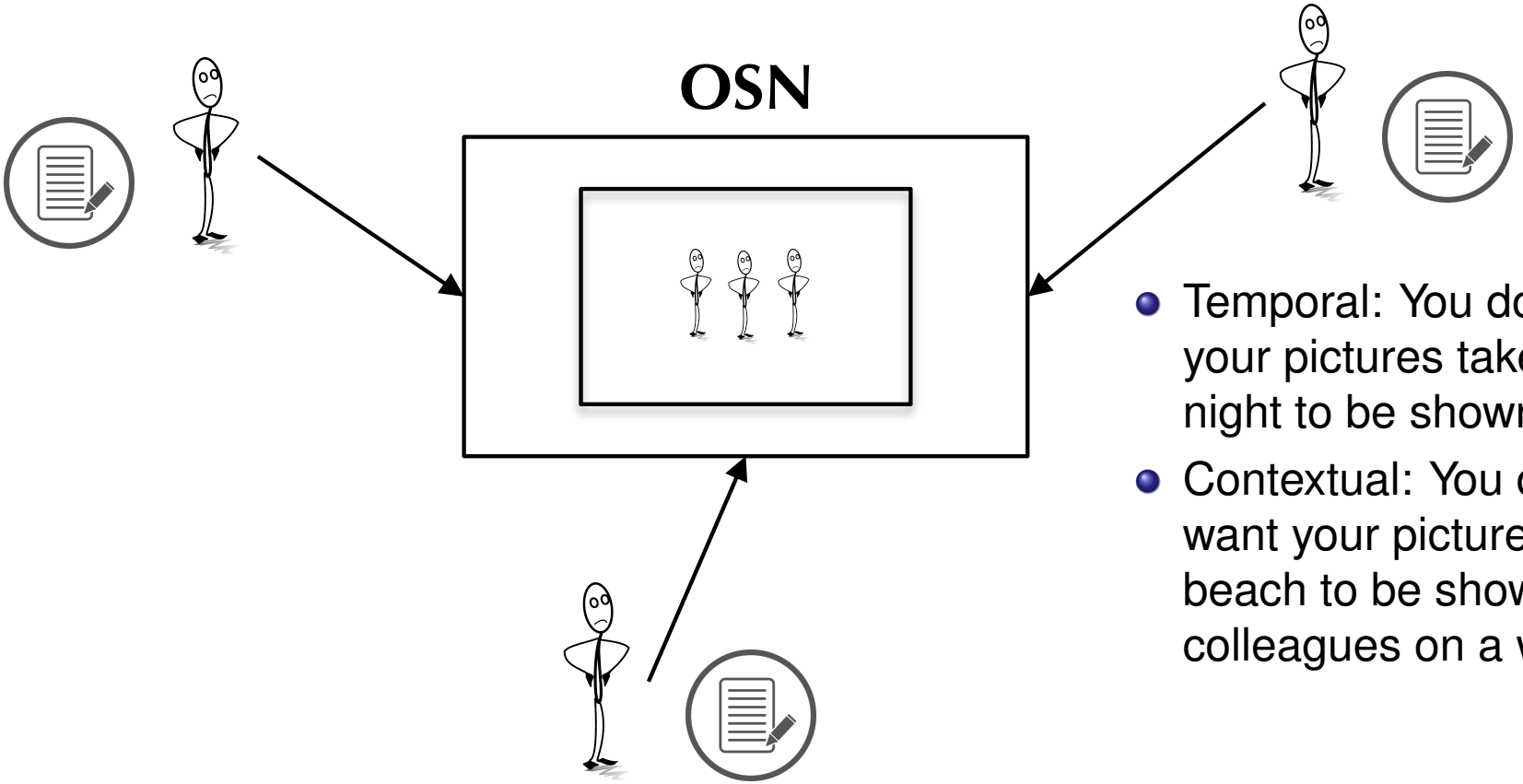
# Agent-Based Privacy Management for Social Media

Pınar Yolum
Email: p.yolum@uu.nl

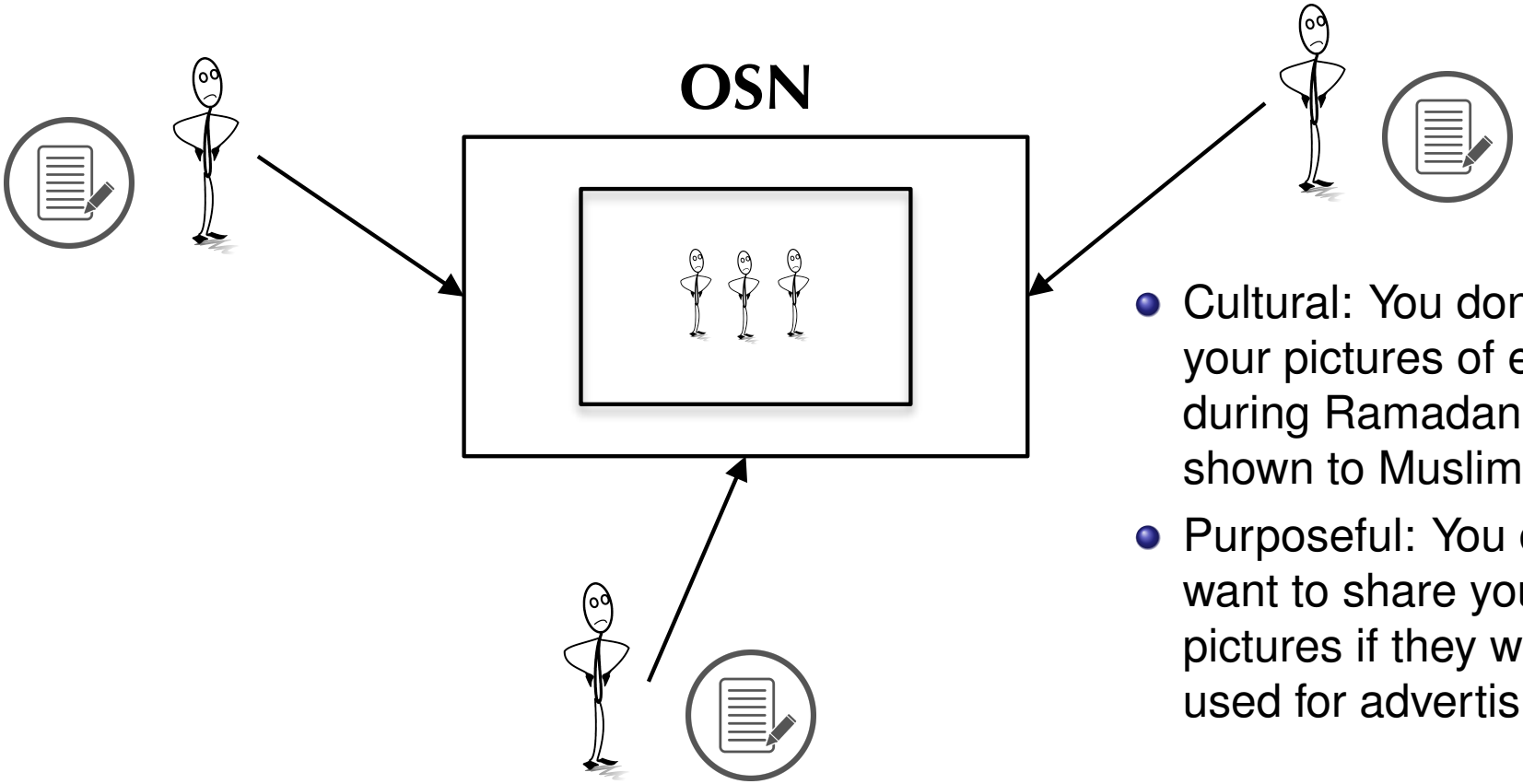
Department of Information and Computing Sciences
Utrecht University

