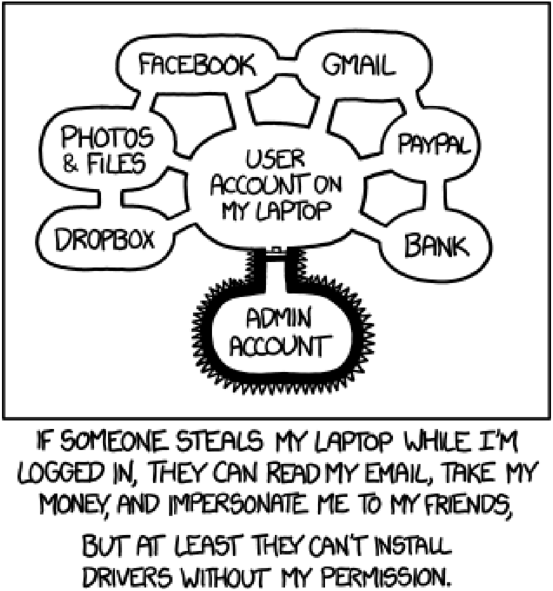
Policy-Governed Secure Collaboration

A Sociotechnical Perspective on Security

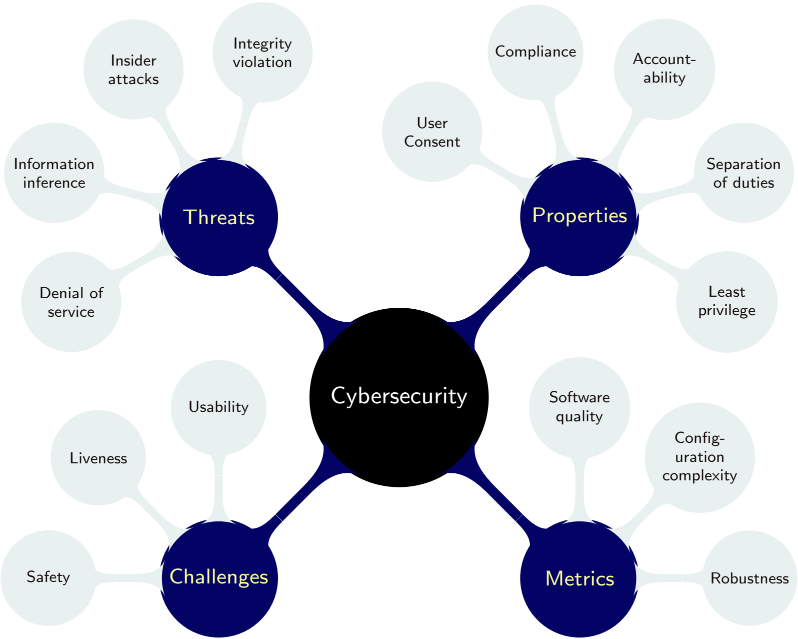
Munindar P. Singh
singh@ncsu.edu

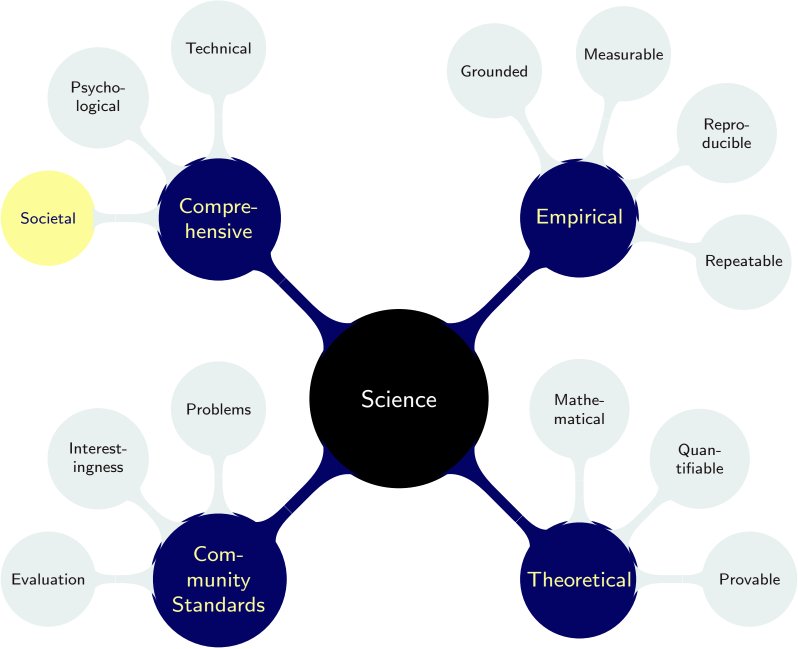
Department of Computer Science
North Carolina State University

XKCD's Assessment of Security Today


FACEBOOK
GMAIL
PHOTOS & FILES
PAYPAL
USER ACCOUNT ON MY LAPTOP
DROPBOX
BANK
ADMIN ACCOUNT
IF SOMEONE STEALS MY LAPTOP WHILE I'M LOGGED IN, THEY CAN READ MY EMAIL, TAKE MY MONEY, AND IMPERSONATE ME TO MY FRIENDS,
BUT AT LEAST THEY CAN'T INSTALL DRIVERS WITHOUT MY PERMISSION.

©Randall Munroe
http://xkcd.com/1200/


Cybersecurity
Threats
Insider attacks
Integrity violation
Information inference
Denial of service
Properties
Compliance
Accountability
User Consent
Separation of duties
Least privilege
Challenges
Usability
Liveness
Safety
Metrics
Software quality
Configuration complexity
Robustness


Science
Comprehensive
Technical
Psychological
Societal
Empirical
Grounded
Measurable
Reproducible
Repeatable
Community Standards
Problems
Interestingness
Evaluation
Theoretical
Mathematical
Quantifiable
Provable

