# Agent-Based Privacy Management for Social Media

Pınar Yolum
Email: p.yolum@uu.nl
Twitter: @pyolum

Department of Information and Computing Sciences
Utrecht University

# Privacy

- ”Right to be let alone” (Warren and Brandeis, Harvard Law Review, 1890)
- ”A state in which one is not observed or disturbed by other people.” (Merriam-Webster Dictionary)
- ”Someone’s right to keep their personal matters and relationships secret” (Cambridge Dictionary)
- ”The right and ability of an individual to define and live his or her life in a self-determined fashion” (substantive privacy) (Dennedy, Fox, and Finneran, ”The Privacy Engineer’s Manifesto”, 2014)
  - By the individual
  - By others
  - By using the data about the person

# Data Privacy

- What data are private?
  - Name, email
  - Financial information
  - Political opinions
  - Sexual orientation
  - Racial or ethnic origin
  - Medical conditions
- Privacy vs. Security

# The Organization for Economic Cooperation and Development (OECD) Guidelines

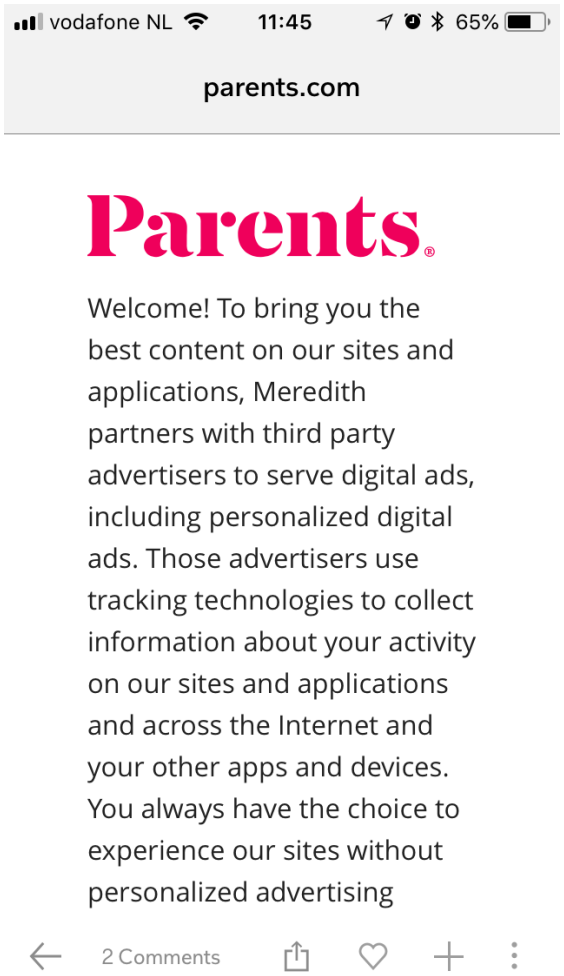
- Collection Limitation: Should be collected lawfully and fairly
- Data Quality Limitation: Should be
  - relevant: Allow appropriate content to be accessed (e.g., age is relevant for checking credit history but phone number is not)
  - accurate: Allow owners to update if necessary
- Purpose Definition Required: Specify explicitly why that information is being shared
- Use Limitation Principle: How it is going to be used (e.g., share with third parties?)
- Accountability Principle: Data sharer will be kept accountable for not abiding with rules

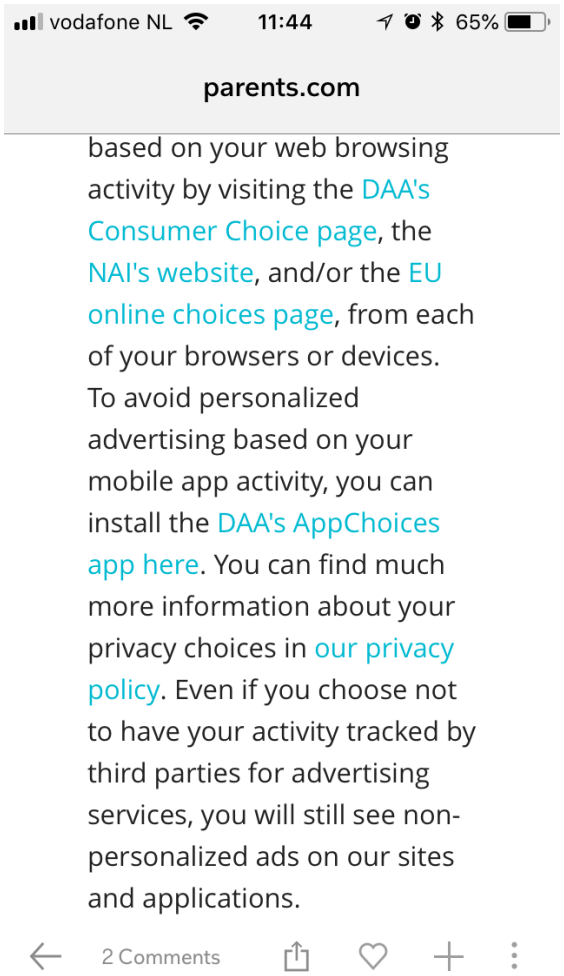
# Authorization Types

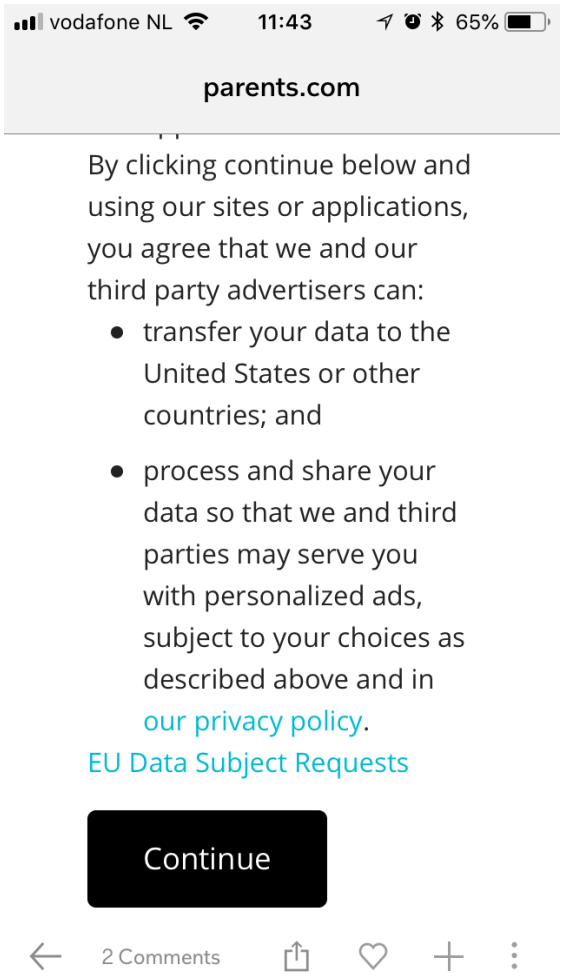
- Opt out (Default is to share)/Opt in (Default is not to share)
- Implied Consent (Your email address appearing on the instructor's list for possible future communication)
- Informed Consent (Explicitly explained how and which information will be used)
- Expressed Consent (Explicitly specified by the user by checking a box or similar)

# General Data Protection Regulation (GDPR)

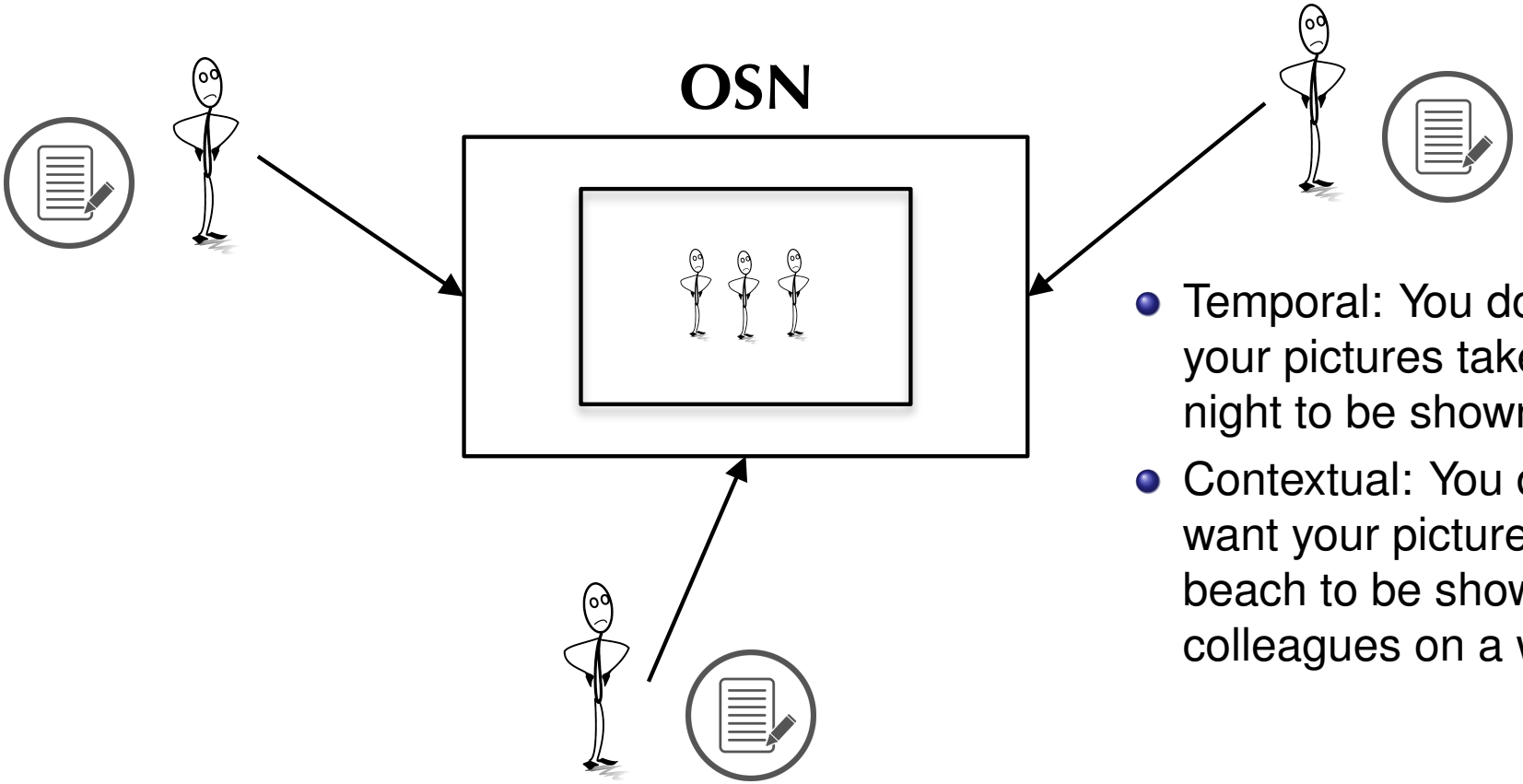
## Informed Consent: Explains what and how information is used



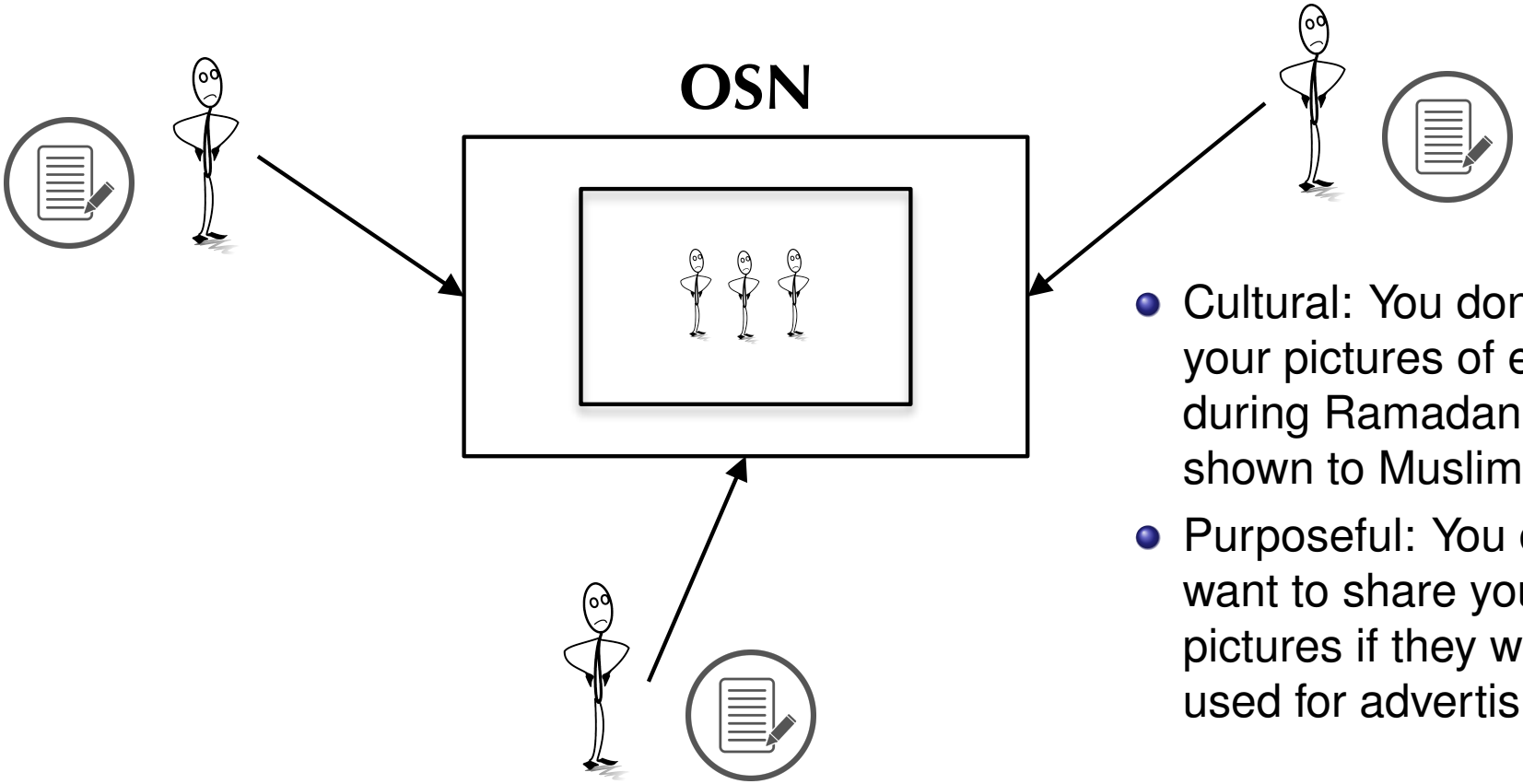



# Privacy in Online Social Networks



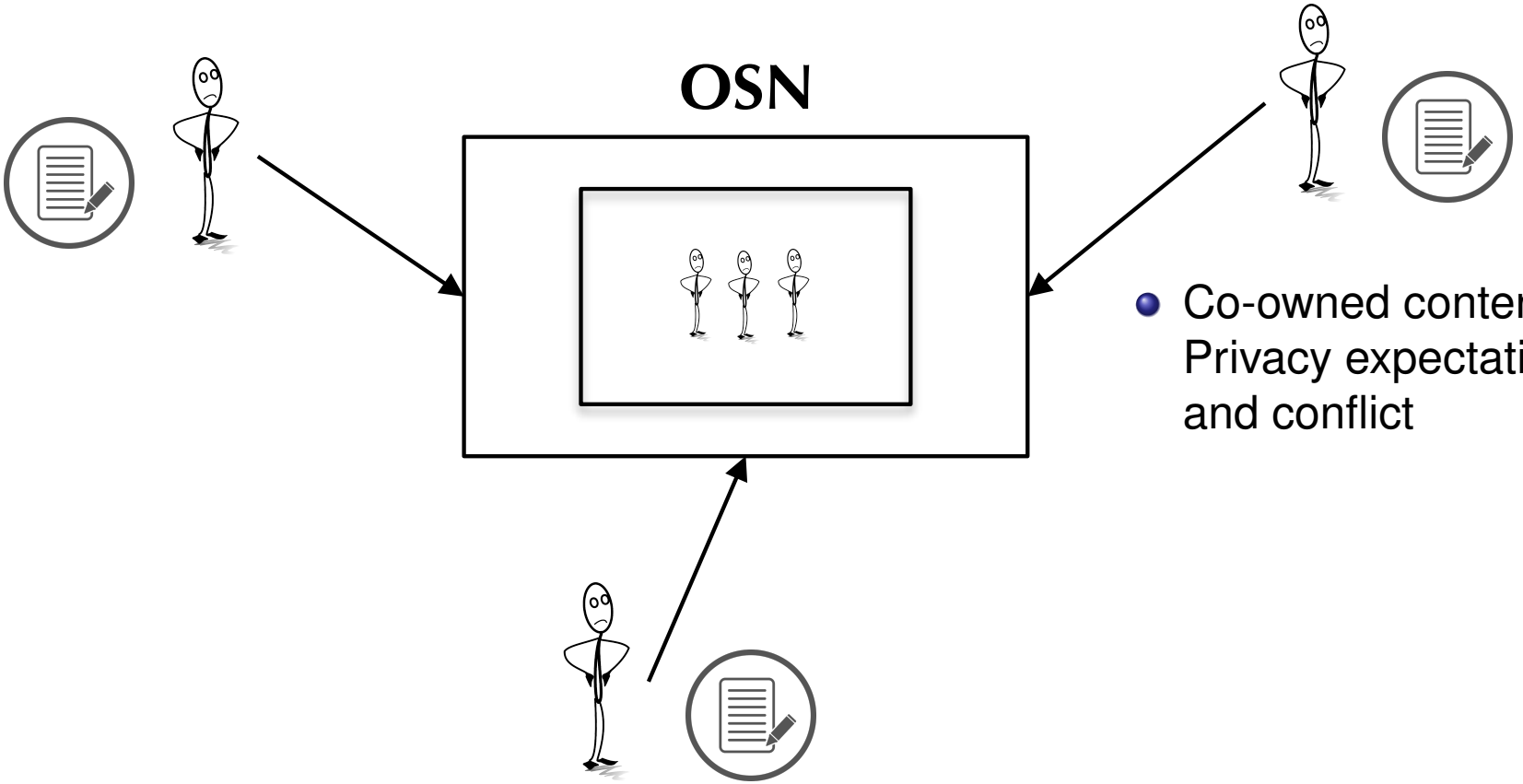

- Temporal: You don't want your pictures taken at night to be shown
- Contextual: You don't want your pictures at the beach to be shown to colleagues on a work day

# Privacy in Online Social Networks




- Cultural: You don't want your pictures of eating during Ramadan to be shown to Muslim friends
- Purposeful: You don't want to share your pictures if they will be used for advertising