# Privacy in Online Social Networks



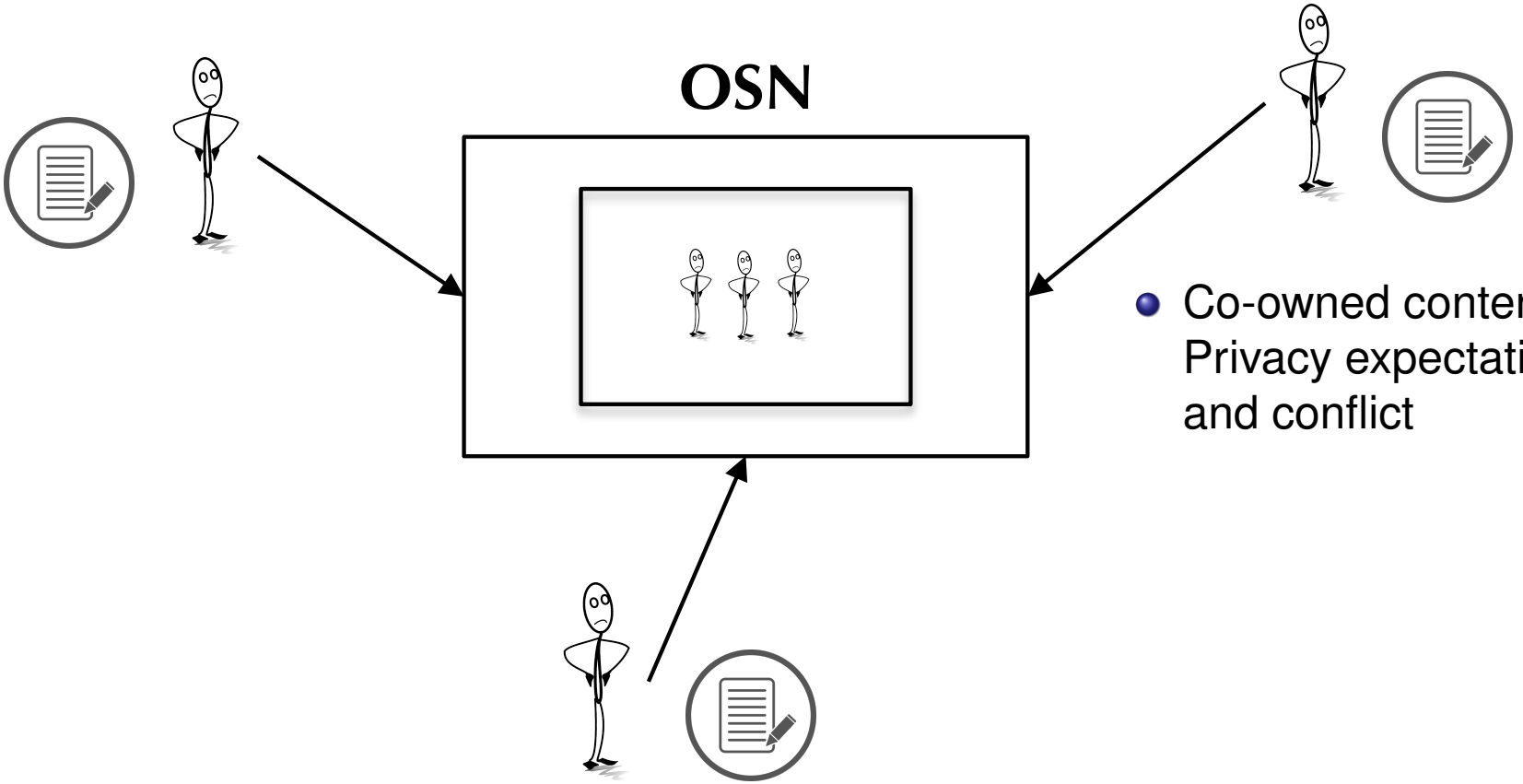

- Temporal: You don't want your pictures taken at night to be shown
- Contextual: You don't want your pictures at the beach to be shown to colleagues on a work day

