Consent & Access Control for Patient Information

Atif Khan

Ian McKillop

PhD Seminar

David R. Cheriton School of Computer Science
University of Waterloo

Jun 10th, 2013

Motivation

Patients are empowered to play an **active** role in the **management** of their medical information

- increasing awareness of individual privacy
- legislative-based measures
  - HIPAA – U.S. Health Insurance Portability and Accountability, 96
  - PIPEDA – Personal Information Protection and Electronic Documents Act, 00
  - PHIPA – Personal Health Information Protection Act, 04
  - EU-DPD – European Union Data Protection Directive, 95

  - punish not prevent

Motivation

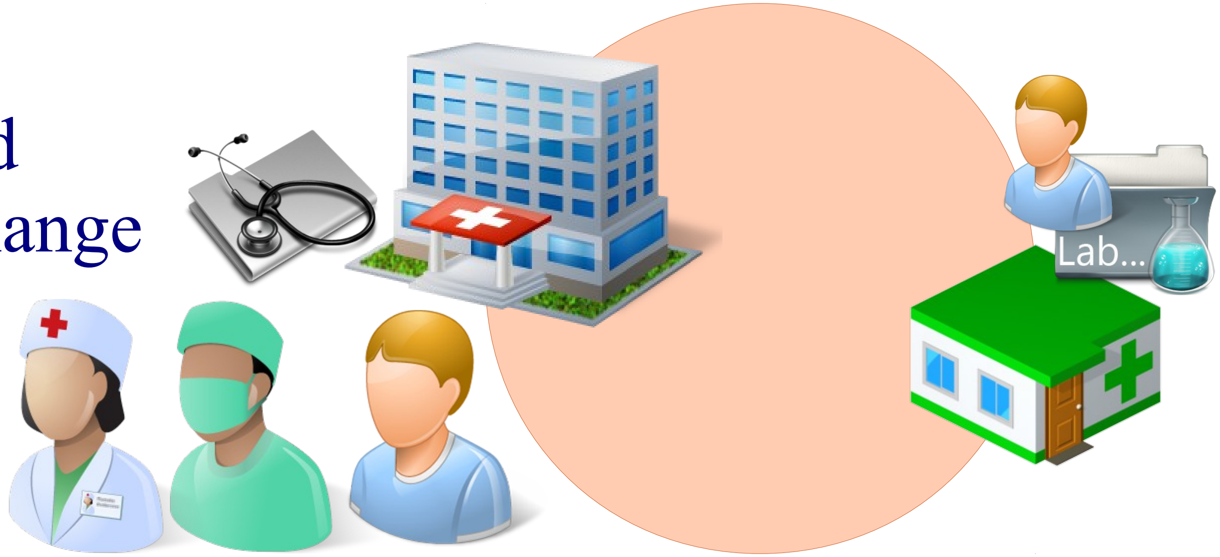
Patient information is

- **dispersed** over **heterogeneous** health information systems under the administration of **different security** domains
- managed by **multiple owners**
- **dynamic** *circle-of-care* membership with **high churn**

Motivation

Considerations

- privacy & security policies
- heterogeneous security models across different administrative domains
- multiple owners
- trusted links required for information exchange



Outline

Background

- challenges of consent application
- knowledge engineering

Proposed Model

- control primitives
- *POC* ontological model
- example scenarios
- handshake protocol

Conclusion

Outline

Background

- challenges of consent application
- knowledge engineering

Proposed Model

- control primitives
- *POC* ontological model
- example scenarios
- handshake protocol

Conclusion

House Keeping

- breakdown roughly 40/20
- questions

Outline

Background

- challenges of consent application
- knowledge engineering

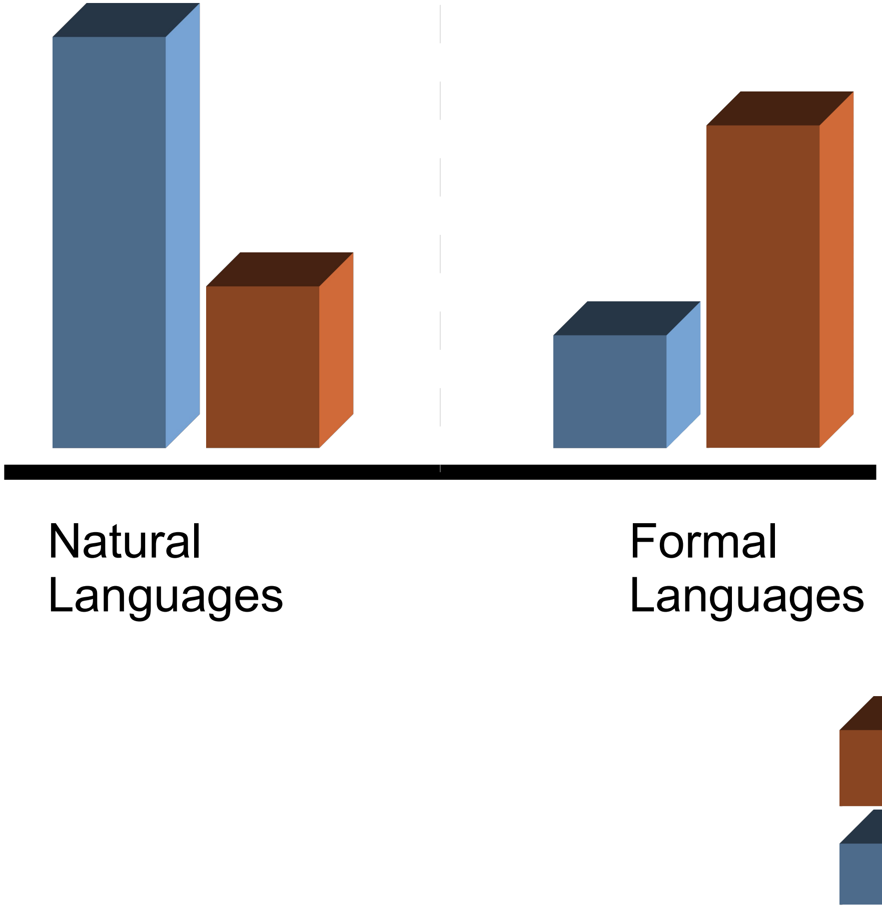
Proposed Model

- control primitives
- *POC* ontological model
- example scenarios
- handshake protocol

Conclusion

Integrating Consent is Challenging

Precision vs. Expressivity



Integrating Consent is Challenging

Precision vs. Expressivity

- ***high degree*** of ***precision*** (of interpretation) of ***consent*** preferences is a ***must***

- ***precision*** can be ***enhanced*** by use of ***formal languages***

  'cost of precision' exponentially grows with an increase in required expressiveness

Integrating Consent is Challenging

Precision vs. Expressivity

- ***high degree*** of ***precision*** (of interpretation) of ***consent*** preferences is a ***must***

- ***precision*** can be ***enhanced*** by use of ***formal languages***

  'cost of precision' exponentially grows with an increase in required expressiveness

consent expression and its application requires **maximizing** both **expression** and **precision**

Integrating Consent is Challenging

Global Interpretation – Universal Semantics

- given the ***variances*** in data and ***security primitives***, it is important to ***establish*** and ***preserve*** the ***meaning*** of ***consent*** rules and ***policies***

__e.g.__, considering RBAC, different security domains can describe the same role "physician" with different levels of access privileges

Integrating Consent is Challenging

Coverage

- given a ***finite expression space*** of a language, it is difficult to express consent for all possible scenarios

  instead, it is preferable to have ***consent primitives offer transference*** properties

Integrating Consent is Challenging

Coverage

- transferring existing consent

$$applies(c, s_1)$$

Integrating Consent is Challenging

Coverage

- transferring existing consent

$$applies(c, s_1) \wedge \underbrace{f(s_1, s_2)}_{\text{similarity function}} > \overbrace{t}^{\text{threshold}}$$

Integrating Consent is Challenging

Coverage

- transferring existing consent

$$applies(c, s_1) \wedge f(s_{1,} s_2) > t \quad \rightarrow \quad applies(c, s_2)$$

Integrating Consent is Challenging

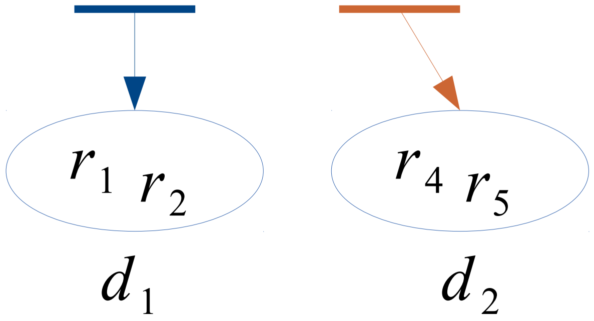
Universal Enforcement

$$R=\{r_1, r_2, r_3, r_4, r_5, \ldots, r_k\} \quad \wedge \quad belongsTo(R, P)$$

Integrating Consent is Challenging

Universal Enforcement

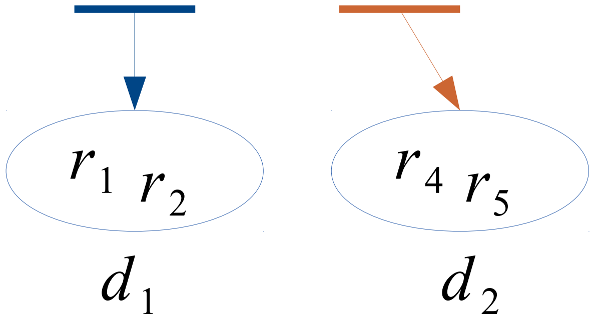
$$R=\{r_1,r_2,r_3,r_4,r_5,\ldots,r_k\} \quad \wedge \quad belongsTo(R,P)$$



Integrating Consent is Challenging

Universal Enforcement

$$R=\{r_1,r_2,r_3,r_4,r_5,\ldots,r_k\} \quad \wedge \quad belongsTo(R,P)$$



$$enforce\ \boldsymbol{C}\ across\ \boldsymbol{D},\ where\ D=\{d_1,d_2,\cdots\}$$

Integrating Consent is Challenging

Enforcement Guarantees

- validation
  were the consent preferences applied properly?