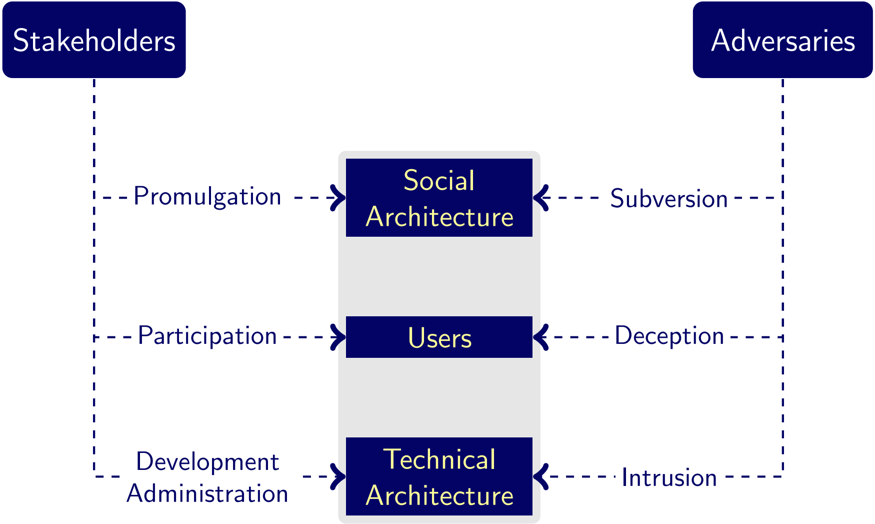
Sociotechnical Systems

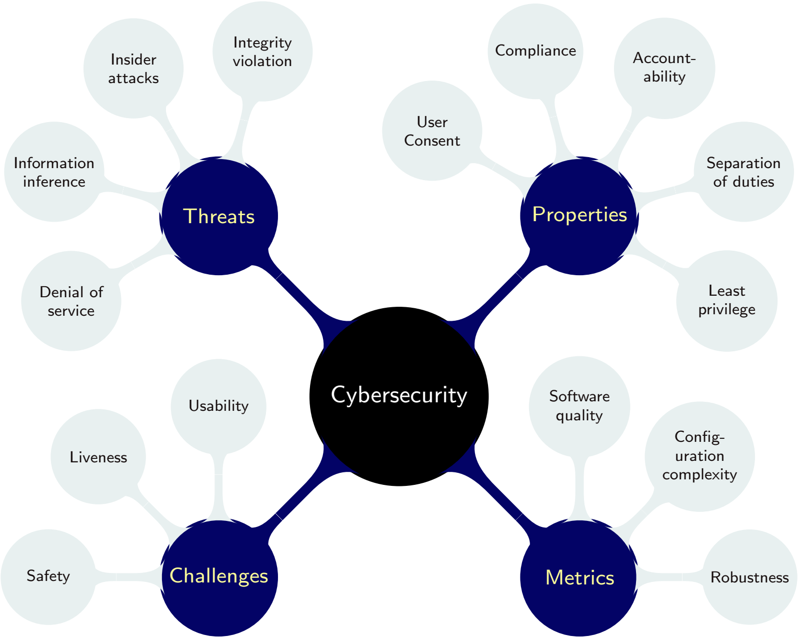
Combine IT with real-life societal considerations

- System characteristics
  - Longevity and identity
  - Autonomy
  - Essentially a society
  - Characterized via norms, not operationally
- Member characteristics
  - Longevity and identity
  - Autonomy
  - Heterogeneity
  - Ability to deal with norms, e.g., via goals realized in policies
- Realization
  - Top down: Members fit into existing system
    - Adopt suitable goals given system norms
- Bottom up: Members design new system
  - Negotiate suitable norms given individual goals

Participants and Artifacts in Security

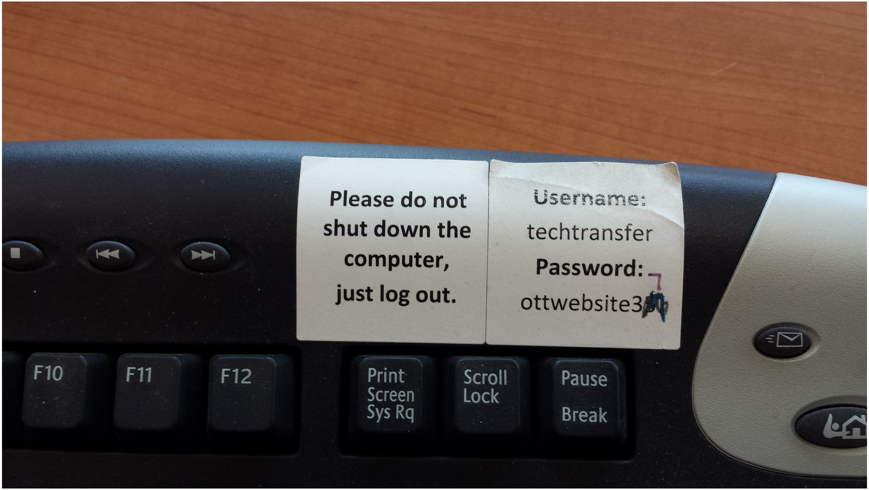
Greatest challenges arise in the upper two; most past effort is on technical architecture

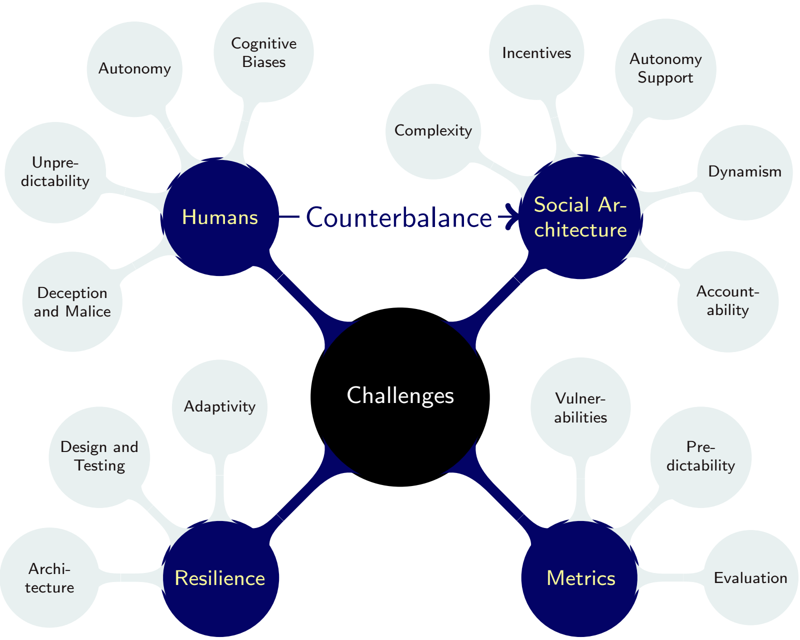



Cybersecurity
Threats
Insider attacks
Integrity violation
Information inference
Denial of service
Properties
User Consent
Compliance
Accountability
Separation of duties
Least privilege
Challenges
Usability
Liveness
Safety
Metrics
Software quality
Configuration complexity
Robustness

Usability and Strange User Behavior

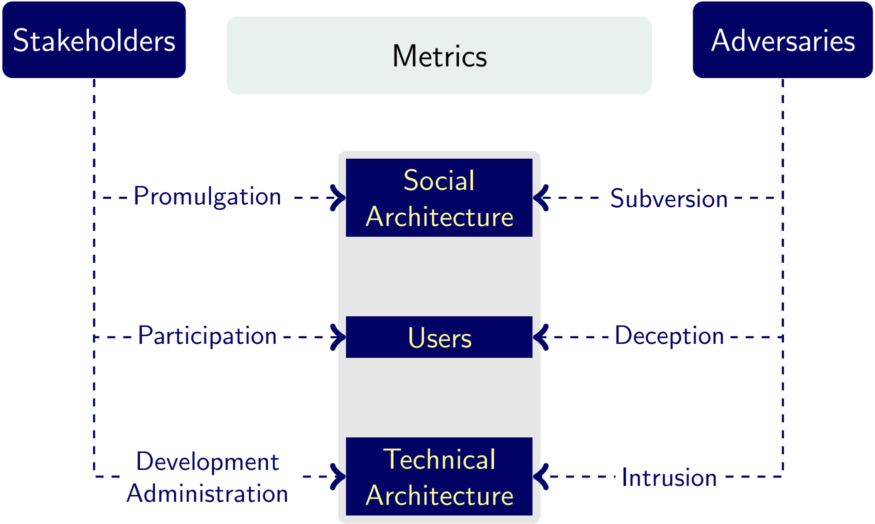
Can we protect users from themselves?