# 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

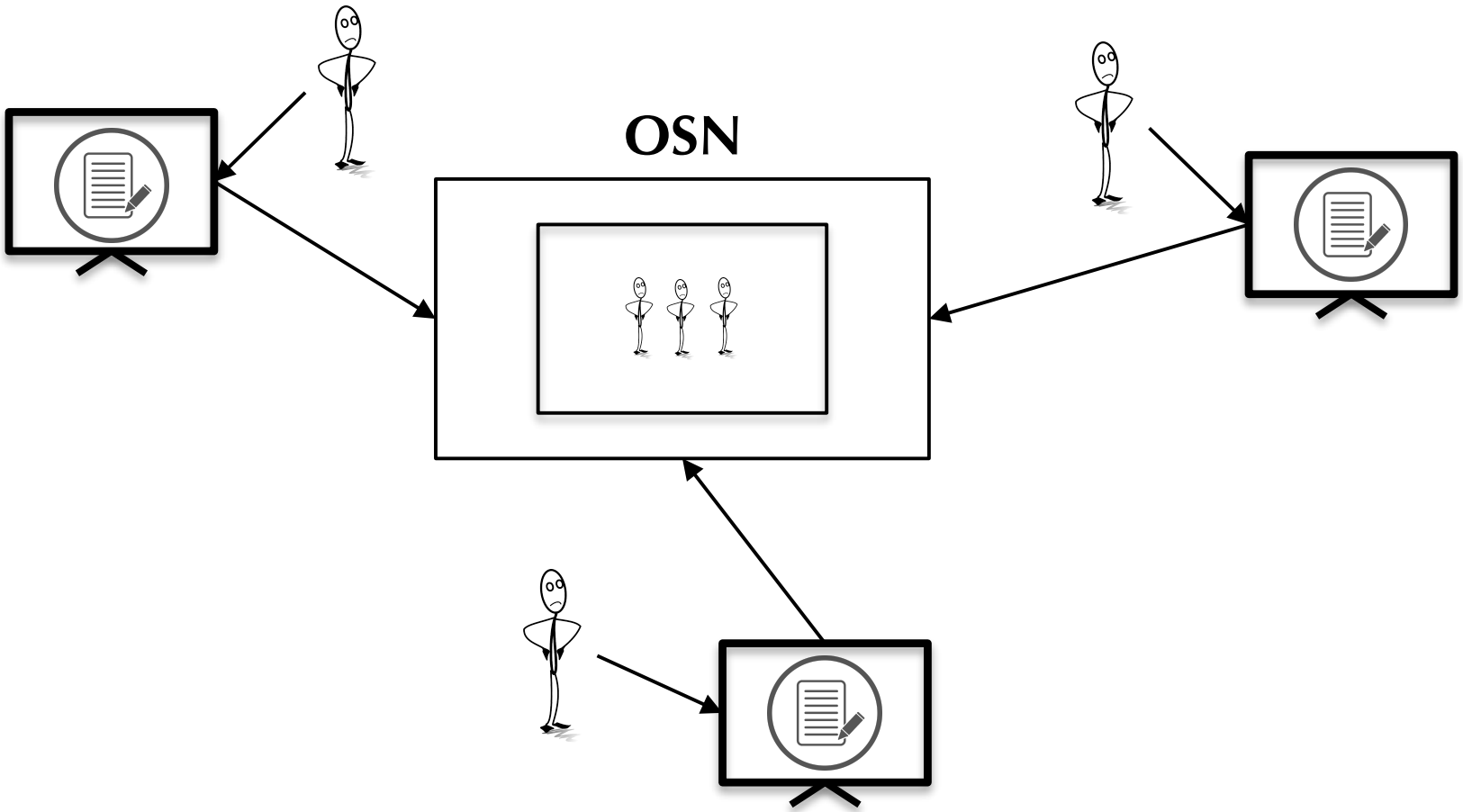



- Co-owned content: Privacy expectations vary and conflict

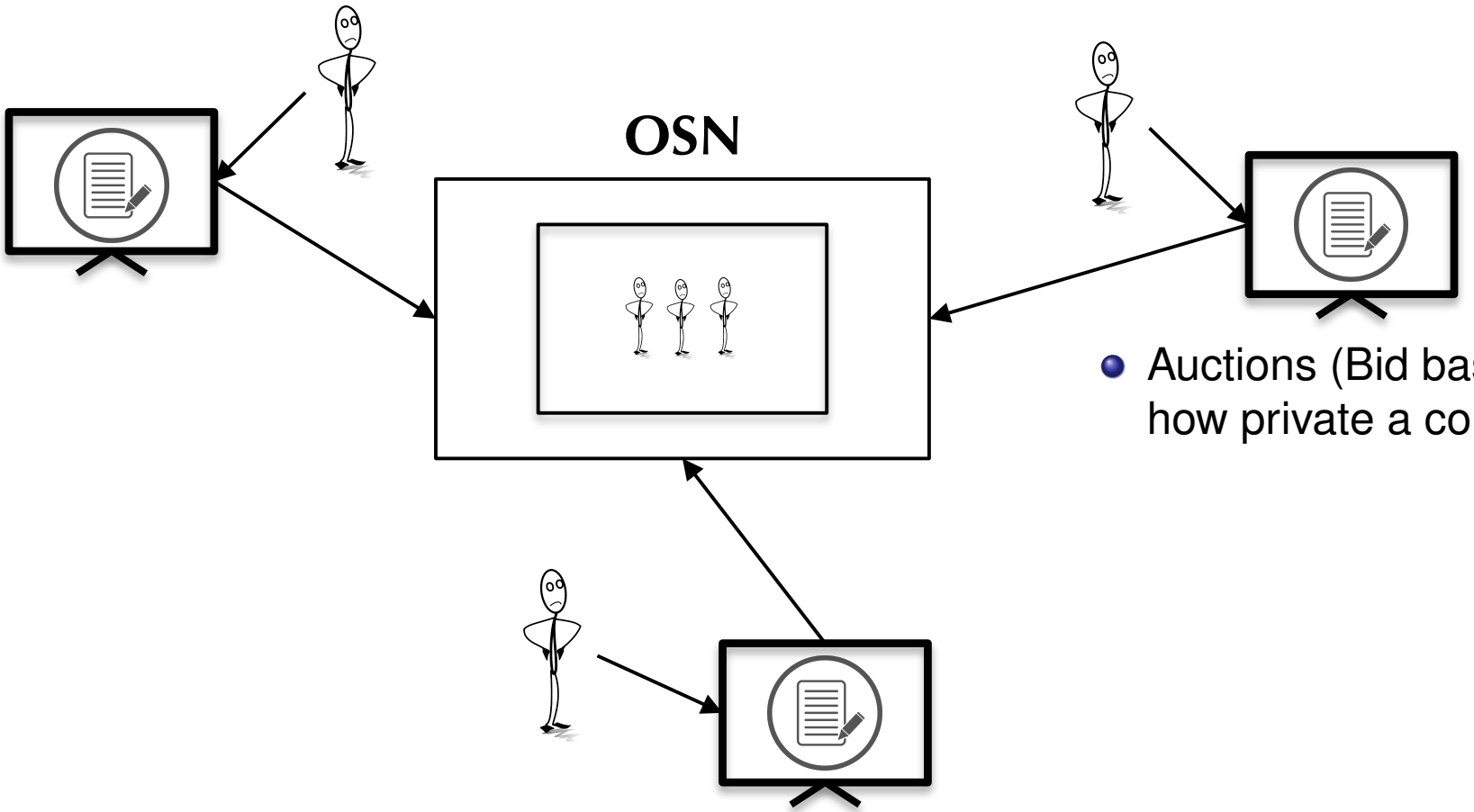
# How to manage the privacy of co-owned data?

