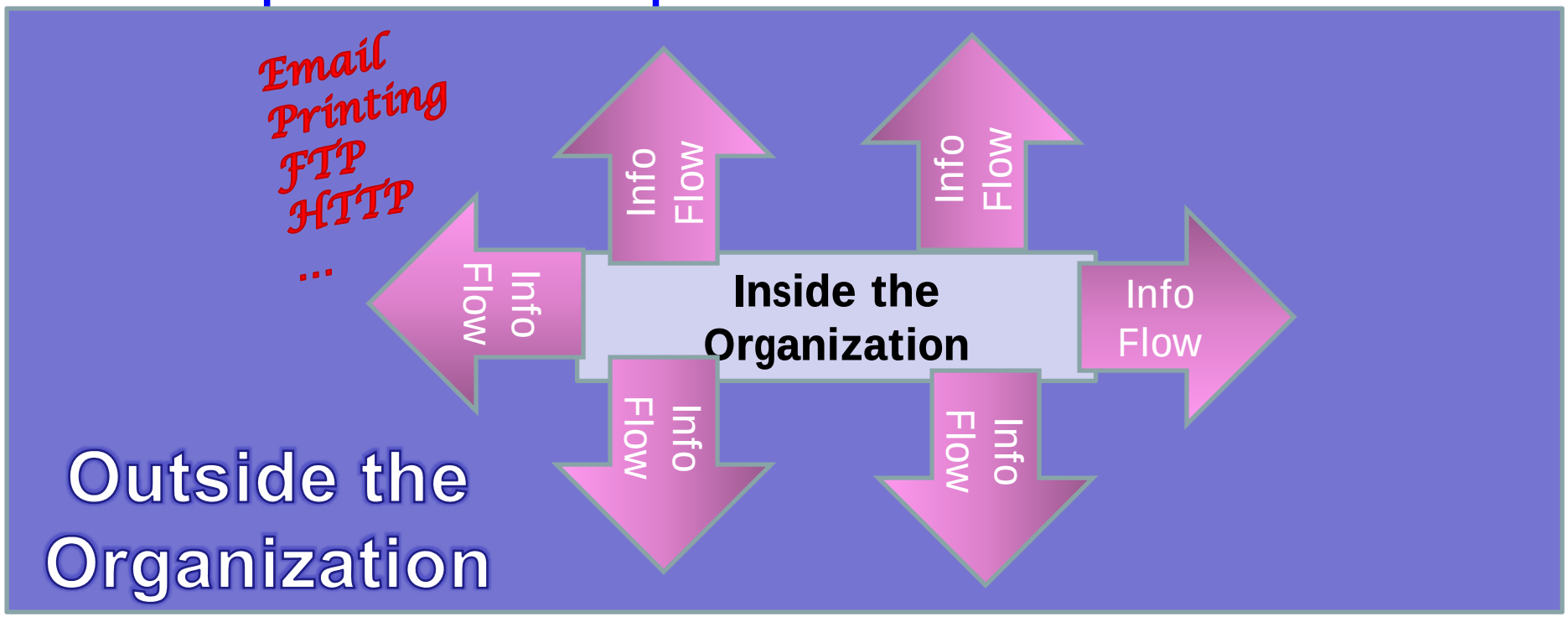
Example: low-cost high-integrity long-term retention of data, documents, logs for SOX



Research challenges:

- Provide trustworthy search, indexing, query answers, & shredding
- Develop/exploit cheap new trustworthy hardware
- Recover from vandalism
- Support fast audits & forensic analysis (what/when/where/how)
- Supporting technology (e.g., de-duplication)

Example: release policies



Research challenges in controlling release:

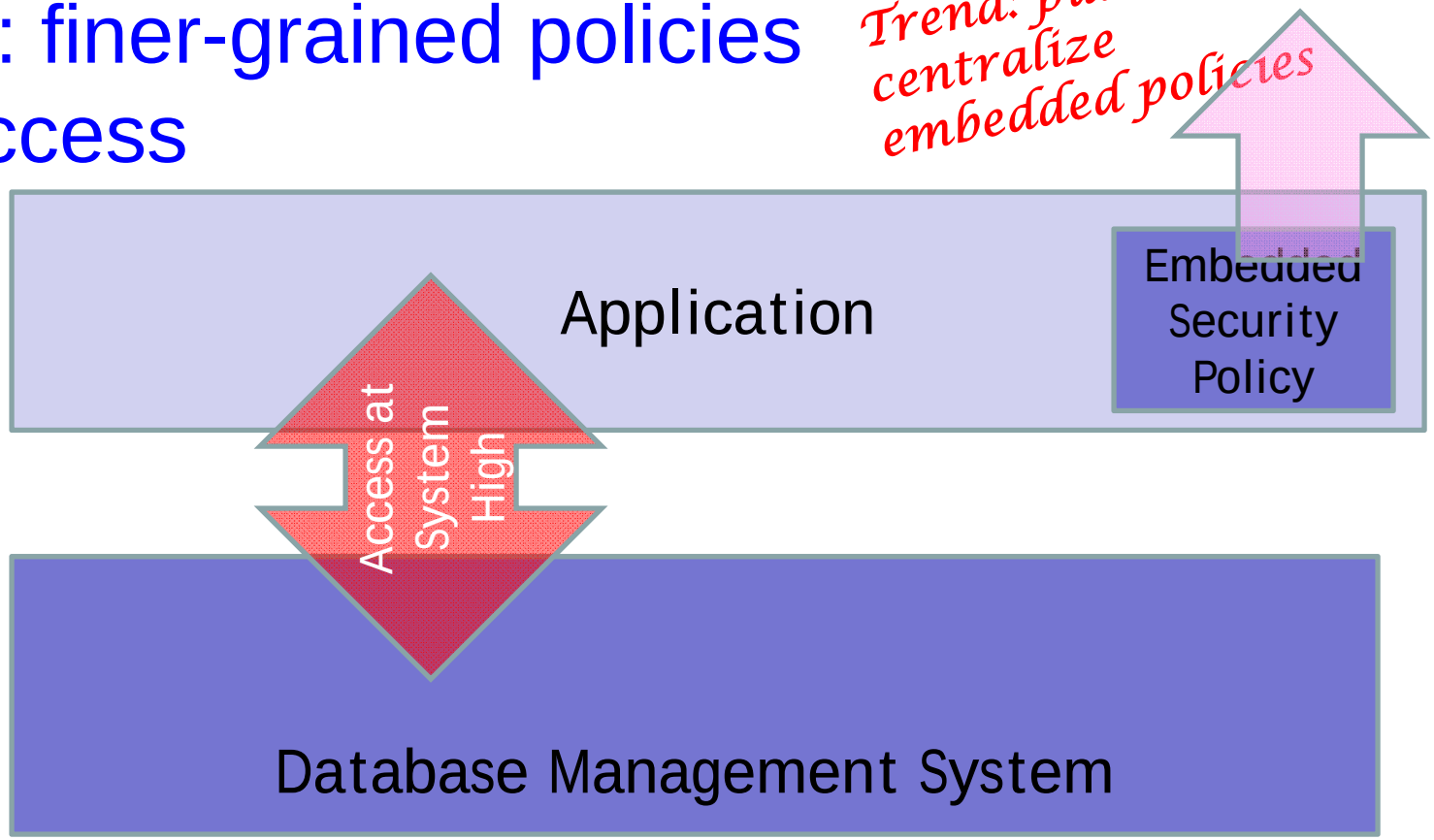
- Fast classification of text, including topic and sentiment identification
- Appropriate handling of encrypted content, tables, figures, images, speech, …
- How to deal with use of outside resources: gmail, clouds, … (often adopted because security is not usable)

Example: auditing cloud SLA compliance

Research / usability issues:

- Where is my data and how is it being stored? (determines regulations, compliance, (sticky) policies to comply with)
- What cloud promises are amenable to user verification, and how can we perform that verification?