Challenges
Humans
Autonomy
Cognitive Biases
Unpredictability
Deception and Malice
Counterbalance
Social Architecture
Complexity
Incentives
Autonomy Support
Dynamism
Accountability
Resilience
Adaptivity
Design and Testing
Architecture
Metrics
Vulnerabilities
Predictability
Evaluation

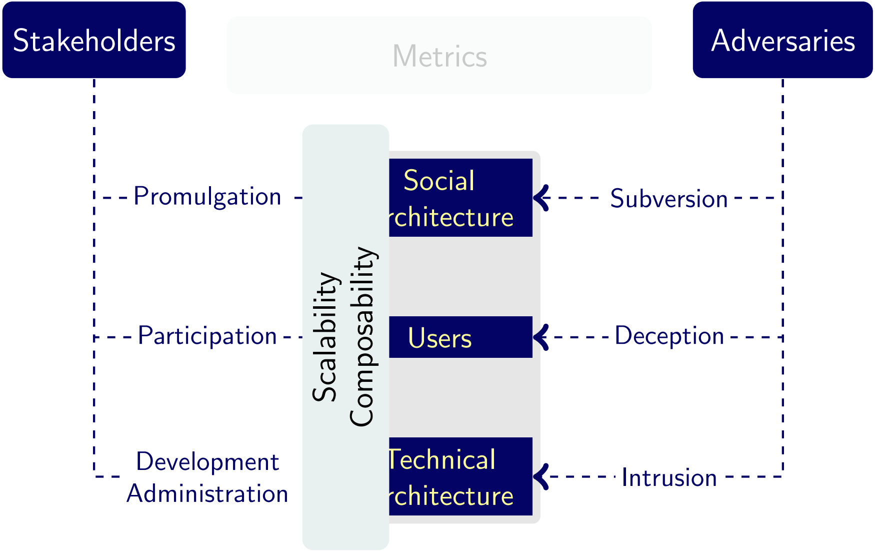
Framing the Hard Problems in Security

Motivations for an emerging research program



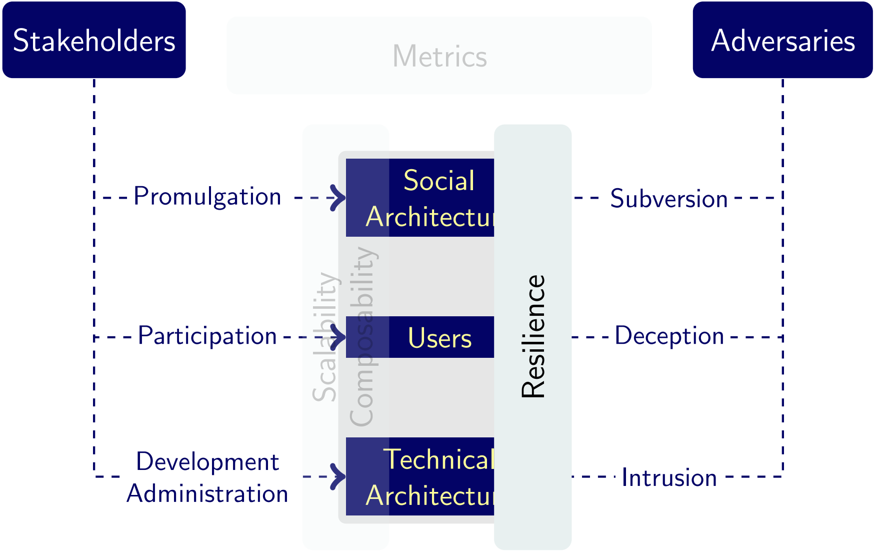
Framing the Hard Problems in Security

Motivations for an emerging research program



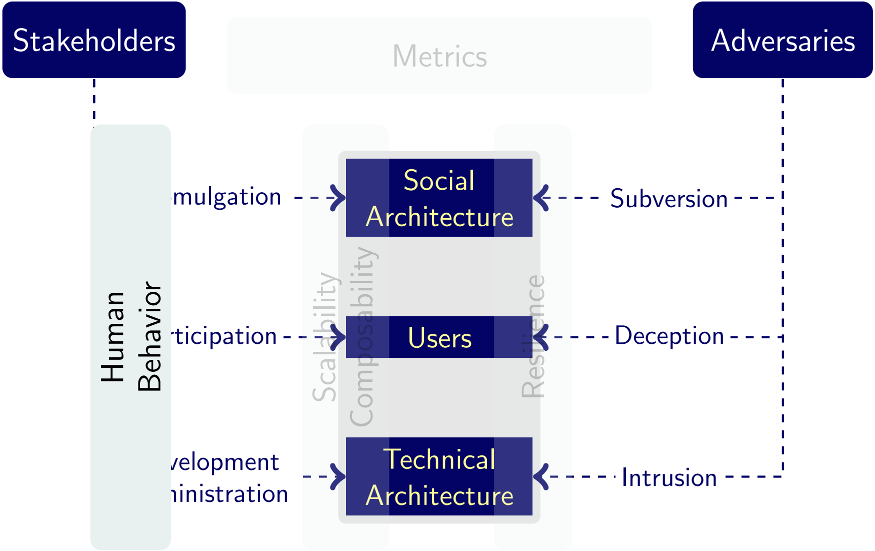
Framing the Hard Problems in Security

Motivations for an emerging research program



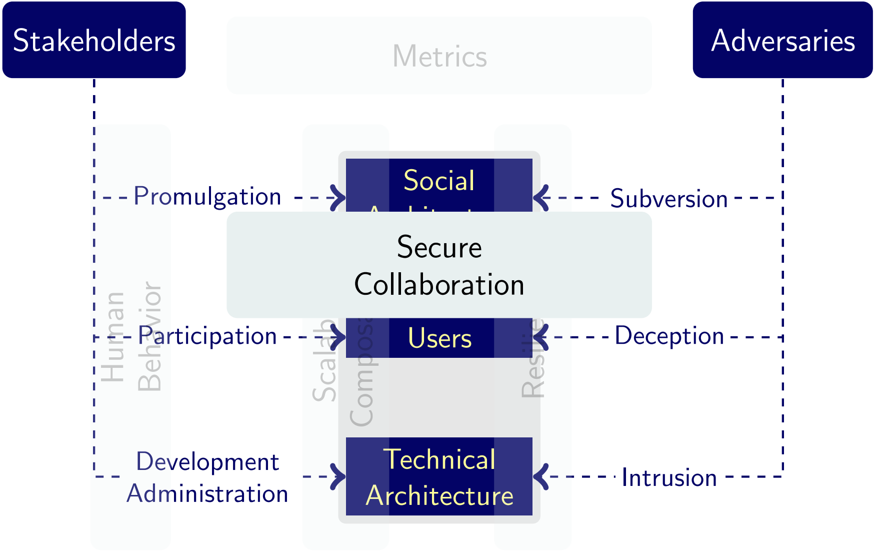
Framing the Hard Problems in Security

Motivations for an emerging research program



Framing the Hard Problems in Security

Motivations for an emerging research program



Stakeholders

Metrics

Adversaries

Human Behavior

Scalab

Composa

Resilie

Promulgation

Social

Subversion

Secure Collaboration

Participation

Users

Deception

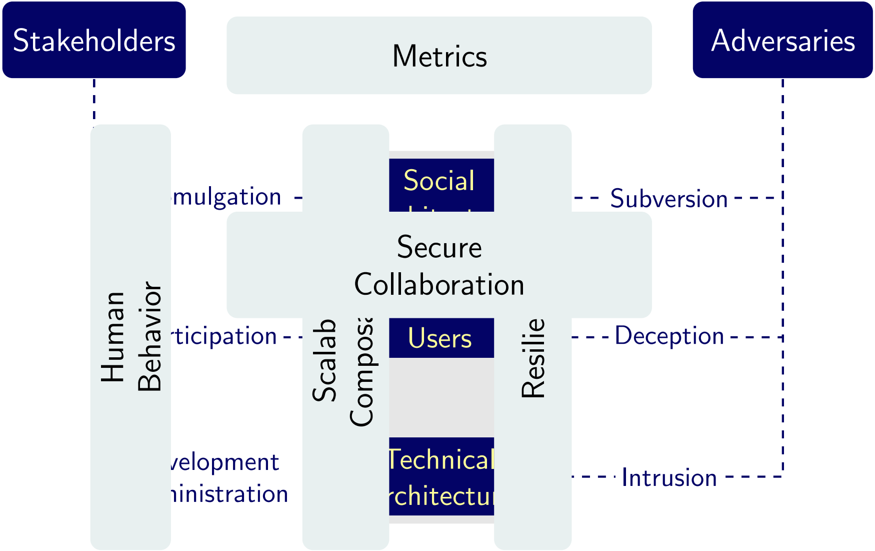
Development Administration

Technical Architecture

Intrusion

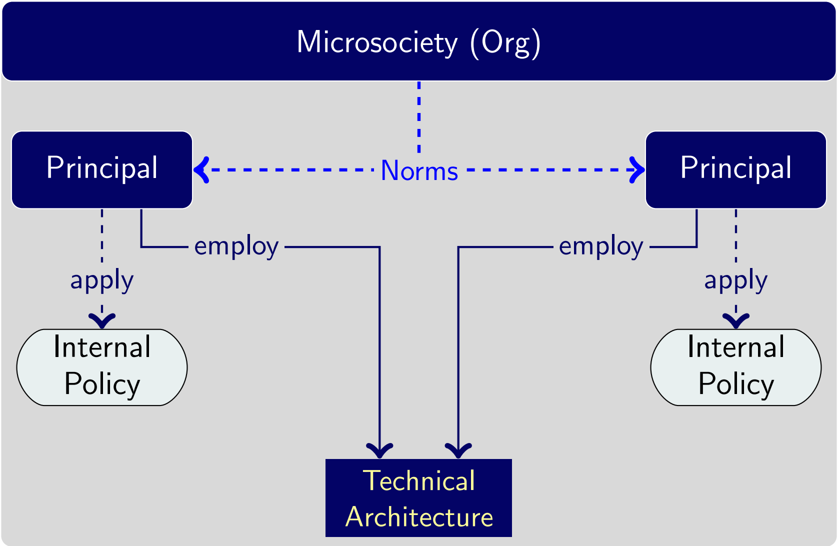
Framing the Hard Problems in Security

Motivations for an emerging research program



A System is a Microsociety

Traditional view: A system is an artifact



Governance for Secure Collaboration

Broadly, administering sociotechnical systems to serve stakeholder needs

- Currently, automated support comes with managerial imposition: by superiors on subordinates
  - Control over managed resources
  - Necessary but not sufficient
  - Unsuited to many settings
    - When user needs aren't met, they subvert managerial diktats
    - Therefore, vulnerabilities
- Currently, governance is manual via out-of-band communications
  - Low productivity
  - Poor scalability to fine-grained, real-time governance decisions
  - Hidden, implicit considerations yield low confidence in correctness and poor maintainability
    - Lead to errors
    - Therefore, vulnerabilities

Governance Challenges in Secure Collaboration

Accommodating autonomy, heterogeneity, and dynamism

- Support configurational adaptation
  - Resource sharing: Offer ocean instrument for sharing
  - Affiliation: Add new laboratories
  - Sanction: Allow external sharing of results to fulfill deliverables
- Support operational adaptation
  - Resource sharing: Preempt low-priority users in case of oil spill
  - Affiliation: Forbid unilateral publishing of results
  - Sanction: Absolve researcher who reveals results to prevent public endangerment (extenuating circumstances)
- Research challenges
  - Abstractions to capture rules of encounter
