# Resource Access Control in the Facebook Model

K. Chronopoulos[1] M. Gouseti[1] A. Kiayias[2]

[1]University of Amsterdam, The Netherlands

[2]Department of Informatics & Telecommunications
University of Athens, Greece

The 12th International Conference on Cryptology and Network Security, 2013

# Table of Contents


1. Resource Access Control In Social Networks
   - Motivation
   - Related Work
2. RACS Formal Model
   - Protocols
   - Properties
3. Facebook
   - Protocols
   - Attacks
   - How to fix it

# Table of Contents


1. Resource Access Control In Social Networks
   - Motivation
   - Related Work
2. RACS Formal Model
   - Protocols
   - Properties
3. Facebook
   - Protocols
   - Attacks
   - How to fix it

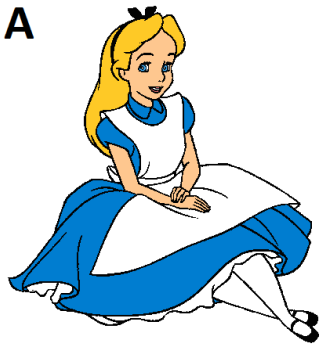
# Motivation

## Formal model?

**Owners**

Server

Clients

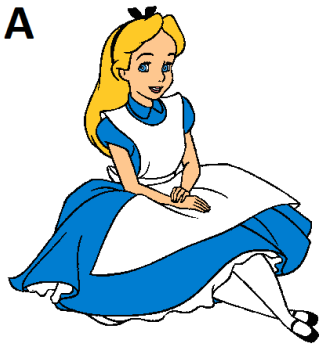



Owners' Resources

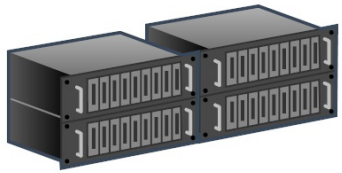
# Motivation

## Formal model?

**Owners**




**Server**



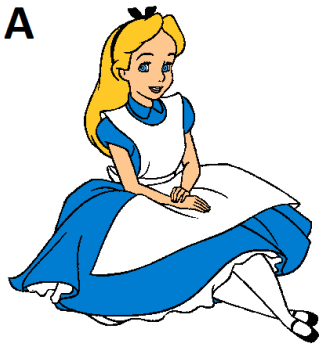
Owners' Resources

Clients

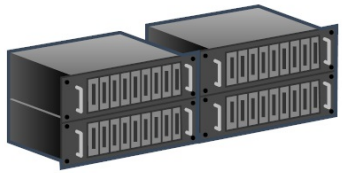
# Motivation

## Formal model?

**Owners**




**Server**



**Owners' Resources**

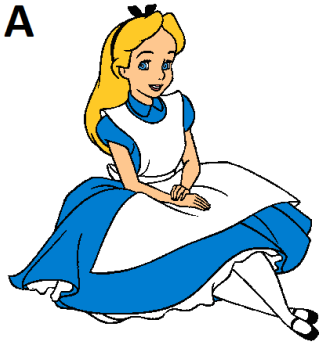

Clients

# Motivation

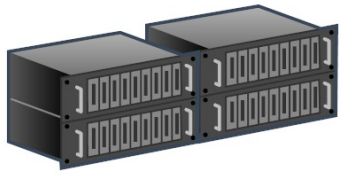
## Formal model?

**Owners**

A

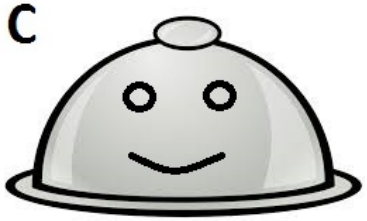


**Server**



**Clients**

C



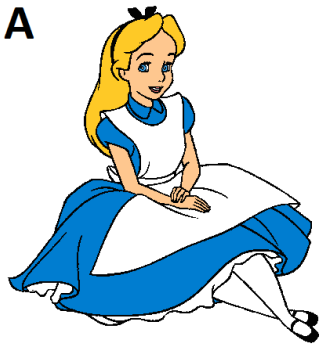
**Owners' Resources**

# Motivation

## Formal model?

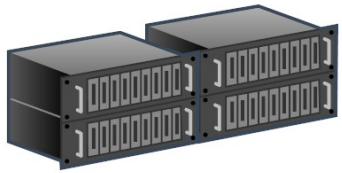
**Owners**

A



B

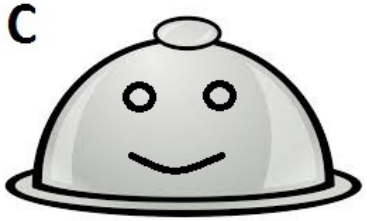
**Server**



**Owners' Resources**


**Clients**

C

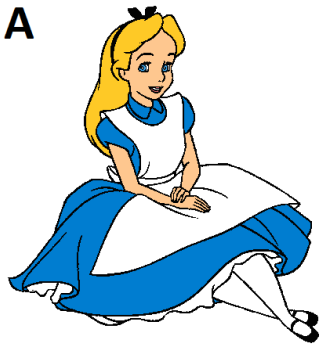


# Motivation

## Formal model?

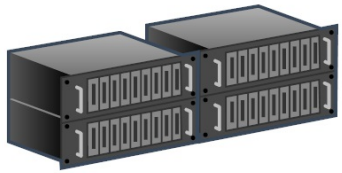
**Owners**

A



B

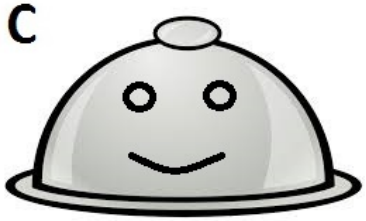
**Server**



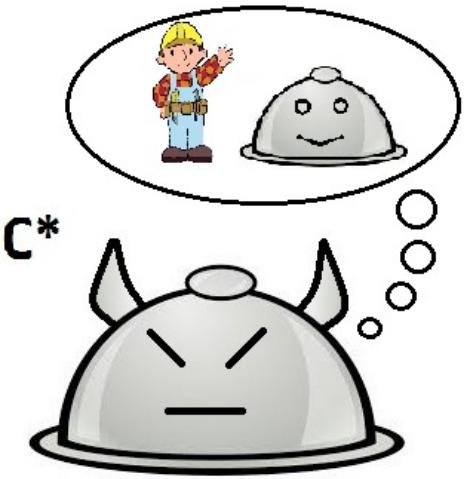
**Owners' Resources**


**Clients**

C



C*



# Table of Contents


1. Resource Access Control In Social Networks
   - Motivation
   - Related Work
2. RACS Formal Model
   - Protocols
   - Properties
3. Facebook
   - Protocols
   - Attacks
   - How to fix it

# Related Work

Previous work includes:

- Security analysis of OAuth
- Resources access control in social networks
  - Expression access control directives
  - Privacy in a untrusted server setting

Our work:

- Define a formal model of social networks in a trusted server setting
- Analyse its security properties

# Table of Contents


1. Resource Access Control In Social Networks
   - Motivation
   - Related Work
2. RACS Formal Model
   - Protocols
   - Properties
3. Facebook
   - Protocols
   - Attacks
   - How to fix it

# Table of Contents


1. Resource Access Control In Social Networks
   - Motivation
   - Related Work
2. RACS Formal Model
   - Protocols
   - Properties
3. Facebook
   - Protocols
   - Attacks
   - How to fix it

# Interactions with the Server

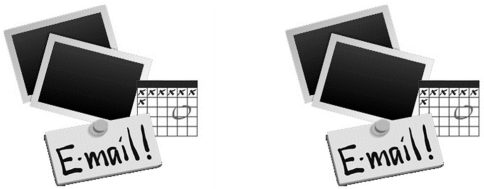
**Owners:**

- Register.
- Authenticate.
- Make connections with other owners.
- Break a connection.
- Authorize clients.
- Use the clients' services.
- Revoke client's authorization.

