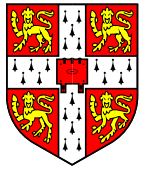
Translating Role-Based Access Control Policy within Context

Jean Bacon, Michael Lloyd, Ken Moody



Computer Laboratory, University of Cambridge

Outline of talk

1. Electronic Health Records (EHRs) a nationwide system
2. OASIS role-based access control
3. Expressing access control policy in pseudo-natural language
4. Automating the translation of policy into code
5. System monitoring and policy change management

1. Electronic Health Records (EHRs) - a nationwide system

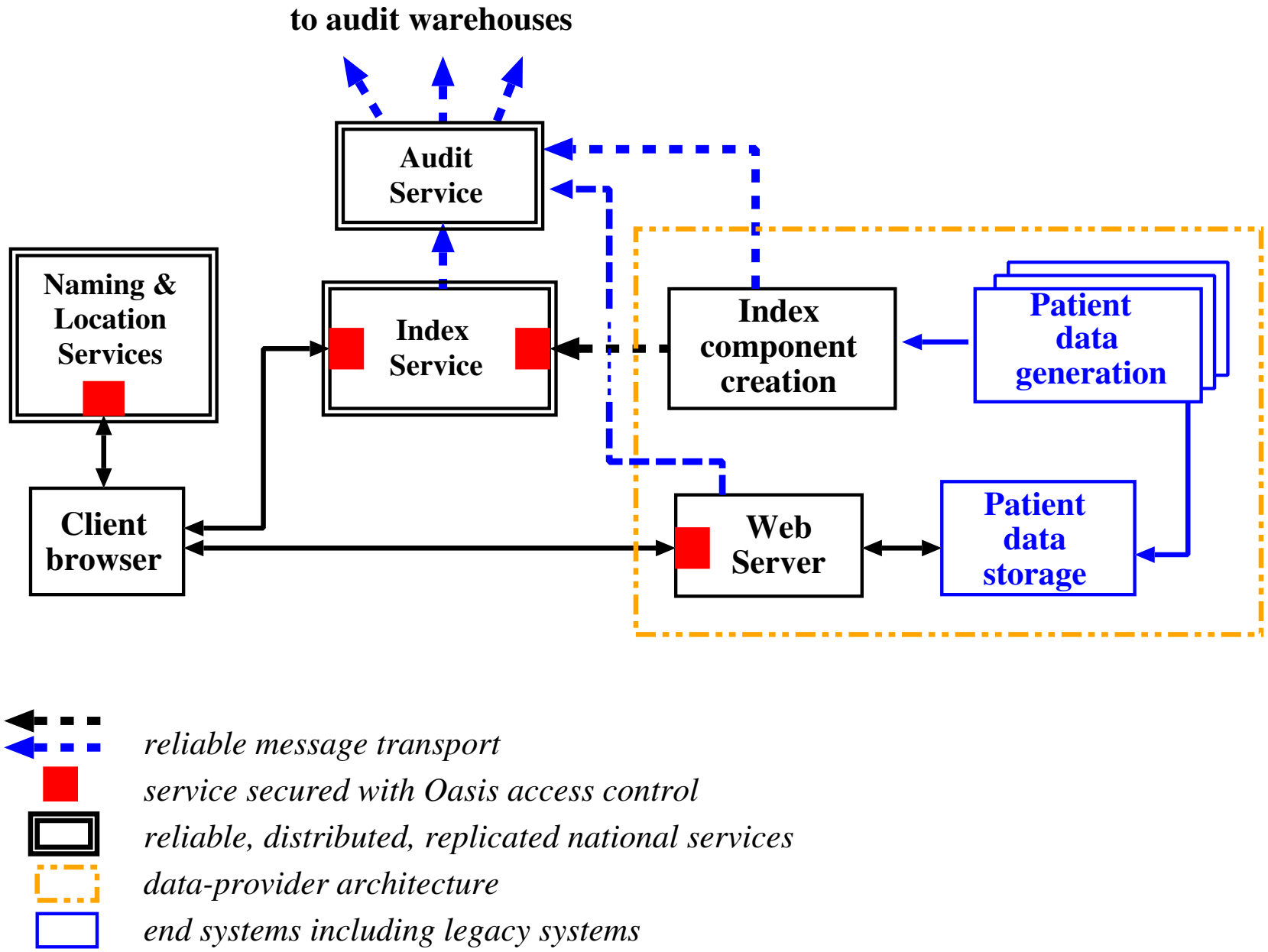
* heterogeneous
* evolving
* independently managed