# Privacy in Online Social Networks




- Co-owned content: Privacy expectations vary and conflict

# Privacy in Online Social Networks

- Lane v. Facebook: A Class-action lawsuit
  - Sean Lane purchases a diamond ring from Overstock.com.
  - This information shows up on the newsfeed of many of his friends, including his fiancee.
  - This was result of Beacon app, with opt-out privacy options.
  - Facebook ended up paying $9.5M
  - Moral: Information propagates
- Celebrity Stalking (from ABC News)
  - iPhones embed picture locations into the picture (known as geotags)
  - Geotags can easily be deciphered by apps, reveling the location even when not intended
  - Not only bad for celebrities (Craiglist pictures)
  - Moral: Information implies other information

# Understanding Privacy Violations

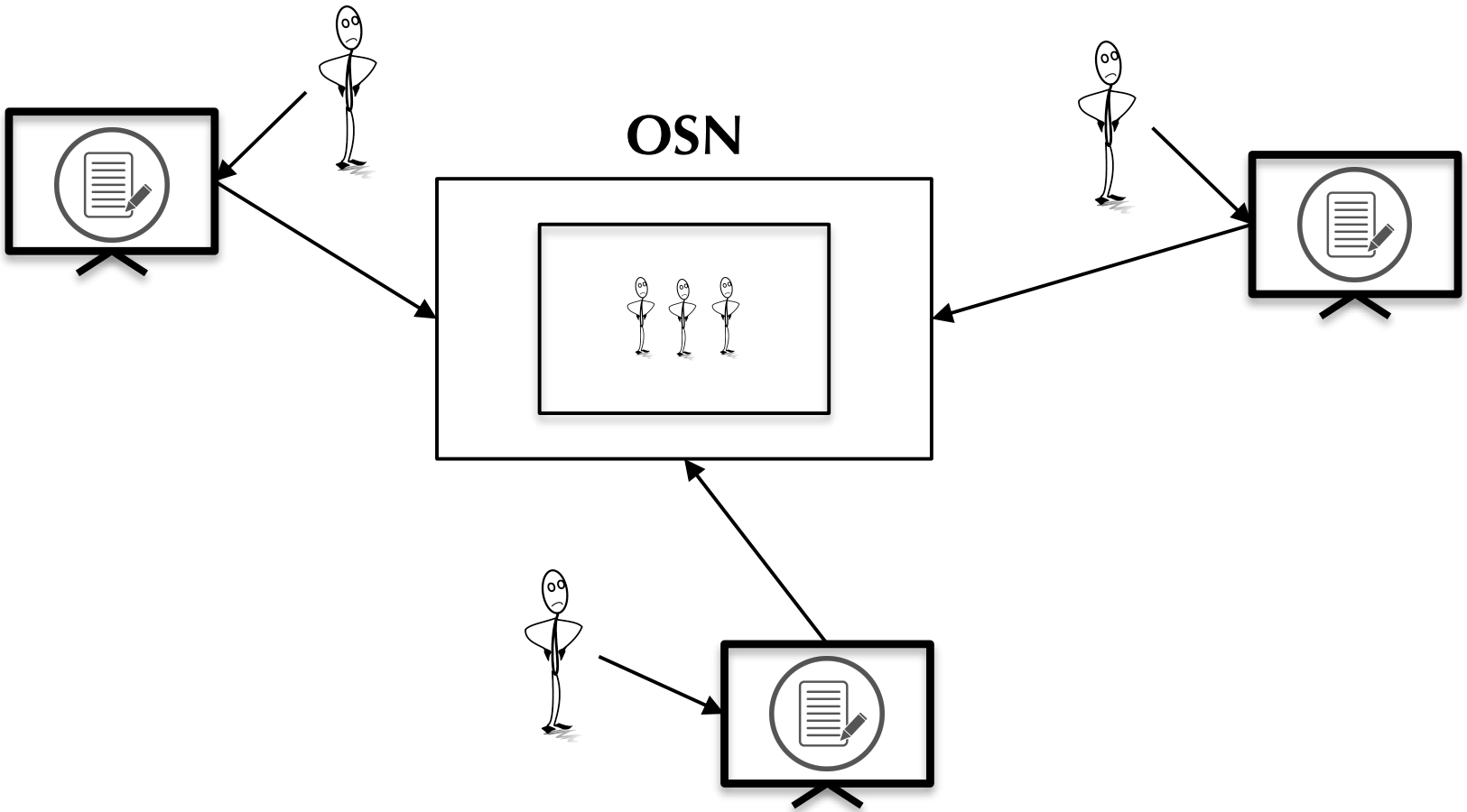
| | **No inference** | **Inference** |
|---|---|---|
| **User** | (i) OSN showing the user's media without consent or user wrongly configuring privacy constraints | (iii) Identifying user's location from a geo-tag in the pictures |
| **Others** | (ii) Friend tags the user and makes the picture public where the user did not want to be seen | (iv) Friend tags the user revealing friendship status even when the user had hid her friend list |

We have conducted an online survey with 330 participants. More than 96% of the participants face privacy violations that occur through inferences.

# Dennis wants his friends to see his pictures but not his location.

| | **No inference** | **Inference** |
|---|---|---|
| **User** | (i) Dennis checks in at a restaurant. | (iii) Dennis shares a picture without declaring his location. It turns out that his picture is geotagged. |
| **Others** | (ii) Charlie shares a picture with everyone. He tags Dennis in it as well. | (iv) Charlie checks in at a restaurant. At the same time, Dennis shares a picture of Charlie. |

# Agent-Based Privacy Management



# How to Manage the Privacy of Users?

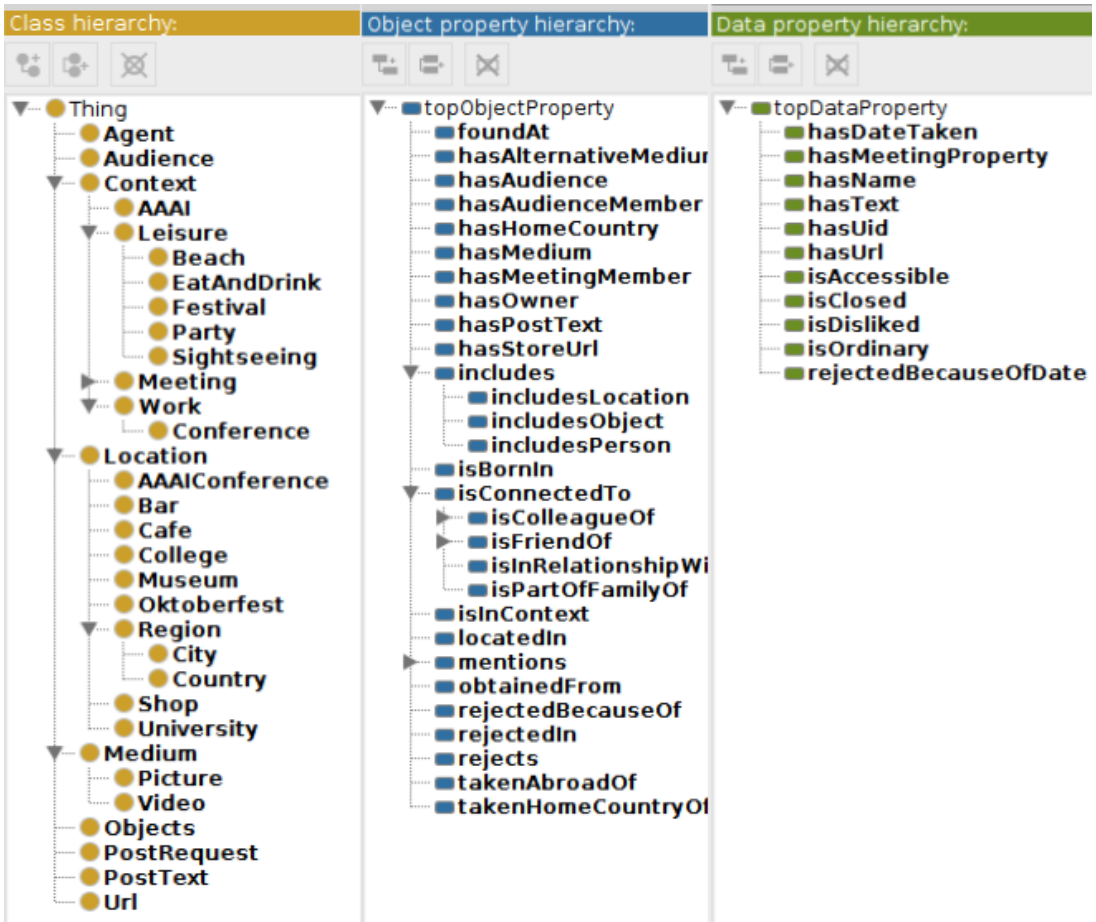
- How to represent the actual privacy preferences of users?
- How to elicit or learn the privacy preferences from users?
- How to advise the users to take actions that are in line with their privacy preferences?
- How to detect potential privacy violations on a user's side?
- How to agree on how a co-owned content will be shared?

# Representations of Privacy Preferences

- Access control: Regulate who can view, edit, use resources
- Role-Based: Users take up roles and act in accordance (RBAC)
- Relation-Based: Capture relations among users
- Attribute-Based: Rules based on values of attributes
- Policy-Based: Enable rules to work in harmony

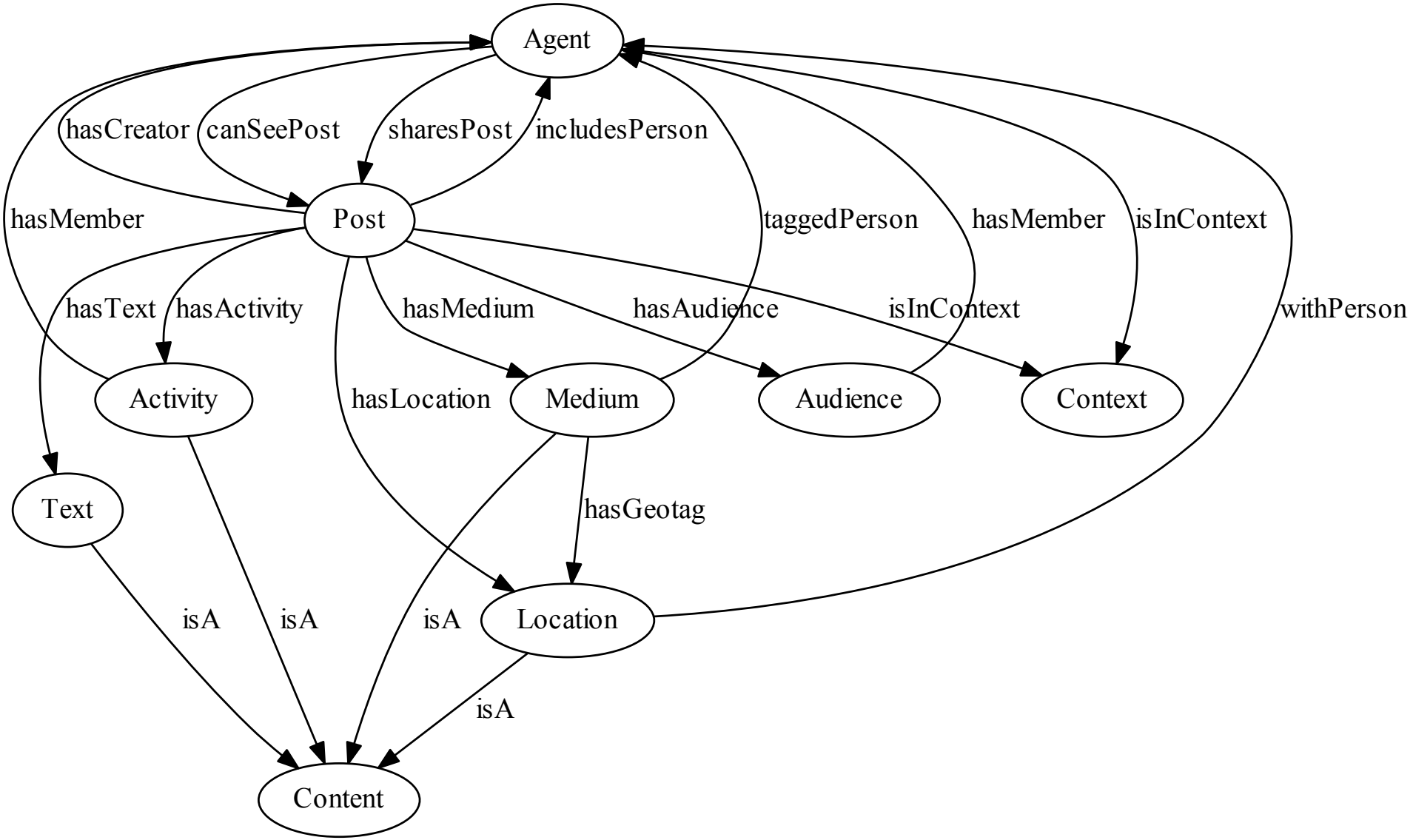
# Semantic Representations

Rely on a knowledge representation, such as an ontology, for reasoning on the content.




- *Concepts* represent a class of individuals (e.g., wig `:wig` is an instance of *Object*).
- Object properties relate different individuals with a specific relation (e.g., includesObject relates a `:Medium` to a `:wig`).
- Data properties relate data values to individuals (e.g., isOrdinary relates `:wig` to either *true* or *false*).

# Content Ontology



# Privacy Rules

"Alice does not want her colleagues to see her leisure photos."

---

$P^6_{A_1}$: *hasAudience*(?postRequest, ?audience), *hasAudienceMember*(?audience, ?audienceMember), `Leisure`(?context), *hasMedium*(?postRequest, ?medium), *isInContext*(?medium, ?context), *isColleagueOf*(?audienceMember, `:alice`) $\rightarrow$ *rejects*(`:alice`, ?postRequest), *rejectedIn*(?audience, ?postRequest), *rejectedBecauseOf*(?audience, ?audienceMember)

---

# Privacy Rules

"Alice does not want her colleagues to see her leisure photos."

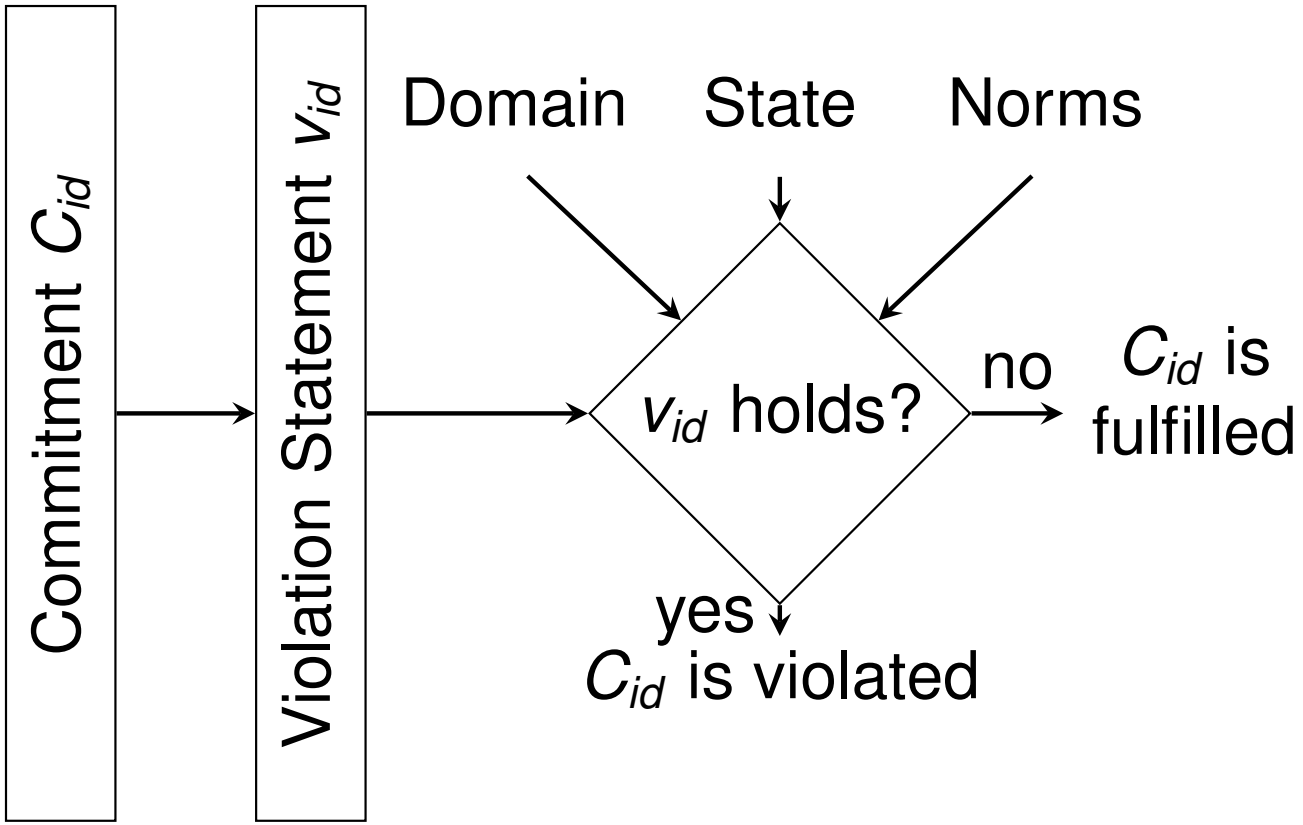
---

$P^6_{A_1}$: *hasAudience*(?postRequest, ?audience), *hasAudienceMember*(?audience, ?audienceMember), `Leisure`(?context), *hasMedium*(?postRequest, ?medium), *isInContext*(?medium, ?context), *isColleagueOf*(?audienceMember, `:alice`) $\rightarrow$ *rejects*(`:alice`, ?postRequest), *rejectedIn*(?audience, ?postRequest), *rejectedBecauseOf*(?audience, ?audienceMember)

---

We can build software agents that can reason on users' privacy preferences.

# Detection Privacy Violations with PriGuard[1]




[1] Nadin Kökciyan and Pınar Yolum. "PriGuard: A Semantic Approach to Detect Privacy Violations in Online Social Networks". In: *IEEE Transactions on Knowledge and Data Engineering* 28.10 (2016), pp. 2724–2737.