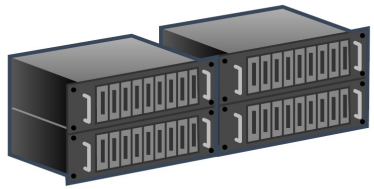
**Clients**

- Register.
- Authenticate.
- Access resources.

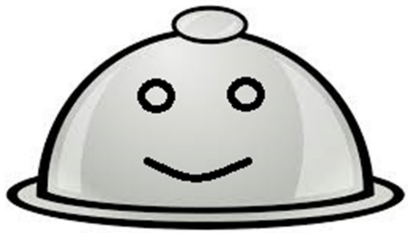
# Client's Authorization



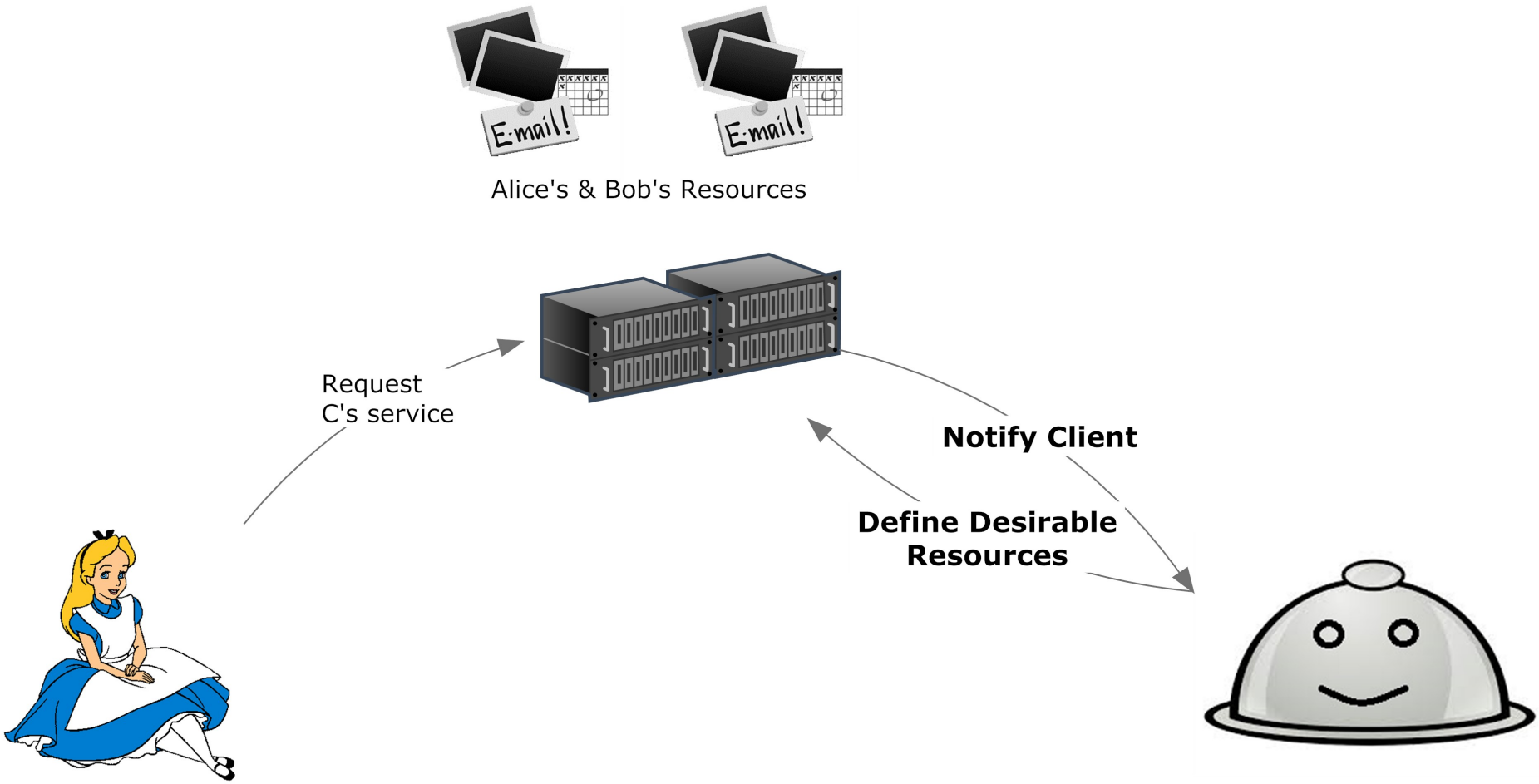

Alice's & Bob's Resources



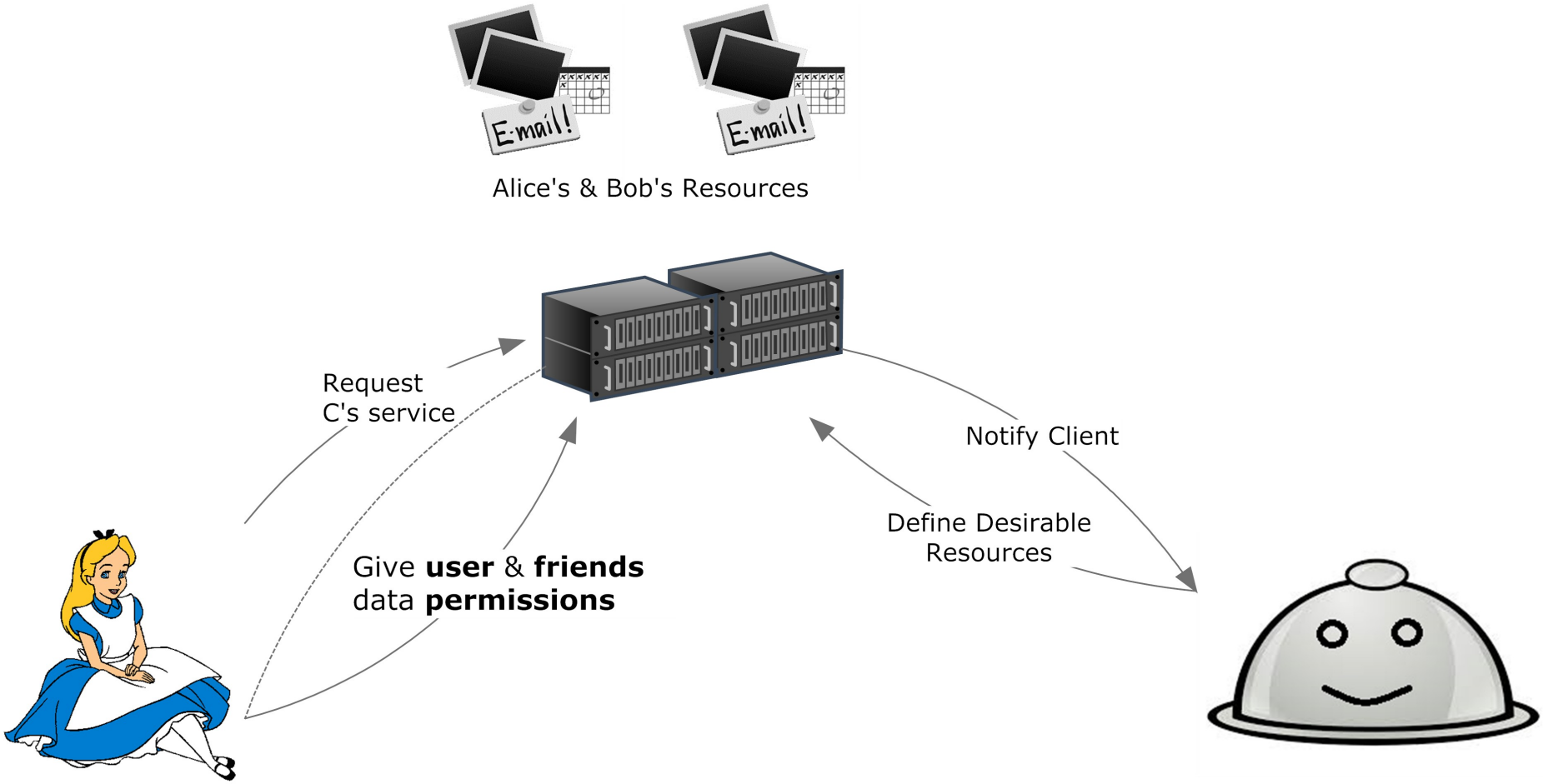
**Request**
**C's service**



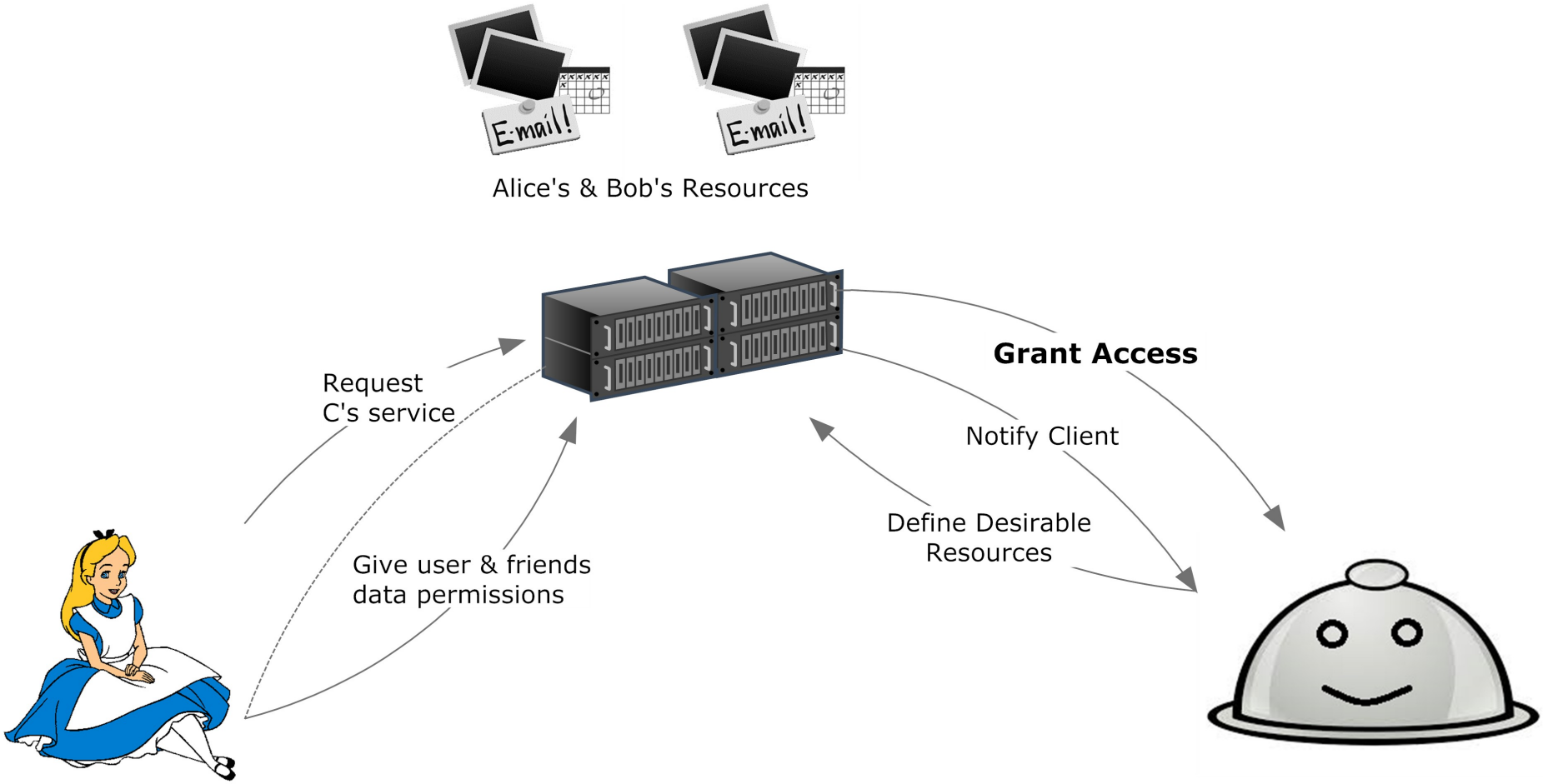
# Client's Authorization



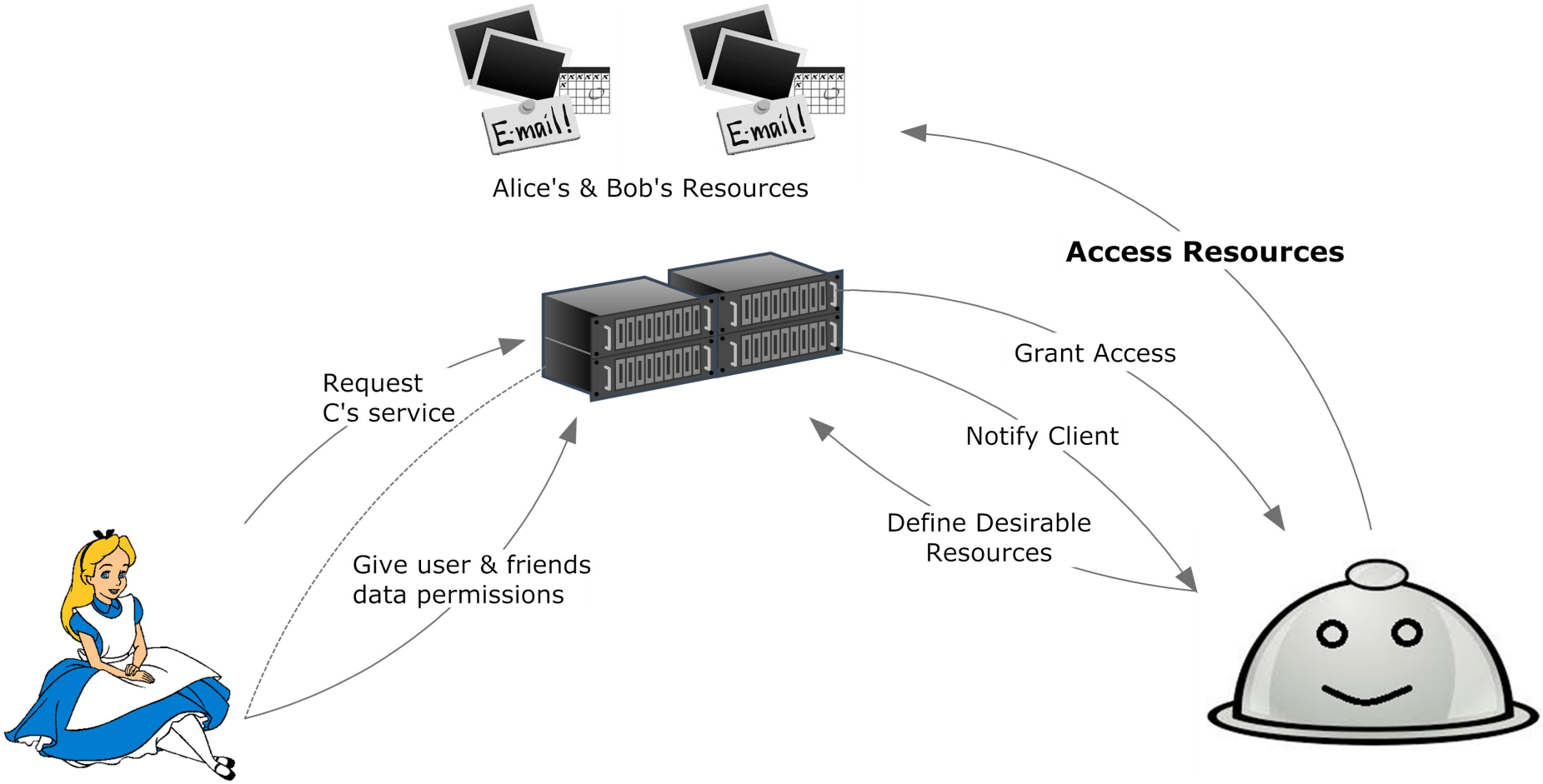
# Client's Authorization



# Client's Authorization



# Client's Authorization



# Access Alice's Resources

- **Direct access**
  When Alice has given the client **user permission** to access her resources.
- **Access through Bob**
  When Bob has given the client **friend permission** to access the resources of Alice that are **visible** to him.

# 2 Modes of Revocation

- **Explicit Revocation**