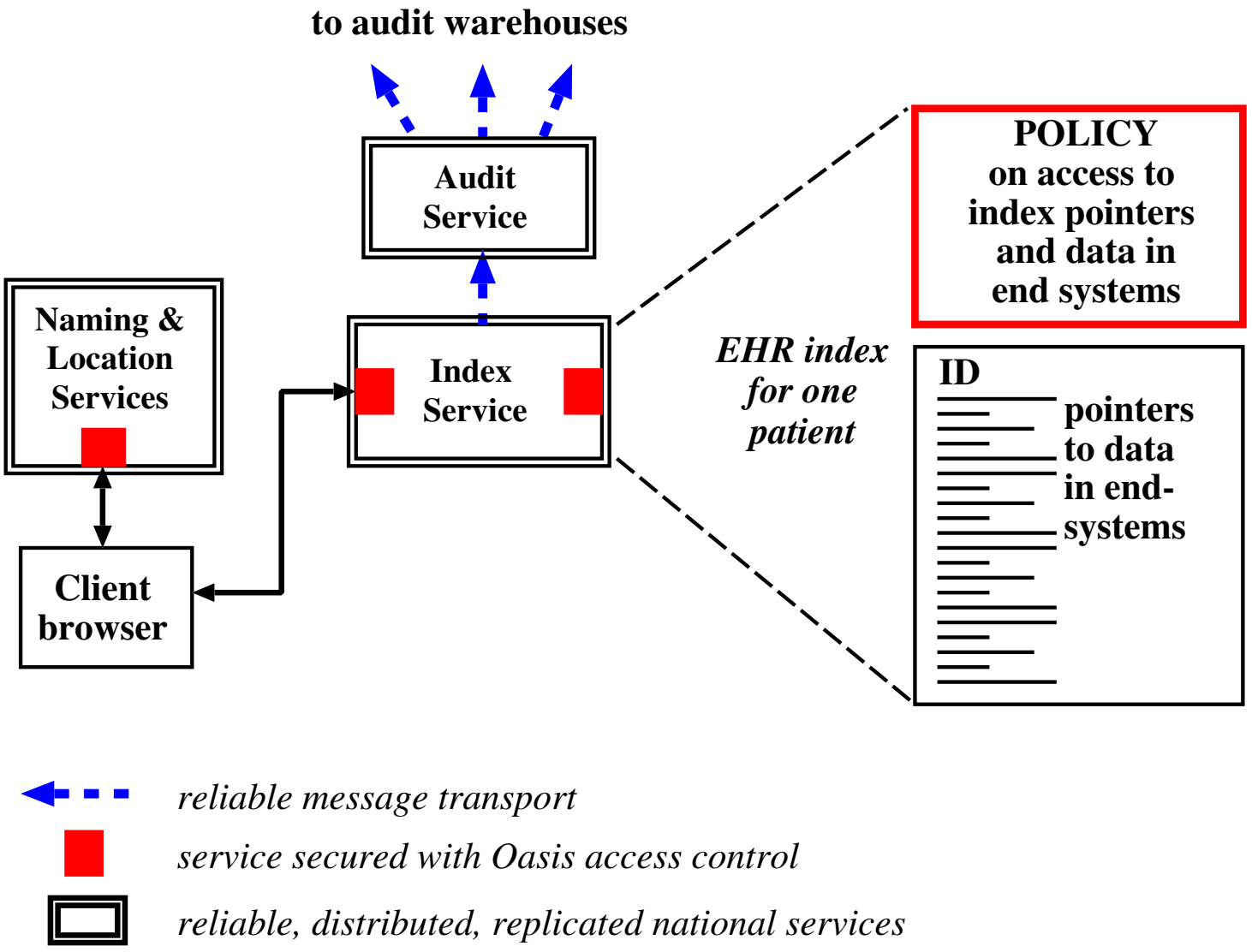
AIMS of our work:

- local expression of management policy
- coordination of local and national policies

NHS call for pan-community proposals, Oct - Nov 1999

NHS Eastern Region Health Authority EHR consortium, (EREHRC) led by Addenbrookes Hospital, including Opera group, Citrix Systems, IBM


to audit warehouses
Audit Service
Naming & Location Services
Index Service
Index component creation
Patient data generation
Client browser
Web Server
Patient data storage
reliable message transport
service secured with Oasis access control
reliable, distributed, replicated national services
data-provider architecture
end systems including legacy systems


to audit warehouses
Audit Service
Naming & Location Services
Index Service
Client browser
POLICY on access to index pointers and data in end systems
EHR index for one patient
ID
pointers to data in end-systems
reliable message transport
service secured with Oasis access control
reliable, distributed, replicated national services

2. OASIS: Open Architecture for Secure Interworking Services

* **Role based access control**

* **Oasis services name their clients in terms of roles**

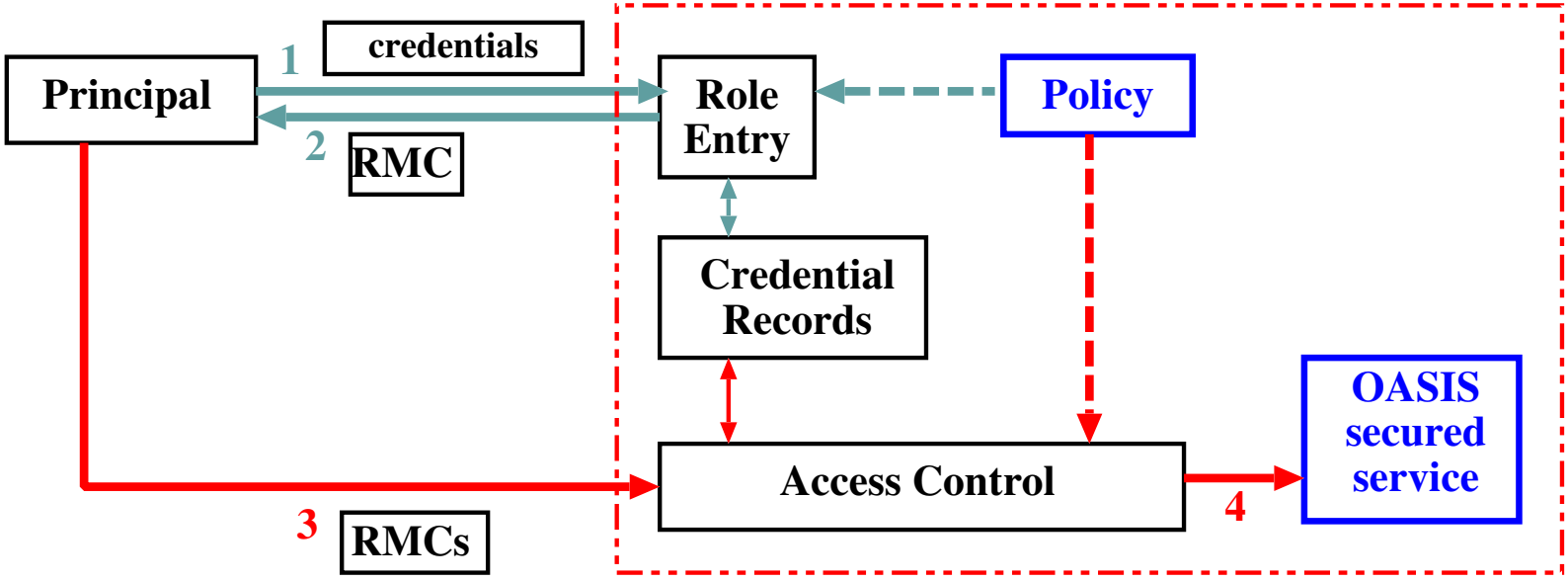
* **Oasis services specify policy in terms of roles**