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

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

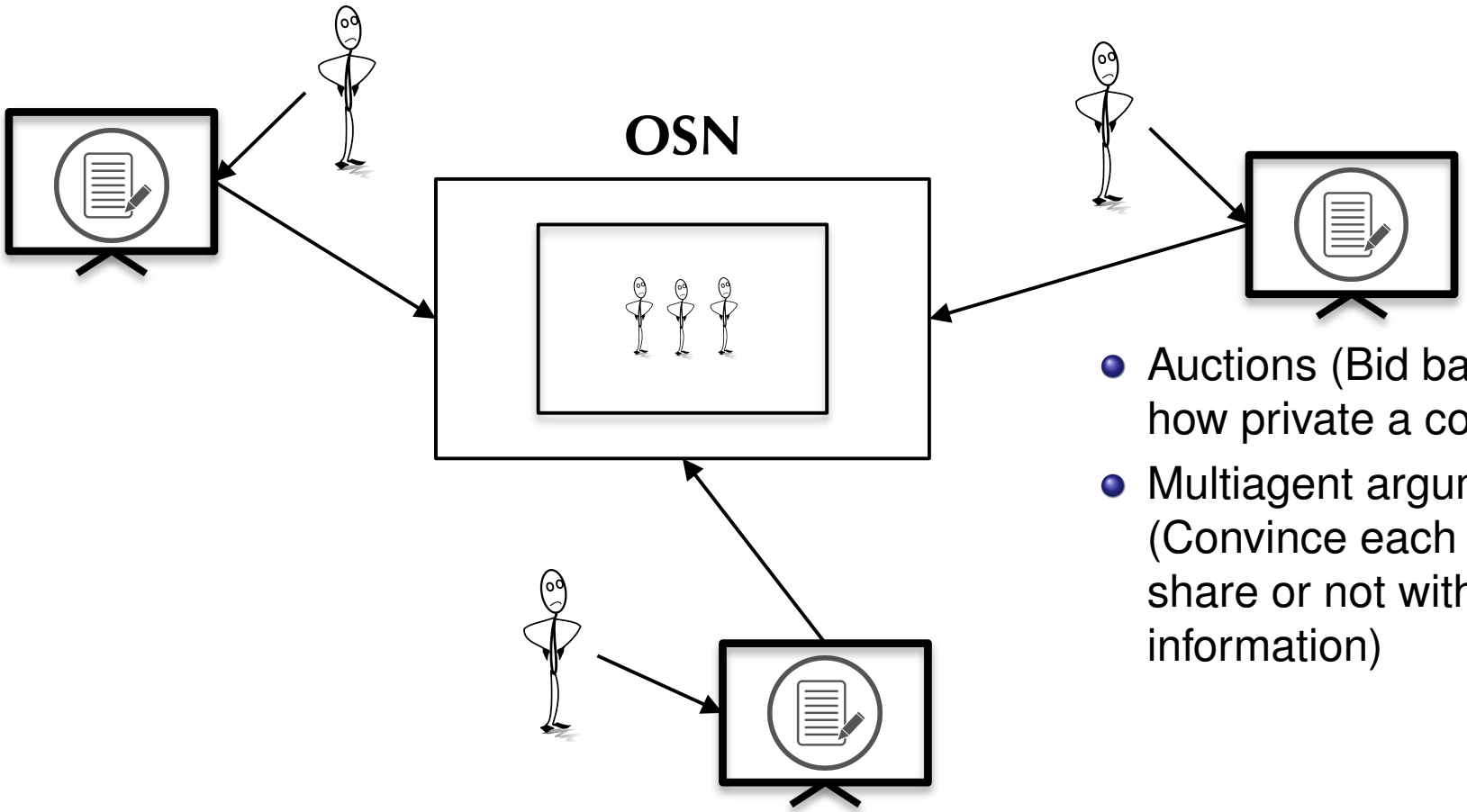


# How to agree on the privacy of co-owned content?



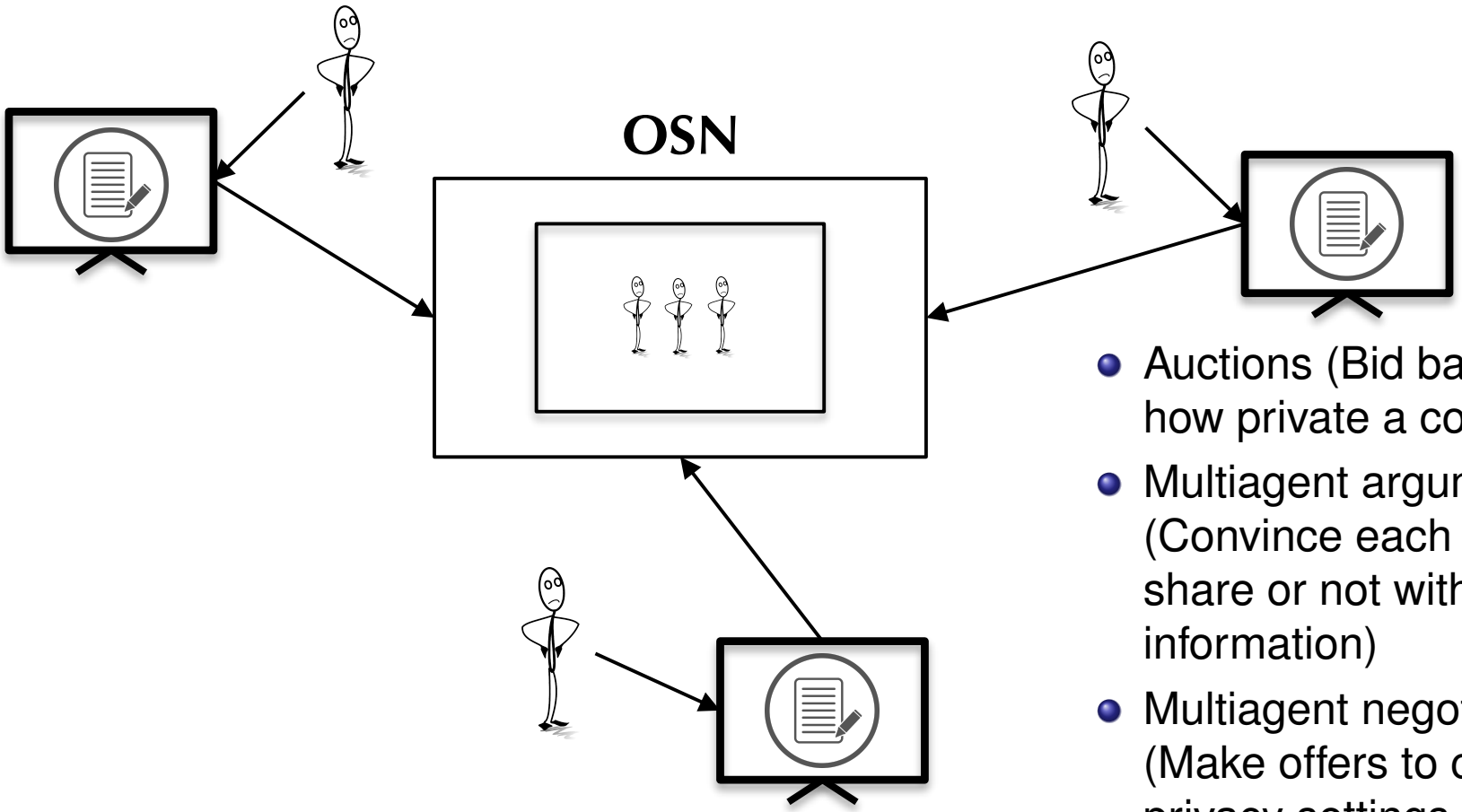

- Auctions (Bid based on how private a content is)

# 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[1]

- Auction mechanism where participants bid for different possible actions in the environment.