# Representation of Privacy Requirements

- Commitments are a powerful representation for modeling multiagent interactions.
- Here used to represent the privacy agreement between a user and the OSN.
- A commitment is denoted as a four-place relation: *C(debtor; creditor; antecedent; consequent)*

| |
|---|
| $C_1$(:osn; :dennis; *isFriendOf*(:dennis,*X*), *sharesPost*(:dennis,*P*), MediumPost(P); *canSeePost*(*X*,*P*)) |
| *Friends of Dennis are allowed to see medium posts of Dennis* |
| $C_2$(:osn; :dennis; *isFriendOf*(:dennis,*X*), *sharesPost*(:dennis,*P*), LocationPost(P); not(*canSeePost*(*X*,*P*))) |
| *Friends of Dennis are not allowed to see location posts of Dennis* |

# Violation Statements

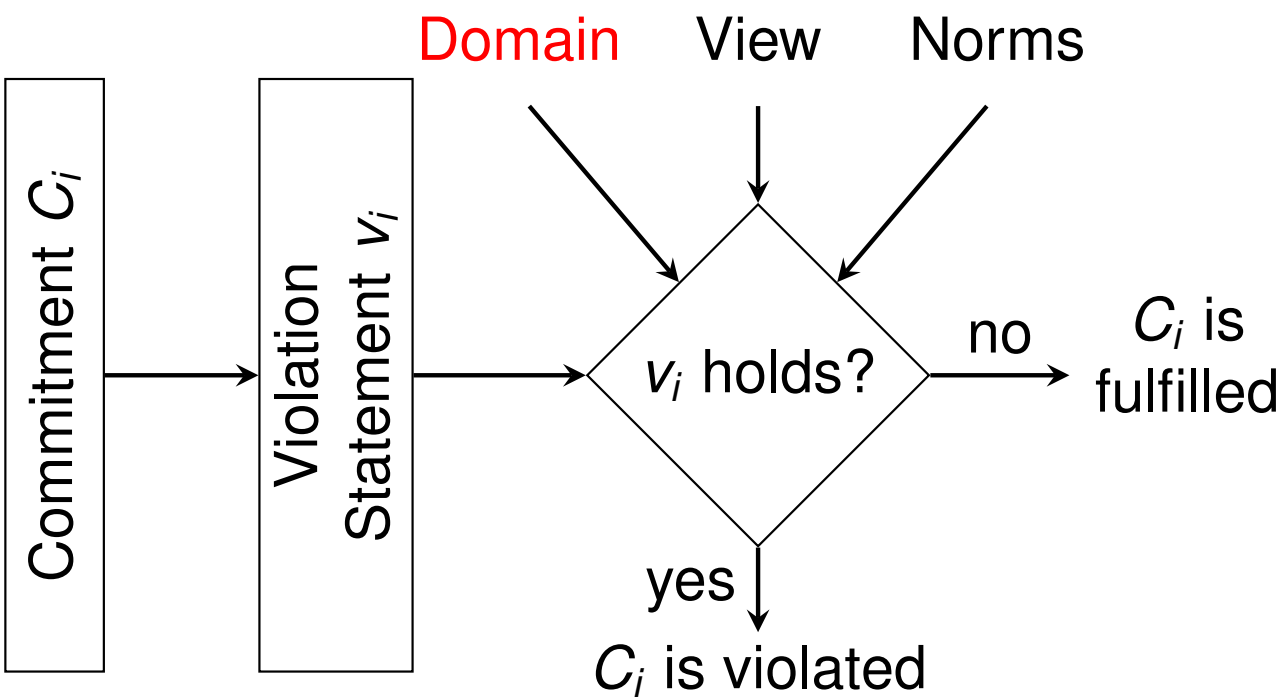
- A violation occurs when the *debtor* fails to bring about the *condition* of a commitment.
- We identify violation statements according to the commitments.
- In a commitment, the *condition* is true if the *antecedent* is true that can be represented as the rule: *precondition* $\rightarrow$ *condition*.
- A violation statement is modeled as the negation of this rule:
  violation: *precondition*,not(*condition*)

---

$C_1$(:osn; :dennis; *isFriendOf*(:dennis,*X*), *sharesPost*(:dennis,*P*), MediumPost(P); *canSeePost*(*X*,*P*))
$v_1$: *isFriendOf*(:dennis,*X*), *sharesPost*(:dennis,*P*), MediumPost(P), not(*canSeePost*(*X*,*P*))

---

# The Social Network Domain



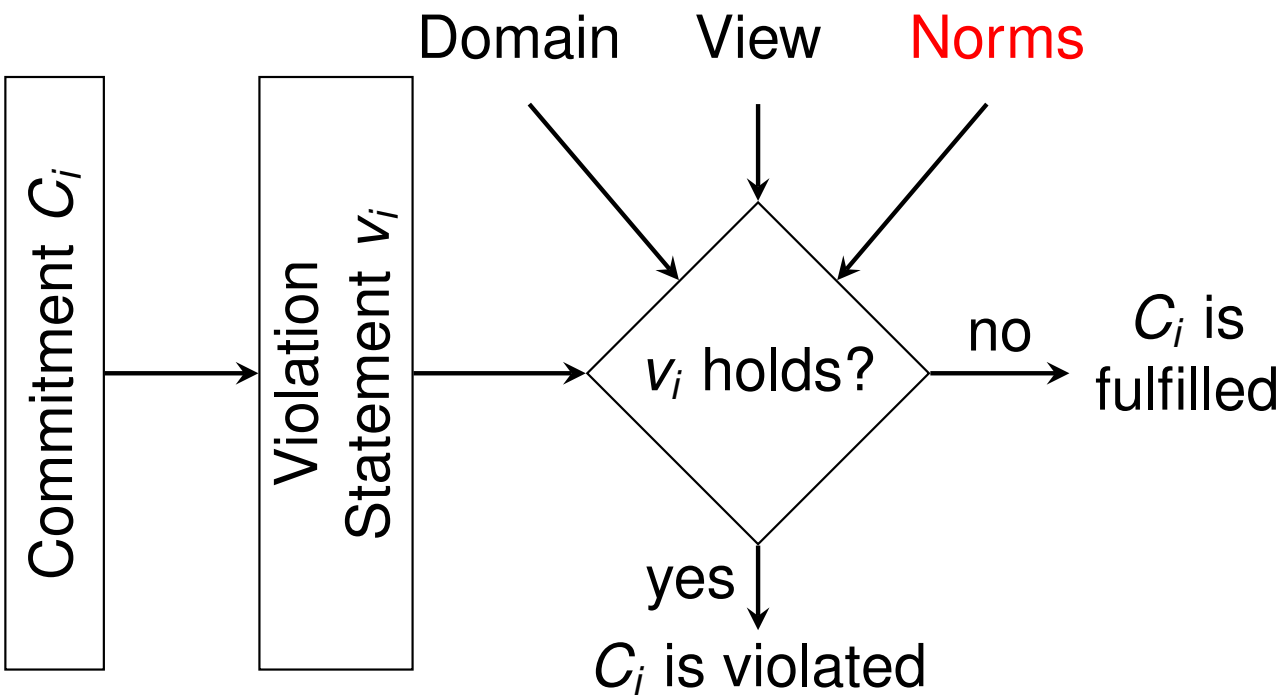
# The Social Network Domain: Axioms

| | |
|---|---|
| $\texttt{Agent}, \texttt{Post}, \texttt{Audience}, \texttt{Context}, \texttt{Content} \sqsubseteq \top$ | $\texttt{Leisure}, \texttt{Meeting}, \texttt{Work} \sqsubseteq \texttt{Context}$ |
| $\texttt{Beach}, \texttt{EatAndDrink}, \texttt{Party}, \texttt{Sightseeing} \sqsubseteq \texttt{Leisure}$ | $\texttt{Bar}, \texttt{Cafe}, \texttt{College}, \texttt{Museum}, \texttt{University} \sqsubseteq \texttt{Location}$ |
| $\texttt{Picture}, \texttt{Video} \sqsubseteq \texttt{Medium}$ | $\texttt{Medium}, \texttt{Text}, \texttt{Location} \sqsubseteq \texttt{Content}$ |
| $\texttt{Post} \sqcap \exists \mathit{sharesPost}^{-}.\texttt{Agent} \equiv \exists \mathit{R\_sharedPost}.\mathit{Self}$ | $\texttt{LocationPost} \equiv \exists \mathit{R\_locationPost}.\mathit{Self}$ |
| $\texttt{LocationPost} \equiv \texttt{Post} \sqcap \exists \mathit{hasLocation}.\texttt{Location}$ | $\texttt{MediumPost} \equiv \texttt{Post} \sqcap \exists \mathit{hasMedium}.\texttt{Medium}$ |
| $\texttt{TaggedPost} \equiv \texttt{Post} \sqcap \exists \mathit{isAbout}.\texttt{Agent}$ | $\texttt{TextPost} \equiv \texttt{Post} \sqcap \exists \mathit{hasText}.\texttt{Text}$ |

# The Social Network Domain: Axioms

| Role Inclusions | Role Restrictions |
|---|---|
| $\mathit{canSeePost} \sqsubseteq U_a$ | $\exists \mathit{canSeePost}.\top \sqsubseteq \texttt{Agent}, \top \sqsubseteq \forall \mathit{canSeePost}.\texttt{Post}$ |
| $\mathit{hasAudience} \sqsubseteq U_a$ | $\exists \mathit{hasAudience}.\top \sqsubseteq \texttt{Post}, \top \sqsubseteq \forall \mathit{hasAudience}.\texttt{Audience}, \top \sqsubseteq \leq 1\, \mathit{hasAudience}.\top$ |
| $\mathit{hasGeotag} \sqsubseteq U_a$ | $\exists \mathit{hasGeotag}.\top \sqsubseteq \texttt{Medium}, \top \sqsubseteq \forall \mathit{hasGeotag}.\texttt{Location}, \top \sqsubseteq \leq 1\, \mathit{hasGeotag}.\top$ |
| $\mathit{hasLocation} \sqsubseteq U_a$ | $\exists \mathit{hasLocation}.\top \sqsubseteq \texttt{Post}, \top \sqsubseteq \forall \mathit{hasLocation}.\texttt{Location}, \top \sqsubseteq \leq 1\, \mathit{hasLocation}.\top$ |
| $\mathit{hasMedium} \sqsubseteq U_a$ | $\exists \mathit{hasMedium}.\top \sqsubseteq \texttt{Post}, \top \sqsubseteq \forall \mathit{hasMedium}.\texttt{Medium}$ |
| $\mathit{hasMember} \sqsubseteq U_a$ | $\exists \mathit{hasMember}.\top \sqsubseteq \texttt{Audience}, \top \sqsubseteq \forall \mathit{hasMember}.\texttt{Agent}$ |
| $\mathit{isAbout} \sqsubseteq U_a$ | $\exists \mathit{isAbout}.\top \sqsubseteq \texttt{Post}, \top \sqsubseteq \forall \mathit{isAbout}.\texttt{Agent}$ |
| $\mathit{isConnectedTo} \sqsubseteq U_a$ | $\exists \mathit{isConnectedTo}.\top \sqsubseteq \texttt{Agent}, \top \sqsubseteq \forall \mathit{isConnectedTo}.\texttt{Agent}, \mathit{isConnectedTo} \equiv \mathit{isConnectedTo}^-$ |
| $\mathit{isFriendOf} \sqsubseteq \mathit{isConnectedTo}$ | $\exists \mathit{isFriendOf}.\top \sqsubseteq \texttt{Agent}, \top \sqsubseteq \forall \mathit{isFriendOf}.\texttt{Agent}, \mathit{isFriendOf} \equiv \mathit{isFriendOf}^-$ |
| $\mathit{taggedPerson} \sqsubseteq U_a$ | $\exists \mathit{taggedPerson}.\top \sqsubseteq \texttt{Medium}, \top \sqsubseteq \forall \mathit{taggedPerson}.\texttt{Agent}$ |

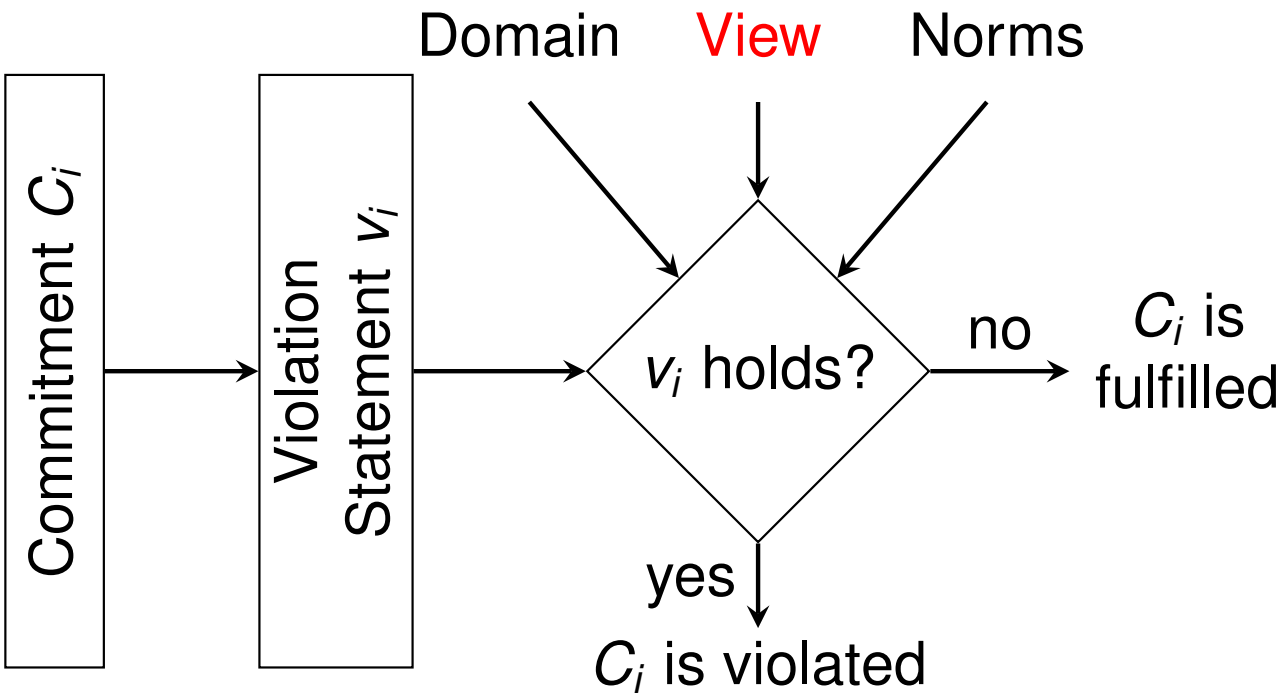
# Norms


Domain
View
Norms
Commitment $C_i$
Violation Statement $v_i$
$v_i$ holds?
no
$C_i$ is fulfilled
yes
$C_i$ is violated

# Norms

| | |
|---|---|
| $N_1$: | $sharesPost(X,P) \rightarrow canSeePost(X,P)$ |
| [Agent can see the posts that it shares.] | |
| $N_2$: $sharesPost(X,P) \wedge hasAudience(P,A) \wedge hasMember(A,M)$ | $\rightarrow canSeePost(M,P)$ |
| [Audience of a post can see the post.] | |
| $N_3$: | $hasMedium(P,M) \wedge taggedPerson(M,X) \rightarrow isAbout(P,X)$ |
| [Post is about agents tagged in a medium.] | |
| $N_4$: | $\texttt{Post}(P) \wedge hasMedium(P,M) \wedge hasGeotag(M,T) \rightarrow \texttt{LocationPost}(P)$ |
| [Geotagged medium gives away the location.] | |

# View


Domain
View
Norms
Commitment $C_i$
Violation Statement $v_i$
$v_i$ holds?
no
$C_i$ is fulfilled
yes
$C_i$ is violated

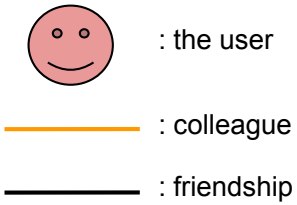
# View

- ABSN view captures a given state of the network.

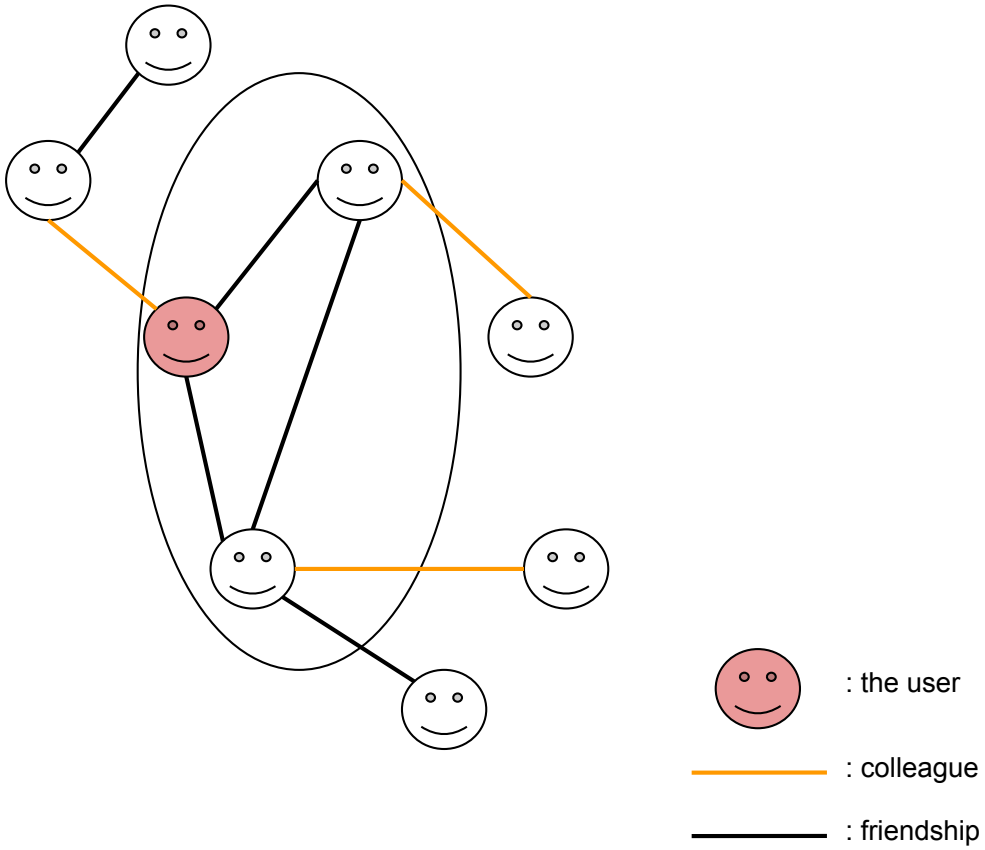
Table: Charlie shares a post `:pc1`

| | |
|---|---|
| ClassAssertion(`Agent :alice`) | ClassAssertion(`Agent :bob`) |
| ClassAssertion(`Agent :charlie`) | ClassAssertion(`Agent :dennis`) |
| ClassAssertion(`Agent :eve`) | ClassAssertion(`Audience :audience`) |
| ClassAssertion(`Post :pc1`) | ClassAssertion(`Picture :pictureConcert`) |
| ObjectPropertyAssertion(*isFriendOf* `:alice :bob`) | ObjectPropertyAssertion(*isFriendOf* `:alice :charlie`) |
| ObjectPropertyAssertion(*isFriendOf* `:bob :charlie`) | ObjectPropertyAssertion(*isFriendOf* `:charlie :dennis`) |
| ObjectPropertyAssertion(*isFriendOf* `:dennis :eve`) | |
| ObjectPropertyAssertion(*sharesPost* `:charlie :pc1`) | ObjectPropertyAssertion(*hasAudience* `:pc1 :audience`) |
| ObjectPropertyAssertion(*hasMedium* `:pc1 :pictureConcert`) | ObjectPropertyAssertion(*taggedPerson* `:pictureConcert :alice`) |
| ObjectPropertyAssertion(*hasMember* `:audience :alice`) | ObjectPropertyAssertion(*hasMember* `:audience :dennis`) |
| ObjectPropertyAssertion(*hasMember* `:audience :eve`) | ObjectPropertyAssertion(*hasMember* `:audience :bob`) |