- audit
  can it be confirmed for correctness?

Traditional Access Control Models

Limitations

- enforcement across security domains
  - requires pre-established trust relationship
  - predefined mapping of security models

- multiple owners
  - traditional models are ***system-centric*** → single owner where a ***user-centric*** approach is required

Traditional Access Control Models

Limitations

- traditional models result in implementations with static configurations
  - access control parameters/policies are predefined
  - leads to ***'breaking-of-the-glass'*** scenarios

Outline

Background

- challenges of consent application
- knowledge engineering

Proposed Model

- control primitives
- *POC* ontological model
- example scenarios
- handshake protocol

Conclusion

Semantic Knowledge Representation

Ontology

> *an ontology is a formal, explicit specification of a shared conceptualization* (R. Struder 98)

- a ***'domain-of-discourse'*** is described using ontological concepts and the relationships between these concepts
- each ontological concept and relationship is unique (*precision of interpretation* = very high)
- the *expressivity* of an ontology ≡ construction (concepts and relationships linking the concepts)

Semantic Knowledge Representation

Ontology

AccessControl

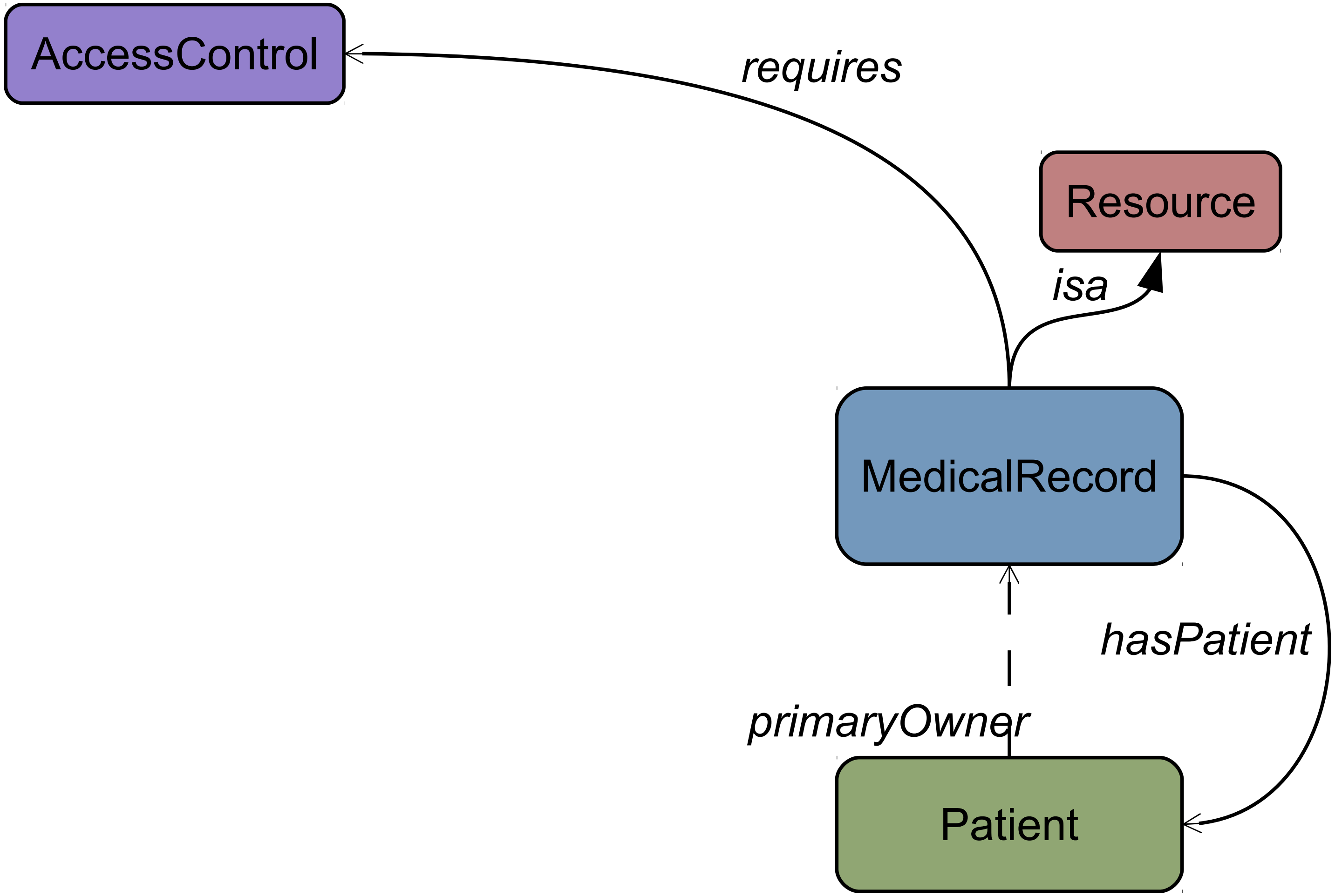
Resource

MedicalRecord

Patient

Semantic Knowledge Representation

Ontology



Semantic Knowledge Representation

Ontology

Let $\mathcal{V}$ be a set of structured vocabulary, and $\mathcal{A}$ axioms about $\mathcal{V}$, which are formulated in a formal language $\mathcal{L}$. Then an ontology $\mathcal{O}$ is a sign-system:

$$\mathcal{O} = \langle \mathcal{L}, \mathcal{V}, \mathcal{A} \rangle$$

(Hussain 09)

Knowledge Inference

Inference

- axioms are defined to infer ***implicit knowledge*** from ***explicitly*** stated ***facts***

- axiom classes *(not discussed)*
- entailment rules *(in N3)*

Knowledge Inference

Inference

- axioms are defined to infer ***implicit knowledge*** from ***explicitly*** stated ***facts***

- entailment rules

$$\begin{tikzcd}
\{f_1, f_2, f_3, \cdots, f_n\} \arrow[r] \arrow[d] & \{a_1, a_2, \cdots\} \arrow[d] \\
\{f_1 \wedge f_2 \wedge f_3 \wedge \cdots \wedge f_n\} & \{a_1 \wedge a_2 \cdots\}
\end{tikzcd}$$

monotonic process

Knowledge Inference

Inference

- axioms are defined to infer ***implicit knowledge*** from ***explicitly*** stated ***facts***

- entailment rules *(in N3)*

triple representation: `subject verb object.`

```
{?R :isa :MedicalRecord; :hasPolicy ?POL.
 ?POL :hasScope :OptIn; :hasOverride :None.
 ?DR :isa :Doctor.

} => { ?DR :hasAccess ?R}
```

Knowledge Inference

Inference

- axioms are defined to infer ***implicit knowledge*** from ***explicitly*** stated ***facts***

- entailment rules *(in N3)*

explicit facts

```
{?R :isa :MedicalRecord; :hasPolicy ?POL.
 ?POL :hasScope :OptIn; :hasOverride :None.
 ?DR :isa :Doctor.
-----------------------------------------------
} => { ?DR :hasAccess ?R}
```

implied knowledge

Knowledge Inference

Inference

- axioms are defined to infer ***implicit knowledge*** from ***explicitly*** stated ***facts***

