Clark-Wilson Policies in ACP: Controlling Information Flow Between Solid Apps

SoSy'24: Privacy @ the Solid Symposium, May 2-3, 2024, Leuven, Belgium

Ellie Forsyth and Ross Horne

University of Strathclyde, Glasgow

Towards a policy model for Solid

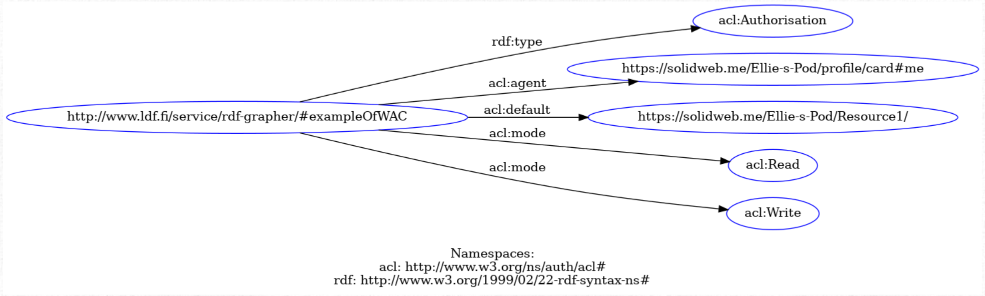
This is WAC, a language for describing access control rules:

```
@prefix acl: <http://www.w3.org/ns/auth/acl#>

<#exampleOfWAC>
    a acl:Authorisation;
    acl:agent <https://solidweb.me/Ellie-s-Pod/profile/card#me>;
    acl:default <https://solidweb.me/Ellie-s-Pod/Resource1/>;
    acl:mode acl:Read, acl:Write.
```

Towards a policy model for Solid

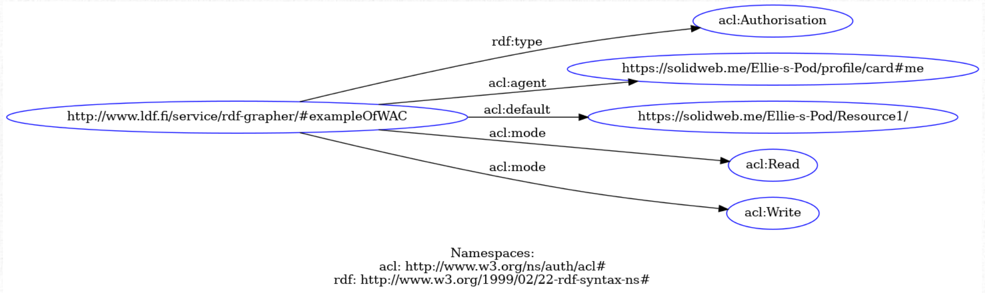
This is WAC, a language for describing access control rules:



http://www.ldf.fi/service/rdf-grapher/#exampleOfWAC
rdf:type
acl:Authorisation
acl:agent
https://solidweb.me/Ellie-s-Pod/profile/card#me
acl:default
https://solidweb.me/Ellie-s-Pod/Resource1/
acl:mode
acl:Read
acl:mode
acl:Write
Namespaces:
acl: http://www.w3.org/ns/auth/acl#
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#

Towards a policy model for Solid

This is WAC, a language for describing access control rules:


http://www.ldf.fi/service/rdf-grapher/#exampleOfWAC
rdf:type
acl:Authorisation
acl:agent
https://solidweb.me/Ellie-s-Pod/profile/card#me
acl:default
https://solidweb.me/Ellie-s-Pod/Resource1/
acl:mode
acl:Read
acl:mode
acl:Write
Namespaces:
acl: http://www.w3.org/ns/auth/acl#
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#

Research questions:

- Is WAC enough to describe policies with adequate security guarantees?
- What **policy model** suits Solid for determining when a policy is secure?

Client constraints

“The Authorization Panel is undertaking the following initiatives, in priority order:

1. Document use cases and requirements for authorization.
2. Produce an authorization system specification to satisfy those use cases and requirements.
3. Propose mechanism(s) for **client constraints**.”

Client constraints

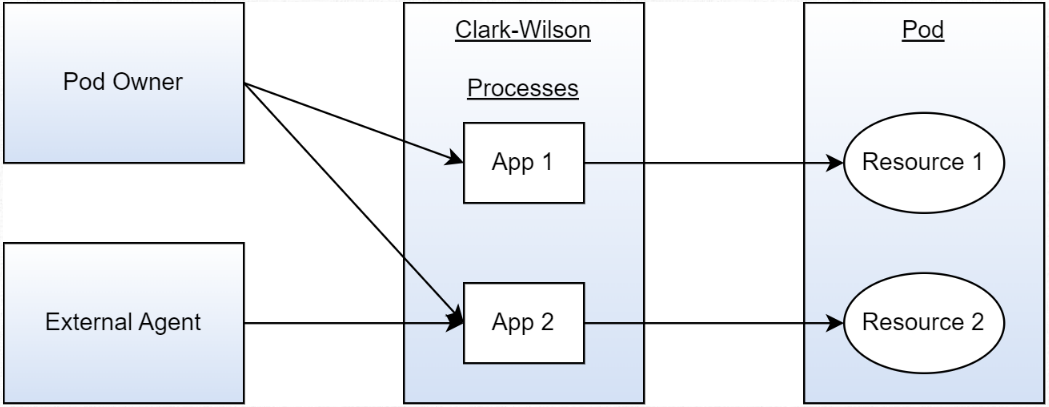
"The Authorization Panel is undertaking the following initiatives, in priority order:

1. Document use cases and requirements for authorization.
2. Produce an authorization system specification to satisfy those use cases and requirements.
3. Propose mechanism(s) for **client constraints**."