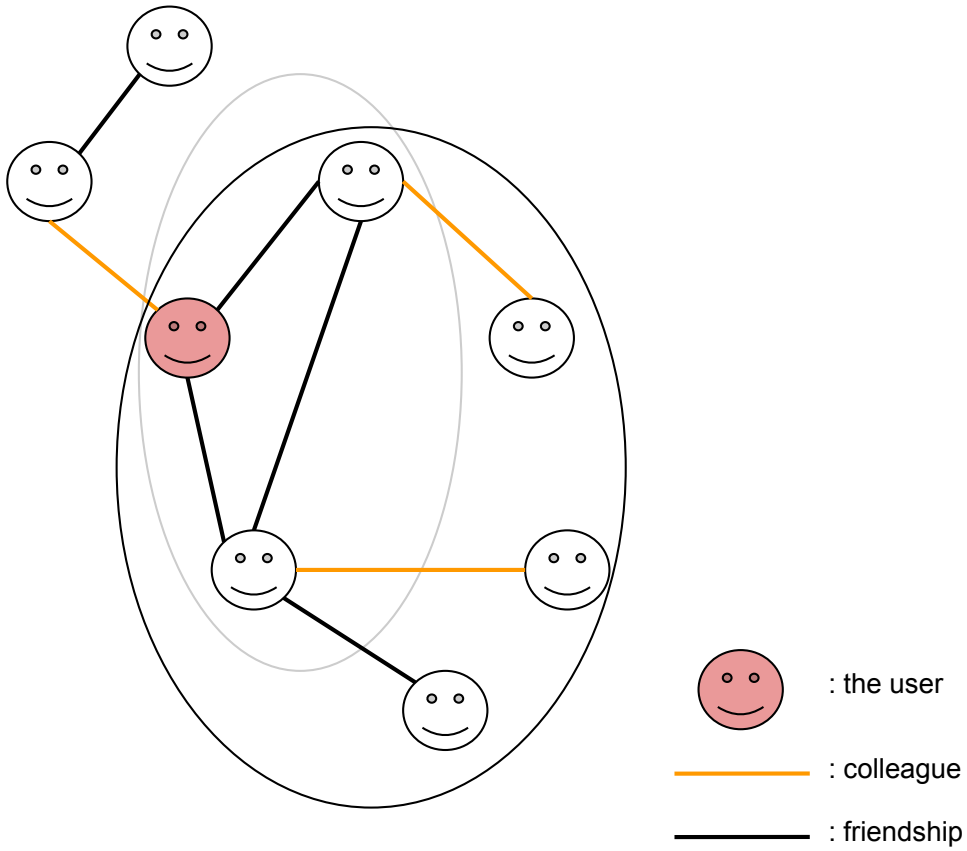
# Views



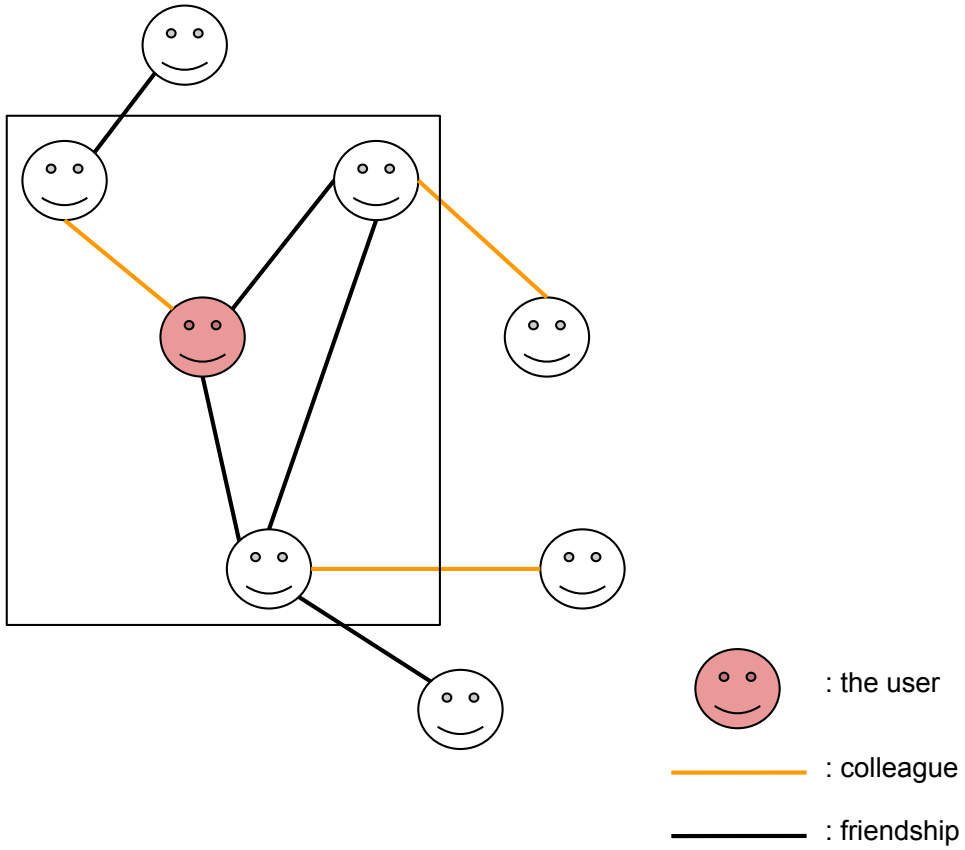
# Views



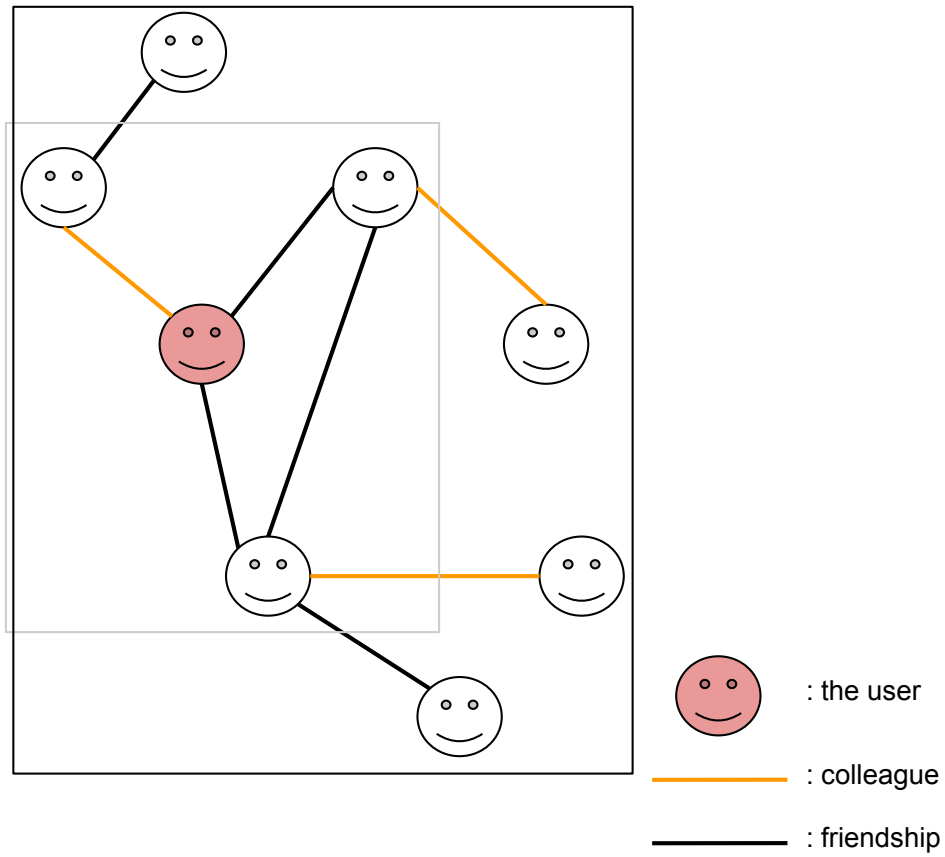
# Views



# Views



# Views




Used to extend the current view. At the final extension, we have the *global view*.

# Detection Algorithm

**Algorithm 1:** DEPTHLIMITEDDETECTION ($C$, $m$=$MAX$)

**Input**: $C$, the commitment to be checked
**Input**: $m$, the maximum number of iterations
**Output**: $V$, the set of privacy violations
**Data**: $KB$, the knowledge base (domain + norms)

1 $S \leftarrow \mathsf{initView}(C.creditor)$;
2 $V \leftarrow \{\}$, $iterno \leftarrow 0$;
3 $vstatement \leftarrow C.antecedent, not(C.consequent)$;
4 **while** $iterno < m$ **do**
5 $\quad KB \leftarrow \mathsf{updateKB}(KB, S)$;
6 $\quad V \leftarrow V \cup \mathsf{checkViolations}(KB, vstatement)$;
7 $\quad iterno \leftarrow iterno + 1$;
8 $\quad$ **if** $V = \{\}$ **then**
9 $\quad\quad S \leftarrow \mathsf{extendView}(S)$;
10 $\quad$ **else**
11 $\quad\quad$ **return** $V$;
12 **return** $V$;

# Theoretical Results

## Theorem (Soundness)

*Given an ABSN that is correctly represented with a KB, and a commitment C that represents a privacy requirement* $PR^t_{a,i}$*, if* DEPTHLIMITEDDETECTION *returns a violation, then* isViolated*(*$PR^t_{a,i}$*,ABSN) holds.*

# Theoretical Results

## Theorem (Soundness)

*Given an ABSN that is correctly represented with a KB, and a commitment C that represents a privacy requirement* $PR^t_{a,i}$*, if* DEPTHLIMITEDDETECTION *returns a violation, then* isViolated*(*$PR^t_{a,i}$*,ABSN) holds.*

*Proof*: Assume that DEPTHLIMITEDDETECTION detects a violation, which is not true. This may occur only if one of the following holds:

- S contains incorrect information.

# Theoretical Results

## Theorem (Soundness)

*Given an ABSN that is correctly represented with a KB, and a commitment C that represents a privacy requirement* $PR^t_{a,i}$*, if* DEPTHLIMITEDDETECTION *returns a violation, then* isViolated*(*$PR^t_{a,i}$*,ABSN) holds.*

*Proof*: Assume that DEPTHLIMITEDDETECTION detects a violation, which is not true. This may occur only if one of the following holds:

- S contains incorrect information.
- KB does not contain the necessary information.

# Theoretical Results

## Theorem (Soundness)

*Given an ABSN that is correctly represented with a KB, and a commitment C that represents a privacy requirement* $PR^t_{a,i}$*, if* DEPTHLIMITEDDETECTION *returns a violation, then* isViolated$(PR^t_{a,i}$*,ABSN) holds.*

*Proof*: Assume that DEPTHLIMITEDDETECTION detects a violation, which is not true. This may occur only if one of the following holds:

- S contains incorrect information.
- KB does not contain the necessary information.
- *vstatement* is computed incorrectly so that it does not reflect a privacy violation.

# Completeness

**Theorem (Completeness)**

*Given a commitment C,* DEPTHLIMITEDDETECTION *always returns a privacy violation, if one exists.*

# Completeness

## Theorem (Completeness)

*Given a commitment C,* DEPTHLIMITEDDETECTION *always returns a privacy violation, if one exists.*

## Lemma

*Given a violation statement of a commitment* $v_i$ *and a knowledge base KB, if there is a privacy violation in KB,* checkViolations *returns it.*

# Completeness

## Theorem (Completeness)

*Given a commitment C,* DEPTHLIMITEDDETECTION *always returns a privacy violation, if one exists.*

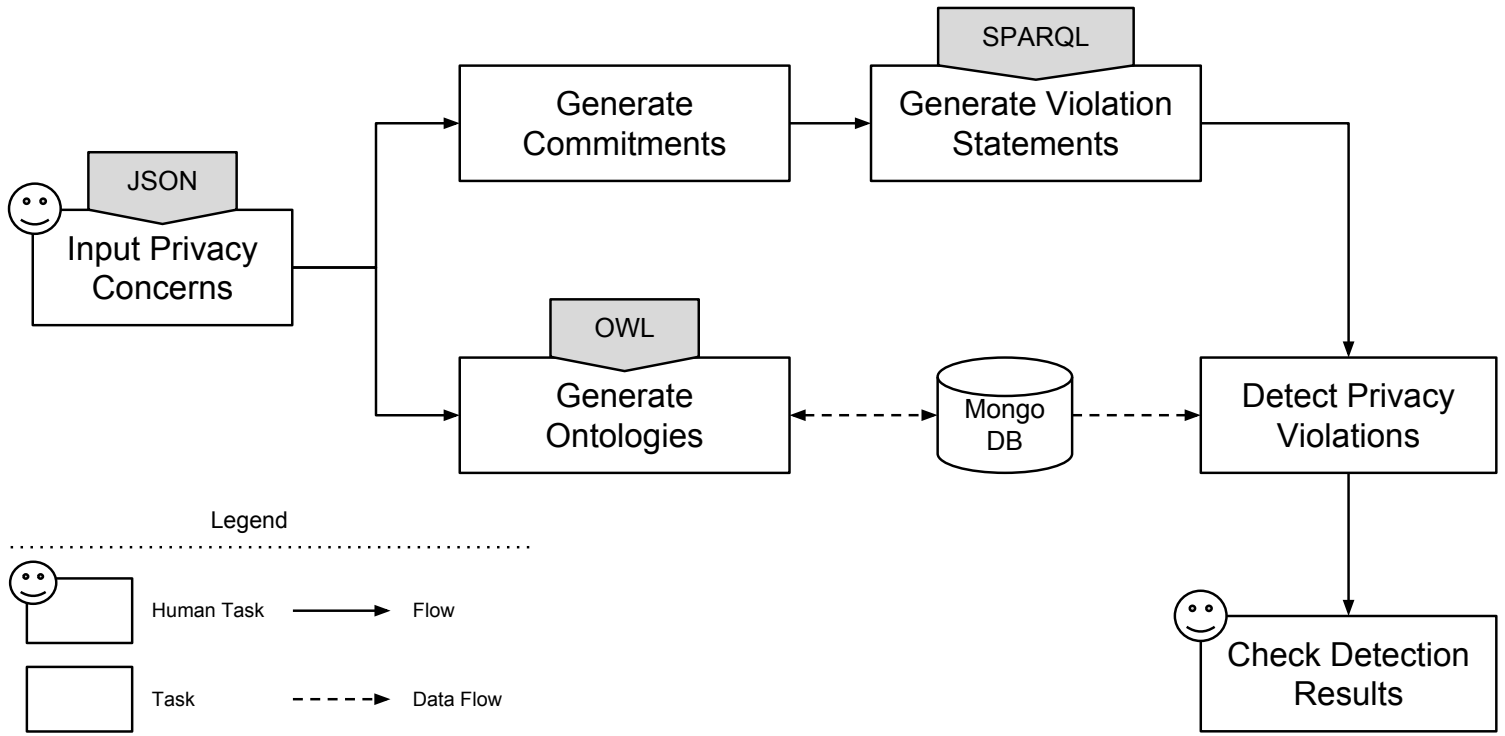
## Lemma

*Given a violation statement of a commitment* $v_i$ *and a knowledge base KB, if there is a privacy violation in KB,* checkViolations *returns it.*

## Lemma

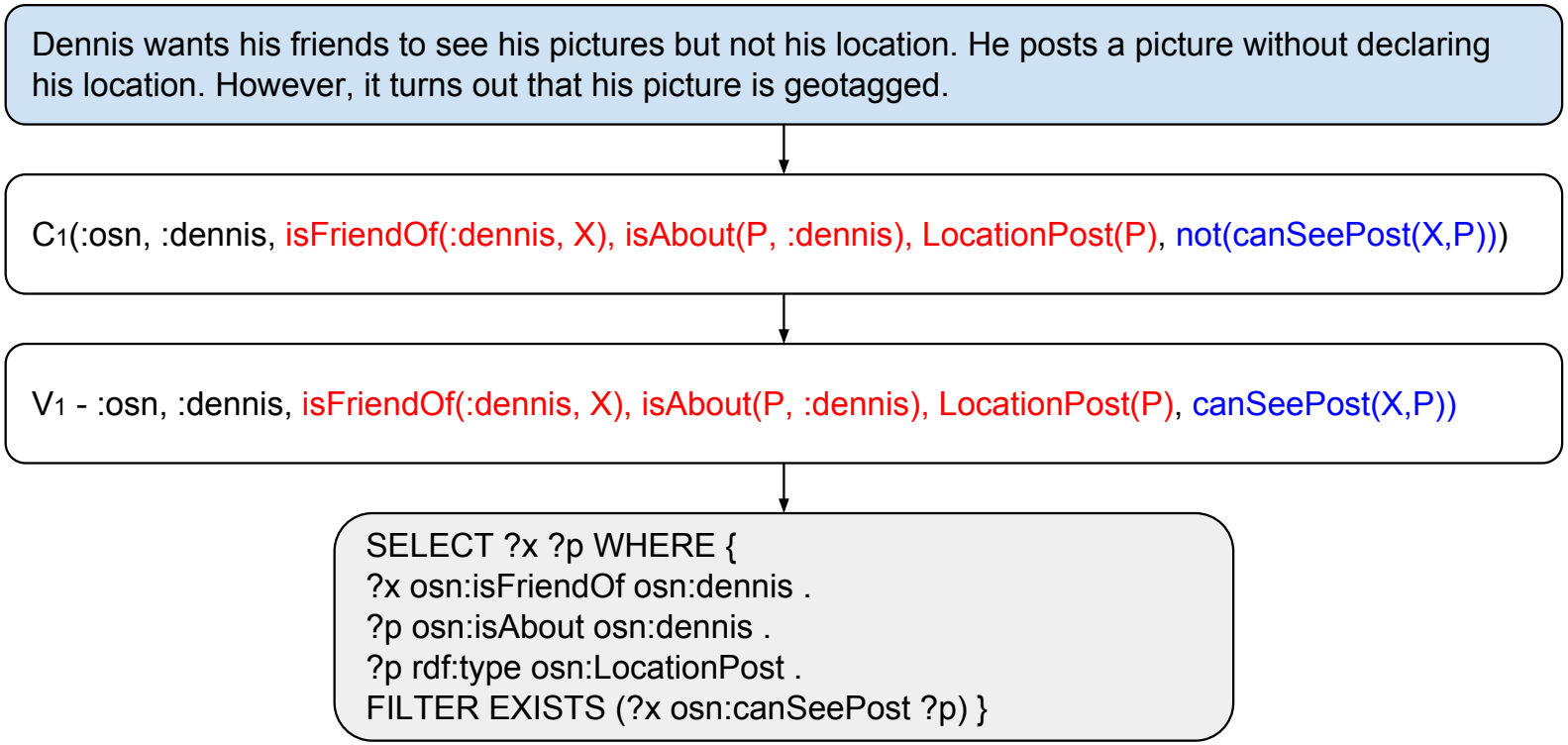
extendView *can eventually create the global view.*

# A Facebook Application: PriGuardTool[2]




[2] Nadin Kökciyan and Pınar Yolum. "PriGuardTool: A Web-Based Tool to Detect Privacy Violations Semantically". In: *Engineering Multi-Agent Systems: 4th International Workshop, EMAS 2016, Singapore, Singapore, May 9-10, 2016, Revised, Selected, and Invited Papers*. Ed. by Matteo Baldoni et al. Springer International Publishing, 2016, pp. 81–98.