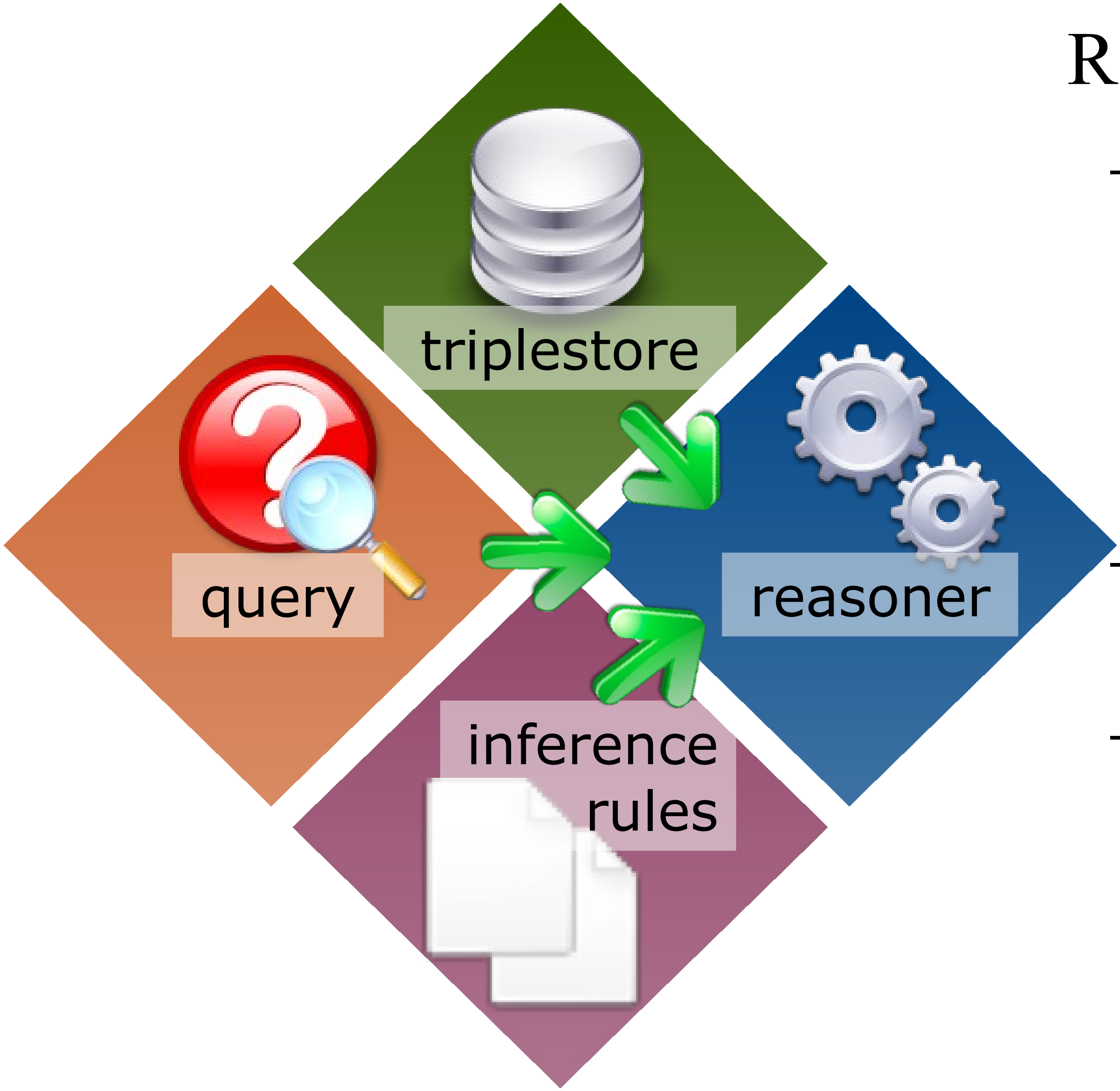
- entailment rules *(in N3)*

form an ontology

```
{ ?R :isa :MedicalRecord; :hasPolicy ?POL.
  ?POL :hasScope :OptIn; :hasOverride :None.
  ?DR :isa :Doctor.

} => { ?DR :hasAccess ?R}
```

generic rules written using variables

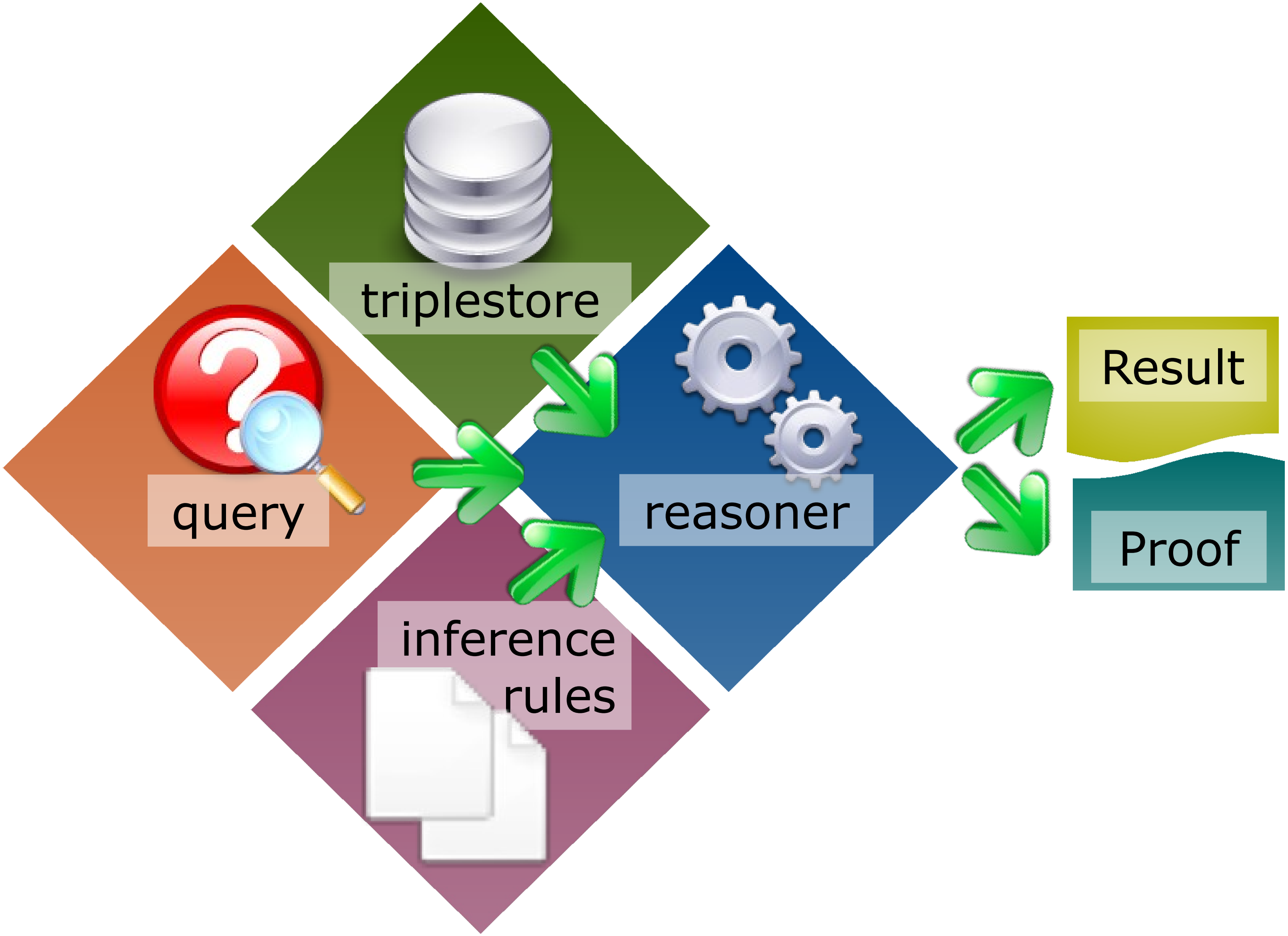
Knowledge Reasoning



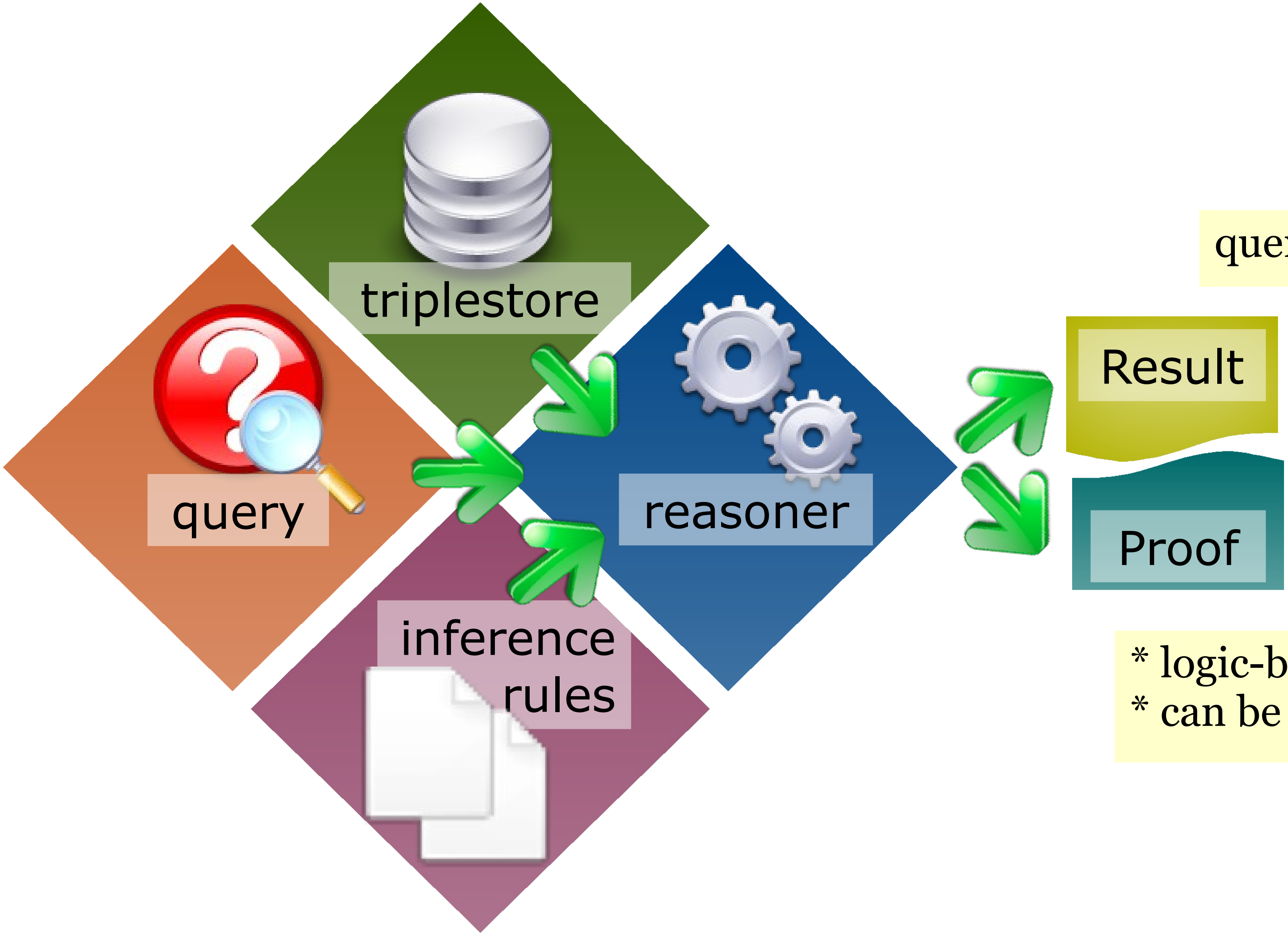
Reasoning Process

- discover new facts by applying inference rules to available *triples* to answer the query
- based on ***first order logic***
- requires a ***semantic reasoner***