Example: finer-grained policies for DB access



Goal:
data-, app-, & user-dependent control over access to each DB cell, to make DB self-protecting

Research challenges:
- Appropriate semantics for policies
- Acceptable performance hit at run time
- Usability
- Sticky policies based on, e.g., data provenance

Example: modern organizations employ *risk management*

Research issues:

- How to evaluate policy effectiveness in reducing risk
- How to reflect risks directly in policies (e.g., variants of risk-based access control)

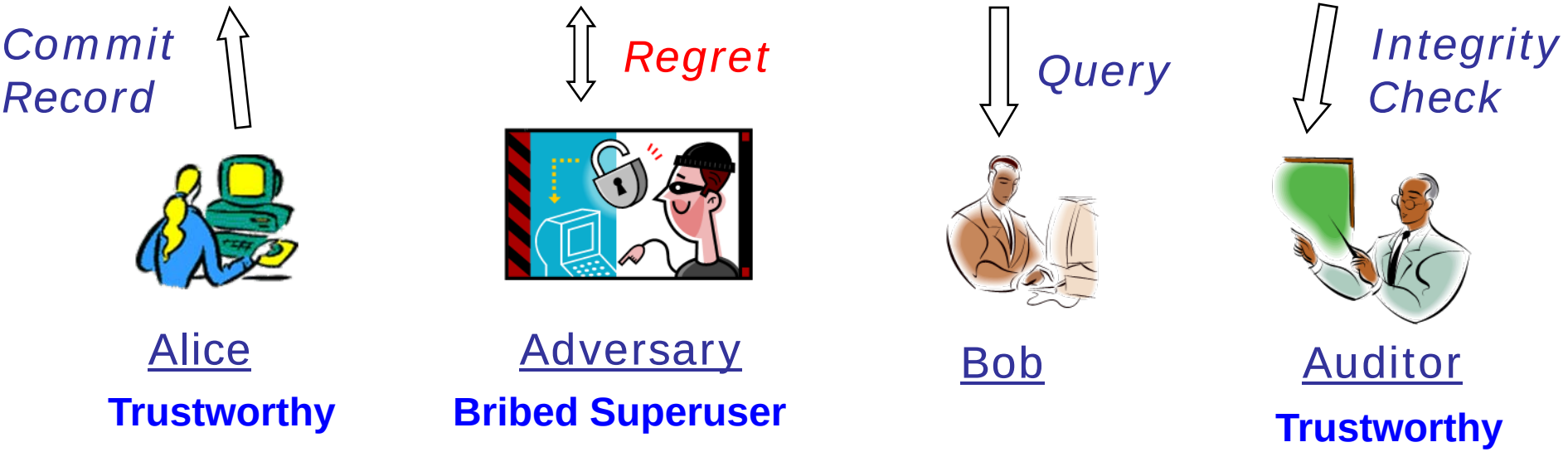
There are many interesting research issues in regulatory compliance beyond SOX, SEC Rule 17a-4, & HIPAA.

Methodology:

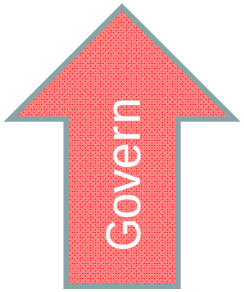
- Understand the regulation and *how it is currently enforced*
- Understand what threats the regulation targets
- Translate those threats into IT-level threats
- Devise novel low-cost IT to address those threats
- *Tech transfer*: Convince policymakers to require its use

Example potential targets: e-govt vital statistics (birth, death, marriage, voter, etc.); stronger assurances for FERPA, GLBA, FISMA at minimal cost

SOX targets CFOs tampering with the info that goes into quarterly reports. Translate that to IT threats:



IT governance means knowing what your assets and policies are.



- Industry sells tools for asset discovery; *what are the open IT-level problems*?
- Policy discovery: how to extract policies embedded in legacy software?
- Role engineering/mining/discovery: how to mine roles from activity logs?
- Permission provisioning: how to assign permissions to new users?
- Can we use cutting-edge info integration techniques to *understand* the information that we find (e.g., determine the meaning of schemas, find PII)?

In conclusion: there's a lot of interesting research to do in policy-based systems.