# Running Example



# PRIGUARD: Performance Results

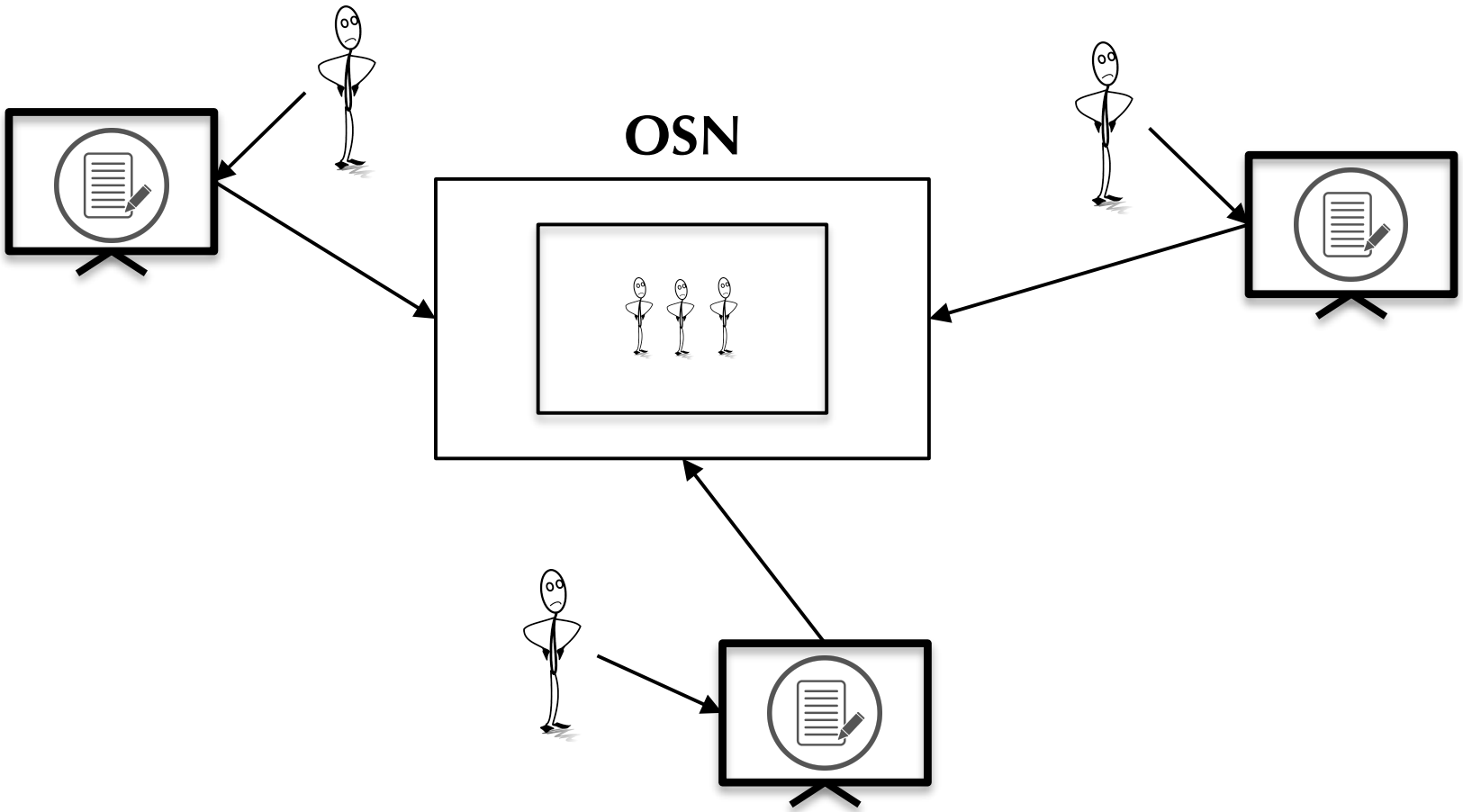
| ABSN | depth=0 | depth=1 | depth=2 | $G$ |
|---|---|---|---|---|
| (#$\mathcal{A}$,#$\mathcal{R}$) | (1,0) | (39,412) | (535,5347) | (535,5347) |
| $G_1$: #Axioms | 2175 | 4267 | 29959 | 29959 |
| Time | 3ms | 4.74ms | 30.19ms | 29.79ms |
| (#$\mathcal{A}$,#$\mathcal{R}$) | (1,0) | (51,579) | (1035,27783) | (1035,27783) |
| $G_2$: #Axioms | 2175 | 5079 | 125703 | 125703 |
| Time | 2.96ms | 5.49ms | 123.95ms | 122.46ms |
| (#$\mathcal{A}$,#$\mathcal{R}$) | (1,0) | (123,4199) | (1046,27795) | (4039,88234) |
| $G_3$: #Axioms | 2175 | 20423 | 125883 | 403555 |
| Time | 3.09ms | 18.01ms | 121.15ms | 530.01ms |
| (#$\mathcal{A}$,#$\mathcal{R}$) | (1,0) | (37,235) | (848,8543) | (60001,728596) |
| $G_4$: #Axioms | 2175 | 3535 | 46463 | 3636547 |
| Time | 3.07ms | 4.13ms | 47.09ms | 18397.26ms |
| (#$\mathcal{A}$,#$\mathcal{R}$) | (1,0) | (157,2669) | (2787,74217) | (65328,1435168) |
| $G_5$: #Axioms | 2175 | 14711 | 332463 | 6526759 |
| Time | 3.11ms | 19.03ms | 406.91ms | 25890.27ms |

# PROTOSS[3]

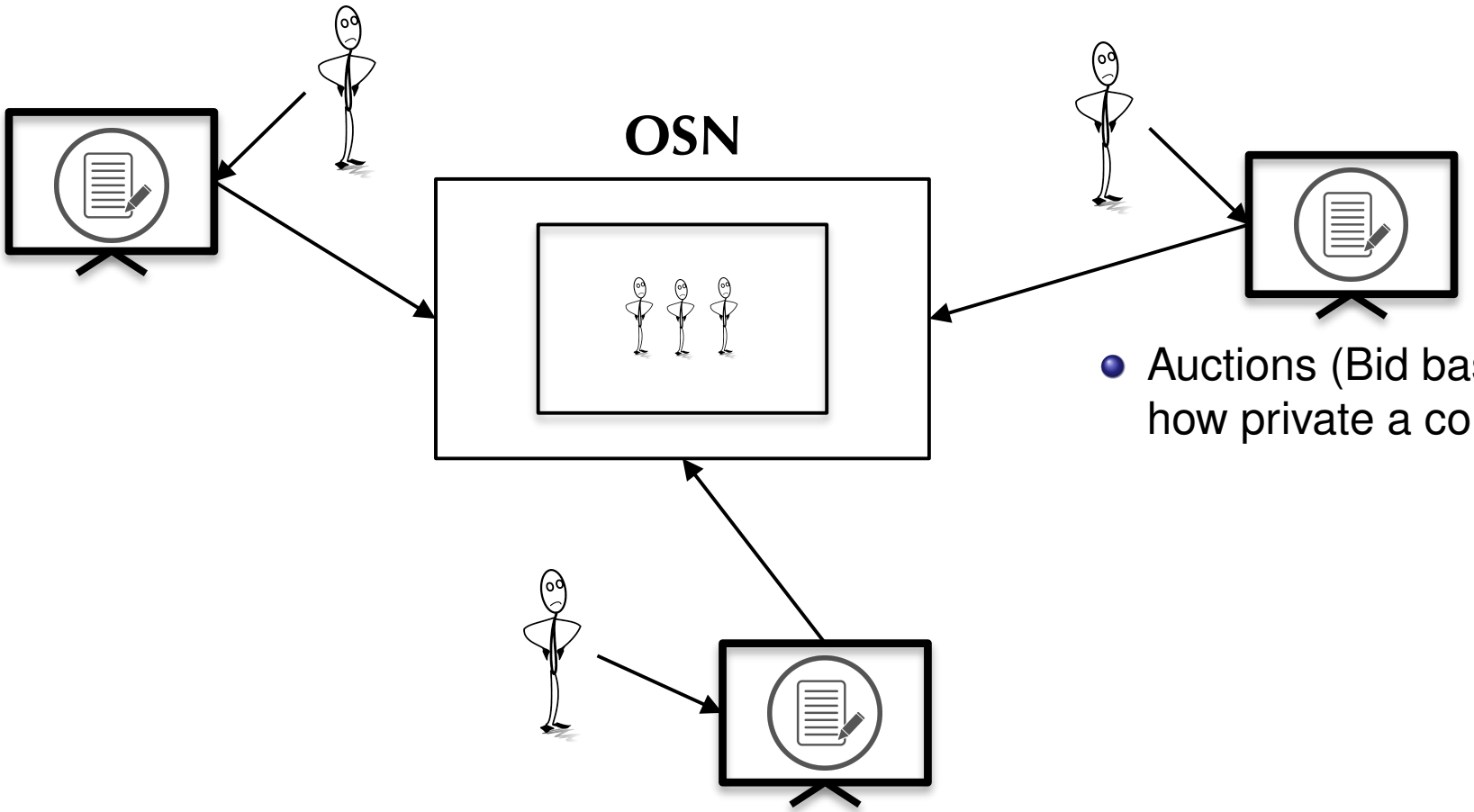
- Rather than checking a single state check the model
- Automatic verification of properties based on the model of a system
- The system is represented as a state transition graph
- The property is represented as a logic formula
- Model checker verifies whether the property holds for the system
- Example:
  - System: the social network (with entire details of relations, agreements, and so on)
  - Property: location of user *X* is accessible by user *Y*
  - Model checking will say whether this property holds

[3] Özgür Kafalı, Akın Günay, and Pınar Yolum. "Detecting and predicting privacy violations in online social networks". In: *Distributed and Parallel Databases* 32.1 (2014), pp. 161–190.

# How to agree on the privacy of co-owned content?

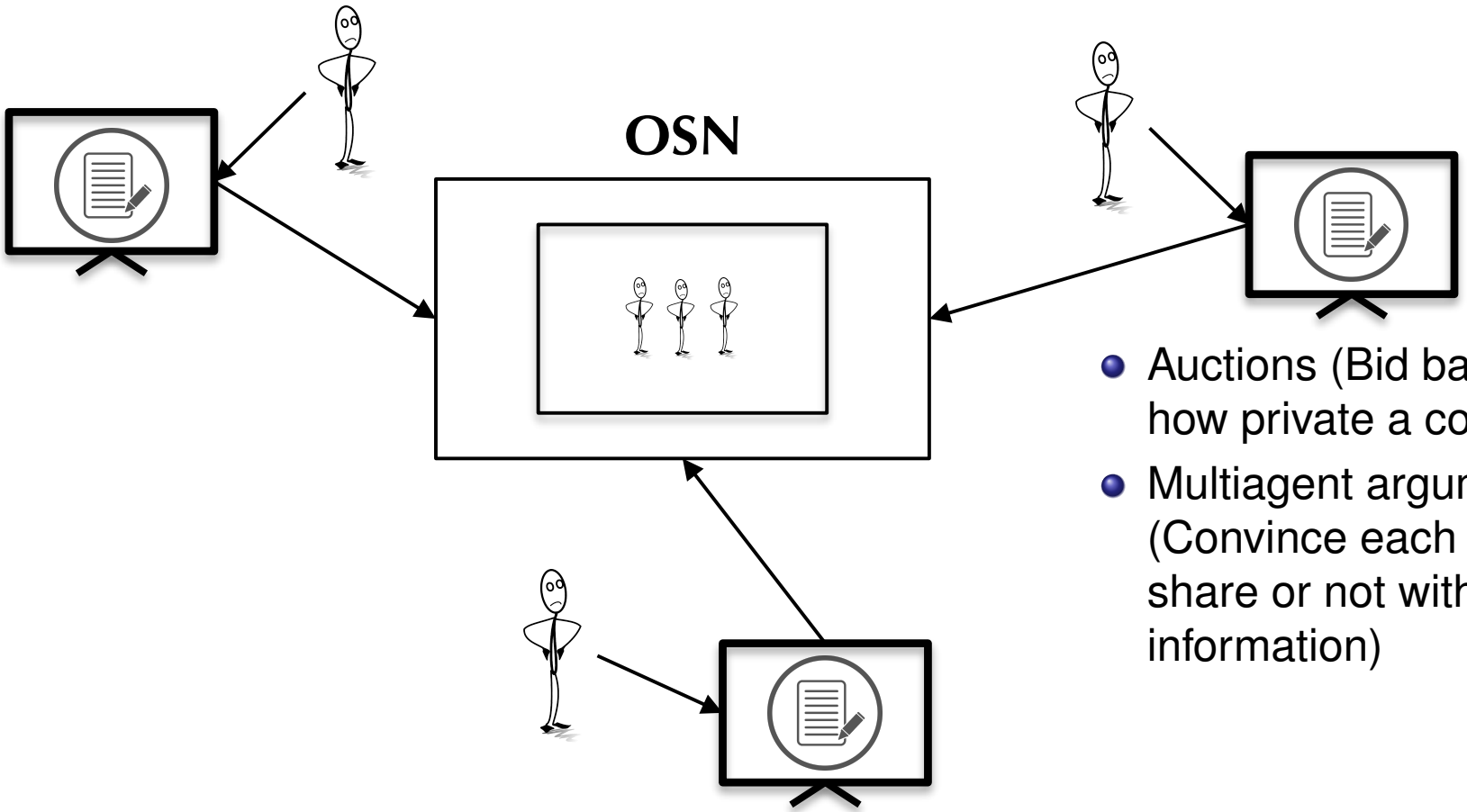


# How to agree on the privacy of co-owned content?



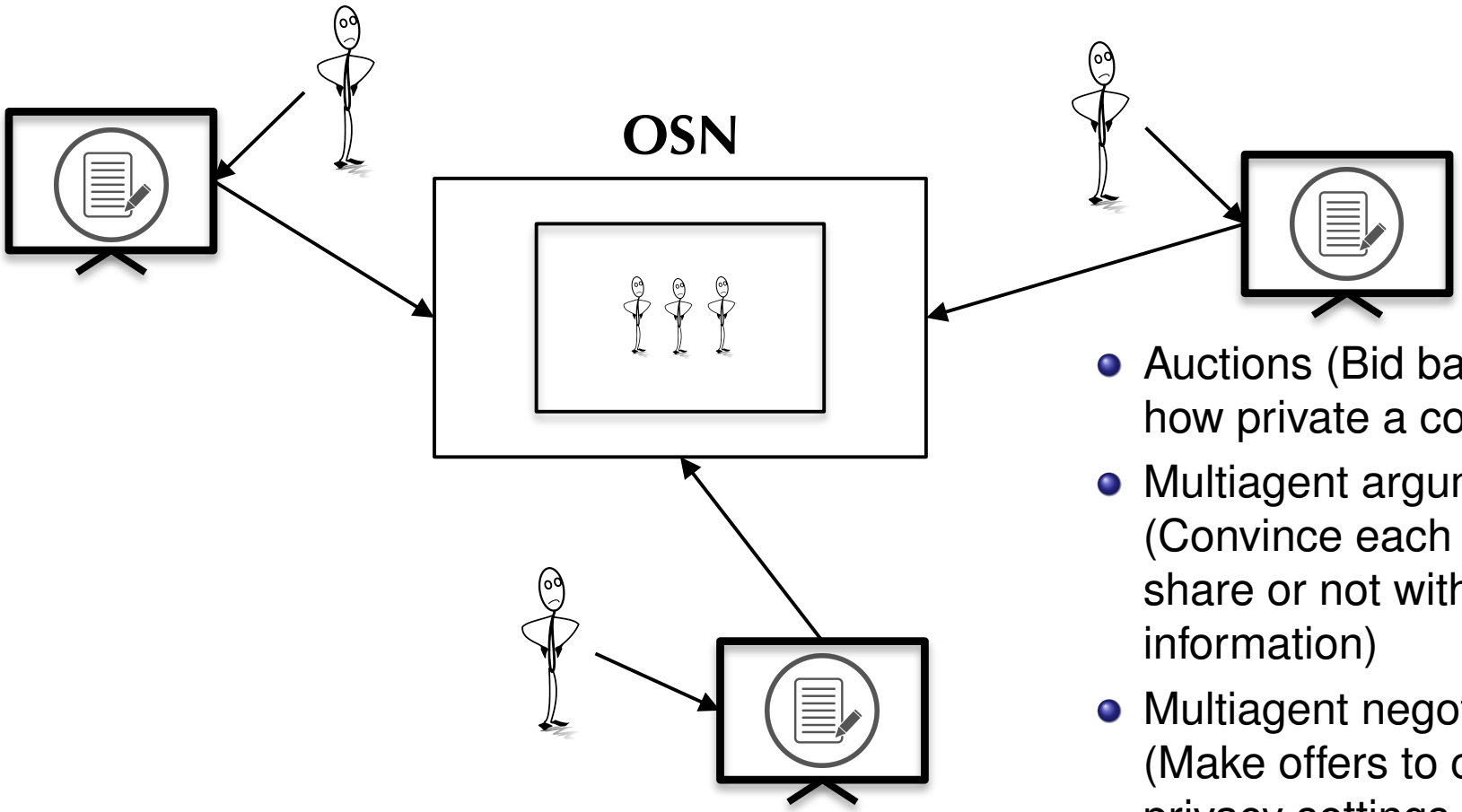

- Auctions (Bid based on how private a content is)

# How to agree on the privacy of co-owned content?




- Auctions (Bid based on how private a content is)
- Multiagent argumentation (Convince each other to share or not with information)

# How to agree on the privacy of co-owned content?




- Auctions (Bid based on how private a content is)
- Multiagent argumentation (Convince each other to share or not with information)
- Multiagent negotiation (Make offers to change privacy settings to meet in between)

# A Privacy Decision Mechanism

- Should consider the privacy of relevant users.