  - **for role entry (role activation)**

  - **for service use (method invocation, including object access)**

  both in Horn Clause form

A service secured by OASIS access control



RMC = role membership certificate

= role entry

= use of service

entering (activating) a role:

* Clients present credentials to a service to enter a named role

* Service checks credentials against policy specification
  - roles already activated
  - other certificates held
  - environmental constraints (side conditions)

* If successful, client is issued a role membership certificate (RMC)

using services:

* Clients present credentials to use services
  - RMCs of this and other services

* Services, according to their access control policy,
  - validate clients' credentials
  - check any environmental constraints (side conditions)

OASIS role-based access control

for more information see:

IEEE Symposium on Security and Privacy, Oakland 1998
OASIS: Access Control in an Open Distributed Environment
Hayton, Bacon, Moody

IEEE Computer March 2000
Generic Support for Asynchronous, Secure Distributed Applications
Bacon, Moody et al.

Middleware 2000
An Architecture for Distributed OASIS Services
Hine, Yao, Bacon and Moody

3. Expressing access control policy in pseudo-natural language

Policies derive from:

National and International Law

UK and European Data Protection Acts

Government Regulations

Patients' Charters for England, Wales etc.
which enable patient preference

Functional Policies expressed at care centres:
Hospital Departments
GP practices (primary care)
Specialist Clinics

The last group are our target!

expressing policy for role entry

RoleName (arg , ...) <=
RoleRef (arg , ...) [^ RoleRef (arg , ...)] [: SideCondition (arg , ...) ....]

expressing policy for method invocation authorisation

MethodRef (attribute, owner, invoker) <=
RoleRef (arg , ...) [^ RoleRef (arg , ...)] [: SideCondition (arg , ...) ....]

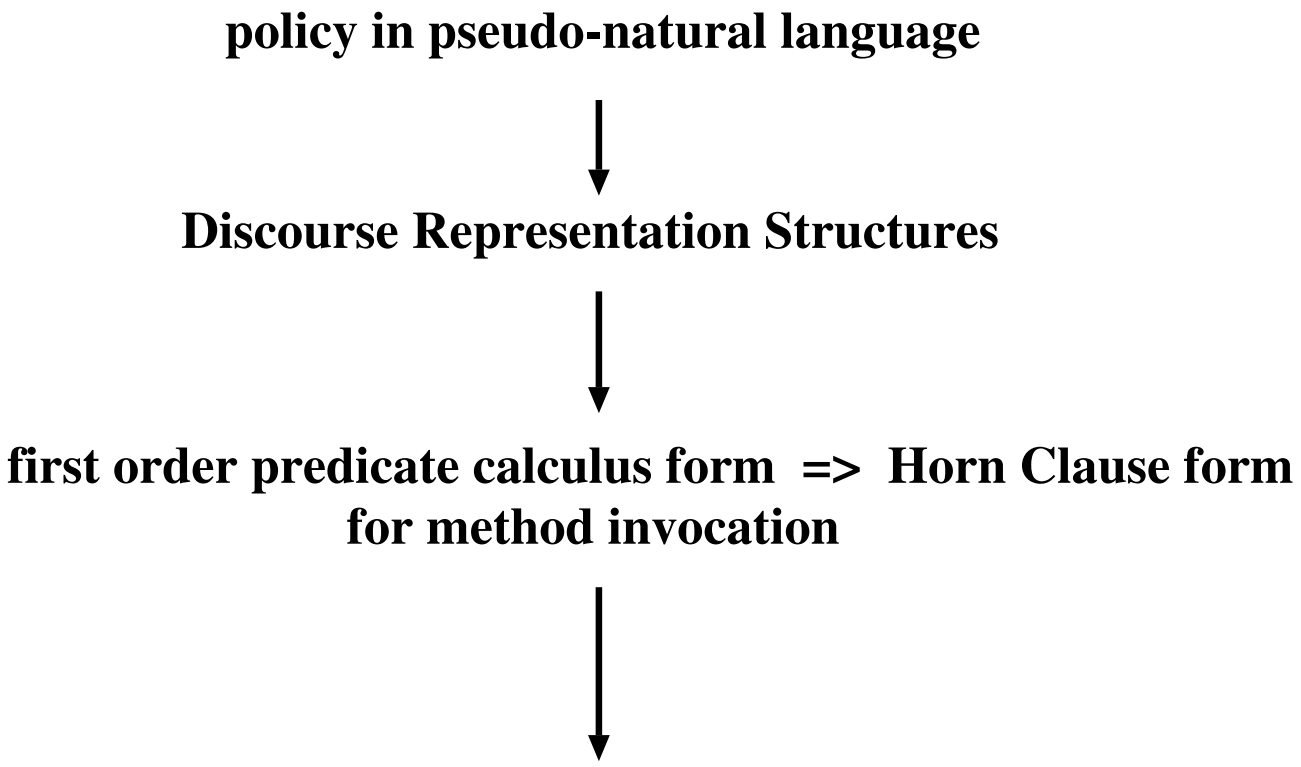
for EHR, side conditions will typically be looked up in a database, e.g.