  - Methods to design and analyze such abstractions
  - Methods to implement such abstractions

Foundations of Secure Collaboration

Social perspective that complements technical (data, application, infrastructure) perspectives

- *Normative basis:* Key relationships are reflected in norms
- *Management of social context:* An Org (as a microsociety) recursively provides the context for the norms among and policies of its members
- *Policy:* An implementation-independent model of decision making and operational semantics
- *Interaction orientation:* How agents apply policies to enter into, monitor, and enact normative relationships

Principles of Governance: What Policies Give Us

Administration that is intelligent and intelligible

- Vividness of modeling
    - Grounded in applications; modeled entities are real
- Minimality of operational specifications
    - Leaving restrictions unstated except where essential to correctness
- Reification of representations
    - Explicit: hence, inspectable, sharable, and manipulable

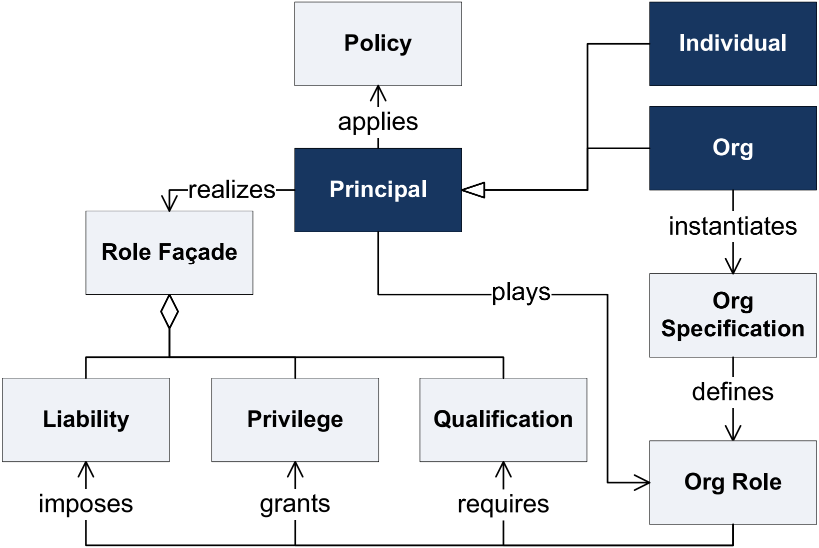
Principles of Governance: What Norms Give Us

Administration that is intelligent and intelligible

- Autonomy and interdependence of participants
  - Stating rules of encounter; omitting policies from specifications
- Centrality of organizations
  - Modeling businesses, communities of practice; specifying rules of encounter; monitoring contracts; sanctioning violators
- Institutional actions
  - Creation and manipulation of commitments; granting or denying powers, authorizations; effecting sanctions
  - Separation of concerns from those of operational interactions

Overview of Policy-Governed Secure Collaboration

Conceptual Model



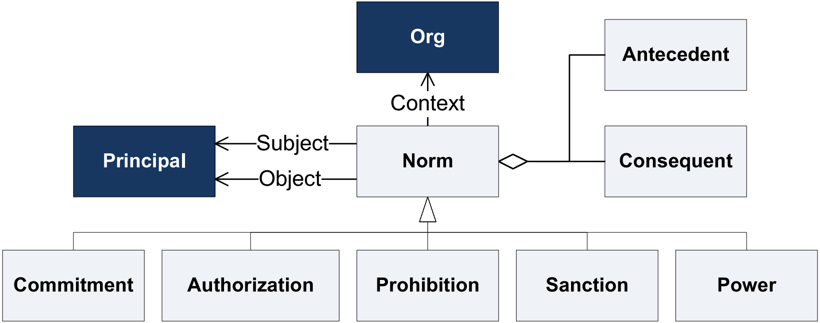
Achieving Governance: Principals and Orgs

Put collaboration in organizations center stage

- Principals are the stakeholders: people and organizations
  - Provide a locus for interaction
- Orgs are like *institutions:* have an identity and life time distinct from their members; also principals
  - Examples: NCSU, DoD, . . .
  - Provide a locus for roles
  - Characterized via norms
  - Potentially enforce norms on members playing specific roles
    - An Org's main hold over its members is the threat of expulsion

Types of Norms

Unified logical form: Norm(subject, object, context, antecedent, consequent)

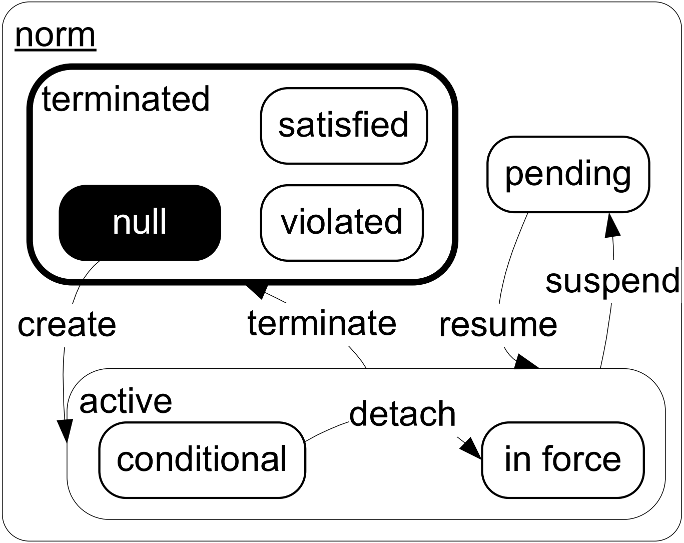


- Directed
- Declarative
- Composable
- Manipulable

Norms as Façades

Norm	**Subject's Façade**	**Object's Façade**
Commitment	Liability	Privilege
Authorization	Privilege	Liability
Power	Privilege	Liability
Prohibition	Liability	Privilege
Sanction	Liability	Privilege