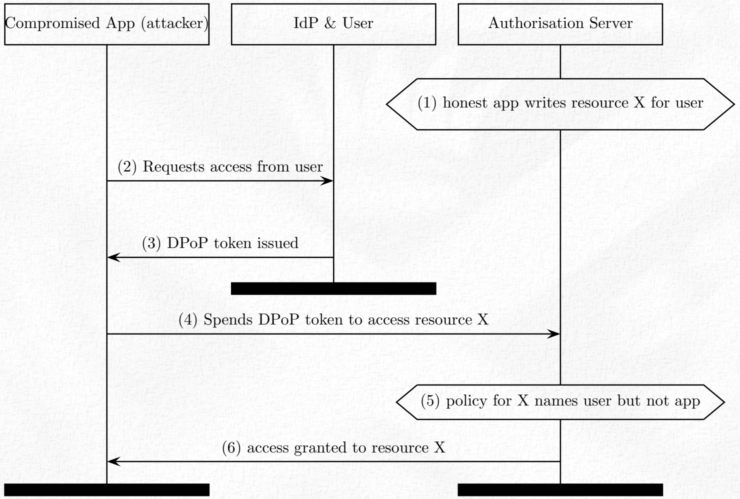
Requirements transcending Solid use cases:

- Different entities run apps and pods.
- Different entities run apps that connect to the same pod.
- Entities are not mutually trustworthy (conflicts-of-interest, exposure to cyber attacks, etc.).

Enterprise policy models since 1987



No client constraints; no confidentiality


Compromised App (attacker)
IdP & User
Authorisation Server
(1) honest app writes resource X for user
(2) Requests access from user
(3) DPoP token issued
(4) Spends DPoP token to access resource X
(5) policy for X names user but not app
(6) access granted to resource X

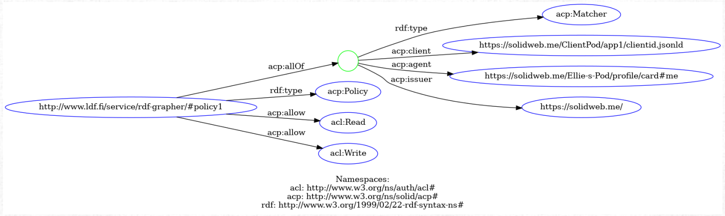
Clark-Wilson in ACP

```
@prefix acl: <http://www.w3.org/ns/auth/acl#> .
@prefix acp: <http://www.w3.org/ns/solid/acp#> .

<#exampleOfACP_1>
  a acp:AccessControlResource;
  acp:resource <https://solidweb.me/Ellie-s-Pod/Resource1/>;
  acp:accessControl <#ownerAccess1>;
  acp:memberAccessControl <#ownerAccess1>.

<#ownerAccess1>
  a acp:AccessControl;
  acp:apply [
    a acp:Policy;
    acp:allow acl:Read, acl:Write;
    acp:allOf [
            a acp:Matcher;
            acp:client <https://solidweb.me/ClientPod/app1/clientid.jsonld>;
            acp:agent <https://solidweb.me/Ellie-s-Pod/profile/card#me>;
            acp:issuer <https://solidweb.me/> ] ].
```

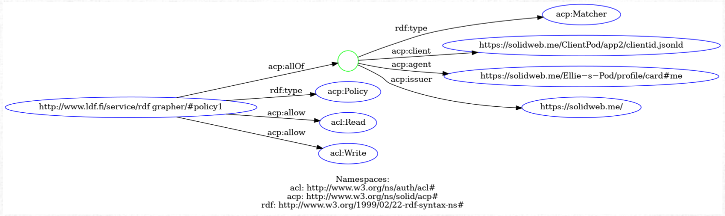
Clark-Wilson in ACP


acp:Matcher
rdf:type
https://solidweb.me/ClientPod/app1/clientid.jsonld
acp:client
acp:allOf
acp:agent
https://solidweb.me/Ellie-s-Pod/profile/card#me
acp:issuer
rdf:type
acp:Policy
http://www.ldf.fi/service/rdf-grapher/#policy1
https://solidweb.me/
acp:allow
acl:Read
acp:allow
acl:Write
Namespaces:
acl: http://www.w3.org/ns/auth/acl#
acp: http://www.w3.org/ns/solid/acp#
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#

```
@prefix acl: <http://www.w3.org/ns/auth/acl#> .
@prefix acp: <http://www.w3.org/ns/solid/acp#> .

<#policy1>
    a acp:Policy;
    acp:allow acl:Read, acl:Write;
    acp:allOf [
            a acp:Matcher;
            acp:client <https://solidweb.me/ClientPod/app1/clientid.jsonld>;
            acp:agent <https://solidweb.me/Ellie-s-Pod/profile/card#me>;
            acp:issuer <https://solidweb.me/> ] .
```