  Alice can revoke a client's access by explicitly instructing the server.
- **Implicit Revocation**
  The Facebook model suggests that a client's access should be revoked if an owner has not used its services after a certain time period ($dt$ units of time).

> If you haven't used an app in a while, it won't be able to continue to update the additional information you've given them permission to access.
>
> — Facebook, Data Use Policy

# Table of Contents


1. Resource Access Control In Social Networks
   - Motivation
   - Related Work
2. RACS Formal Model
   - Protocols
   - Properties
3. Facebook
   - Protocols
   - Attacks
   - How to fix it

# Notation

We will use the following notation:

- $O$, $C$: unique id that identifies owners and clients respectively.
- $f$: projection ($D^n \rightarrow D^k$) where $k \leq n$ and $D$ is the space of the owner's resources. Also used as a set of indices.
- `oos_ac()`, `ocs_ac()`, `ocg_ac()`, `expt()`, $r()$: server's matrices
- $\lambda$: level of security associated with our proposed solutions

# Correctness

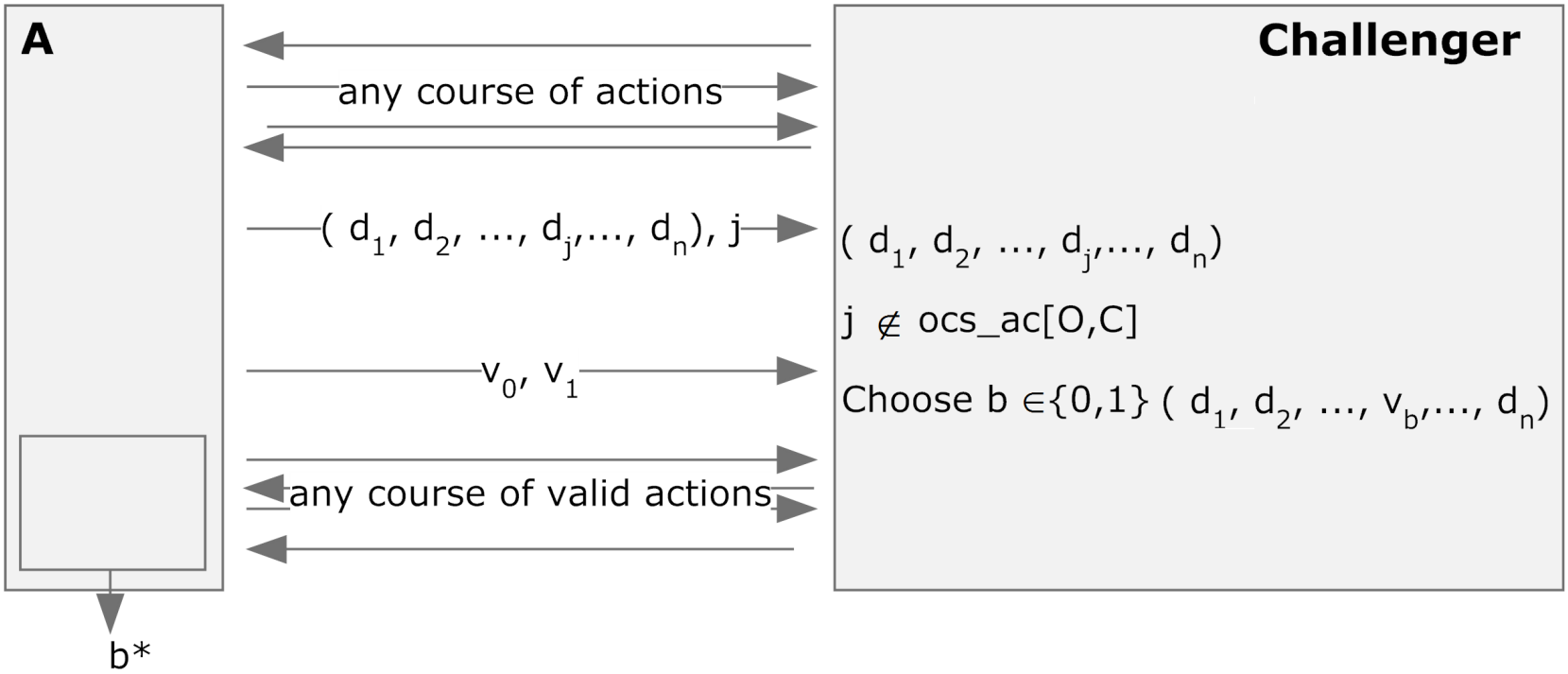
## Definition

For all $O$, $O' \neq O$, $C$, $f : D^n \rightarrow D^k$ where $k \leq n$, if

$$\Big((f \subseteq \texttt{ocs\_ac}[O, C]) \wedge (\mathit{server\_time} < \texttt{expt}[O, C])\Big) \vee$$
$$\Big((f \subseteq (\texttt{ocg\_ac}[O', C] \cap \texttt{oos\_ac}[O, O'])) \wedge (\mathit{server\_time} < \texttt{expt}[O', C])\Big),$$

then $C$, by running the "Client Access Resources Protocol", will receive the resources $f(\texttt{r}[O])$ and the server will record the action $\mathrm{access\_resources}(C, O, f)$.