Norm Life Cycle: 1



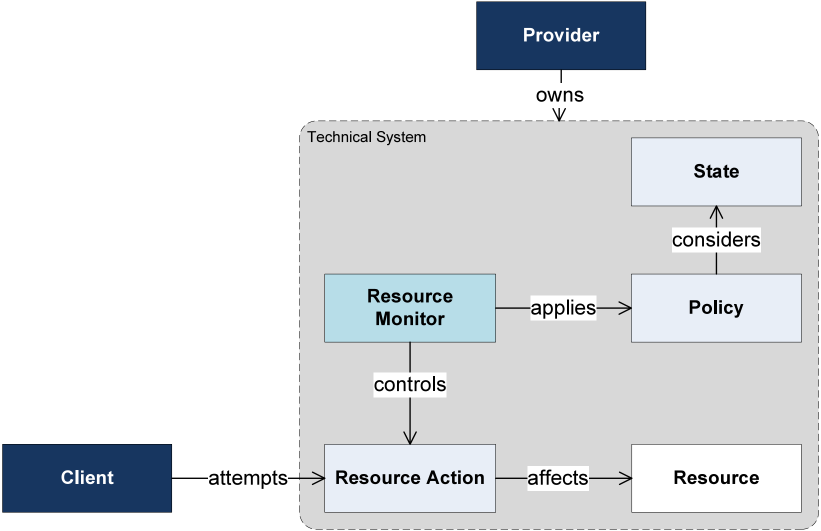
Norm Life Cycle: 2

Substate of a terminated norm

If terminated in		**Then**				
ant	**con**	**Com**	**Aut**	**Pro**	**San**	**Pow**
false	false	null	null	null	null	null
false	true	sat	vio	null	null	null
true	false	vio	null	sat	null	vio
true	true	sat	sat	vio	sat	sat

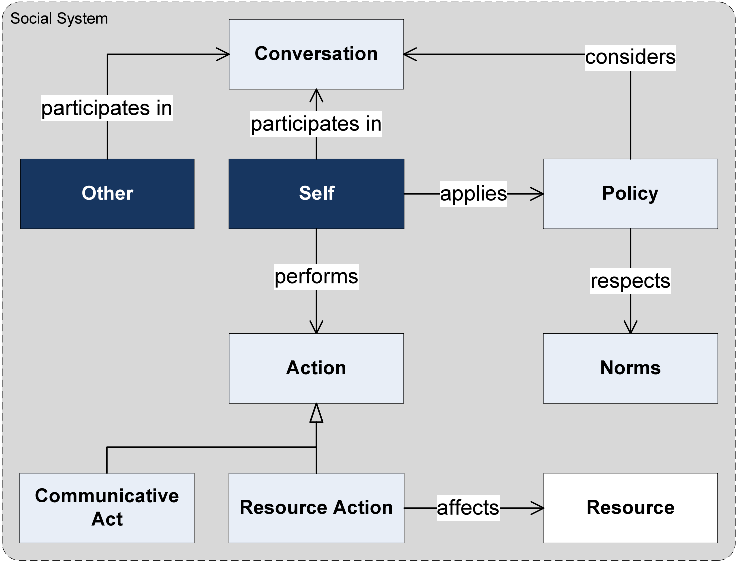
Traditional View: Systems as Artifacts

Traditional application of policies



Proposed View: Systems as Societies

Norm-driven application of policies



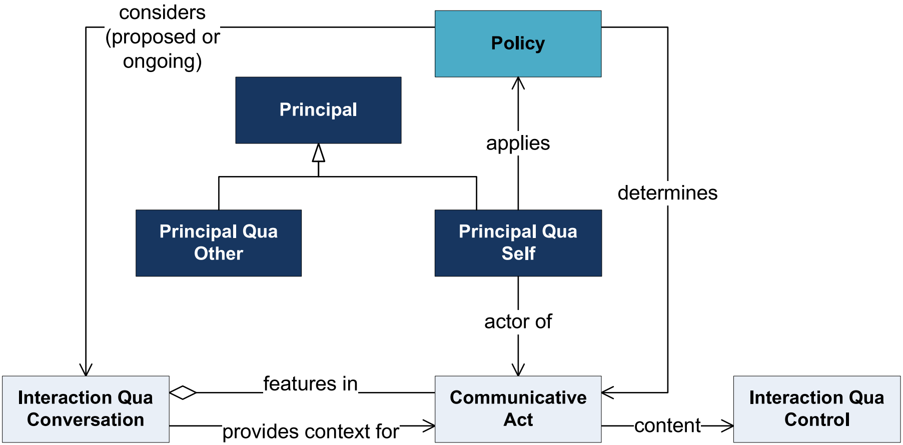
Unifying Norms and Policies for Governance

Promoting precision, verifiability, modularity, and reusability for secure collaboration

- Norms characterize interactions in terms of expectations and accountability
  - Provide the standards of correctness for governance
  - Packaged as role façades
  - Adopted by an agent to support its goals and concomitant policies
  - Help identify *policy points*: where policies apply
- An agent adopts policies that, given its role façades and goals,
  - Support discharging its liabilities
  - Potentially exploit its privileges
  - May not individually or collectively comply with norms
  - May thus violate some security expectations

Governance and Policies: Two Kinds of Interaction

Conversations with autonomous parties; control over resources



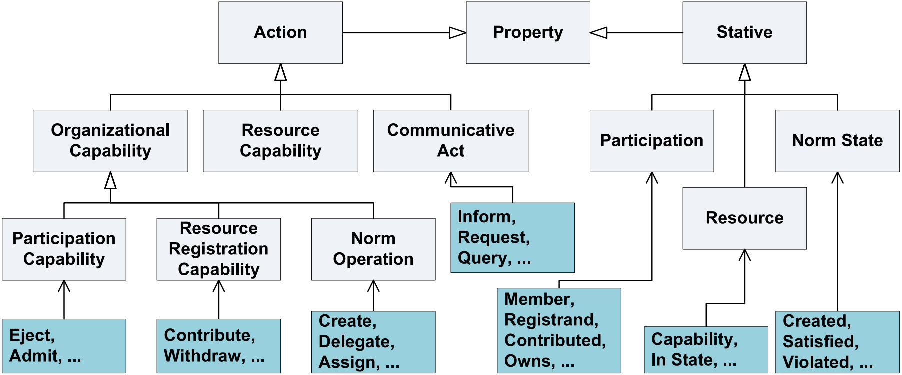
Governance and Policies: Information Model

Relevant information

- Attributes of the parties involved
    - Qualifications, affiliations
- Attributes of the capabilities involved
    - Interactions to be carried out upon resources
    - Collated as interaction types and resource types
- Attributes of the relationships among the parties involved
    - Participations in different Orgs
    - Arrangements among Orgs (captured as participations)
    - Ongoing interactions