Knowledge Reasoning



Knowledge Reasoning



query answer

* logic-based,
* can be verified

Outline

Background

- challenges of consent application
- knowledge engineering

Proposed Model

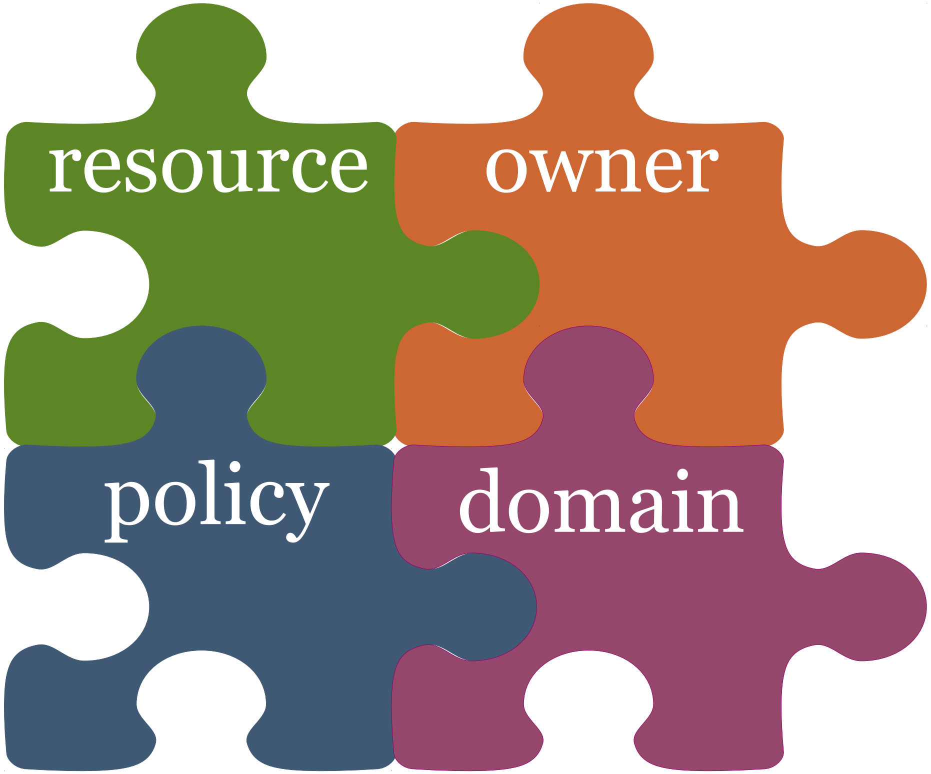
- control primitives
- *POC* ontological model
- example scenarios
- handshake protocol

Conclusion

Proposed Access Control Model

Control Primitives

define the building blocks of the proposed access control model



Control Primitives

Resource

resource represents any "***entity***" for which **access control** is **required**

$$\forall t : requires(t, AccessControl) \rightarrow resource(t)$$

e.g.
medical records
diagnostic images
medical procedures

Control Primitives

Resource

resource represents any "***entity***" for which **access control** is **required**

- a resource can be identified by
  - a specific resource identifier (such as record number)
  - a logical grouping of things

$$R_M = \{ r \mid r \, is \, a \, mental \, health \, record \}$$

Control Primitives

Resource

resource represents any "***entity***" for which **access control** is **required**

- a resource can be identified by
  - a specific resource identifier (such as record number)
  - a logical grouping of things

composite resource

$$R_M = \{ r \mid r \text{ is a mental health record} \}$$

this can be **statically defined** or can be **inferred dynamically**

concept hierarchy along ***isa*** *relationship*

rule-based inference

Control Primitives

Resource

resource represents any "***entity***" for which **access control** is **required**

- resource *provenance*

$$created(r, d) \rightarrow originatesFrom(r, d)$$

a resource may be transferred over

$$created(r, d_l) \wedge transferedTo(r, d_l, d_h)$$
$$\rightarrow owner(d_h, r) \wedge originatesFrom(r, d_l)$$

Control Primitives

Owner

an ***entity*** that is *allowed* to define rules for a resource

$$canDefine(e, r, p) \rightarrow owner(e, r)$$

where:
e is an entity
r is a resource
p is a policy

Control Primitives

Owner

an ***entity*** that is *allowed* to define rules for a resource

$$canDefine(e, r, p) \rightarrow owner(e, r)$$

- multiple (*resource*) owners
  - patients are primary owners of their information
  - all other owners are considered secondary owners

Control Primitives

Domain

an ***administrative abstraction*** for an entity governing a set of resources

- example,
  - a hospital, or a family physician's office
- a *domain* is an ***implied owner*** for all resources that originate within its administrative control

Control Primitives

Policy

a ***set of rules*** that ***must be fulfilled*** to grant access to a resource

- coarse-grained policy expressions
  - *all treating physicians have access to my medical records*
  - *no one has access unless there is a life threatening emergency*
- fine-grained policy expressions
  - *Dr. Smith cannot access my mental health records*

Control Primitives

Policy

a ***set of rules*** that ***must be fulfilled*** to grant access to a resource

- linking ***policies*** to ***resources***

$$hasPolicy(r, p)$$

where:
r is a resource
p is a policy

$$hasPolicy(R, P)$$

where:
R is a set of resources
P is a set of policies

Control Primitives

Policy

a ***set of rules*** that ***must be fulfilled*** to grant access to a resource

- linking ***policies*** to ***resources***

$$hasPolicy(r, p)$$

$$hasPolicy(xary1, OptIn)$$

$$hasPolicy(R, P)$$

$$P = \{c_{pat}, p_{institutional}, p_{provincial}\}$$
$$R = \{r \mid r \ is\, a\, mental\, health\, record\}$$

Control Primitives

Policy

a ***set of rules*** that ***must be fulfilled*** to grant access to a resource

- policy ***conflict resolution***
  - *resources* are protected by *policies* defined by multiple *owners*

$$f_{resolve}: \quad p_i \circ p_{i+1} \circ \cdots \circ p_{i+k} \quad \rightarrow \quad p_{effective}$$

Control Primitives

Policy

a ***set of rules*** that ***must be fulfilled*** to grant access to a resource

- policy ***conflict resolution***
  - *resources* are protected by *policies* defined by multiple *owners*

$$f_{resolve}: \underbrace{p_i \underset{\text{aggregation operator}}{\circ} p_{i+1} \circ \cdots \circ p_{i+k}}_{\text{functional aggregation of policies}} \rightarrow \underset{\text{policy enforced}}{p_{effective}}$$

Control Primitives

Policy

a ***set of rules*** that ***must be fulfilled*** to grant access to a resource

- policy ***conflict resolution***
  - *resources* are protected by *policies* defined by multiple *owners*

$$f_{resolve}: \quad C_{patient} \vee \left(p_{hospital} \wedge p_{province}\right) \quad \rightarrow \quad p_{effective}$$

Outline

Background

- challenges of consent application
- knowledge engineering

Proposed Model

- control primitives
- POC ontological model
- example scenarios
- handshake protocol

Conclusion

A POC Access Control Ontology

$$\mathcal{O}=\langle \mathcal{L}, \mathcal{V}, \mathcal{A} \rangle$$

- choosing $\mathcal{L}$
  - high degree of expressivity and precision
  - computational completeness

    all decisions are guaranteed to be computable
  - decidability

    all computations will finish in finite time

A POC Access Control Ontology

$\mathcal{O} = \langle \mathcal{L}, \mathcal{V}, \mathcal{A} \rangle$

- choosing $\mathcal{L}$ = $\mathcal{OWL}$-$\mathcal{DL}$

web ontology language (OWL)

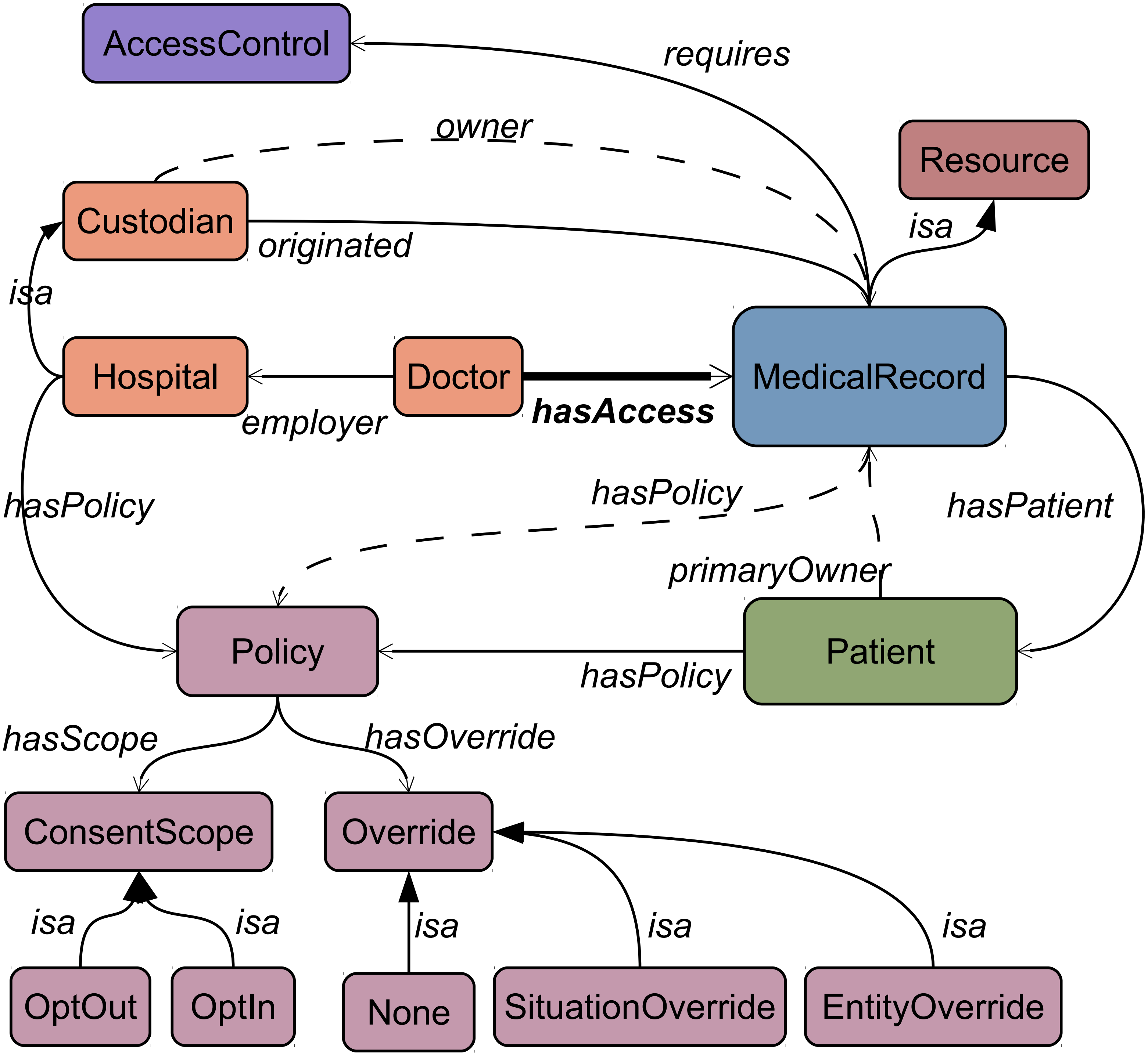
- formal semantics
- machine processable serialization
- based on description logic (DL)