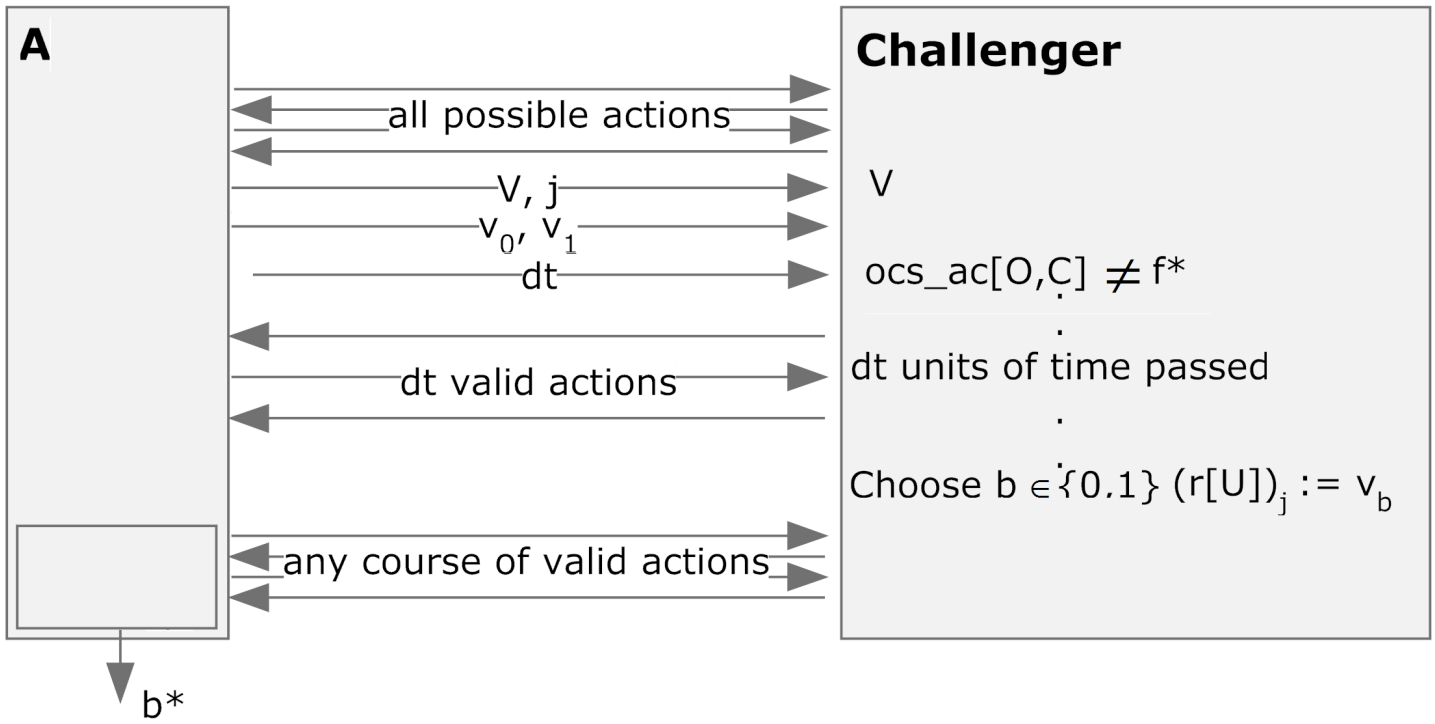
# Owner Privacy - Explicit Revocation




## Definition

For all PPT adversaries $A$, $Pr[WIN^A] = \frac{1}{2} + negl(\lambda)$, where $WIN^A$ is the event $b = b^*$ while playing the above game.

# Owner Privacy - Implicit Revocation




## Definition

For all PPT adversaries $A$, $Pr[WIN^A] = \frac{1}{2} + negl(\lambda)$, where $WIN^A$ is the event $b = b^*$ while playing the above game.

# Server Consistency

We define a predicate $P(\texttt{log\_file}, dt)$ that is true when the server can justify a resource access, i.e:

1. $\text{authenticate}(O), t_0$
2. $\text{authorize\_client}(O, C, f_s, f_g), t_1,$
3. any of $\text{authenticate}(O)$ or $\text{use}(O, C), t_2$
4. $\text{authenticate}(C) t_3$
5. $\text{access\_resources}(C, O, f_s'), t_4$ where $f_s' \subseteq f_s \wedge (t_4 - t_{1,2}) < dt$

## Definition

For all PPT adversaries $A$, $Pr[P(\texttt{log\_file}, dt) = 0] = negl(\lambda)$, where $\texttt{log\_file}$ is a random variable that reflects the log file given the activity of $A$ as described above.

# Table of Contents


1. Resource Access Control In Social Networks
   - Motivation
   - Related Work
2. RACS Formal Model
   - Protocols
   - Properties
3. Facebook
   - Protocols
   - Attacks
   - How to fix it

# Table of Contents


1. Resource Access Control In Social Networks
   - Motivation
   - Related Work
2. RACS Formal Model
   - Protocols
   - Properties
3. Facebook
   - Protocols
   - Attacks
   - How to fix it

# Client Access Resources Protocol (part 1)

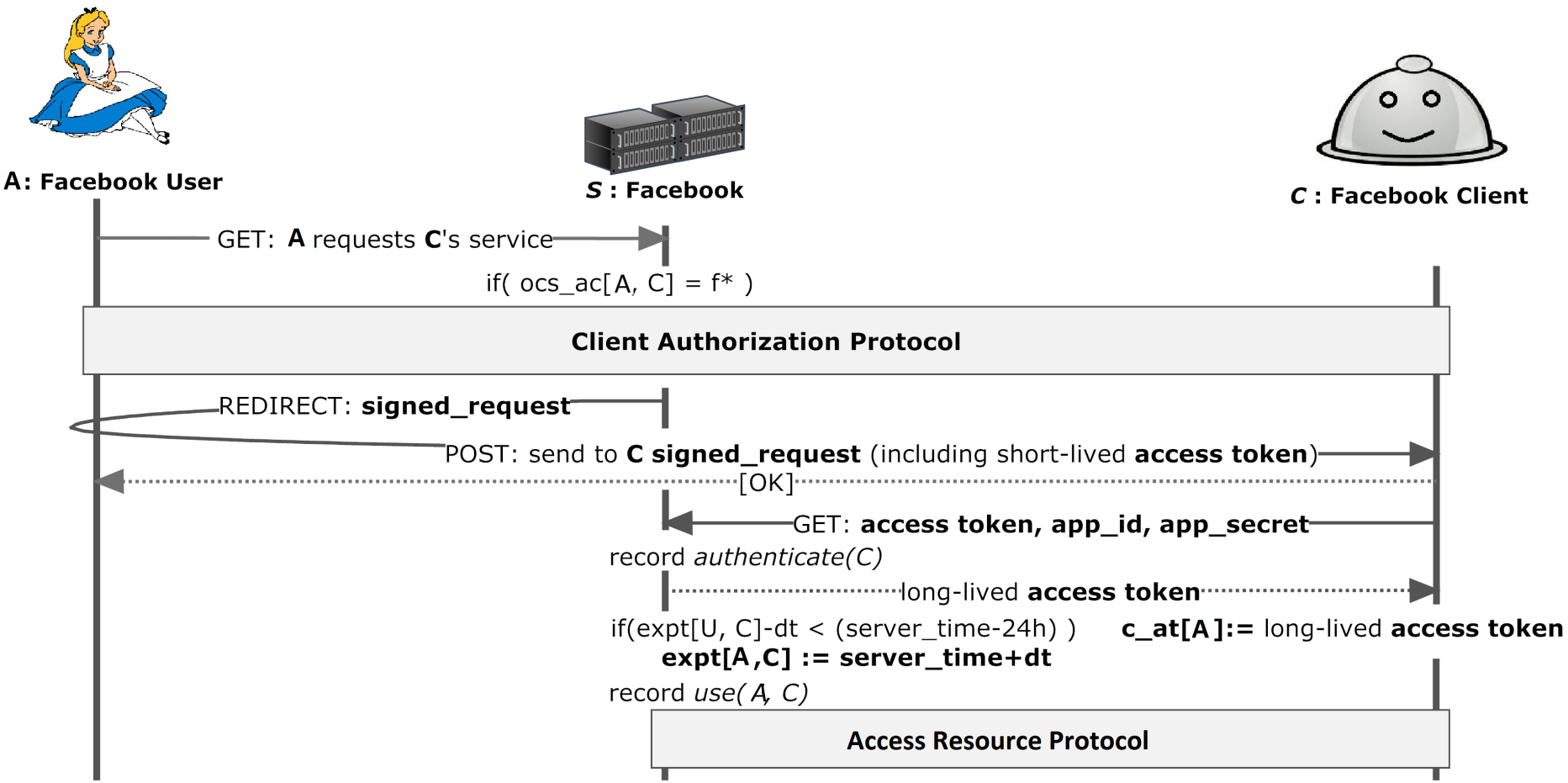



Figure : Only when the protocol is initiated by a user, i.e. Alice, the authorization protocol can be executed.