MethodRef *attribute, owner, invoker*

Invoke-read (EHR-contact-details-field, patient-y, gp-x) <=
GP-role (gp-x) : gp-of (gp-x, patient-y)

RoleRef *side condition looked up in local database*

4. Automating the translation of policy into code



Use of Discourse Representation Theory (DRT)

Advice from Steve Pulman of the Computer Lab NL group:

use Attempto controlled English, which translates to Discourse Representation Structures
can tailor specific pseudo NLs for expressing policy in particular healthcare applications

acted as supervisor for:

Mike Lloyd: MPhil in Speech and NL Processing
- **project explored the use of controlled English for expressing access control policy for EHRs in primary healthcare**

OASIS RBAC is permissive only, so inappropriate for EHR index records

the examples we give do not use the full expressive power of DRT but illustrate the translation process

Expressing policy in pseudo-natural language

Semantics of RBAC require universal rather than existential quantification

how to express?

e.g. NOT " a GP can read his/her patients' contact details"

- **ambiguous**
- **traditional natural language semantics are existential**

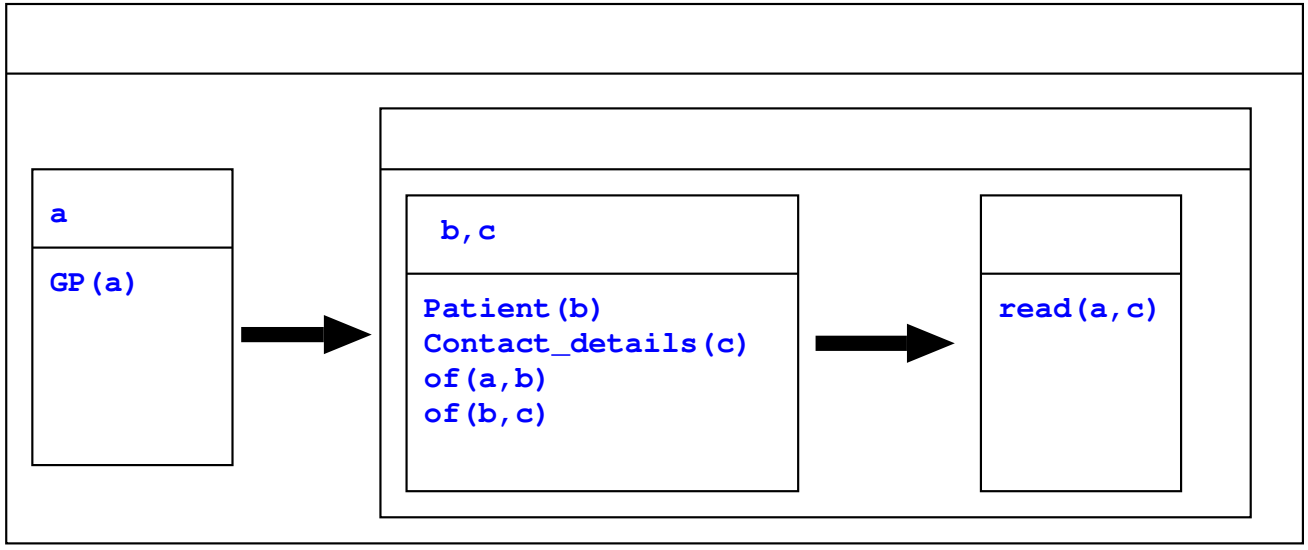
Therefore we restrict:

- **must use ALL or EVERY, not A**
  - ∀ **principals in a role**
  - ∀ **arguments of a certain type**

"EVERY GP can read the contact details of ALL his/her patients"

"Every GP can read the contact details of all his/her patients"

The Discourse Representation Structure for this example is:



subject / invoker
and any side conditions
role **GP**
argument **a**

object / owner
argument instantiations and relations
owner type **Patient**, *argument* **b**
data object :
EHR field type **Contact_details**
argument **c**
relations
gp-of **of(a,b)**
EHR-field-of **of(b, c)**

method (invoker, object)

Translation of access control policy into FOPC

DRS => FOPC produces :

subject - invoker *owner & object* *relations* *method (invoker, object)*

$\forall$ a. (GP(a) => $\forall$ bc. (Patient(b) ^ Contact-details(c) ^ of(a,b) ^ of(b,c) => read(a,c)))

which is equivalent to :

$\forall$ abc. (GP(a) ^ Patient(b) ^ Contact-details(c) ^ of(a,b) ^ of(b,c) => read(a,c))

Translation into Horn Clause form with side conditions

subject - invoker *owner & object* *relations* *method (invoker, object)*

$\forall$ **abc. (GP(a) ^ Patient(b) ^ Contact-details(c) ^ of(a,b) ^ of(b,c) => read(a,c))**

MethodRef (attribute, owner, invoker) *RoleRef* *side conditions*

Invoke-read (contact-details, b, a) <= GP(a) [Patient(b) ^ of(a,b)]

Invoke-read (contact-details, b, a) <= GP(a) [Patient(b) ^ gp-of (a,b)]

redundant if we know the database schema

Invoke-read (contact-details, b, a) <= GP(a) [gp-of (a,b)]

MethodRef (attribute, owner, invoker) *RoleRef* *side condition to be looked up in local database*

5. System monitoring and policy change management

Service-level policies can be deployed in many environments with lookup in local databases

Pseudo-natural language policy translation generates a formal representation of policy in first order logic

OASIS proves that role entries and method invocations are authorised by the service-level policy in place at the time

Role entries and method invocations are:

- **tagged with the policy version they were authorised by**
- **logged for audit, with the credentials presented**