**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 |

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

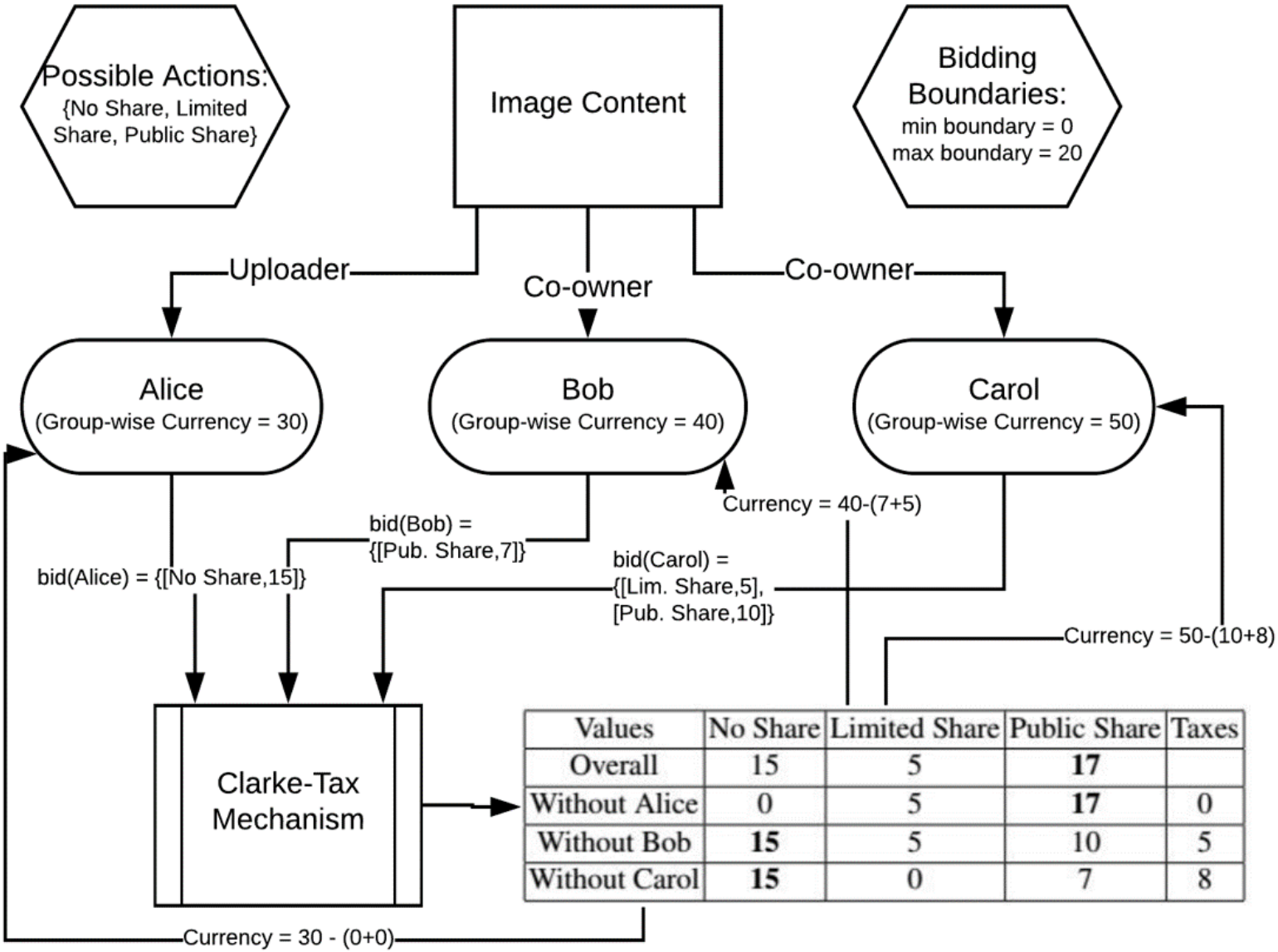
# PANO Example

- 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.