# Client Access Resources Protocol (part 2.1)

## Direct Access

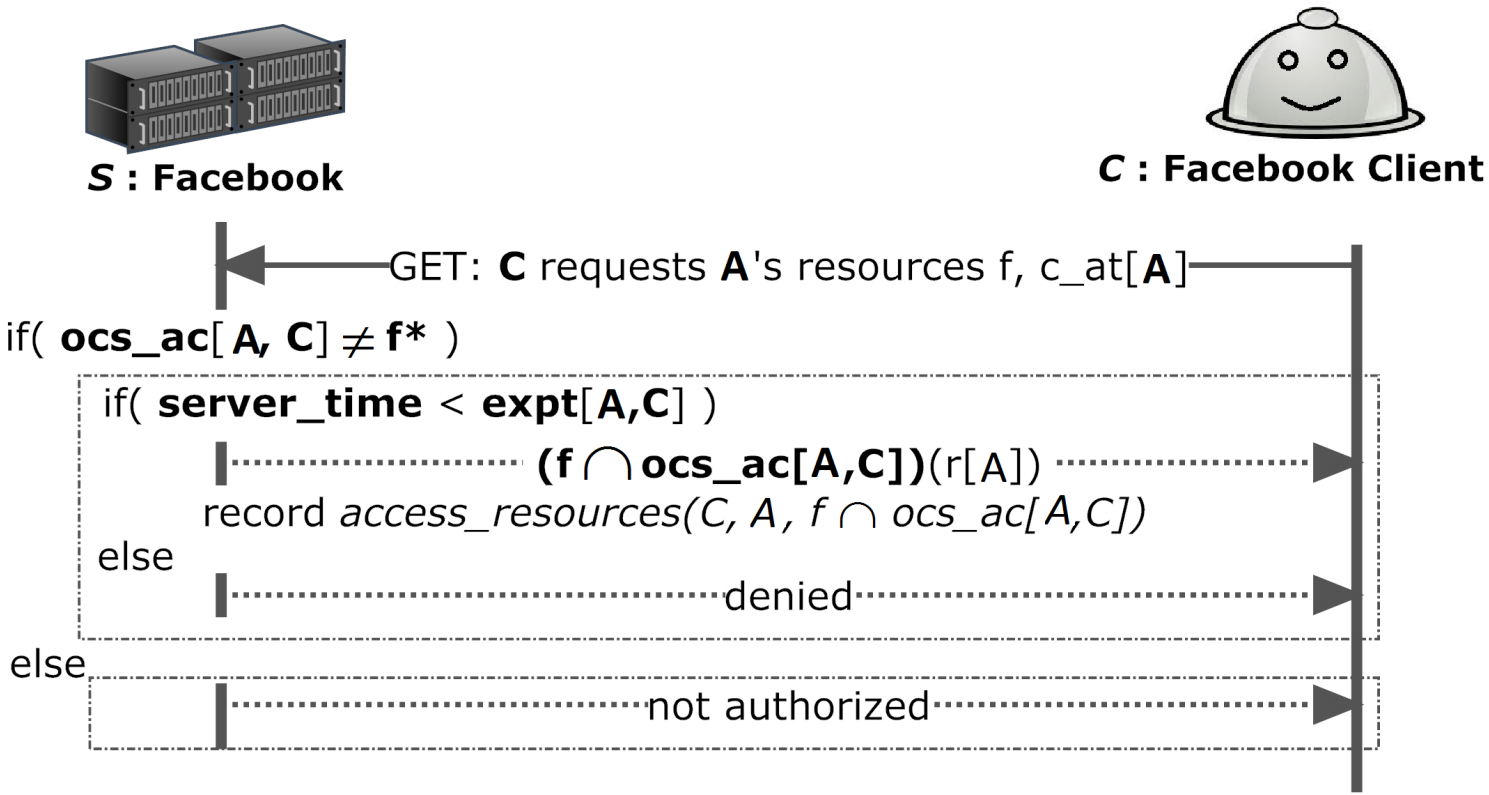



Figure : $C$ accesses Alice's resources using her access token.

# Client Access Resources Protocol (part 2.2)

## Indirect Access

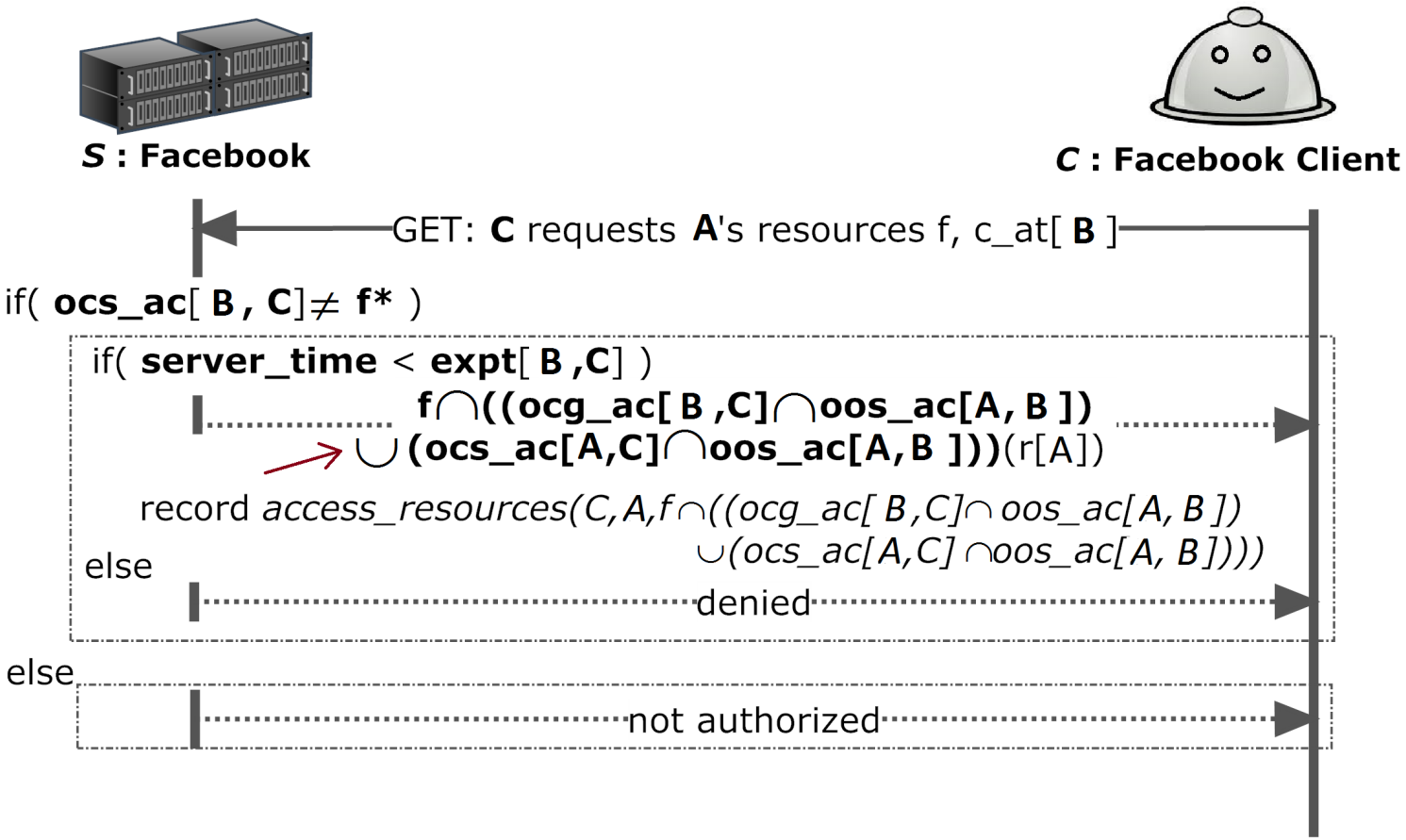



Figure : $C$ accesses Alice's resources using Bob's access token.