A POC Access Control Ontology

$\mathcal{O} = \langle \mathcal{L}, \mathcal{V}, \mathcal{A} \rangle$

- choosing $\mathcal{V}$

any structured vocabulary can be utilized as long as monotonic reasoning is not violated


AccessControl
requires
owner
Resource
Custodian
originated
isa
isa
Hospital
Doctor
MedicalRecord
employer
hasAccess
hasPolicy
hasPolicy
hasPatient
primaryOwner
Policy
Patient
hasPolicy
hasScope
hasOverride
ConsentScope
Override
isa
isa
isa
isa
isa
OptOut
OptIn
None
SituationOverride
EntityOverride

A POC Access Control Ontology

$O = \langle \mathcal{L}, \mathcal{V}, \mathcal{A} \rangle$

- defining $\mathcal{A}$
  - all ***access control rules*** defined ***as entailment rules*** using $\mathcal{V}$

- consent preferences expressed as a *policy*
  - a *policy* is a set of access control rules

Outline

Background

- challenges of consent application
- knowledge engineering

Proposed Model

- control primitives
- POC ontological model
- example scenarios
- handshake protocol

Conclusion

Proposed Model

Example Scenarios

- medical records as resources requiring access control
- discovering resource owners
- OptIn
- OptIn & OptOut with conditions
- consent transitivity
- validation of system made decisions
- *breaking-of-the-glass*

Example *Domain of Discourse*

```
1 :H1   a  :Hospital.
2 :Dr1 a :Doctor; :employer :H1.
3 :Dr2 a :Doctor; :employer :H1.

4 :P1 a :Patient.
5 :R1 a :MedicalRecord;
     :hasPatient :P1; :originated :H1.

6 :P2 a :Patient.
7 :R2 a :MedicalRecord;
     :hasPatient :P2; :originated :H1.
```

Example *Domain of Discourse*

Rule:
anything that is a medical record requires access control

```
{?R a :MedicalRecord}
    =>{?R :requires :AccessControl}.
```

Example *Domain of Discourse*

Rule:

anything that is a medical record requires access control

```
{?R a :MedicalRecord}
   =>{?R :requires :AccessControl}.
```

Query

what entities require access control?

```
:R1 :requires :AccessControl.
:R2 :requires :AccessControl.
```

Resource Owners

```
{?H a :Hospital.} => {?H a :Custodian}.

{?C a :Custodian.
 ?R a :MedicalRecord.
 ?R :originated ?C } => {?C :owner ?R}.

{?P a :Patient.
 ?R a :MedicalRecord.
 ?R :hasPatient ?P } => {?P :primaryOwner ?R}.

{?X :primaryOwner ?Y}=>{?X :owner ?Y}.

{?X :owner ?Y}=>{?Y :isOwnedBy ?X}.
```

Resource Owners

```
{?H a :Hospital.} => {?H a :Custodian}.

{?C a :Custodian.
 ?R a :MedicalRecord.
 ?R :originated ?C } => {?C :owner ?R}.

{?P a :Patient.
 ?R a :MedicalRecord.
 ?R :hasPatient ?P } => {?P :primaryOwner ?R}.

{?X :primaryOwner ?Y}=>{?X :owner ?Y}.

{?X :owner ?Y}=>{?Y :isOwnedBy ?X}.
```

Resource Owners

```
{?H a :Hospital.} => {?H a :Custodian}.

{?C a :Custodian.
 ?R a :MedicalRecord.
 ?R :originated ?C } => {?C :owner ?R}.

{?P a :Patient.
 ?R a :MedicalRecord.
 ?R :hasPatient ?P } => {?P :primaryOwner ?R}.

{?X :primaryOwner ?Y}=>{?X :owner ?Y}.

{?X :owner ?Y}=>{?Y :isOwnedBy ?X}.
```

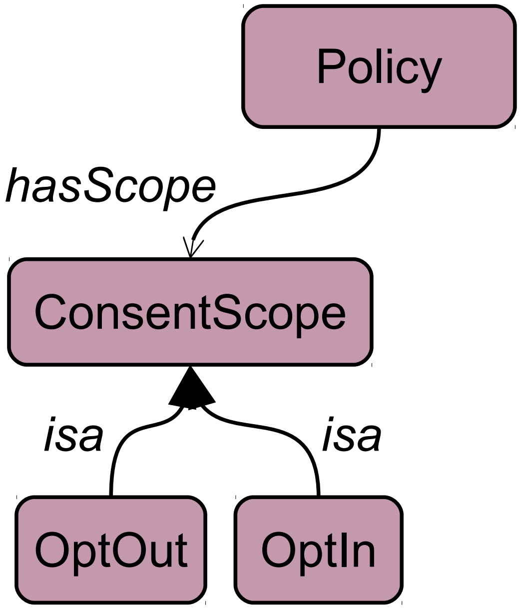
Resource Owners

```
{?H a :Hospital.} => {?H a :Custodian}.

{?C a :Custodian.
 ?R a :MedicalRecord.
 ?R :originated ?C } => {?C :owner ?R}.

{?P a :Patient.
 ?R a :MedicalRecord.
 ?R :hasPatient ?P } => {?P :primaryOwner ?R}.

{?X :primaryOwner ?Y}=>{?X :owner ?Y}.

{?X :owner ?Y}=>{?Y :isOwnedBy ?X}.
```

Resource Owners

```
:P1 :primaryOwner :R1;:owner :R1.
:P2 :primaryOwner :R2;:owner :R2.
:H1 :owner :R1, :R2.

:R1 :isOwnedBy :P1,:H1.
:R2 :isOwnedBy :P2,:H2.
```

Consent Expression



OptOut
share nothing

OptIn
share all

OptIn Consent

```
C1 a :Policy;
   :hasScope :OptIn; :hasOverride :None.


:P1 :hasPolicy :C1.

:P2 :hasPolicy :C1.
```

```
{?P :owner ?R; :hasPolicy ?P.}
   =>{?R :hasPolicy ?P}.
```

Consent Transference: *if a patient has a policy, then a record has the same policy by default if owned by the patient*

OptIn Consent

```
C1 a :Policy;
   :hasScope :OptIn; :hasOverride :None.

:P1 :hasPolicy :C1.
:P2 :hasPolicy :C1.
```

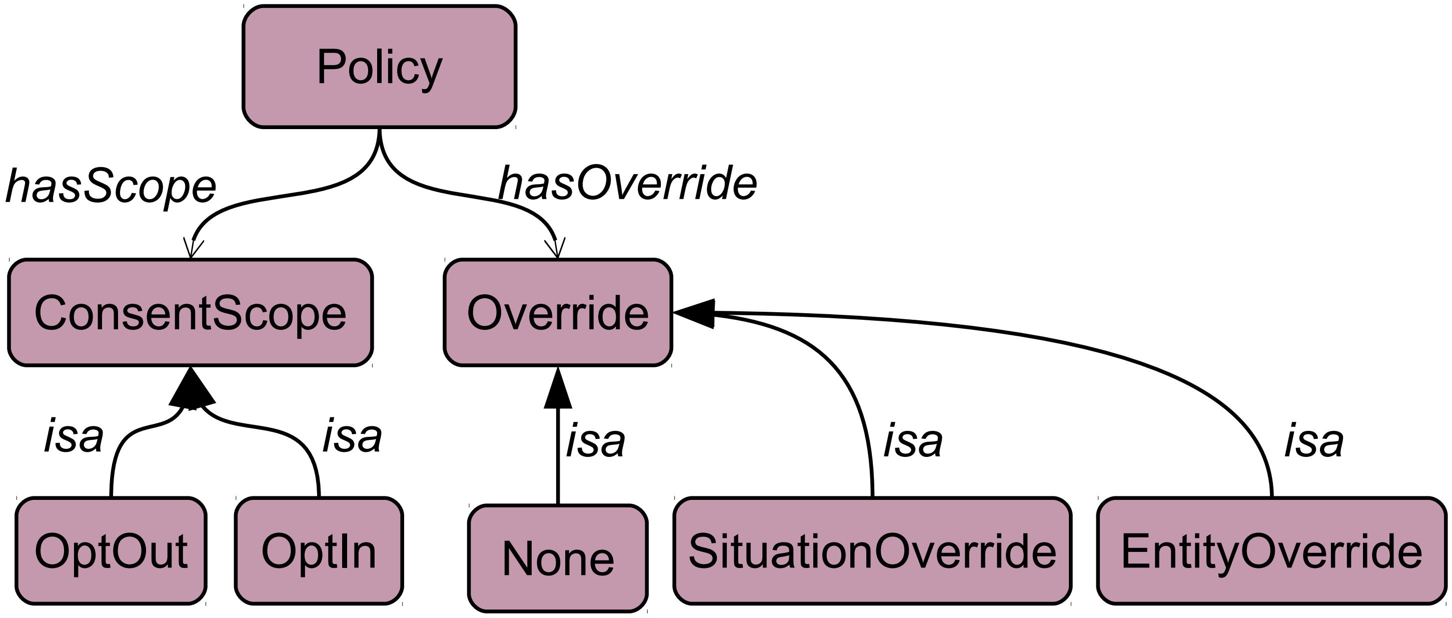
```
{?P :owner ?R; :hasPolicy ?P.}
   =>{?R :hasPolicy ?P}.
```

```
{?R a :MedicalRecord; :hasPolicy ?POL.
 ?POL :hasScope :OptIn; :hasOverride :None.
 ?DR a :Doctor.
} => {?DR :hasAccess ?R}.
```