**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?



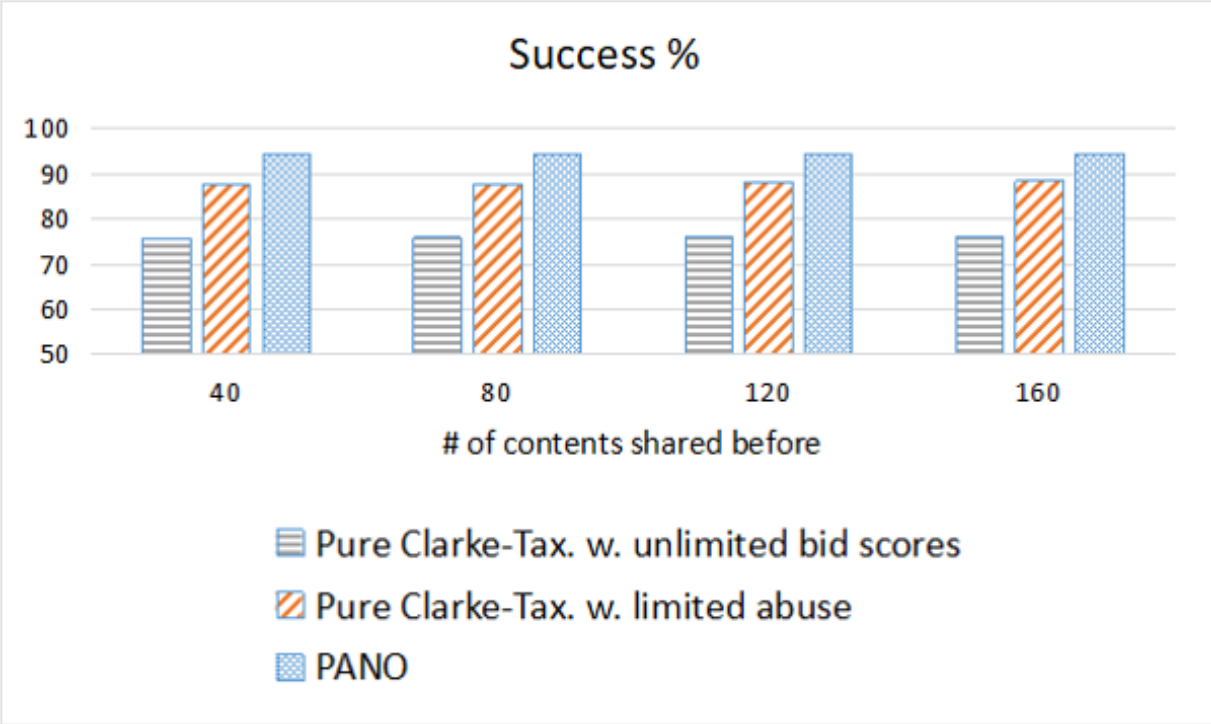

| Values | No Share | Limited Share | Public Share | Taxes |
|---|---|---|---|---|
| Overall | 15 | 5 | **17** | |
| Without Alice | 0 | 5 | **17** | 0 |
| Without Bob | **15** | 5 | 10 | 5 |
| Without Carol | **15** | 0 | 7 | 8 |

# 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



# Convince Others to Keep Private[2]

- 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.

[2] 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.

# 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) |

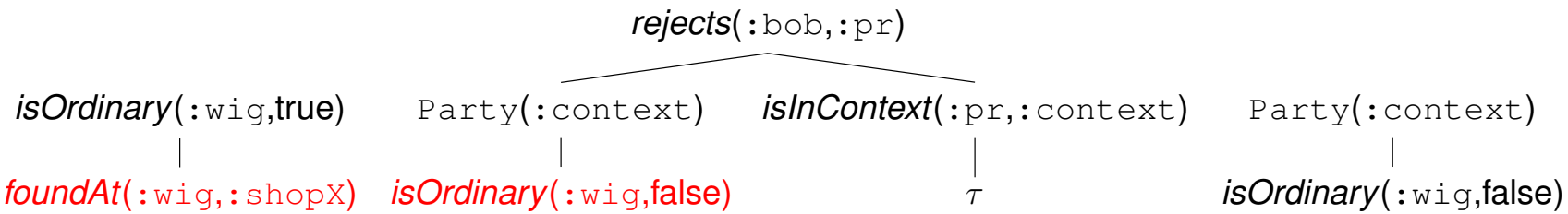


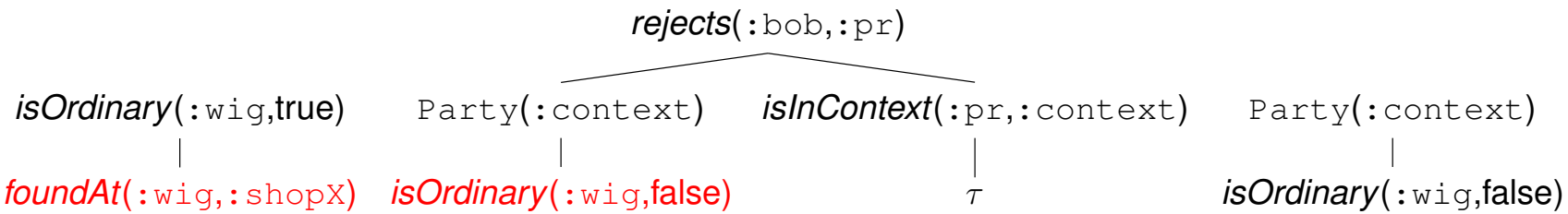

Figure: Deduction Trees

# 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

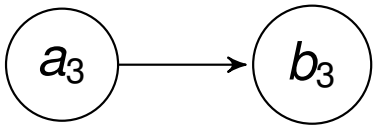
An argument $S_1 \vdash \sigma_1$ can attack another argument $S_2 \vdash \sigma_2$ if and only if $\sigma_1$ is the contrary of one of the assumptions in $S_2$

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

# 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) |

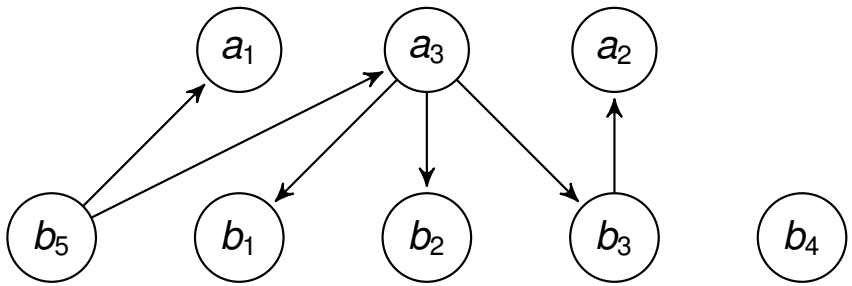
# Argumentation in Action

## 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) |



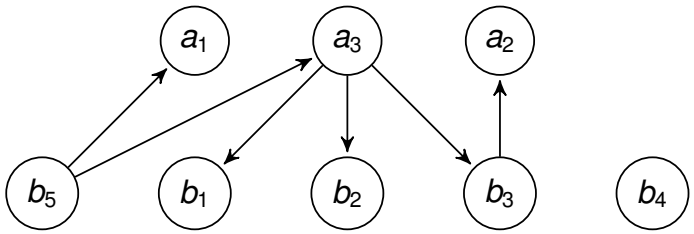

Attacks

# 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



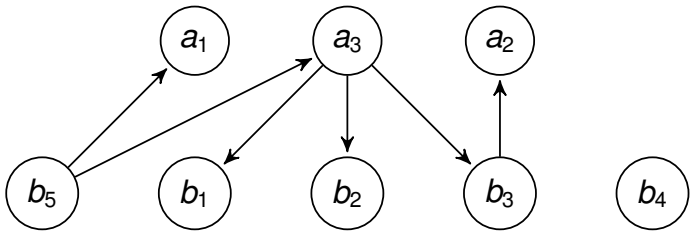

**Justified Argument Sets**

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

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**

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

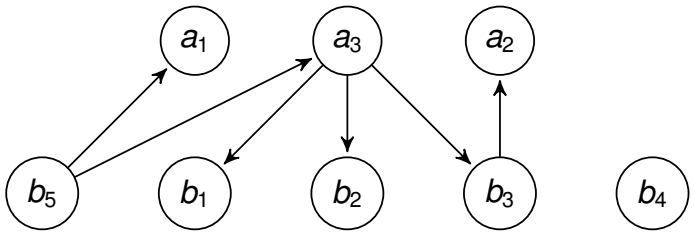
Credulous semantics allow for alternative argument sets