# Table of Contents


1. Resource Access Control In Social Networks
   - Motivation
   - Related Work
2. RACS Formal Model
   - Protocols
   - Properties
3. Facebook
   - Protocols
   - Attacks
   - How to fix it

# Owner Privacy with Implicit Revocation

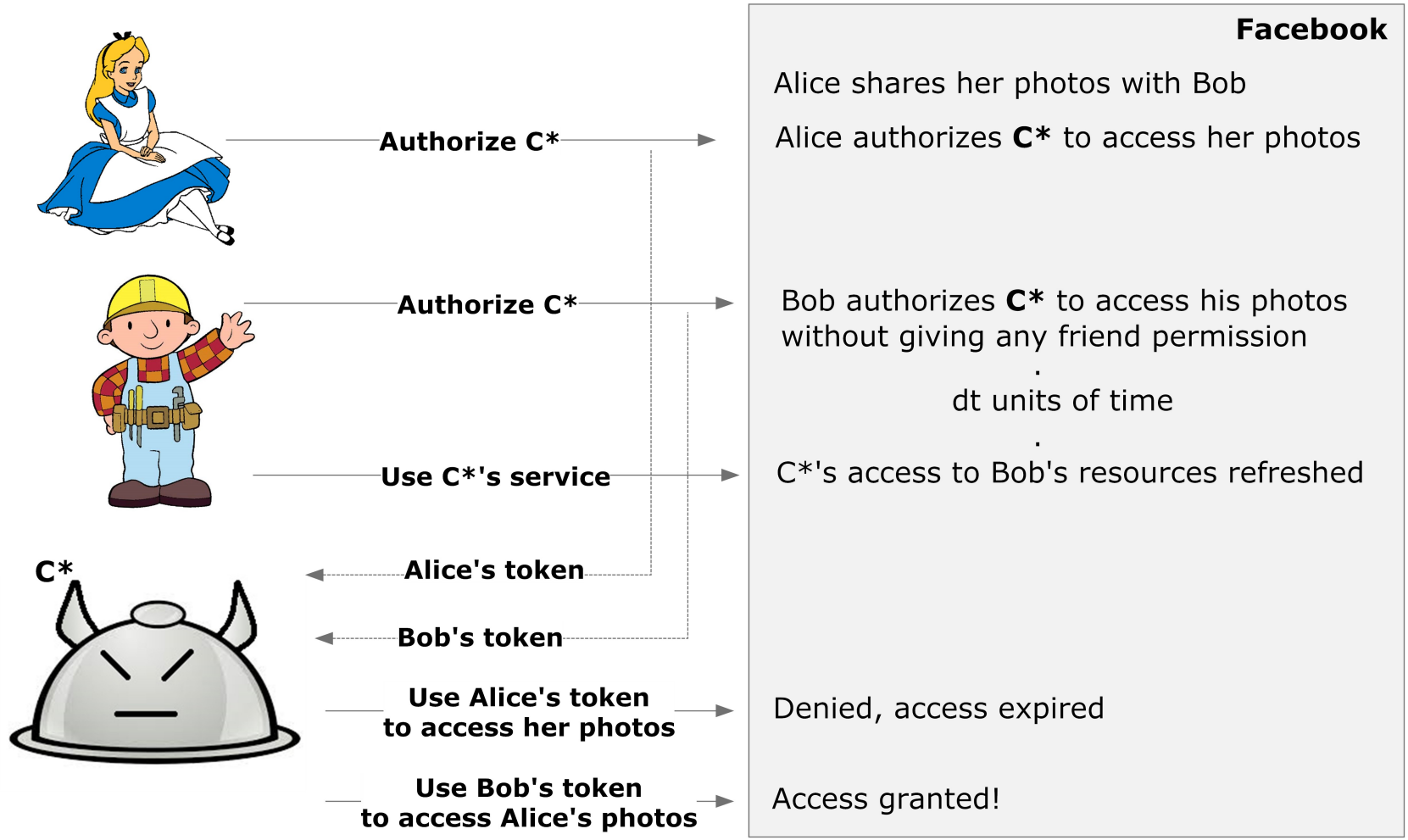
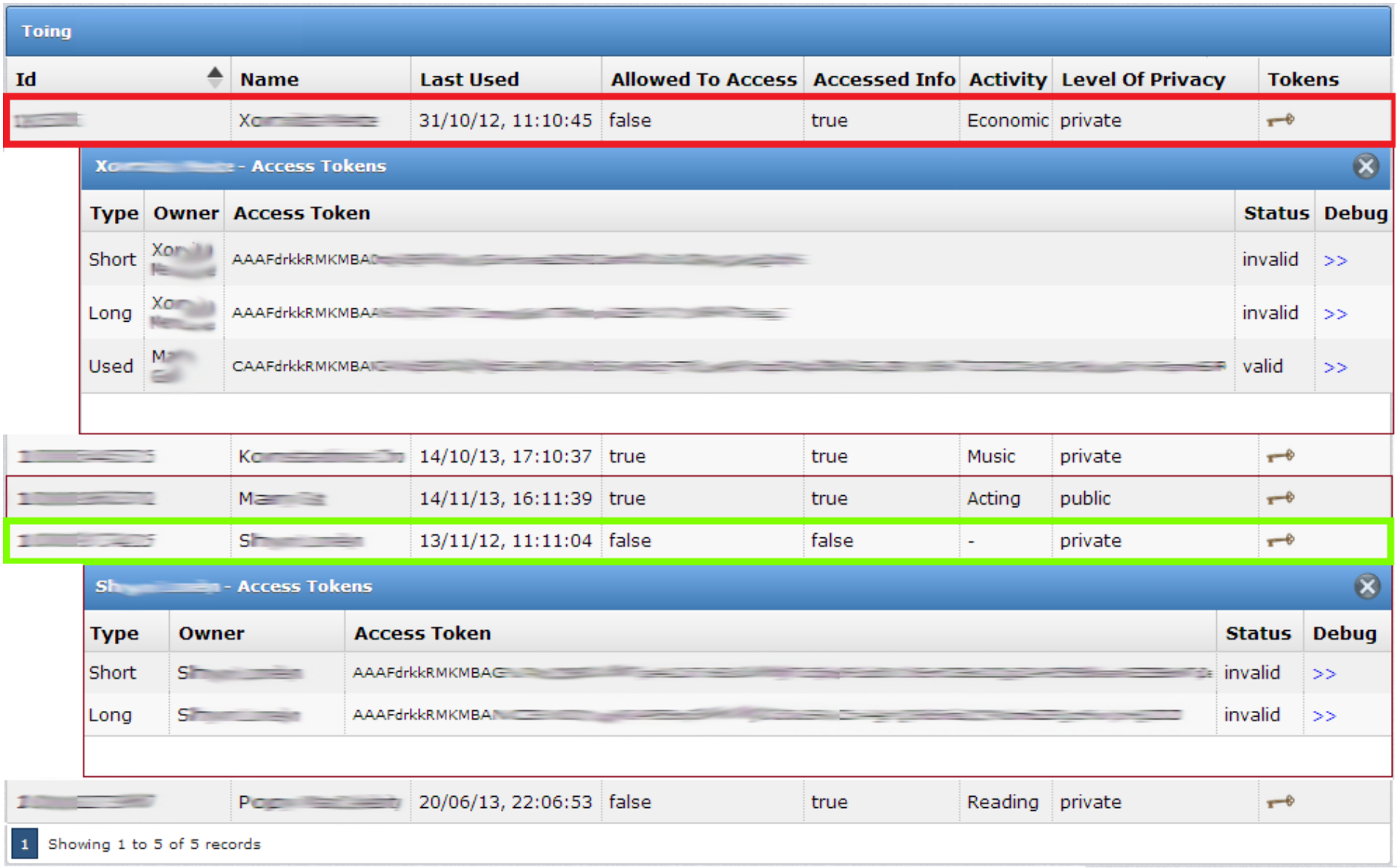