# A Privacy Decision Mechanism

- Should consider the privacy of relevant users.
- Should enable relevant users express opinions on a post *before* it is revealed.

# A Privacy Decision Mechanism

- Should consider the privacy of relevant users.
- Should enable relevant users express opinions on a post *before* it is revealed.
- Should be automatic.

# A Privacy Decision Mechanism

- Should consider the privacy of relevant users.
- Should enable relevant users express opinions on a post *before* it is revealed.
- Should be automatic.
- Should enable customized privacy constraints.

# A Privacy Decision Mechanism

- Should consider the privacy of relevant users.
- Should enable relevant users express opinions on a post *before* it is revealed.
- Should be automatic.
- Should enable customized privacy constraints.
- Should protect against violations that occur with inference.

# PANO: Privacy Auctioning[4]

- Clarke-Tax mechanism provides an auction mechanism where participants bid for different possible actions in the environment.
- Participants whose bids are decisive on the final action are taxed according to the value they put on it.
- Extentions:
  - Group-wise spending: Earned currencies can only be used in new contents with same co-owners to overcome abuse.
  - Boundaries: Limitations to minimum and maximum bids in order to prevent richer users dominating the decisions.

[4] Onuralp Ulusoy and Pınar Yolum. "PANO: Privacy Auctioning for Online Social Networks". In: *AAMAS*. 2018, pp. 2103–2105.

# PANO Example

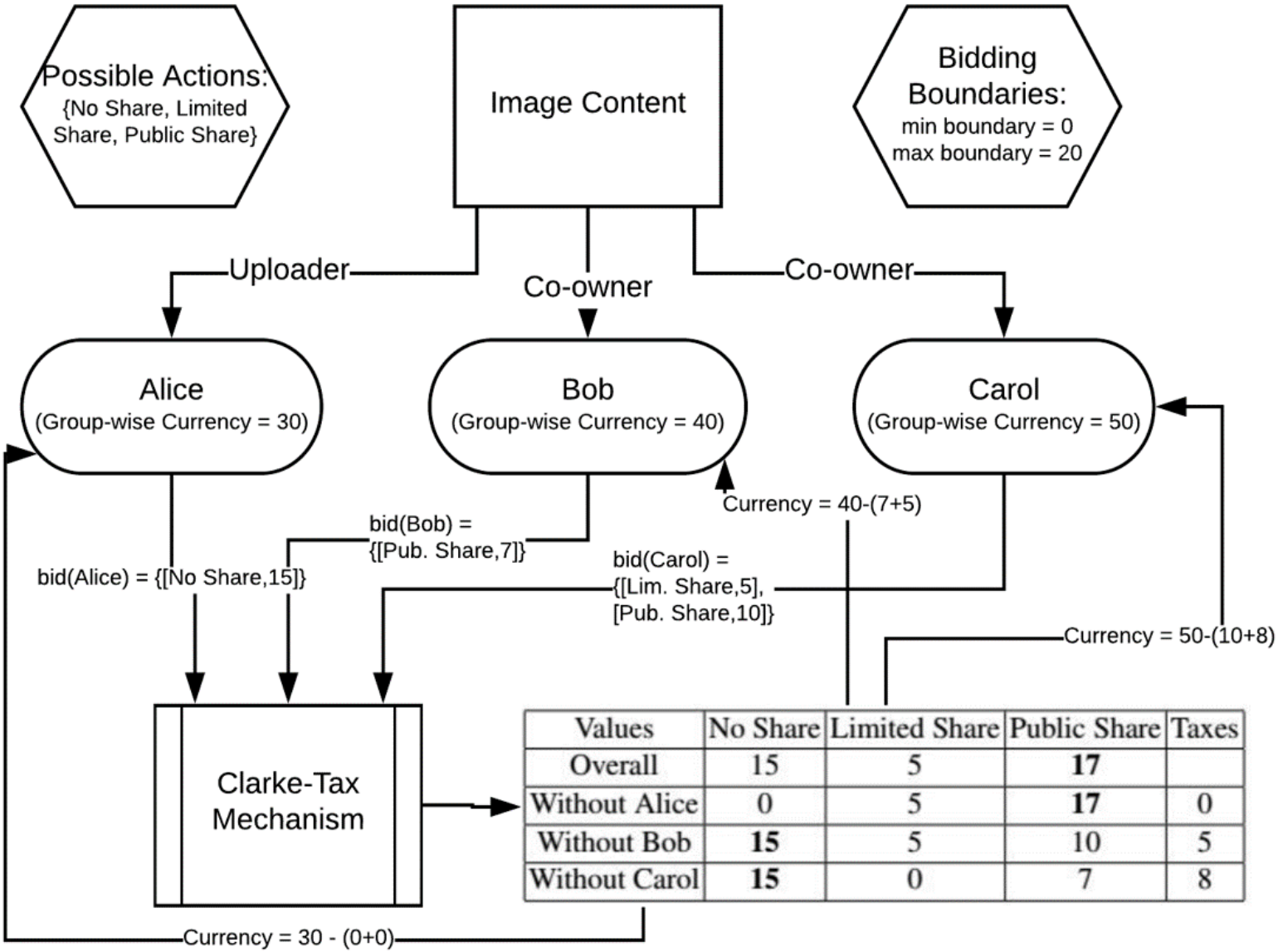
**Table 1: Four User Bids for Sharing an Image**

| Users | No Share | Limited Share | Public Share |
|---|---|---|---|
| Alice | 3 | 5 | 0 |
| Bob | 15 | 2 | 0 |
| Carol | 5 | 8 | 5 |
| Dave | 2 | 6 | 18 |

**Table 2: Clarke-Tax Mechanism Example - Decision and Taxes**

| Values | No | Limited | Public | Taxes |
|---|---|---|---|---|
| Overall | **25** | 21 | 23 | |
| Without Alice | 22 | 16 | **23** | 1 |
| Without Bob | 10 | 19 | **23** | 13 |
| Without Carol | **20** | 13 | 18 | 0 |
| Without Dave | **23** | 15 | 5 | 0 |

# How does PANO work?



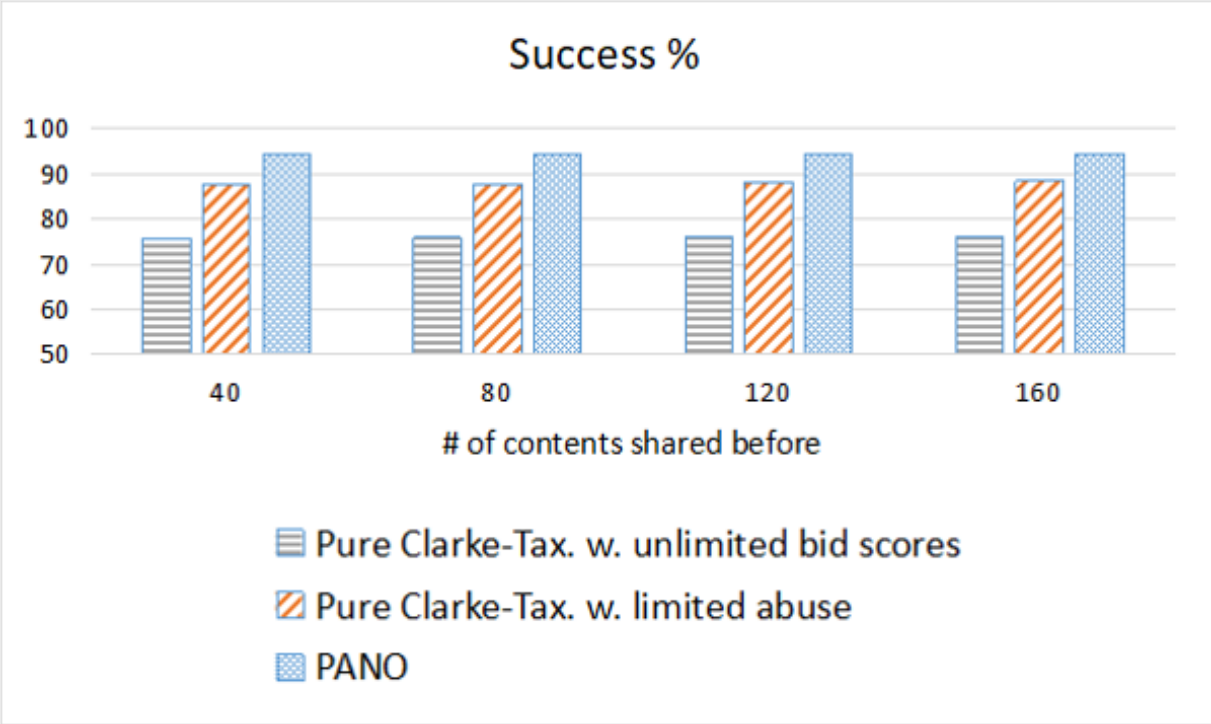

| Values | No Share | Limited Share | Public Share | Taxes |
|---|---|---|---|---|
| Overall | 15 | 5 | **17** | |
| Without Alice | 0 | 5 | **17** | 0 |
| Without Bob | **15** | 5 | 10 | 5 |
| Without Carol | **15** | 0 | 7 | 8 |

# Challenges

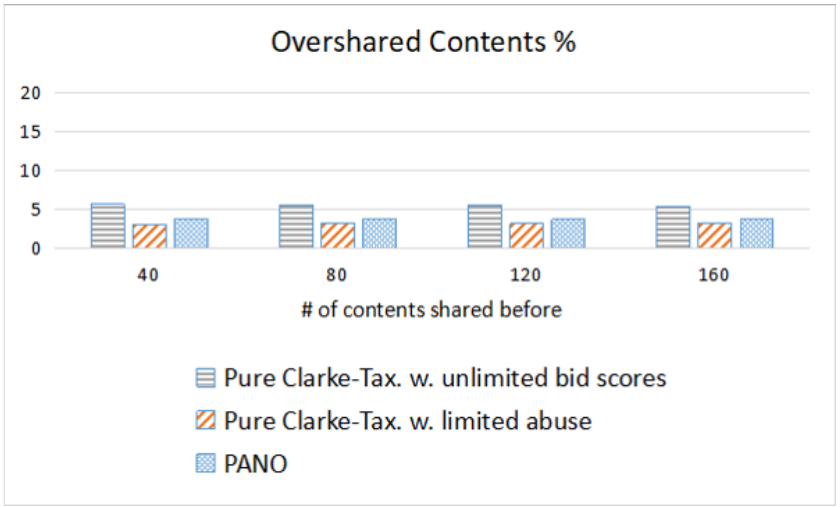
- How to generate a bid given the privacy requirements of the user and importance of a content?
- How to model others to see what bids they give?
- Is it necessary to go into an auction every single time?

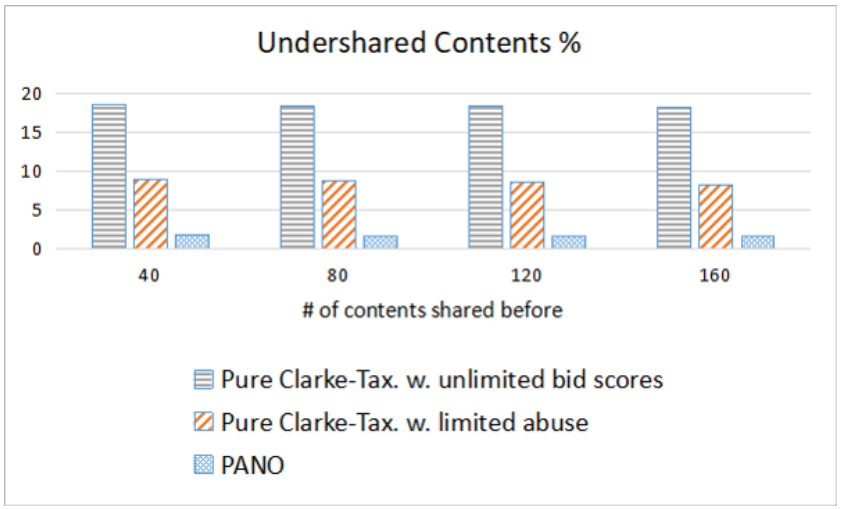
# How Well is the Privacy Preserved?

Success is defined as the percentage of users who view the content as specified in the privacy policies of the agents



# Evaluation - Oversharing & Undersharing



# Convince Others to Keep Private[5]

- Protecting privacy collaboratively
- Users discuss on a post *before* it is shared.
- Discussion is conducted automatically.
  - Each agent is equipped with an ontology and the semantic rules.
  - Agents discuss on a post by providing each other with arguments using a distributed algorithm.
  - At the end of the discussion, we find the justified arguments.

[5] Nadin Kökciyan, Nefise Yaglikci, and Pınar Yolum. "An Argumentation Approach for Resolving Privacy Disputes in Online Social Networks". In: *ACM Transactions on Internet Technology* (2017).

# Assumption-Based Argumentation (ABA)

- ABA framework is a four-tuple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{C} \rangle$ (Dung *et al.*, 2009).
- Each rule in $\mathcal{R}$ consists of a body $\sigma_1,...,\sigma_m$ and a head $\sigma_0$ where $\sigma_1,...,\sigma_m \rightarrow \sigma_0$ ($m \geq 0$, $\sigma_i \in \mathcal{L}$).
  - Facts are rules with an empty body (e.g., $\{\rightarrow$ *includesObject*(`:medium`,`:wig`)$\}$ ).
- Assumption set $\mathcal{A}$ includes the weak points of arguments.
- Contrary mapping $\mathcal{C}$ includes the contraries of the assumptions.

# How to manage the privacy of co-owned data?

Alice would like to share a picture taken with Bob. Bob does not like to share party pictures online.

# Derivation of Arguments

An argument has the form $S \vdash^{R} \sigma$ where $S \subseteq \mathcal{A}$, $R \subseteq \mathcal{R}$, $\sigma \in \mathcal{L}$.

# Derivation of Arguments

An argument has the form $S \vdash^{R} \sigma$ where $S \subseteq \mathcal{A}$, $R \subseteq \mathcal{R}$, $\sigma \in \mathcal{L}$.

Table: SWRL Rules

| | |
|---|---|
| An object that can found in a shop is an ordinary object. | |
| $I_{A_1}$: | *foundAt*(?object, ?shop) → *isOrdinary*(?object, true) |
| If a post request has a medium including an unordinary object given at `ChristmasParty`, then it is in `Party` context. | |
| $I_{B_1}$: | *isInContext*(?postRequest, ?context), *hasMedium*(?postRequest, ?medium), *includesObject*(?medium, ?object), `ChristmasParty`(?location), *obtainedFrom*(?object, ?location), *isOrdinary*(?object, false) → `Party`(?context) |
| Bob rejects all the post requests in `Party` context. | |
| $P_{B_1}$: | `Party`(?context), *isInContext*(?postRequest, ?context) → *rejects*(`:bob`, ?postRequest) |

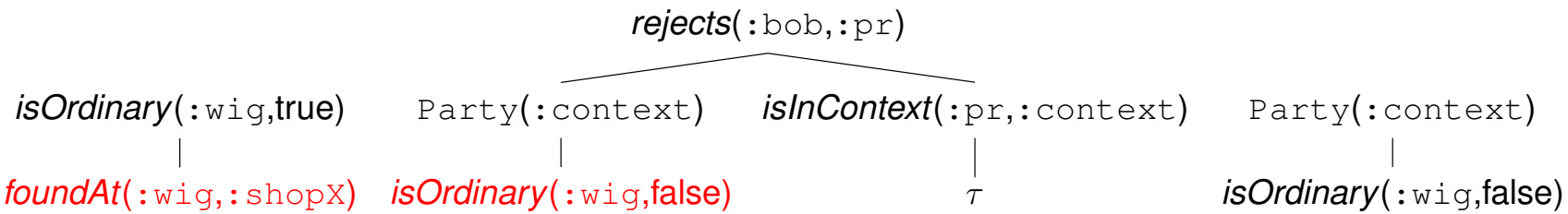


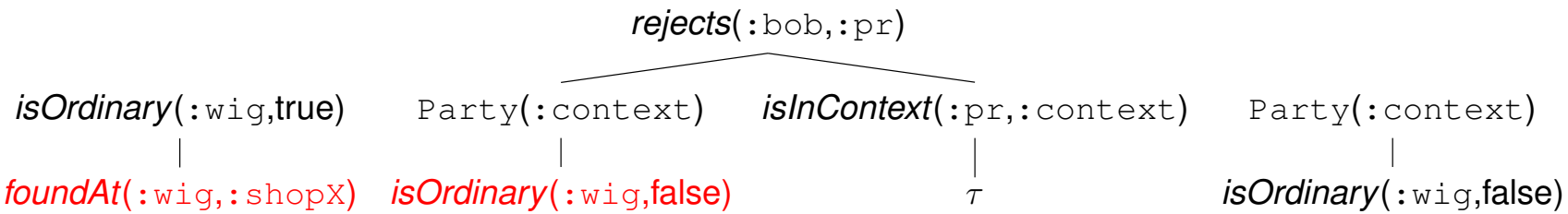

Figure: Deduction Trees

# Derivation of Arguments

An argument has the form $S \vdash^{R} \sigma$ where $S \subseteq \mathcal{A}$, $R \subseteq \mathcal{R}$, $\sigma \in \mathcal{L}$.