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

*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**

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

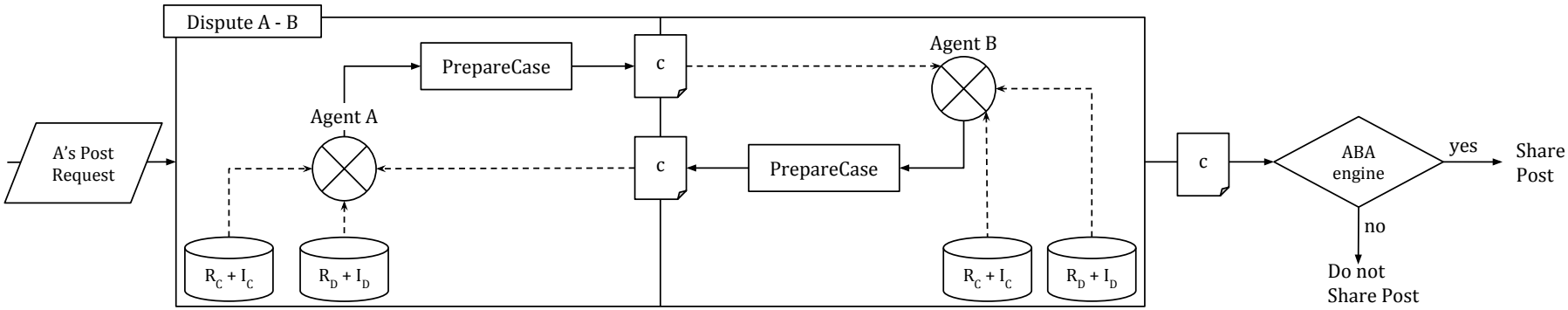
Credulous semantics allow for alternative argument sets

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

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

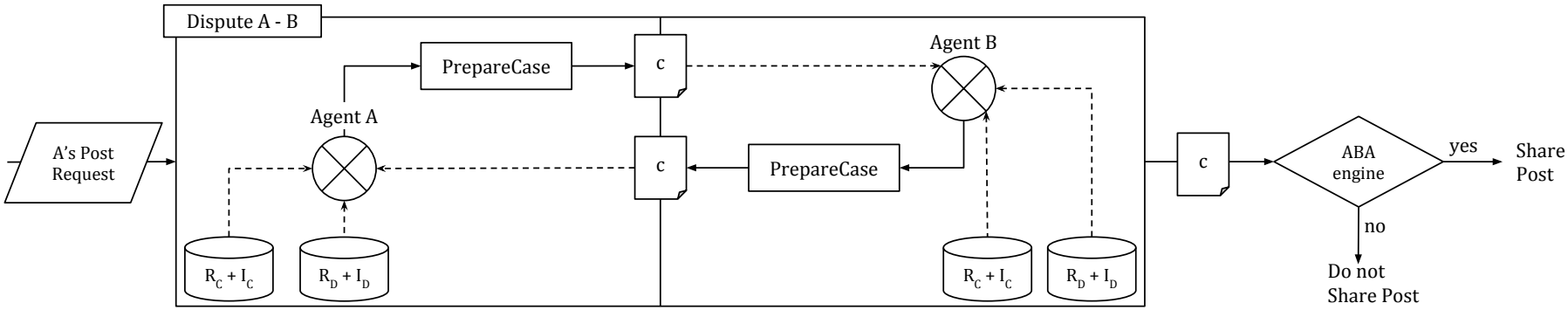
Distributed argumentation to create ABA specification in a turntaking fashion.

# 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



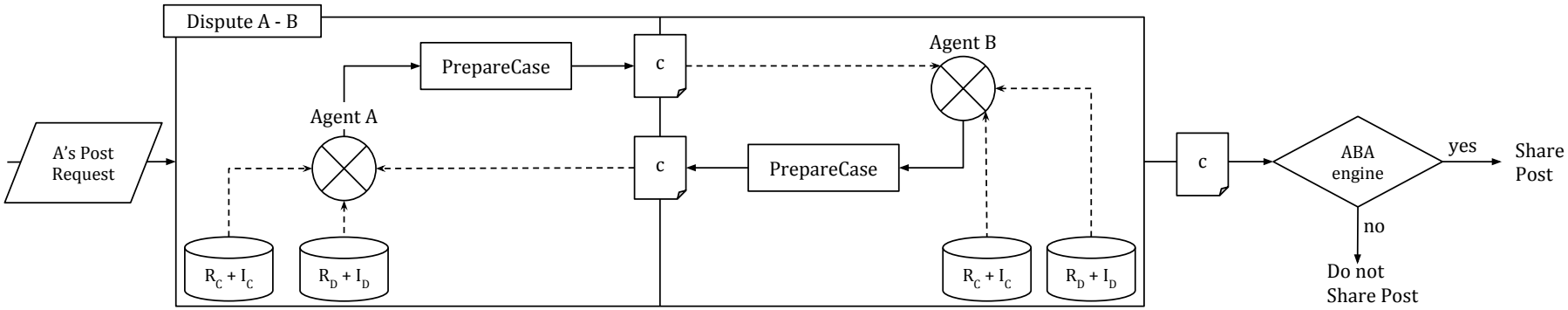

- A case is a tuple $\langle \mathcal{R}, \mathcal{A}, \mathcal{F}, \mathcal{C}, \mathit{status} \rangle$
- $R$ is a set of rules, $A$ is a set of assumptions, $F$ is a set of facts, $C$ is the assumption contrary mapping and *status* is either *ongoing* or *stop*.
- A case includes an ABA specification,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



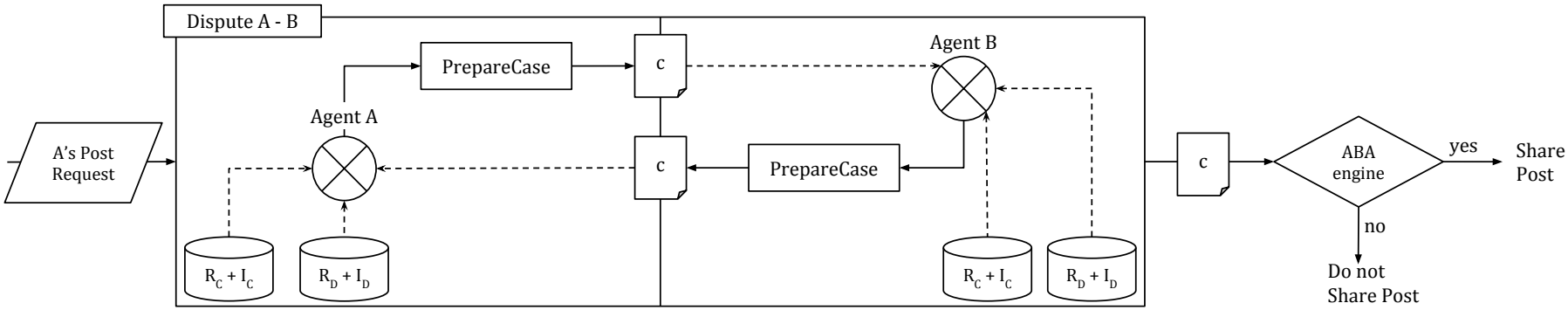

- A case is a tuple $\langle \mathcal{R}, \mathcal{A}, \mathcal{F}, \mathcal{C}, \mathit{status} \rangle$
- $R$ is a set of rules, $A$ is a set of assumptions, $F$ is a set of facts, $C$ is the assumption contrary mapping and *status* is either *ongoing* or *stop*.
- A case includes an ABA specification,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