High level policy changes trigger recomputation of service-level policies

Consistency checking is needed on every policy installation

Future work

policy store management using active database technology

EPSRC DIM grant Oct 2000 - Sept 2003

the goal: automatic management of expressed policy

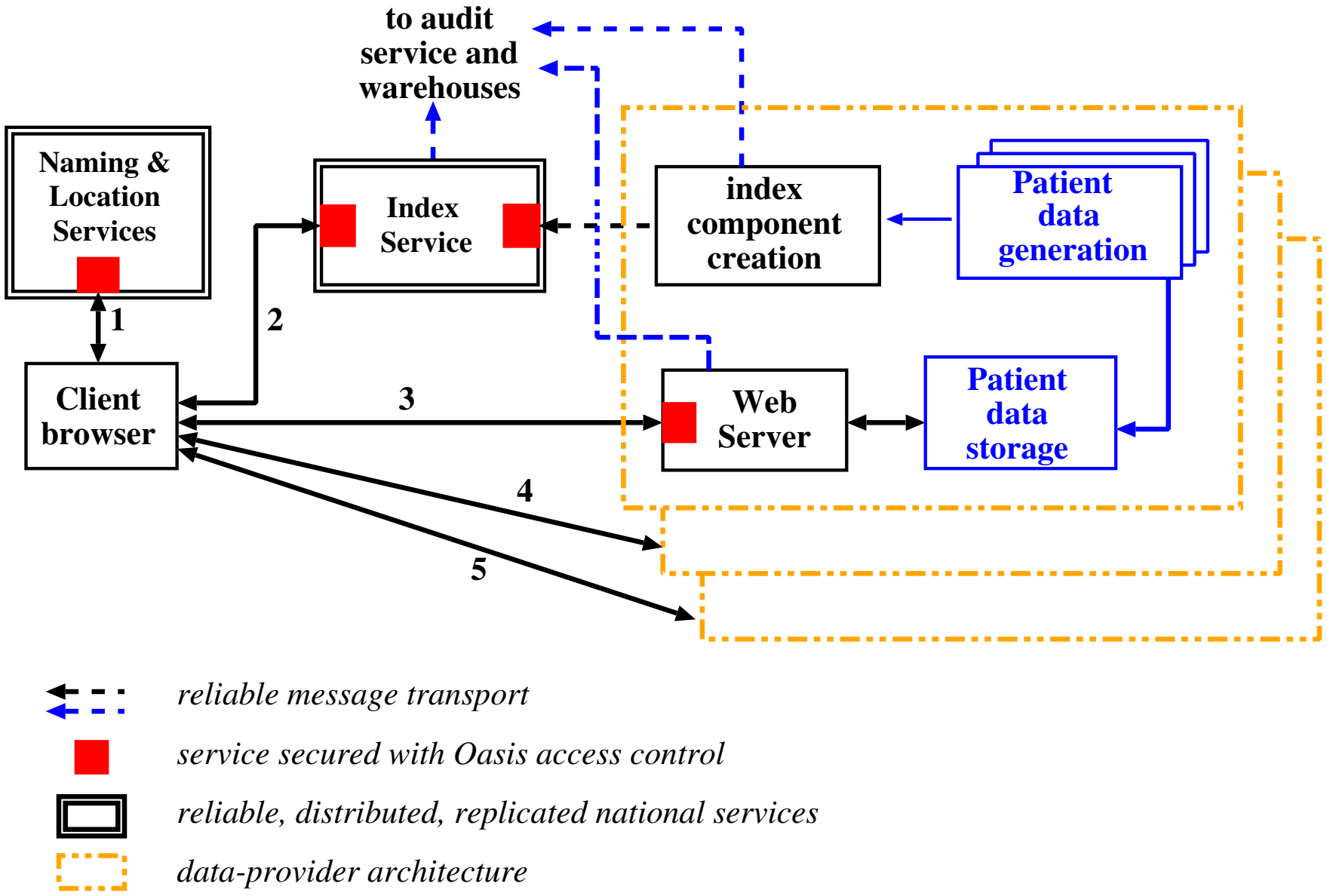
- **version control**
- **consistency checking**
- **enforcement**

Acknowledgements

Steve Pulman, now Professor of Computational Linguistics at Oxford

members of the Opera research group

Professor John Hine, sabbatical visitor from Wellington NZ


to audit
service and
warehouses
Naming &
Location
Services
Index
Service
index
component
creation
Patient
data
generation
1
2
Client
browser
3
Web
Server
Patient
data
storage
4
5
reliable message transport
service secured with Oasis access control
reliable, distributed, replicated national services
data-provider architecture