Figure : $C^*$ can access Alice's photos using Bob's token even if its access has expired.

# Owner Privacy with Implicit Revocation



Toing

| Id | Name | Last Used | Allowed To Access | Accessed Info | Activity | Level Of Privacy | Tokens |
|---|---|---|---|---|---|---|---|
| [redacted] | Xc[redacted] | 31/10/12, 11:10:45 | false | true | Economic | private | |

Xc[redacted] - Access Tokens

| Type | Owner | Access Token | Status | Debug |
|---|---|---|---|---|
| Short | Xo[redacted] | AAAFdrkkRMKMBAC[redacted] | invalid | >> |
| Long | Xo[redacted] | AAAFdrkkRMKMBAA[redacted] | invalid | >> |
| Used | M[redacted] | CAAFdrkkRMKMBAI[redacted] | valid | >> |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| [redacted] | Kc[redacted] | 14/10/13, 17:10:37 | true | true | Music | private | |
| [redacted] | M[redacted] | 14/11/13, 16:11:39 | true | true | Acting | public | |
| [redacted] | Sh[redacted] | 13/11/12, 11:11:04 | false | false | - | private | |

Sh[redacted] - Access Tokens

| Type | Owner | Access Token | Status | Debug |
|---|---|---|---|---|
| Short | Sh[redacted] | AAAFdrkkRMKMBAG[redacted] | invalid | >> |
| Long | Sh[redacted] | AAAFdrkkRMKMBAI[redacted] | invalid | >> |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| [redacted] | Pc[redacted] | 20/06/13, 22:06:53 | false | true | Reading | private | |

1 Showing 1 to 5 of 5 records

# Server Consistency

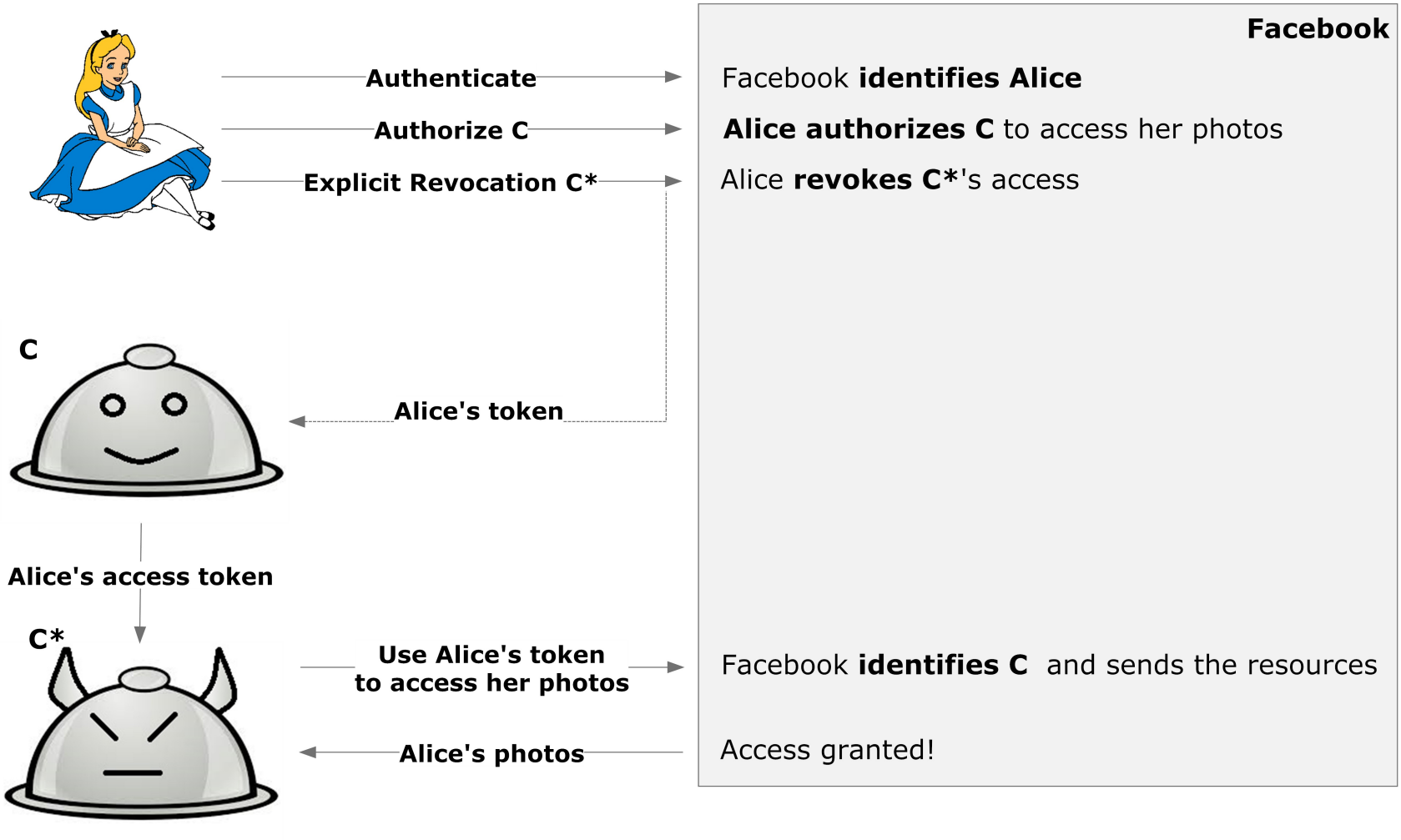



Figure : Inconsistency between Facebook's view and reality. Facebook has recorded that the resources were accessed by $C$ while they were accessed by $C^*$.

# Table of Contents


1. Resource Access Control In Social Networks
   - Motivation
   - Related Work
2. RACS Formal Model
   - Protocols
   - Properties
3. Facebook
   - Protocols
   - Attacks
   - How to fix it

# How to fix it

- **Owner Privacy with Implicit Revocation**
  When $C$ requests Alice's resources using Bob's access token, Facebook should respond with the intersection of Alice's resources that Bob can access and the *friends data permissions* that Bob has given to $C$ i.e. $(\mathtt{oos\_ac}[\mathit{Alice}, \mathit{Bob}] \cap \mathtt{ocg\_ac}[\mathit{Bob}, C])$.

# How to fix it

- **Server Consistency**
  Various ways, Facebook can:
  - Support sign in functionality for applications.
  - Filter IP address of an access resource request.
  - Request that the client signs the token and a random value with its app_secret.

# Questions?


This work was performed while at the National and Kapodistrian University of Athens. Research partly supported by ERC project CODAMODA.