Clark-Wilson in ACP


acp:Matcher
rdf:type
https://solidweb.me/ClientPod/app2/clientid.jsonld
acp:client
acp:allOf
acp:agent
https://solidweb.me/Ellie−s−Pod/profile/card#me
acp:issuer
rdf:type
acp:Policy
http://www.ldf.fi/service/rdf-grapher/#policy1
https://solidweb.me/
acp:allow
acl:Read
acp:allow
acl:Write
Namespaces:
acl: http://www.w3.org/ns/auth/acl#
acp: http://www.w3.org/ns/solid/acp#
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#

```
@prefix acl: <http://www.w3.org/ns/auth/acl#> .
@prefix acp: <http://www.w3.org/ns/solid/acp#> .

<#policy1>
    a acp:Policy;
    acp:allow acl:Read, acl:Write;
    acp:allOf [
        a acp:Matcher;
        acp:client <https://solidweb.me/ClientPod/app2/clientid.jsonld>;
        acp:agent <https://solidweb.me/Ellie−s−Pod/profile/card#me>;
        acp:issuer <https://solidweb.me/> ] .
```

Clark-Wilson in ACP

```
<#exampleOfACP_2>
  a acp:AccessControlResource;
  acp:resource <https://solidweb.me/Ellie-s-Pod/Resource2/>;
  acp:accessControl <#ownerAccess2>, <#externalAgent>;
  acp:memberAccessControl <#ownerAccess2>, <#externalAgent>.

<#ownerAccess2>
  a acp:AccessControl;
  acp:apply [
    a acp:Policy;
    acp:allow acl:Read, acl:Write;
    acp:allOf [
            a acp:Matcher;
            acp:client <https://solidweb.me/ClientPod/app2/clientid.jsonld>;
            acp:agent <https://solidweb.me/Ellie-s-Pod/profile/card#me>;
            acp:issuer <https://solidweb.me/> ] ].
```

Requirement: identify apps and name them in the policy.

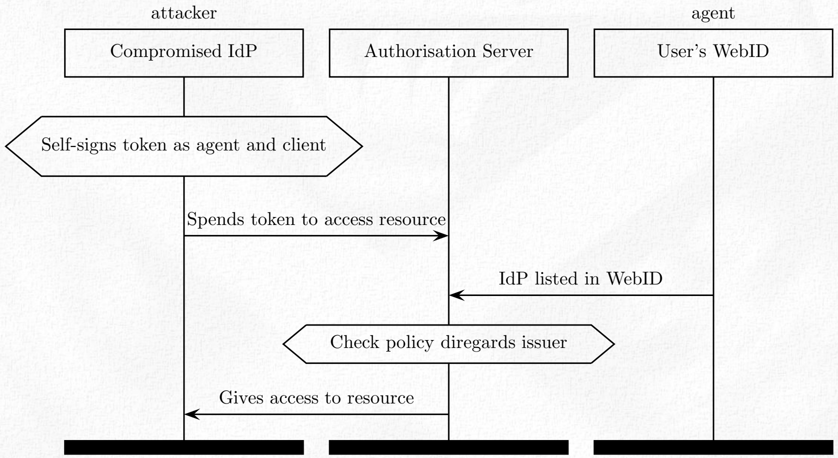
Policy model enforced by security app

```
<#root>
 a acp:AccessControlResource;
  acp:resource <./>;
  acp:accessControl <#secureApp>;
  acp:memberAccessControl <#secureApp>.

<#secureApp>
  a acp:AccessControl;
  acp:apply [
    a acp:Policy;
    acp:allow acl:Control;
    acp:allOf [
      a acp:Matcher;
      acp:client <https://solidweb.me/ClientPod/demoApp/clientid.jsonld>;
      acp:agent <https://solidweb.me/Ellie-s-Pod/profile/card#me>;
      acp:issuer <http://localhost:3000/>
    ]
  ].
```

Policy: entity of pod entrusts entity providing security app.

Attack without `acp:issuer`



Challenges

Cyber-resillience: isolate cyber risks by filtering access by organisational boundaries? Draw inspiration from Clark-Wilson, Android `developer`, `same-origin`, etc.

Lattice-based policy model:

- Explicit *confidentiality* and *integrity* goals.
- *Conflicts-of-interests* between entities (Brewer-Nash).
- *Sanitized data* flows freely between entities permited to access pod.
- *Dynamics:* pod (and security app) validates state transitions (e.g., don't give more than `acl:Control` to security app, require consent to enable new accesses, etc.)

Community effort: specify **policy models** for Solid with guidelines for use cases.

Policy model can encompass legal aspects of policy (e.g., is entity linked with contact details of controller).