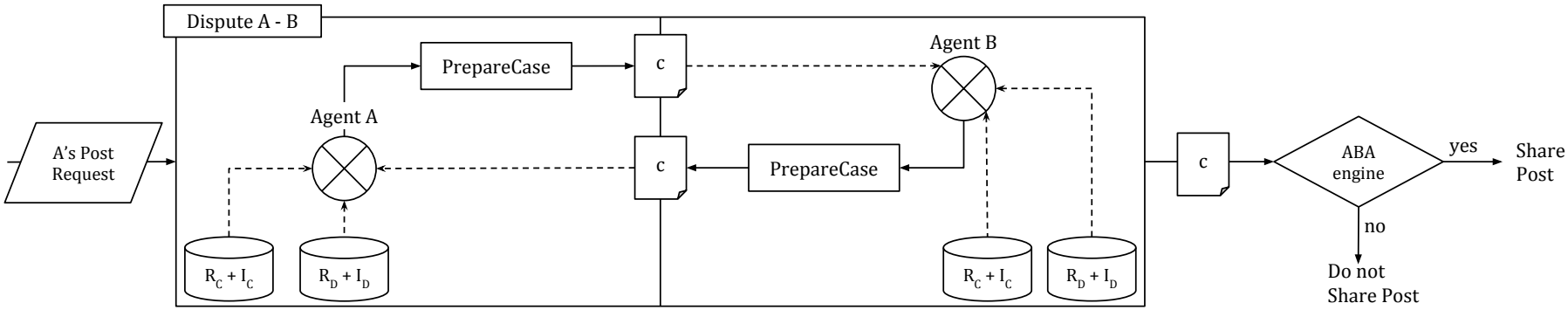

- A case is a tuple $\langle \mathcal{R}, \mathcal{A}, \mathcal{F}, \mathcal{C}, \mathit{status} \rangle$
- $R$ is a set of rules, $A$ is a set of assumptions, $F$ is a set of facts, $C$ is the assumption contrary mapping and *status* is either *ongoing* or *stop*.
- A case includes an ABA specification,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

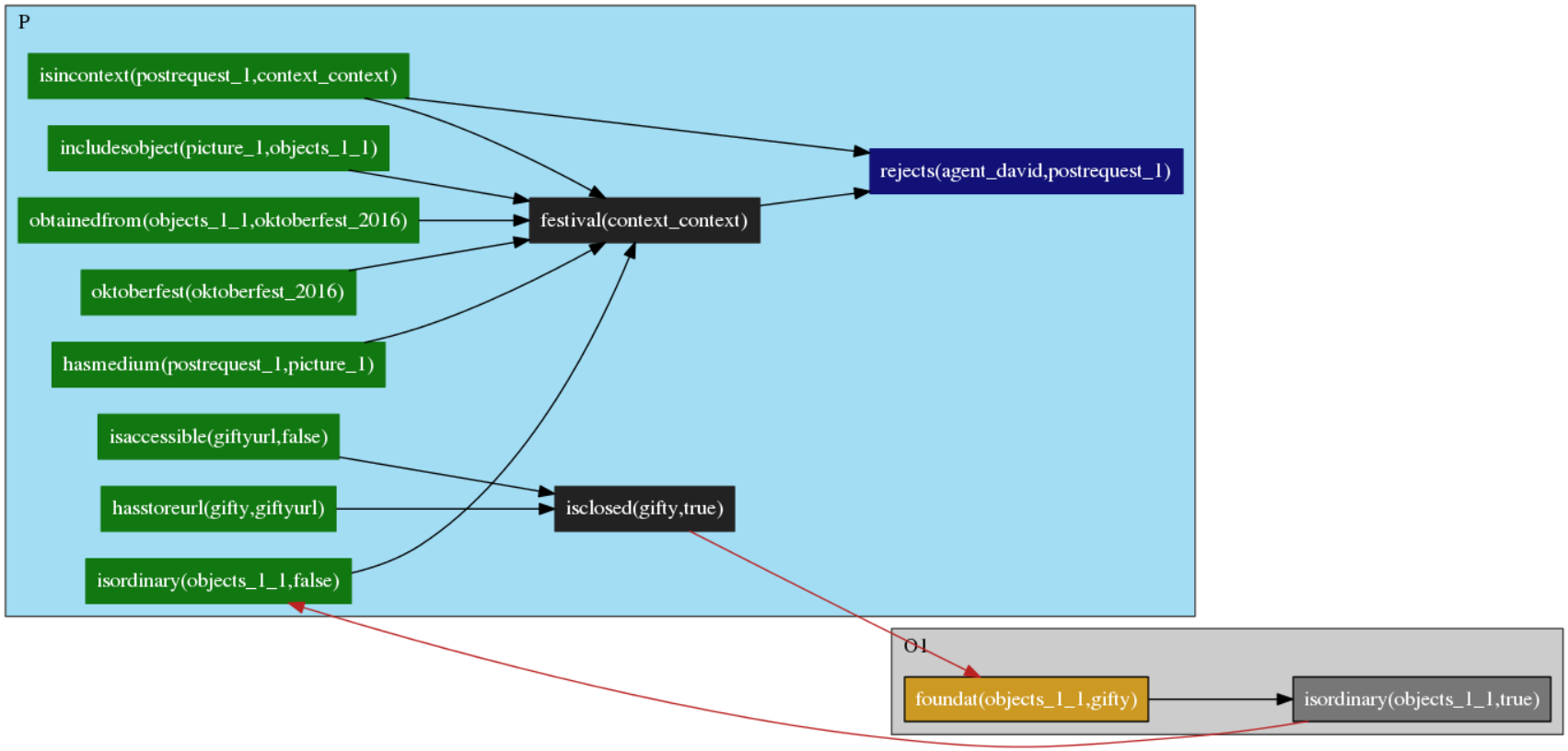
# Evaluation

- 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[3]

- 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

[3] 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[4]
  - 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[5]
  - 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

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

[5] 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[6]
  - 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

[6] 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