OptIn Consent

```
:P1 :hasPolicy :C1. :R1 :hasPolicy :C1.
:P2 :hasPolicy :C1. :R2 :hasPolicy :C1.

:Dr1 :hasAccess :R1,:R2.
:Dr2 :hasAccess :R1,:R2.
```

Consent Expression



Policy

hasScope

hasOverride

ConsentScope

Override

isa

isa

isa

isa

isa

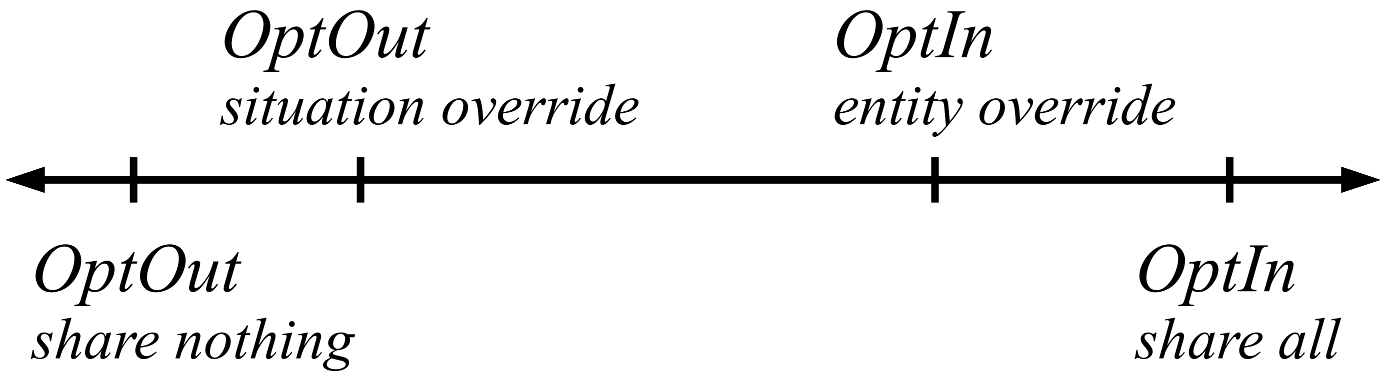
OptOut

OptIn

None

SituationOverride

EntityOverride



OptOut
situation override

OptIn
entity override

OptOut
share nothing

OptIn
share all

OptIn/OptOut with Overrides

```
:C2 :hasScope :OptOut; :hasOverride :LifeEmergency.
:C3 :hasScope :OptIn; :hasOverride :Dr2.

:P1 :hasPolicy :C2.
:P2 :hasPolicy :C3.
```

OptIn with Entity Override

```
{ ?R a :MedicalRecord; :hasPolicy ?POL.
  ?POL :hasScope :OptIn;
        :hasOverride ?DR_NOT_ALLOWED.

  ?DR_NOT_ALLOWED a :Doctor.
  ?DR a :Doctor.
  ?DR :notEqualTo ?DR_NOT_ALLOWED.

} => {?DR :hasAccess ?R}.
```

OptOut with Situation Override

```
{ ?R a :MedicalRecord; :hasPolicy ?POL.
  ?POL :hasScope :OptOut;
       :hasOverride :LifeEmergency.

  ?P a :Patient; :owner ?R;
      :hasCondition :LifeEmergency.

  ?DR a :Doctor.

} => {?DR :hasAccess ?R}.
```

OptOut with Situation Override

```
:P1 :hasPolicy :C2. :R1 :hasPolicy :C2.
:P2 :hasPolicy :C3. :R2 :hasPolicy :C3.

:Dr1 :hasAccess :R2.
```

Observations

- no one has access to medical record R1
- Dr2 does not have access to medical record R2

OptOut with Situation Override

```
:P1 :hasCondition :LifeEmergency.
```

```
:P1 :hasPolicy :C2. :R1 :hasPolicy :C2.
:P2 :hasPolicy :C3. :R2 :hasPolicy :C3.

:Dr1 :hasAccess :R1, :R2.
:Dr2 :hasAccess :R1.
```

Observations

- Dr2 has access to medical record R1
  but does not have access to medical record R2

Consent Transitivity

Extending existing consent

```
1  :C4 a :Policy;
      :hasScope :OptOut; :hasOverride :P1CareTeam.


2  :P1 :hasPolicy :C4.;
      :careTeam :P1CareTeam.


3  :Dr1 :isMemeber :P1CareTeam.
```

Consent Transitivity

Extending existing consent

```
{?D1 a :Doctor. ?D2 a :Doctor.
 ?D1 :consults [:with ?D2; :about ?P].
 ?P :careTeam ?CT.
} => {?D2 :isMemeber ?CT}.
```

Consent Transitivity

Extending existing consent

```
{?D1 a :Doctor. ?D2 a :Doctor.
 ?D1 :consults [:with ?D2; :about ?P].
 ?P :careTeam ?CT.
} => {?D2 :isMemeber ?CT}.
```

```
{?R a :MedicalRecord; :hasPolicy ?POL.
?POL :hasScope :OptOut; :hasOverride ?CT.
?DR a :Doctor; :isMemeber ?CT.
} => {?DR :hasAccess ?R}.
```

Consent Transitivity

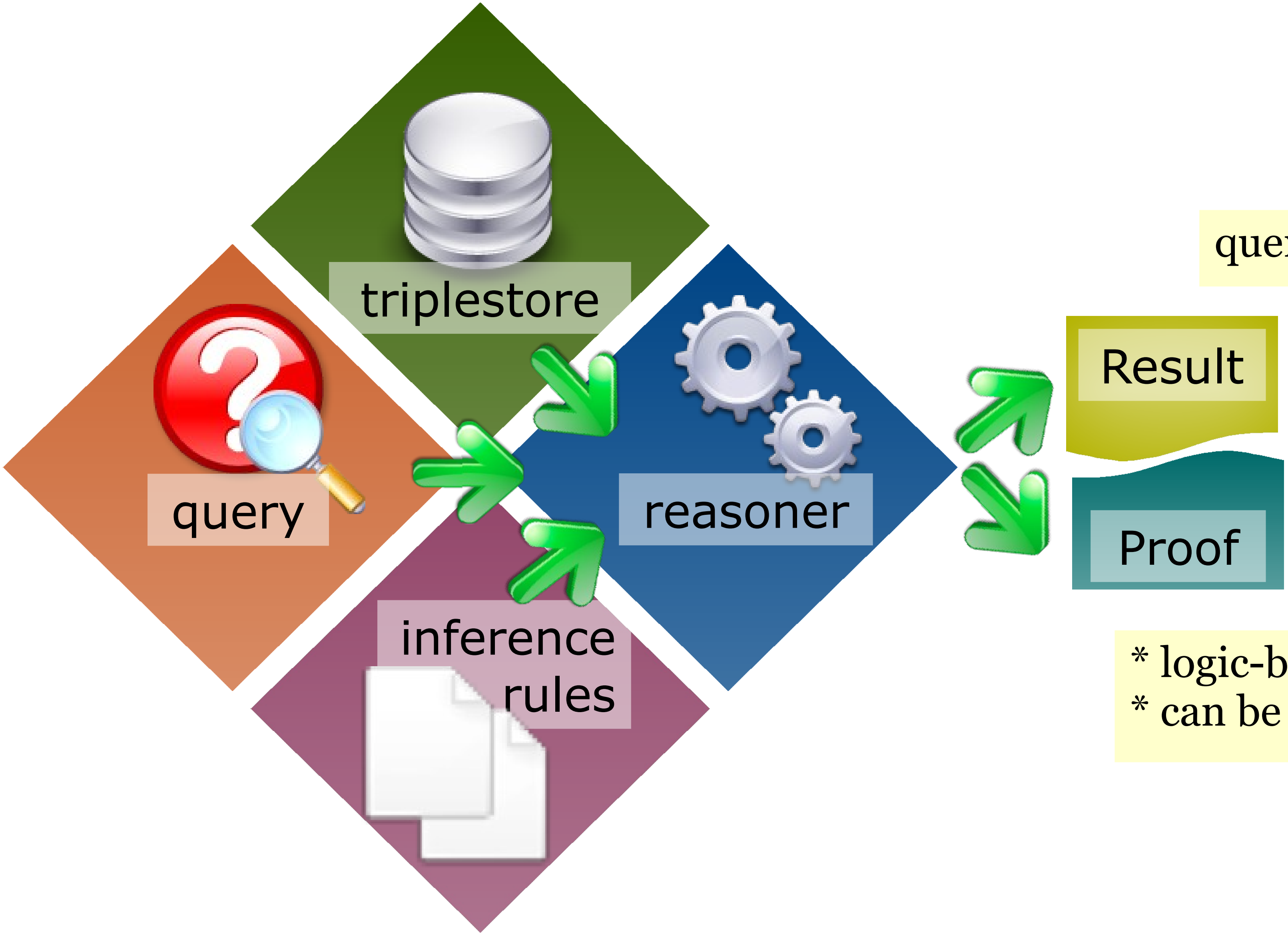
Extending Existing Consent

```
:C4 a :Policy;
   :hasScope :OptOut; :hasOverride :P1CareTeam.

:P1 :hasPolicy :C4.;
   :careTeam :P1CareTeam.

:Dr1 :isMemeber :P1CareTeam.
```

```
:Dr1 :hasAccess :R1.
:Dr2 :hasAccess :R2.
```