Vocabulary for Governance and Policies

Norms and Orgs

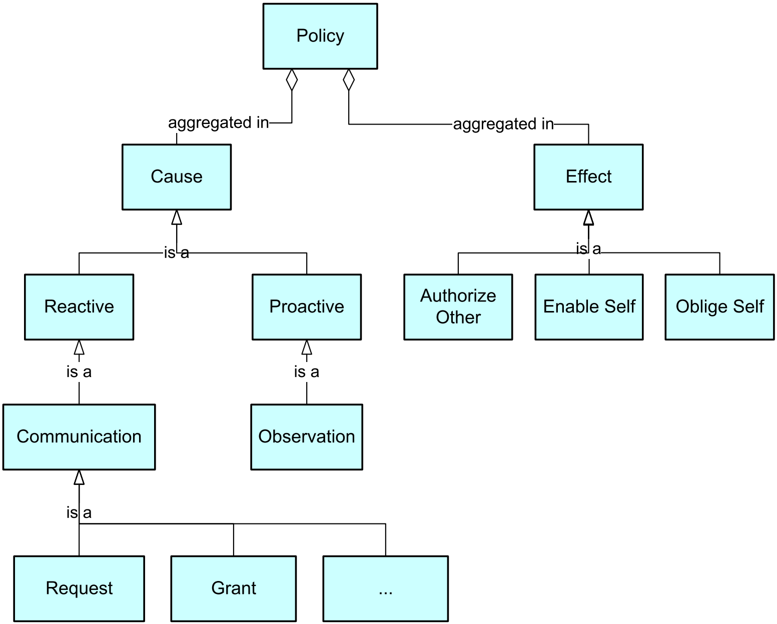


Action
Property
Stative
Organizational Capability
Resource Capability
Communicative Act
Participation
Norm State
Resource
Participation Capability
Resource Registration Capability
Norm Operation
Inform, Request, Query, ...
Eject, Admit, ...
Contribute, Withdraw, ...
Create, Delegate, Assign, ...
Member, Registrand, Contributed, Owns, ...
Capability, In State, ...
Created, Satisfied, Violated, ...

Policy Types

The policy interactions need to go beyond traditional access control

- Each policy can be understood in terms of its cause and its effect
- Cause
    - *Reactive:* triggered by a request from another stakeholder
    - *Proactive:* triggered by local observations
- Effect
    - *Authorization* of action to be taken on behalf of requester
    - *Enablement* of action, which would otherwise not be taken
    - *Obligation* of action, which would now be performed

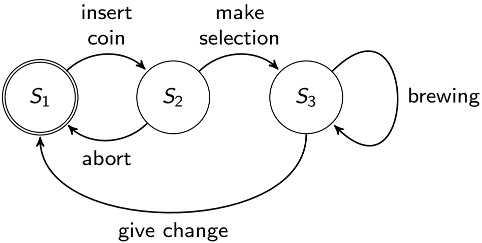

Policy
aggregated in
aggregated in
Cause
Effect
is a
is a
Reactive
Proactive
Authorize Other
Enable Self
Oblige Self
is a
is a
Communication
Observation
is a
Request
Grant
...

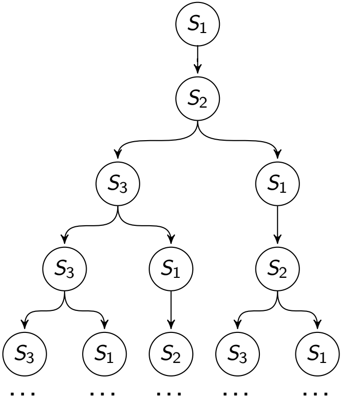
Regulation versus Regimentation

- Regimentation: preventing bad behavior
  - Fits a closed system
  - Reflects a pessimistic stance
  - Presumes a regimenting infrastructure
- Regulation: discouraging and correcting—though *allowing*—bad behavior
  - Fits an open system
  - Reflects an optimistic stance
  - Presumes a regulating social system

Vending Machine in Vienna

Conventional formal methods assume regimentation, i.e., a technical service





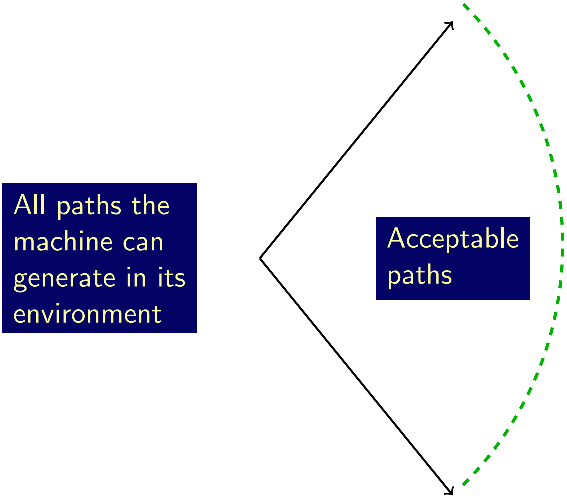
AF[Brew]: On every path, coffee is eventually brewed

A[¬Brew U Coin]: On every path, no coffee is brewed prior to payment

`http://embsys.technikum-wien.at/projects/decs/verification/formalmethods.php`

Regimentation: Violations Aren't Possible

Viable assumption in a closed system

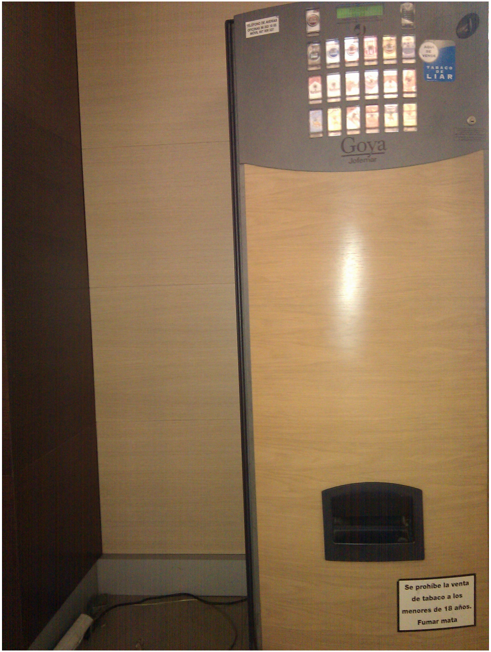


All paths the machine can generate in its environment

Acceptable paths

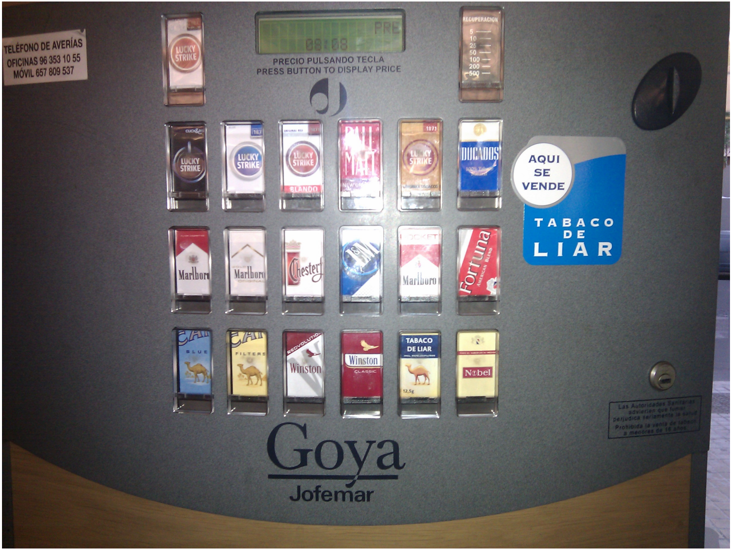
Vending Machine in Valencia

A business service

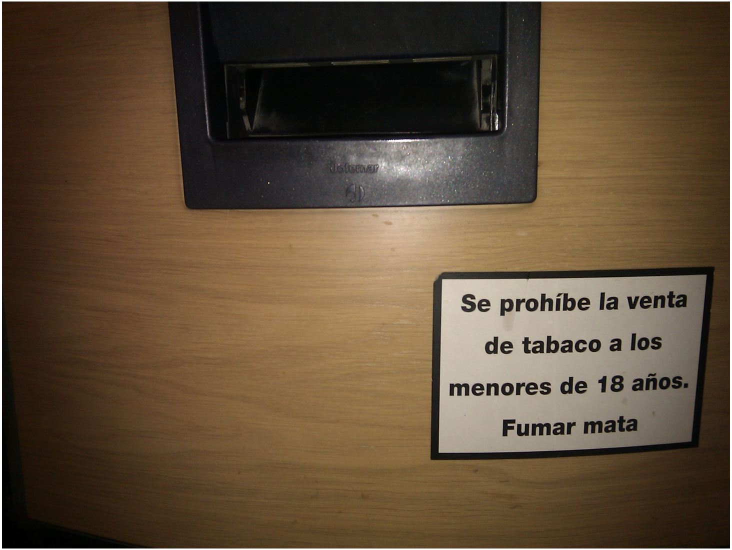

AQUI SE VENDE
TABACO DE LIAR
Goya
Jofemar
Se prohíbe la venta de tabaco a los menores de 18 años. Fumar mata