## Table: SWRL Rules

| | |
|---|---|
| An object that can found in a shop is an ordinary object. | |
| $I_{A_1}$: | *foundAt*(?object, ?shop) → *isOrdinary*(?object, true) |
| If a post request has a medium including an unordinary object given at `ChristmasParty`, then it is in `Party` context. | |
| $I_{B_1}$: | *isInContext*(?postRequest, ?context), *hasMedium*(?postRequest, ?medium), *includesObject*(?medium, ?object), `ChristmasParty`(?location), *obtainedFrom*(?object, ?location), *isOrdinary*(?object, false) → `Party`(?context) |
| Bob rejects all the post requests in `Party` context. | |
| $P_{B_1}$: | `Party`(?context), *isInContext*(?postRequest, ?context) → *rejects*(`:bob`, ?postRequest) |




## Figure: Deduction Trees

$a_3 : \{\mathit{foundAt}(\texttt{:wig},\texttt{:Gifty})\} \vdash^{I_{A_1}} \mathit{isOrdinary}(\texttt{:wig},\text{true})$

$b_2 : \{\mathit{isOrdinary}(\texttt{:wig},\text{false})\} \vdash^{I_{B_1} \cup_{i=1}^{5} r_i} \texttt{Party}(\texttt{:context})$

$b_3 : \{\mathit{isOrdinary}(\texttt{:wig},\text{false})\} \vdash^{I_{B_1} \cup P_{B_1} \cup_{i=1}^{5} r_i} \mathit{rejects}(\texttt{:bob},\texttt{:pr})$

# Attacks between Arguments

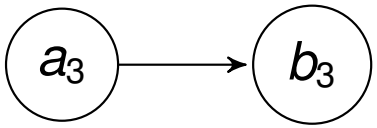
An argument $S_1 \vdash \sigma_1$ can attack another argument $S_2 \vdash \sigma_2$ if and only if $\sigma_1$ is the contrary of one of the assumptions in $S_2$

$$\frac{\begin{array}{l} a_3 : \{\mathit{foundAt}(\texttt{:wig},\texttt{:Gifty})\} \vdash^{I_{A_1}} \mathit{isOrdinary}(\texttt{:wig},\text{true}) \\ b_3 : \{\mathit{isOrdinary}(\texttt{:wig},\text{false})\} \vdash^{I_{B_1} \cup P_{B_1} \cup_{i=1}^{5} r_i} \mathit{rejects}(\texttt{:bob},\texttt{:pr}) \end{array}}{c_3 = (\mathit{isOrdinary}(\texttt{:wig},\text{false}), \mathit{isOrdinary}(\texttt{:wig},\text{true}))}$$

# Attacks between Arguments

An argument $S_1 \vdash \sigma_1$ can attack another argument $S_2 \vdash \sigma_2$ if and only if $\sigma_1$ is the contrary of one of the assumptions in $S_2$

$$\frac{\begin{array}{l} a_3 : \{\mathit{foundAt}(\texttt{:wig},\texttt{:Gifty})\} \vdash^{I_{A_1}} \mathit{isOrdinary}(\texttt{:wig},\text{true}) \\ b_3 : \{\mathit{isOrdinary}(\texttt{:wig},\text{false})\} \vdash^{I_{B_1} \cup P_{B_1} \cup_{i=1}^{5} r_i} \mathit{rejects}(\texttt{:bob},\texttt{:pr}) \end{array}}{c_3 = (\mathit{isOrdinary}(\texttt{:wig},\text{false}), \mathit{isOrdinary}(\texttt{:wig},\text{true}))}$$



# Argumentation in Action

## Table: ABA Specification

| |
|---|
| $\mathcal{R} = I_{A_1} \cup I_{B_1} \cup I_{B_2} \cup P_{B_1} \cup_{i=1}^{7} r_i$ |
| $r_1 = \{\rightarrow$ *isInContext*(:pr,:context)$\}$ |
| $r_2 = \{\rightarrow$ *hasMedium*(:pr,:medium)$\}$ |
| $r_3 = \{\rightarrow$ *includesObject*(:medium,:wig)$\}$ |
| $r_4 = \{\rightarrow$ ChristmasParty(:location)$\}$ |
| $r_5 = \{\rightarrow$ *obtainedFrom*(:wig,:location)$\}$ |
| $r_6 = \{\rightarrow$ *taggedPerson*(:medium,:bob)$\}$ |
| $r_7 = \{\rightarrow$ *hasUrl*(:Gifty,:url)$\}$ |
| $\mathcal{A} = \{as_1, as_2, as_3, as_4\}$ |
| $as_1 =$ *foundAt*(:wig,:Gifty) |
| $as_2 =$ not(*rejects*(:alice,:pr)) |
| $as_3 =$ *isOrdinary*(:wig,false) |
| $as_4 =$ *isAccessible*(:url,false) |
| $\mathcal{C} = \{c_1, c_2, c_3, c_4\}$ |
| $c_1 =$ (*foundAt*(:wig,:Gifty)=*isClosed*(:Gifty,true)) |
| $c_2 =$ (not(*rejects*(:alice,:pr))=*rejects*(:bob,:pr)) |
| $c_3 =$ (*isOrdinary*(:wig,false)=*isOrdinary*(:wig,true)) |
| $c_4 =$ (*isAccessible*(:url,false)=*isAccessible*(:url,true)) |

# Argumentation in Action

## Table: ABA Specification

| |
|---|
| $\mathcal{R} = I_{A_1} \cup I_{B_1} \cup I_{B_2} \cup P_{B_1} \cup_{i=1}^{7} r_i$ |
| $r_1 = \{\rightarrow$ *isInContext*(:pr,:context)$\}$ |
| $r_2 = \{\rightarrow$ *hasMedium*(:pr,:medium)$\}$ |
| $r_3 = \{\rightarrow$ *includesObject*(:medium,:wig)$\}$ |
| $r_4 = \{\rightarrow$ ChristmasParty(:location)$\}$ |
| $r_5 = \{\rightarrow$ *obtainedFrom*(:wig,:location)$\}$ |
| $r_6 = \{\rightarrow$ *taggedPerson*(:medium,:bob)$\}$ |
| $r_7 = \{\rightarrow$ *hasUrl*(:Gifty,:url)$\}$ |
| $\mathcal{A} = \{as_1, as_2, as_3, as_4\}$ |
| $as_1 =$ *foundAt*(:wig,:Gifty) |
| $as_2 =$ not(*rejects*(:alice,:pr)) |
| $as_3 =$ *isOrdinary*(:wig,false) |
| $as_4 =$ *isAccessible*(:url,false) |
| $\mathcal{C} = \{c_1, c_2, c_3, c_4\}$ |
| $c_1 =$ (*foundAt*(:wig,:Gifty)=*isClosed*(:Gifty,true)) |
| $c_2 =$ (not(*rejects*(:alice,:pr))=*rejects*(:bob,:pr)) |
| $c_3 =$ (*isOrdinary*(:wig,false)=*isOrdinary*(:wig,true)) |
| $c_4 =$ (*isAccessible*(:url,false)=*isAccessible*(:url,true)) |

## Table: Arguments

| |
|---|
| $f_1 : \{\} \vdash^{r_1}$ *isInContext*(:pr,:context) |
| $f_2 : \{\} \vdash^{r_2}$ *hasMedium*(:pr,:medium) |
| $f_3 : \{\} \vdash^{r_3}$ *includesObject*(:medium,:wig) |
| $f_4 : \{\} \vdash^{r_4}$ ChristmasParty(:location) |
| $f_5 : \{\} \vdash^{r_5}$ *obtainedFrom*(:wig,:location) |
| $f_6 : \{\} \vdash^{r_6}$ *taggedPerson*(:medium,:bob) |
| $f_7 : \{\} \vdash^{r_7}$ *hasUrl*(:Gifty,:url) |
| $a_1 : \{$*foundAt*(:wig,:Gifty)$\} \vdash$ *foundAt*(:wig,:Gifty) |
| $a_2 : \{$not(*rejects*(:alice,:pr))$\} \vdash$ not(*rejects*(:alice,:pr)) |
| $a_3 : \{$*foundAt*(:wig,:Gifty)$\} \vdash^{I_{A_1}}$ *isOrdinary*(:wig,true) |
| $b_1 : \{$*isOrdinary*(:wig,false)$\} \vdash$ *isOrdinary*(:wig, false) |
| $b_2 : \{$*isOrdinary*(:wig,false)$\} \vdash^{I_{B_1} \cup_{i=1}^{5} r_i}$ Party(:context) |
| $b_3 : \{$*isOrdinary*(:wig,false)$\} \vdash^{I_{B_1} \cup P_{B_1} \cup_{i=1}^{5} r_i}$ *rejects*(:bob,:pr) |
| $b_4 : \{$*isAccessible*(:url,false)$\} \vdash$ *isAccessible*(:url,false) |
| $b_5 : \{$*isAccessible*(:url,false)$\} \vdash^{I_{B_2} \cup r_7}$ *isClosed*(:Gifty,true) |

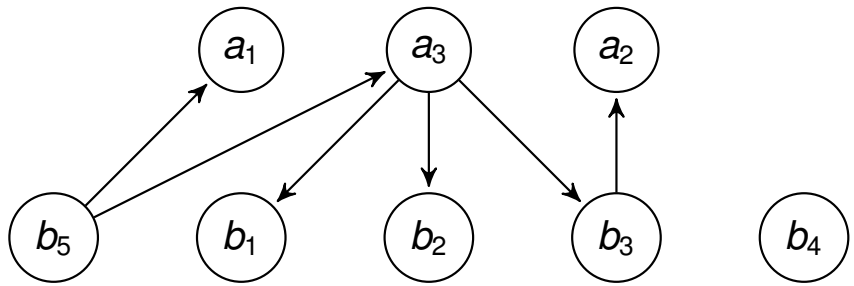
# Argumentation in Action

## Table: ABA Specification

| |
|---|
| $\mathcal{R} = I_{A_1} \cup I_{B_1} \cup I_{B_2} \cup P_{B_1} \cup_{i=1}^{7} r_i$ |
| $r_1 = \{\rightarrow$ *isInContext*(:pr,:context)$\}$ |
| $r_2 = \{\rightarrow$ *hasMedium*(:pr,:medium)$\}$ |
| $r_3 = \{\rightarrow$ *includesObject*(:medium,:wig)$\}$ |
| $r_4 = \{\rightarrow$ ChristmasParty(:location)$\}$ |
| $r_5 = \{\rightarrow$ *obtainedFrom*(:wig,:location)$\}$ |
| $r_6 = \{\rightarrow$ *taggedPerson*(:medium,:bob)$\}$ |
| $r_7 = \{\rightarrow$ *hasUrl*(:Gifty,:url)$\}$ |
| $\mathcal{A} = \{as_1, as_2, as_3, as_4\}$ |
| $as_1 =$ *foundAt*(:wig,:Gifty) |
| $as_2 =$ not(*rejects*(:alice,:pr)) |
| $as_3 =$ *isOrdinary*(:wig,false) |
| $as_4 =$ *isAccessible*(:url,false) |
| $\mathcal{C} = \{c_1, c_2, c_3, c_4\}$ |
| $c_1 =$ (*foundAt*(:wig,:Gifty)=*isClosed*(:Gifty,true)) |
| $c_2 =$ (not(*rejects*(:alice,:pr))=*rejects*(:bob,:pr)) |
| $c_3 =$ (*isOrdinary*(:wig,false)=*isOrdinary*(:wig,true)) |
| $c_4 =$ (*isAccessible*(:url,false)=*isAccessible*(:url,true)) |

## Table: Arguments

| |
|---|
| $f_1 : \{\} \vdash^{r_1}$ *isInContext*(:pr,:context) |
| $f_2 : \{\} \vdash^{r_2}$ *hasMedium*(:pr,:medium) |
| $f_3 : \{\} \vdash^{r_3}$ *includesObject*(:medium,:wig) |
| $f_4 : \{\} \vdash^{r_4}$ ChristmasParty(:location) |
| $f_5 : \{\} \vdash^{r_5}$ *obtainedFrom*(:wig,:location) |
| $f_6 : \{\} \vdash^{r_6}$ *taggedPerson*(:medium,:bob) |
| $f_7 : \{\} \vdash^{r_7}$ *hasUrl*(:Gifty,:url) |
| $a_1 :$ {*foundAt*(:wig,:Gifty)}$\vdash$ *foundAt*(:wig,:Gifty) |
| $a_2 :$ {not(*rejects*(:alice,:pr))}$\vdash$ not(*rejects*(:alice,:pr)) |
| $a_3 :$ {*foundAt*(:wig,:Gifty)}$\vdash^{I_{A_1}}$ *isOrdinary*(:wig,true) |
| $b_1 :$ {*isOrdinary*(:wig,false)}$\vdash$ *isOrdinary*(:wig, false) |
| $b_2 :$ {*isOrdinary*(:wig,false)}$\vdash^{I_{B_1} \cup_{i=1}^{5} r_i}$ Party(:context) |
| $b_3 :$ {*isOrdinary*(:wig,false)}$\vdash^{I_{B_1} \cup P_{B_1} \cup_{i=1}^{5} r_i}$ *rejects*(:bob,:pr) |
| $b_4 :$ {*isAccessible*(:url,false)}$\vdash$ *isAccessible*(:url,false) |
| $b_5 :$ {*isAccessible*(:url,false)}$\vdash^{I_{B_2} \cup r_7}$ *isClosed*(:Gifty,true) |



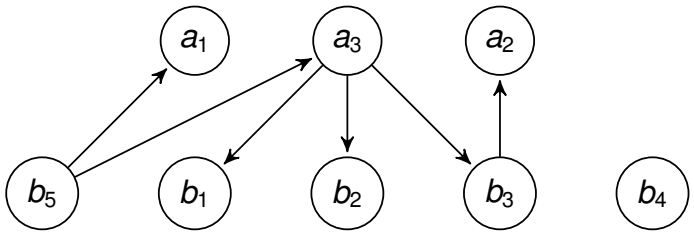

Attacks

# Semantics for ABA

- Finds justified argument sets.
- We use credulously admissible argument sets.
  An argument set is admissible iff,
  1. It does not attack itself and
  2. It can defend itself against all attacks

# Semantics for ABA

- Finds justified argument sets.
- We use credulously admissible argument sets.
  An argument set is admissible iff,
  1. It does not attack itself and
  2. It can defend itself against all attacks



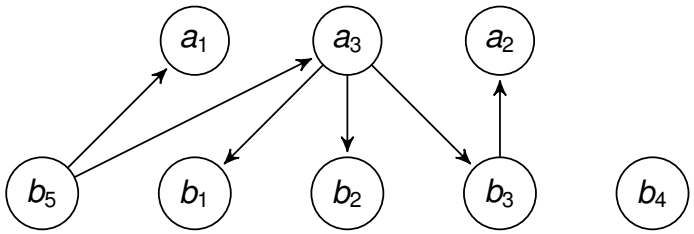

**Justified Argument Sets**

$\{\}, \{b_5\}, \{b_4\}, \{b_4, b_5\}, \{b_3, b_5\}, \{b_3, b_4, b_5\},$
$\{b_2, b_5\}, \{b_2, b_3, b_5\}, \{b_2, b_4, b_5\},$
$\{b_2, b_3, b_4, b_5\}, \{b_1, b_5\}, \{b_1, b_4, b_5\},$
$\{b_1, b_3, b_5\}, \{b_1, b_3, b_4, b_5\}, \{b_1, b_2, b_5\},$
$\{b_1, b_2, b_4, b_5\}, \{b_1, b_2, b_3, b_5\}, \{b_1, b_2, b_3, b_4, b_5\}$

Credulous semantics allow for alternative argument sets

# Semantics for ABA

- Finds justified argument sets.
- We use credulously admissible argument sets.
  An argument set is admissible iff,
  1. It does not attack itself and
  2. It can defend itself against all attacks




**Justified Argument Sets**

$\{\}, \{b_5\}, \{b_4\}, \{b_4, b_5\}, \{b_3, b_5\}, \{b_3, b_4, b_5\},$
$\{b_2, b_5\}, \{b_2, b_3, b_5\}, \{b_2, b_4, b_5\},$
$\{b_2, b_3, b_4, b_5\}, \{b_1, b_5\}, \{b_1, b_4, b_5\},$
$\{b_1, b_3, b_5\}, \{b_1, b_3, b_4, b_5\}, \{b_1, b_2, b_5\},$
$\{b_1, b_2, b_4, b_5\}, \{b_1, b_2, b_3, b_5\}, \{b_1, b_2, b_3, b_4, b_5\}$

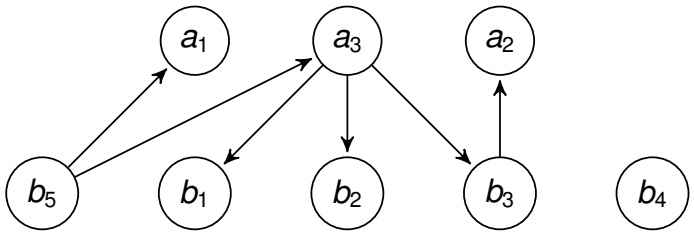
Credulous semantics allow for alternative argument sets

$b_3 : \{isOrdinary(\texttt{:wig}, \text{false})\} \vdash^{I_{B_1} \cup P_{B_1} \cup_{i=1}^{5} r_i}$

*rejects*(:bob,:pr) is justified!

# Semantics for ABA

- Finds justified argument sets.
- We use credulously admissible argument sets.
  An argument set is admissible iff,
  1. It does not attack itself and
  2. It can defend itself against all attacks




**Justified Argument Sets**

$\{\}, \{b_5\}, \{b_4\}, \{b_4, b_5\}, \{b_3, b_5\}, \{b_3, b_4, b_5\},$
$\{b_2, b_5\}, \{b_2, b_3, b_5\}, \{b_2, b_4, b_5\},$
$\{b_2, b_3, b_4, b_5\}, \{b_1, b_5\}, \{b_1, b_4, b_5\},$
$\{b_1, b_3, b_5\}, \{b_1, b_3, b_4, b_5\}, \{b_1, b_2, b_5\},$
$\{b_1, b_2, b_4, b_5\}, \{b_1, b_2, b_3, b_5\}, \{b_1, b_2, b_3, b_4, b_5\}$

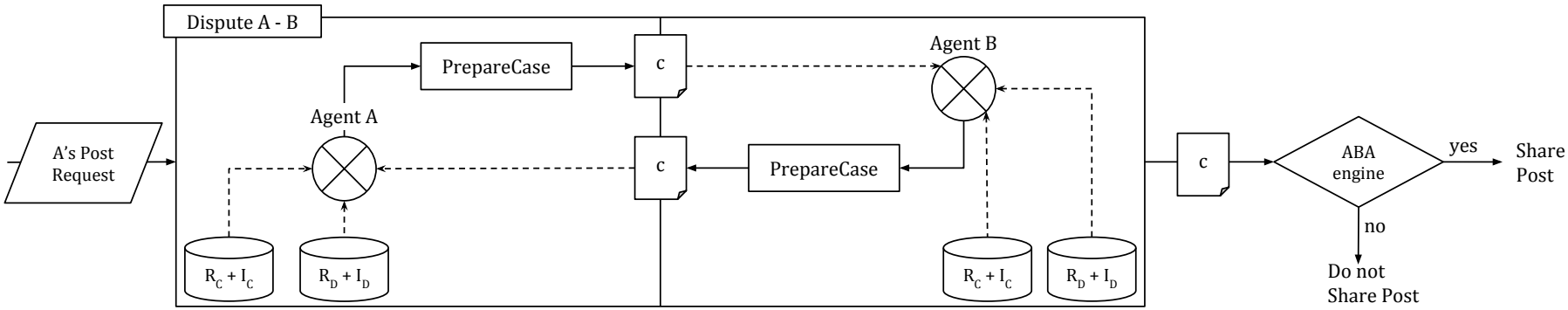
Credulous semantics allow for alternative argument sets

$b_3 : \{isOrdinary(\texttt{:wig}, \text{false})\} \vdash^{I_{B_1} \cup P_{B_1} \cup_{i=1}^{5} r_i}$

*rejects*(:bob,:pr) is justified!