Trusting System Made Decisions

Proof-of-Correctness



query answer

* logic-based,
* can be verified

Trusting System Made Decisions

Proof-of-Correctness

- semantic proof

  access control (inference) ***rules are applied*** *against the knowledge-base graph* ***to find evidence*** *(sub-graphs) towards fulfilment of the query →* ***semantic proofs***

Trusting System Made Decisions

Proof-of-Correctness

- semantic proof

  access control (inference) ***rules are applied*** *against the knowledge-base graph* ***to find evidence*** *(sub-graphs) towards fulfilment of the query →* ***semantic proofs***

- validation

  traverse the sub-graphs *(semantic proofs) to decide if the same decision can be reached*

Trusting System Made Decisions

Recall

```
{ ?R a :MedicalRecord; :hasPolicy ?POL.
  ?POL :hasScope :OptIn;
       :hasOverride ?DR_NOT_ALLOWED.

  ?DR_NOT_ALLOWED a :Doctor.
  ?DR a :Doctor.
  ?DR :notEqualTo ?DR_NOT_ALLOWED.

} => {?DR :hasAccess ?R}.
```

Trusting System Made Decisions

```
{{:R2 a :MedicalRecord} e:evidence <kb.n3#_36>.
 {{{{:P2 a :Patient} e:evidence <kb.n3#_35>.
    {:R2 a :MedicalRecord} e:evidence <kb.n3#_36>.
    {:R2 :hasPatient :P2} e:evidence <kb.n3#_36>}
    => {{:P2 :primaryOwner :R2} e:evidence <kb.n3#_42>}}
   => {{:P2 :owner :R2} e:evidence <kb.n3#_43>}.
  {:P2 :hasPolicy :C3} e:evidence <kb.n3#_35>}
  => {{:R2 :hasPolicy :C3} e:evidence <kb.n3#_48>}.
 {:C3 :hasScope :OptIn} e:evidence <kb.n3#_22>.
 {:C3 :hasOverride :Dr2} e:evidence <kb.n3#_22>.
 {:Dr2 a :Doctor} e:evidence <kb.n3#_27>.
 {:Dr1 a :Doctor} e:evidence <kb.n3#_26>.
 {:Dr1 :notEqualTo :Dr2} e:evidence <log#kb>}
 => {{:Dr1 :hasAccess :R2} e:evidence <kb.n3#_59>}.
```

Trusting System Made Decisions

Observations

- semantic proofs are also represented in *triple* format

  ∴ can also be reasoned about → automated validation

- semantic proofs provide built-in auditing mechanism

'Breaking-of-the-Glass'

Definition

*when it is necessary to **ignore policy** in the interest of **patient safety***

- a common practice
  due to limitations of traditional access control models

'Breaking-of-the-Glass'

Definition

when it is necessary to ***ignore policy*** *in the interest of* ***patient safety***

- using consent transference
  - determine if existing consent policy can be applied
  - *breaking-of-the-glass* is exercised only when absolutely necessary

'Breaking-of-the-Glass'

Definition

when it is necessary to ***ignore policy*** *in the interest of* ***patient safety***

- using semantic proofs
  - extend semantic proofs
    in support of *breaking-of-the-glass* decisions
  - provides logic-based reasons for *"why"*

Outline

Background

- challenges of consent application
- knowledge engineering

Proposed Model

- control primitives
- POC ontological model
- example scenarios
- handshake protocol

Conclusion

Handshake Protocol

Purpose

establish if a requester is allowed to receive patient information

- is not a communication protocol
  - can be integrated into existing protocols as a "*pre-step*"

Handshake Protocol

Purpose

establish if a requester is allowed to receive patient information

- 3 Phases Protocol
  - request for information
  - proof generation
  - proof validation

Handshake Protocol

$$D \xrightarrow{isTreating} P$$

$$\left(\begin{array}{c|c} Pol_{H2} & \\ & R = \{r_1, r_2, \cdots\} \\ Pol_C & \end{array} \right)$$

Handshake Protocol

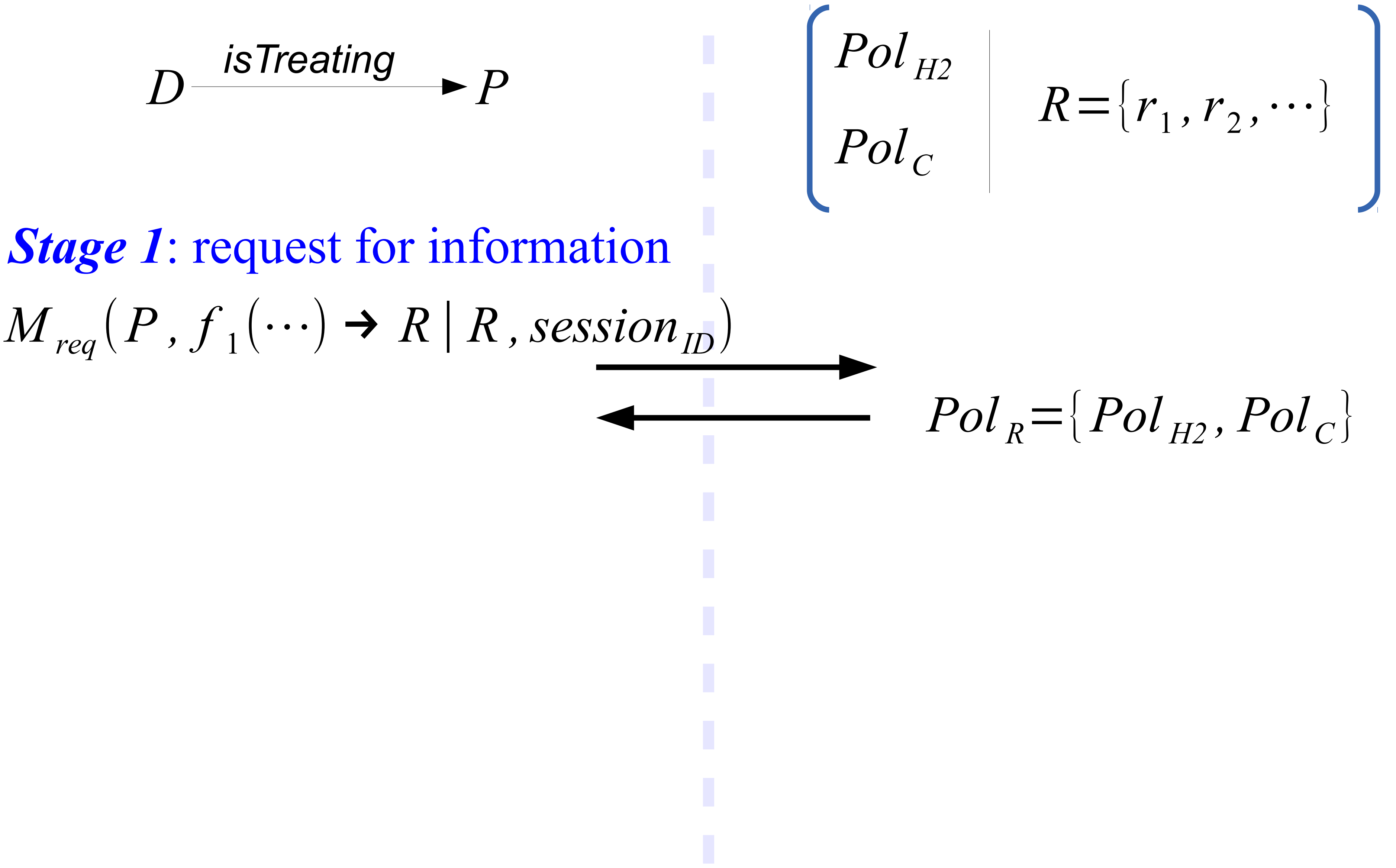
$$D \xrightarrow{isTreating} P$$

$$\left(\begin{array}{c|c} Pol_{H2} & \\ & R=\{r_1, r_2, \cdots\} \\ Pol_C & \end{array}\right)$$

Stage 1: request for information

$$M_{req}\left(P, \underbrace{f_1(\cdots) \rightarrow R}_{resolution\ function} \mid R, session_{ID}\right)$$

Handshake Protocol



$$D \xrightarrow{isTreating} P$$

$$\left[\begin{array}{c|c} Pol_{H2} & \\ & R=\{r_1, r_2, \cdots\} \\ Pol_C & \end{array}\right]$$

Stage 1: request for information

$$M_{req}(P, f_1(\cdots) \rightarrow R \mid R, session_{ID})$$

$$Pol_R = \{Pol_{H2}, Pol_C\}$$

Handshake Protocol

$$D \xrightarrow{isTreating} P$$

$$\left(\begin{array}{c|c} Pol_{H2} & \\ & R=\{r_1, r_2, \cdots\} \\ Pol_C & \end{array}\right)$$