- Tall structure
- Hard to reach for short people
- Is that a bug or a feature?

Vending Machine Close Up: Cigarettes!

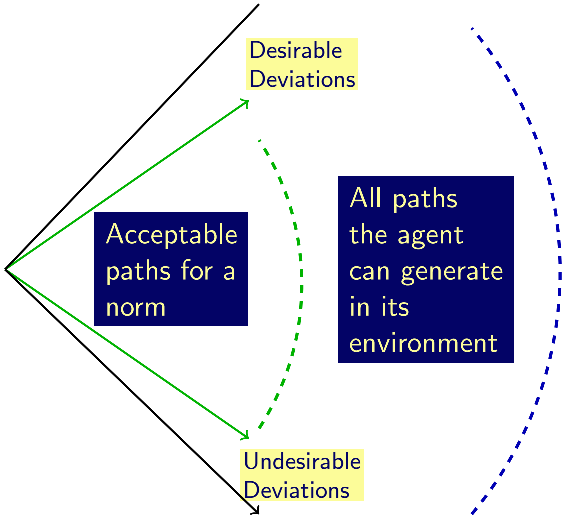


Regulation


Jofemar
Se prohíbe la venta
de tabaco a los
menores de 18 años.
Fumar mata

Regulation: Violations are Possible

Appropriate assumption when dealing with autonomous parties


Desirable Deviations
Acceptable paths for a norm
All paths the agent can generate in its environment
Undesirable Deviations

Outline

Modeling Secure Collaboration

Realizing Secure Collaboration

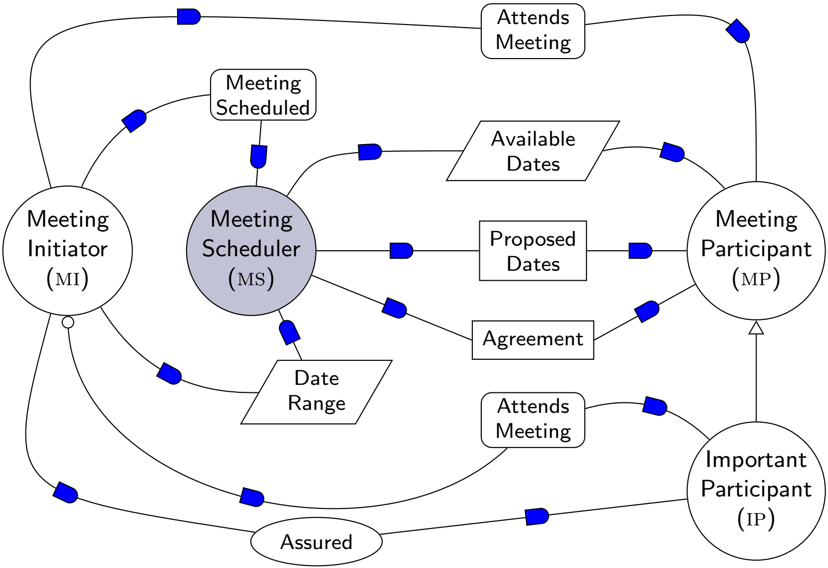
Synthesis

Traditional Software Engineering Approaches

Focus on the technical architecture

- Begin with stakeholders
  - Elicit their goals
  - Determines dependencies between goals
- "Compile out" the stakeholders
  - Producing a *system actor* (specification of a software "machine")
  - That would provide a regimented solution
- Only two parties in the system
  - The software
  - Its environment

Meeting Scheduler: Traditional View



Inadequacy of Traditional SE Approaches

A normative conception matters when engineering STSs

- The system actor is ill-construed
    - Is any actor accountable to it?
    - Is it accountable to any actor?
    - What do these questions even mean for a technical entity?
- What happens when deviations from the imagined scenarios occur?
- What happens when an actor does not have the goal we modeled it with?

Normative Conception of Accountability

A party is *accountable* to another party when the second party has *standing* to expect certain behavior from the first party

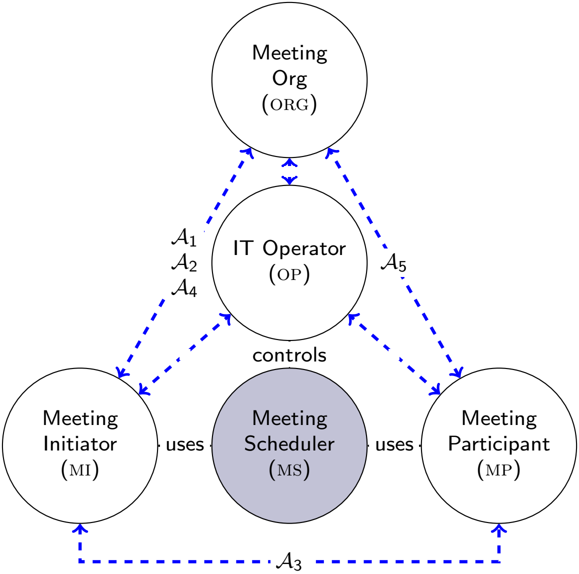
- This definition respects intuitions about accountability in
  - Health care
  - Political science
  - Law
- Every normative relationship creates an accountability
- Autonomy and accountability are two faces of the same coin
  - For any principal: No accountability without autonomy
  - For any society: No autonomy without accountability

Traditional Computing Conceptions of Accountability

Confuse mechanism or outcome with the core concept

- Traceability: actions can be traced to the performer
  - Unnecessary: Alice is a bully and openly commits infractions
  - Insufficient: Alice gets Bob to submit a form for her
  - Insufficient: the tracing mechanism fails silently
  - Insufficient: the traces are not contested
  - Plain wrong: the tracing mechanism is compromised
- Deterrence: specified actions yield a negative utility
  - Simply a more complex norm "N or else penalty" where the penalty voids any accountability
  - In contrast,
    - Nonzero deterrence serves as sanctioning—subsequent to accountability
    - Zero deterrence doesn't absolve one of accountability