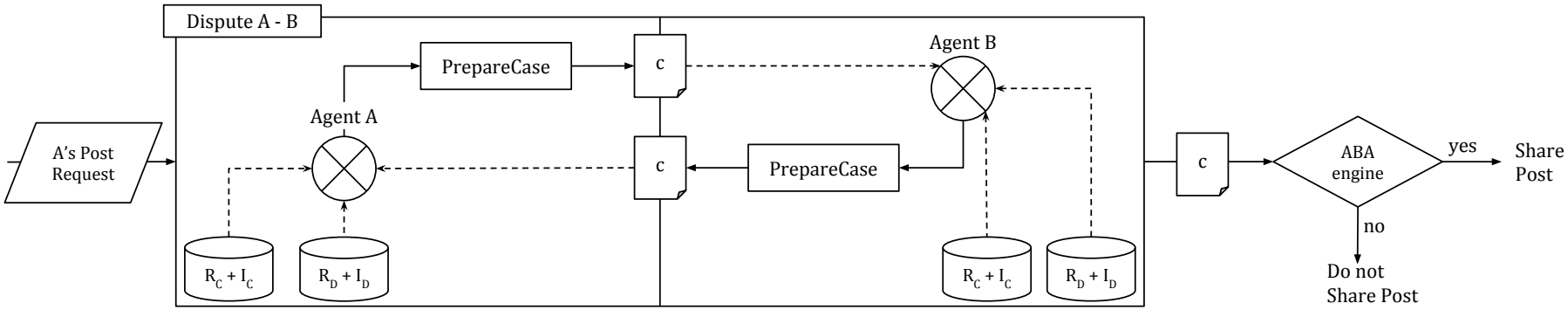
Distributed argumentation to create ABA specification in a turntaking fashion.

# Distributed Privacy Argumentation Framework




- A case is a tuple $\langle \mathcal{R}, \mathcal{A}, \mathcal{F}, \mathcal{C}, \mathit{status} \rangle$
- *R* is a set of rules, *A* is a set of assumptions, *F* is a set of facts, *C* is the assumption contrary mapping and *status* is either *ongoing* or *stop*.
- A case includes an ABA specification, which is updated in each iteration.

# Distributed Privacy Argumentation Framework



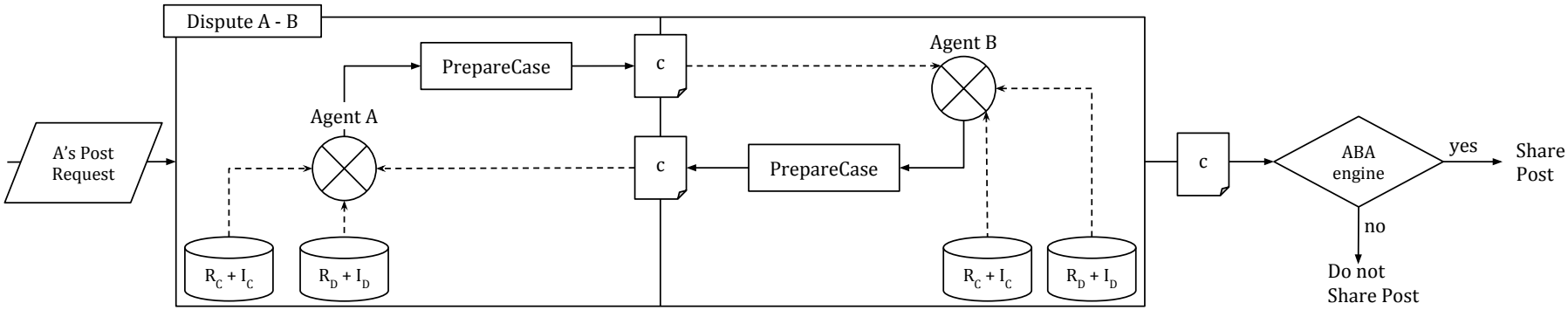

- A case is a tuple $\langle \mathcal{R}, \mathcal{A}, \mathcal{F}, \mathcal{C}, \mathit{status} \rangle$
- $R$ is a set of rules, $A$ is a set of assumptions, $F$ is a set of facts, $C$ is the assumption contrary mapping and *status* is either *ongoing* or *stop*.
- A case includes an ABA specification, which is updated in each iteration.

Centralized Rules $R_C$, Centralized Instances $I_C$

$I_{A_1}$: *foundAt*(?object, ?shop) $\rightarrow$ *isOrdinary*(?object, true)

*foundAt*(`:wig`,`:Gifty`)

# Distributed Privacy Argumentation Framework



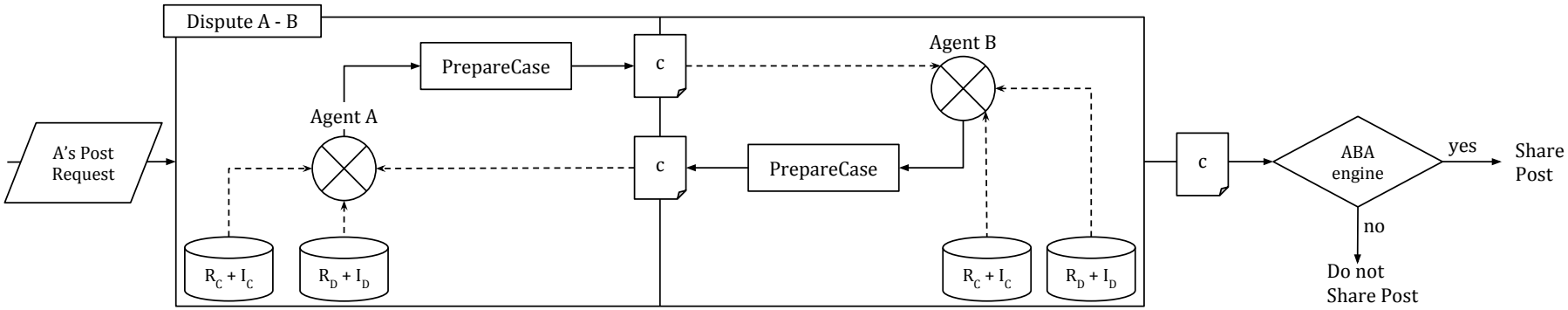

- A case is a tuple $\langle \mathcal{R}, \mathcal{A}, \mathcal{F}, \mathcal{C}, \mathit{status} \rangle$
- $R$ is a set of rules, $A$ is a set of assumptions, $F$ is a set of facts, $C$ is the assumption contrary mapping and *status* is either *ongoing* or *stop*.
- A case includes an ABA specification, which is updated in each iteration.

Centralized Rules $R_C$, Decentralized Instances $I_D$

$I_{A_1}$: *foundAt*(?object, ?shop) $\rightarrow$ *isOrdinary*(?object, true)

*foundAt*(`:wig`,`:Gifty`)

# Distributed Privacy Argumentation Framework



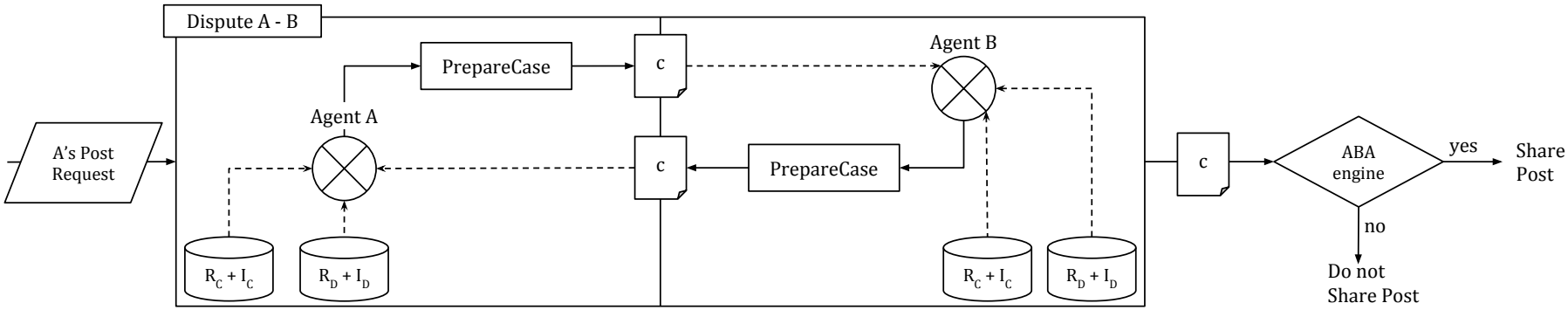

- A case is a tuple $\langle \mathcal{R}, \mathcal{A}, \mathcal{F}, \mathcal{C}, \mathit{status} \rangle$
- $R$ is a set of rules, $A$ is a set of assumptions, $F$ is a set of facts, $C$ is the assumption contrary mapping and *status* is either *ongoing* or *stop*.
- A case includes an ABA specification, which is updated in each iteration.

Decentralized Rules $R_D$, Centralized Instances $I_C$

$I_{B_2}$:*hasUrl*(?shop,?url),*isAccessible*(?url,false)$\rightarrow$*isClosed*(?shop,true)

*hasUrl*(`:Gifty,:url`)

# Distributed Privacy Argumentation Framework




- A case is a tuple $\langle \mathcal{R}, \mathcal{A}, \mathcal{F}, \mathcal{C}, \mathit{status} \rangle$
- $R$ is a set of rules, $A$ is a set of assumptions, $F$ is a set of facts, $C$ is the assumption contrary mapping and *status* is either *ongoing* or *stop*.
- A case includes an ABA specification, which is updated in each iteration.

Decentralized Rules $R_D$, Decentralized Instances $I_D$

$I_{B_2}$:*hasUrl*(?shop,?url),*isAccessible*(?url,false)$\rightarrow$*isClosed*(?shop,true)

*isAccessible*(:`url`,false)

# Evaluation

- Lack of data: Difficult to collect, impossible to share
- User study
  - Online survey and personal interviews to gather privacy requirements and outcome expectations
  - Participants evaluate the scenarios as neutral observers or by impersonation
  - Example scenarios are shown in stages
  - User expectations are compared with the algorithms outcomes
- Multiagent simulations

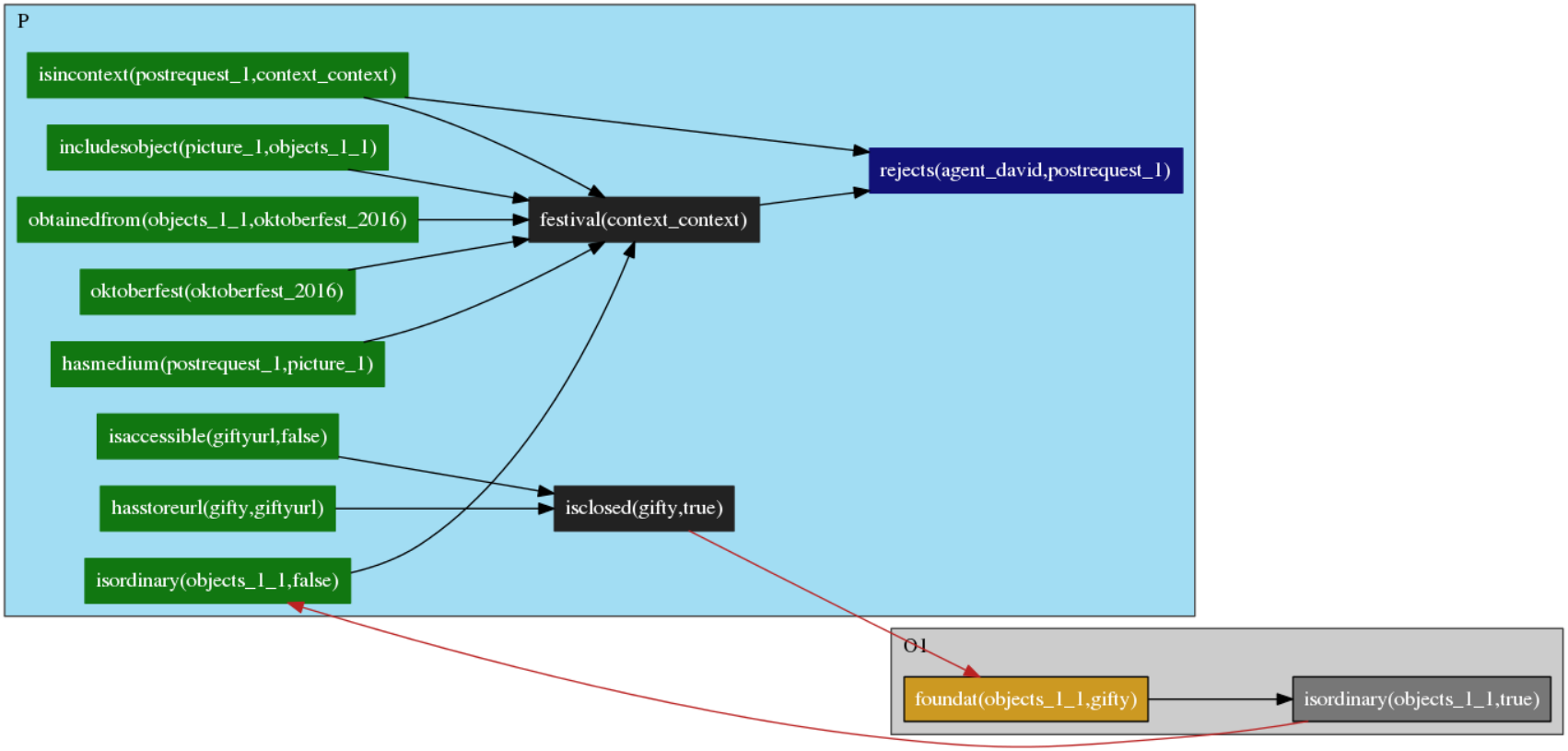
# Evaluation

- Lack of data: Difficult to collect, impossible to share
- User study
  - Online survey and personal interviews to gather privacy requirements and outcome expectations
  - Participants evaluate the scenarios as neutral observers or by impersonation
  - Example scenarios are shown in stages
  - User expectations are compared with the algorithms outcomes
- Multiagent simulations

Table: Personal Interviews and Online Survey Results

| Stage | Personal Interviews (36 participants) | | Online Survey (68 participants) | | PriArg |
|---|---|---|---|---|---|
| | Share | Not Share | Share | Not Share | |
| 1 | 5.55% | **94.44%** | 7.35% | **92.65%** | **Not Share** |
| 2 | **52.77%** | 47.22% | 20.59% | **79.41%** | **Share** |
| 3 | 2.77% | **97.22%** | 7.35% | **92.65%** | **Not Share** |

# Explanation



# Negotiation

- Negotiation is mostly used in e-commerce.
- Agents try to reach a mutually acceptable agreement.
- Negotiation technique consists of various components:
  - A *protocol* is a set of rules allowing agents to interact.
  - A *strategy* (mostly private) is used by agents to make *offers* and *counter-offers*.
  - An *agreement rule* determines when an agreement has been reached.

# How to use negotiation technique in privacy context?

- Given a protocol, an agent starts a negotiation with other agents to publish a post.
- Each agent evaluates this post according to its own strategy.
  - It gives a response (accept or deny). The negotiator agent analyzes responses and take an action.
  - It proposes a counter-offer (e.g., a new post), which should be agreed on by agents involved in the counter-offer.

**In privacy context, what is ...**

An agreement? A protocol? A strategy? An offer? A counter-offer? An agreement rule?

# Creating a Post Request

- The content owner puts together the content she wants to publish with the potential audience
- Her agent decides with whom the post is related
  - Sends the post request to those agents
  - Asks for feedback
  - Feedback: I don't want to see Bob in the audience; I don't want a picture on this date to be shown, etc.
  - Feedback calculated based on the Privacy Rules
  - Collects the reasons and revises the post request

# Revising a Post Request

- Rejection reasons cannot conflict with each other.
- When a post request is rejected by at least one agent, the negotiator agent:
  - honors every rejection reason,
  - checks whether the resulting post request is reasonable.
- Alternatives: lots of possibilities (using priorities, past experience)

# An Example Execution

| Iter. | Content | Audience | Asked Agents | Evaluate | Response |
|---|---|---|---|---|---|
| 1 | May 1 picture | Bob, Carol, Errol, Filipo | :carol | :carol $\rightarrow P_{C_2}$ | :carol $\rightarrow$ -date |
| 2 | May 28 picture$_1$ | Bob, Carol, Errol, Filipo | :carol, :bob | :carol $\rightarrow$ N/A, :bob $\rightarrow P_{B_2}$ | :carol $\rightarrow$ 3, :bob $\rightarrow$ -self |
| 3 | May 28 picture$_2$ | Bob, Carol, Errol, Filipo | :carol, :bob | :carol $\rightarrow$ N/A, :bob $\rightarrow$ N/A | :carol $\rightarrow$ 4, :bob $\rightarrow$ 4 |

# Preserving Privacy as Social Responsibility[6]

- Exploit reciprocity as a heuristic (e.g., this time you help me, next time I help you)
- Agents negotiate with each other on their users' preferences
- Negotiation strategies to concede on their preferences
- Given incentives through gamification

[6] Dilara Kekulluoglu, Nadin Kökciyan, and Pınar Yolum. "Preserving Privacy as Social Responsibility in Online Social Networks". In: *ACM Transactions on Internet Technology* (2018).

# Important Criteria

- Concealment of privacy constraints (not being have to explain everything)
- Protection before exposure (checking privacy constraints prior to posting)
- Automating privacy protection (using software agents)
- Fairness (partial improvements instead of all-or-nothing approach)

# Research Directions

- Deciphering user's privacy preferences[7]
  - Privacy rules can be identified based on previously shared content using machine learning algorithms
  - Asking other trusted users for privacy recommendation
- Instructing users about privacy preferences[8]
  - User studies show many users do not know what their privacy expectations or even implications
  - Making suggestions based on other trusted users for privacy recommendation or already shared content

[7] Berkant Kepez and Pınar Yolum. "Learning Privacy Rules Cooperatively in Online Social Networks". In: *PrAISe@ECAI*. 2016.

[8] Abdurrahman Can Kurtan and Pinar Yolum. "PELTE: Privacy Estimation of Images from Tags". In: *AAMAS*. 2018, pp. 1989–1991.

# Research Directions

- Managing privacy in IoT
  - Context-Based as opposed to Policy-Based[9]
  - Common-sense reasoning as opposed to personalization
  - Scaling up methods for detection and prediction
- Privacy vs. Utility
  - Agents choosing to violate privacy for a better outcome
  - Metrics to evaluate benefit and cost for privacy
  - Agents learning their evaluations over time

[9] Nadin Kökciyan and Pinar Yolum. "Context-Based Reasoning on Privacy in Internet of Things." In: *IJCAI*. 2017.

# Summary: Agents for Privacy

- Represent Privacy Preferences: Semantic representation of policies as those in knowledge representation
- Elicit Privacy Preferences: Machine learning to understand user behavior over time or gamification for understanding users
- Agent-Based Modeling: Agents act on behalf of users to detect privacy violations or avoid them in the first place
- Multiagent Agreement Technologies: Negotiation or argumentation among software agents to reach an agreement for sharing settings