Stage 1: request for information

$$M_{req}(P, f_1(\cdots) \rightarrow R \mid R, session_{ID}) \longrightarrow$$

$$\longleftarrow \quad Pol_R=\{Pol_{H2}, Pol_C\}$$

Stage 2: proof generation

$$Q=\{Proof(p) \mid p \in Pol_R\}$$

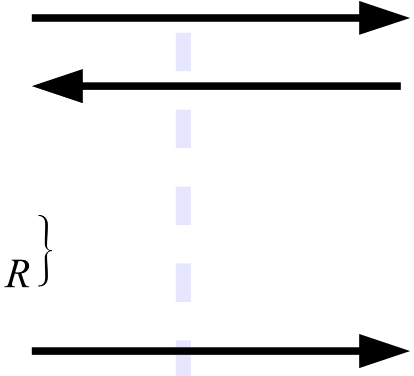
Handshake Protocol

$$D \xrightarrow{isTreating} P$$

$$\left[\begin{array}{c|c} Pol_{H2} & \\ & R=\{r_1, r_2, \cdots\} \\ Pol_C & \end{array}\right]$$

Stage 1: request for information

$$M_{req}(P, f_1(\cdots) \rightarrow R \mid R, session_{ID})$$



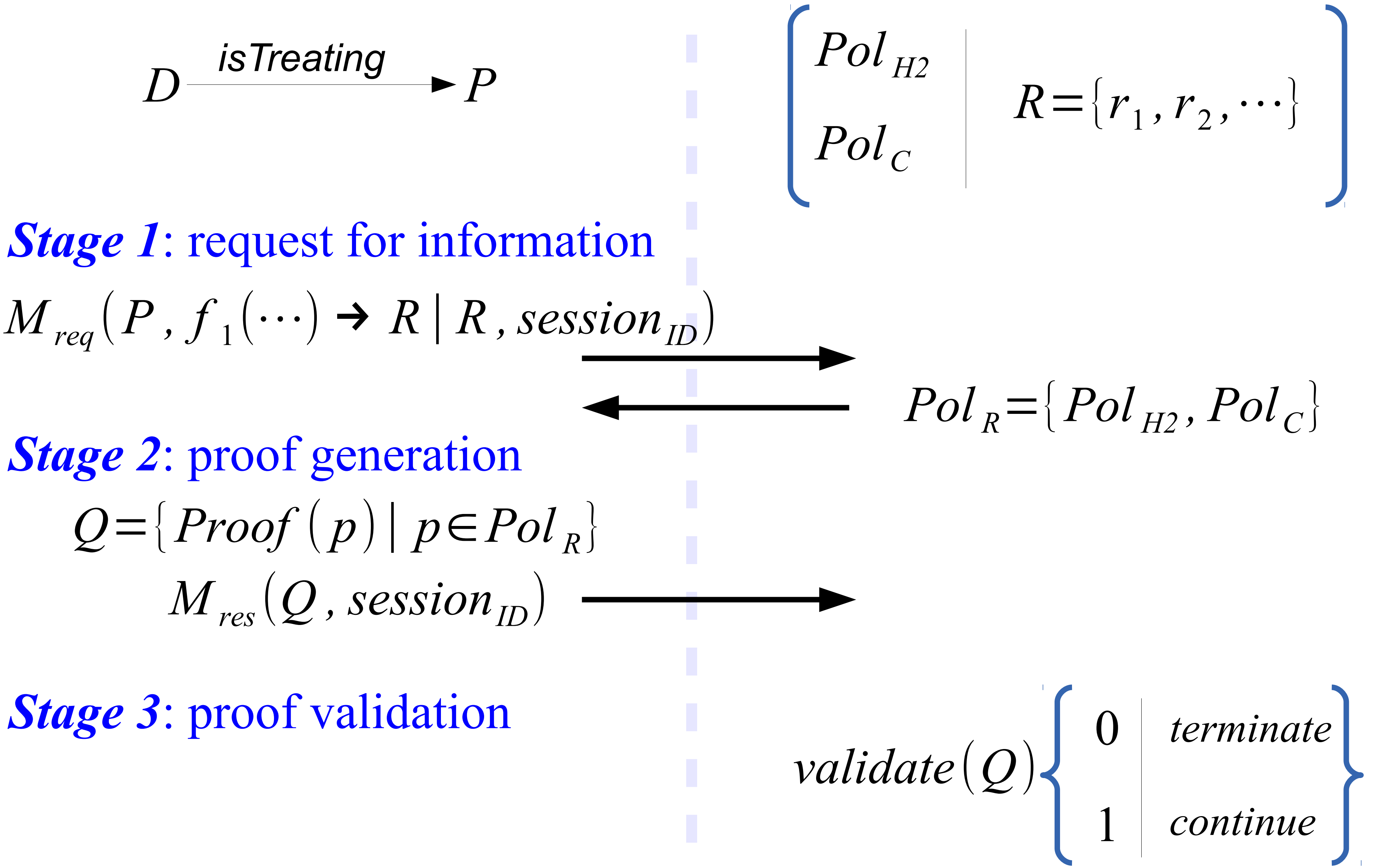
$$Pol_R=\{Pol_{H2}, Pol_C\}$$

Stage 2: proof generation

$$Q=\{Proof(p) \mid p \in Pol_R\}$$

$$M_{res}(Q, session_{ID})$$

Handshake Protocol



$$D \xrightarrow{isTreating} P$$

$$\left(\begin{array}{c|c} Pol_{H2} & \\ & R=\{r_1, r_2, \cdots\} \\ Pol_C & \end{array}\right)$$

Stage 1: request for information

$$M_{req}(P, f_1(\cdots) \rightarrow R \mid R, session_{ID})$$

$$Pol_R=\{Pol_{H2}, Pol_C\}$$

Stage 2: proof generation

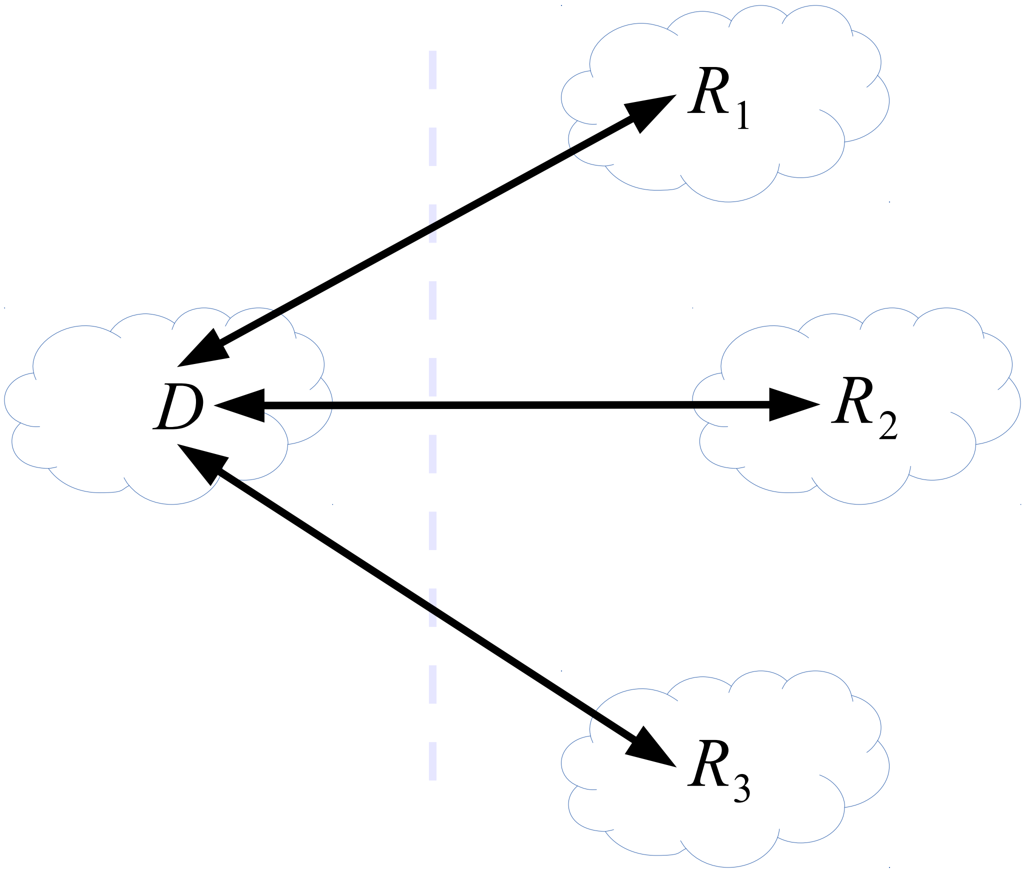
$$Q=\{Proof(p) \mid p \in Pol_R\}$$

$$M_{res}(Q, session_{ID})$$

Stage 3: proof validation

$$validate(Q)\left\{\begin{array}{c|c} 0 & terminate \\ 1 & continue \end{array}\right\}$$

Handshake Protocol



Future Extensions

Trust

- utilizing semantic proofs to dynamically establish trust

Privacy

- validation of semantic proofs requires access to raw information
  - utilize cryptographic primitives for proof generation
    zero knowledge proof, cryptographic commitments, oblivious transfer

Model as MAS (*multi agent system*)

- offload proof generation & validation to agents

Outline

Background

- challenges of consent application
- knowledge engineering

Proposed Model

- control primitives
- POC ontological model
- example scenarios
- handshake protocol

Conclusion

Conclusion

Access Control Model

- *resource, owner, policy, domain*
  - structured knowledge representation (ontology)
  - logic-based reasoning/inference

- offers high degree of expression & precision

- suitable for healthcare domain
  - multiple owners
  - heterogeneous administrative domains
  - built-in validation

Conclusion

Realization

- a simple proof-of-concept ontology
- applied to core healthcare scenarios

Integration

- *hand shake* protocol

Thank You!