Meeting Scheduler: Accountability View



$\mathcal{A}_1$. MI to ORG: valid purpose and invitees

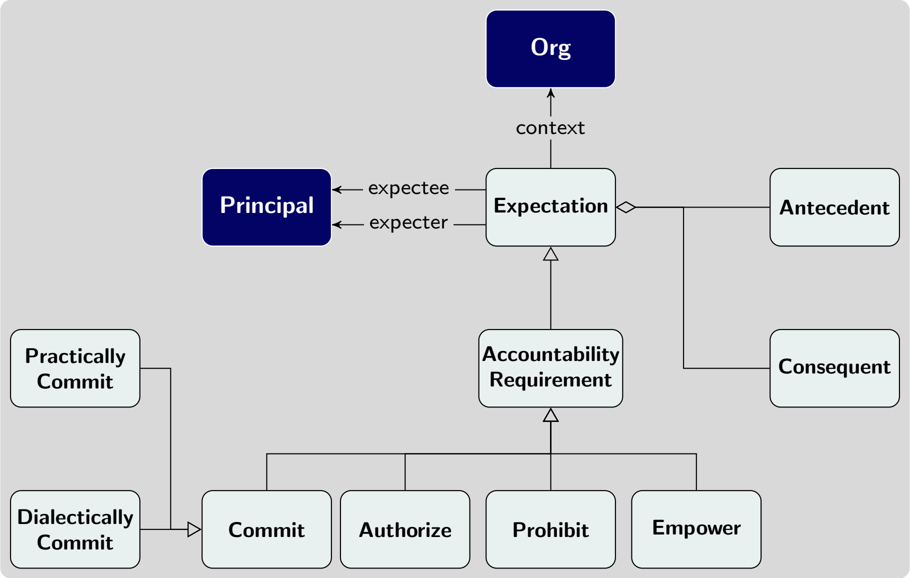
$\mathcal{A}_2$. MI to ORG: meeting times

$\mathcal{A}_3$. MP to MI: attend if accepted

$\mathcal{A}_4$. MI to ORG: power down room

$\mathcal{A}_5$. MP to ORG: clearing the room

Accountability Requirements



Accountability Requirements for Cybersecurity

- Identify the stakeholders
- Identify the normative relationships that would achieve their objectives
  - Functional
  - Security
- Each principal applies its policies to participate in the system
  - Accountable for the normative relationships that are among its liabilities

Challenges and Partial Recent Progress

- Storing and retrieving events to determine the state of a norm
  - Mapping commitments to relational algebra [AAAI 2015]
- Maintaining alignment of views despite decentralization
  - Communications to guarantee (eventual) alignment [AAMAS 2015]
  - TBD: maximizing partial or "quick" alignment
- Designing protocols and Org contexts for monitorability
  - Failure of compositionality of monitorability [IJCAI 2015]
  - Automatically close a context to ensure monitorability
- Designing protocols and Org for robustness and resilience
  - Typology of sanctions and sanctioning processes [Draft]
  - Preliminary simulation study [HotSoS 2015]
  - TBD: Formalization of normative robustness and resilience
  - TBD: Reasoning about sanctions for design of Orgs
- Design processes conducive to autonomy
  - Abstract formal model of a sociotechnical design process [RE 2014]
  - TBD: Methodologies

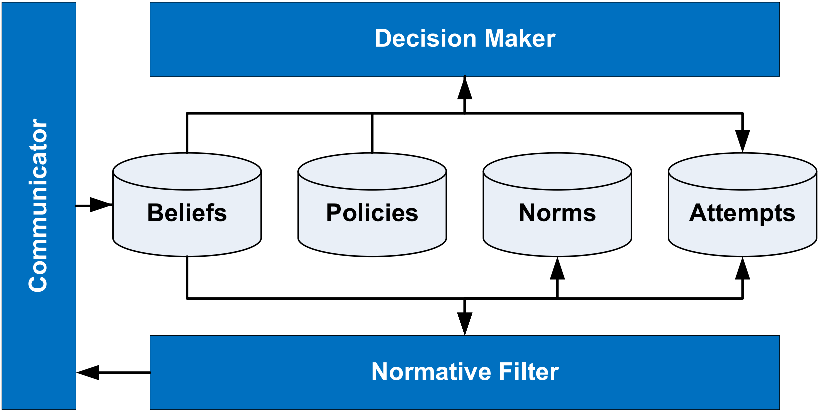
Outline

Modeling Secure Collaboration

Realizing Secure Collaboration

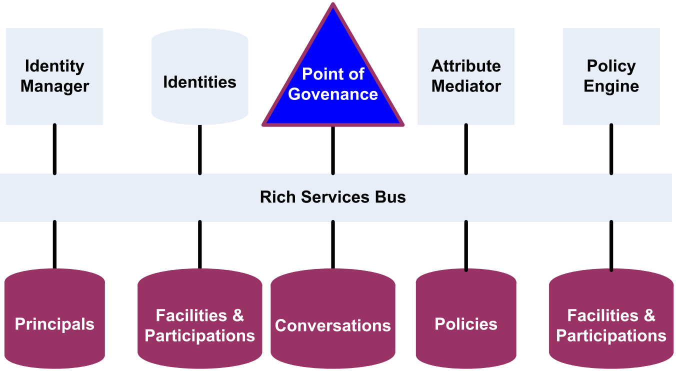
Synthesis

Architecture for a Normative Agent



Point of Governance

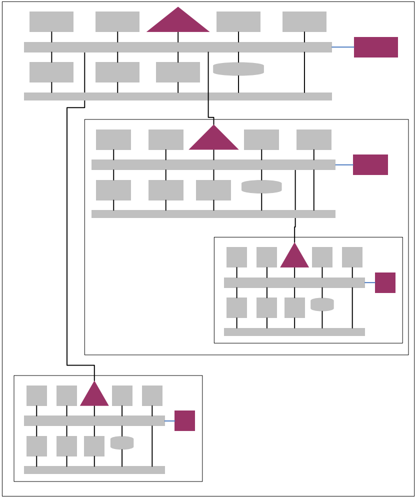
A way to implement an Org in a conventional approach



Normative Conception of Architecture

The connectors are not data or control flow but normative relationships

- Autonomy is key
- Abstraction and encapsulation
  - Norms describe what, not how
  - The rules of encounter differ from strategies for participation
  - Fractal structure of Orgs
    - Turtles all the way
- Dynamism of membership and strategies
- Motivate new architectural styles, e.g.,
  - Make at least one party accountable for each requirement
  - Make exactly one party accountable for each requirement
  - Ensure each Org controls its infrastructure
  - Ensure each Org provides identity for its members
- Motivate new properties for validation, e.g.,
  - The information inference vulnerability is avoided
  - Certain actions cannot be performed unless two agents agree



Fractal structure

Outline

Modeling Secure Collaboration

Realizing Secure Collaboration

Synthesis

Highlights

Differences with some of the literature

- A norm
  - First-class concept, not to be confused with a belief, goal, or policy
  - Directed
  - Manipulable
  - Helps define Orgs and is defined within Orgs
- An Org
  - Active entity, not a specification
  - Lacks any inherent powers
  - Doesn't regiment interactions: members can violate norms
- A role
  - A specification, not an active entity
  - Inherently incomplete: an adopting agent would supply its policies to determine specific decisions
- Enactment of operations
  - Minimize operational restrictions
  - Lie above a declarative language *Blindingly Simple Protocol Language*

What Does Policy-Governed Secure Collaboration Require?

A rich panopoly of research challenges in norms

- Security (and computing) need to think outside the box, literally
- Autonomy means dealing with regulation, not regimentation
- It may be beneficial to violate norms sometimes
    - But without undermining norms altogether
- Normative systems may not be well formed
- Normative systems can be undercut by insider attacks
- Norms demonstrate complex structure
- The unpredictability of user behavior complicates security
    - All the more reason to formulate effective norms

Recommendations for Research

Autonomy, autonomy, autonomy

- Conceptual models
  - Norms and Orgs as bases for policy models
  - Interaction as key
  - Supporting regulation, not just regimentation
- Operational models
  - Architectures that support regulation
    - Monitoring
    - Sanctioning
  - Agent representation and reasoning to support governance
    - Proactive behavior
    - Incorporating goals as duals of norms
  - Declarative language for protocols: Blindingly Simple Protocol Language
- Incorporate tools and methods from computational social science and application areas such as epidemiology

Thanks!

http://www.csc.ncsu.edu/faculty/mpsingh/

http://www.csc.ncsu.edu/faculty/mpsingh/