Authorization-Certificate Access Controlled Resources[1]

(An Application of Public-key Infrastructure and Digitally Signed Certificates)

William E. Johnston[2], Srilekha Mudumbai, Mary Thompson
Information and Computing Sciences Division
Ernest Orlando Lawrence Berkeley National Laboratory
University of California

[1] This work is supported by the Director, Office of Energy Research, Office of Computation and Technology Research, Mathematical, Information, and Computational Sciences Division, of the U. S. Department of Energy under Contract No. DE-AC03-76SF00098 with the University of California

[2] wejohnston@lbl.gov, 510-486-5014, http://www-itg.lbl.gov/~johnston

Outline:

1.0 Security for Widely Distributed Systems - Overall Approach

2.0 The WALDO Digital Library Environment (illustrative prototype)

3.0 The Digital Library Information Model

4.0 The Digital Library Policy Model

5.0 The General Security Model for Access Control

6.0 Implementation of the Policy Model for the Digital Library (ImgLib)

7.0 Certificate Infrastructure

8.0 The Security Architecture and Implementation

1.0 Security for Widely Distributed Systems - Overall Approach

Widely distributed systems and collaborative environments that involve

- multi-user instruments at national facilities
- widely distributed supercomputers and large-scale storage systems
- data sharing in restricted collaborations
- network-based multimedia collaboration channels

give rise to a range of requirement for distributed access control.

In all of these scenarios, the resource (data, instrument, computational and storage capacity, communication channel) has multiple stakeholders (typically the intellectual principals and policy makers), and each stakeholder will impose use-conditions on the resource. All of the use-conditions must be met simultaneously in order to satisfy the requirements for access. This idea is illustrated in the first figure.

Further, it is common that scientific collaboration tends to be diffuse, with the principals and stakeholders being geographically distributed, and multi-organizational. Therefore the access control mechanism must accommodate these circumstances by providing:

- distributed management of policy-based access control for all resources
- security (authentication, information integrity, confidentiality, etc.)
- mechanisms supporting the internal integrity of distributed systems

We also anticipate that the resulting infrastructure will support automated brokering and other policy-based negotiation for resources.

Goals

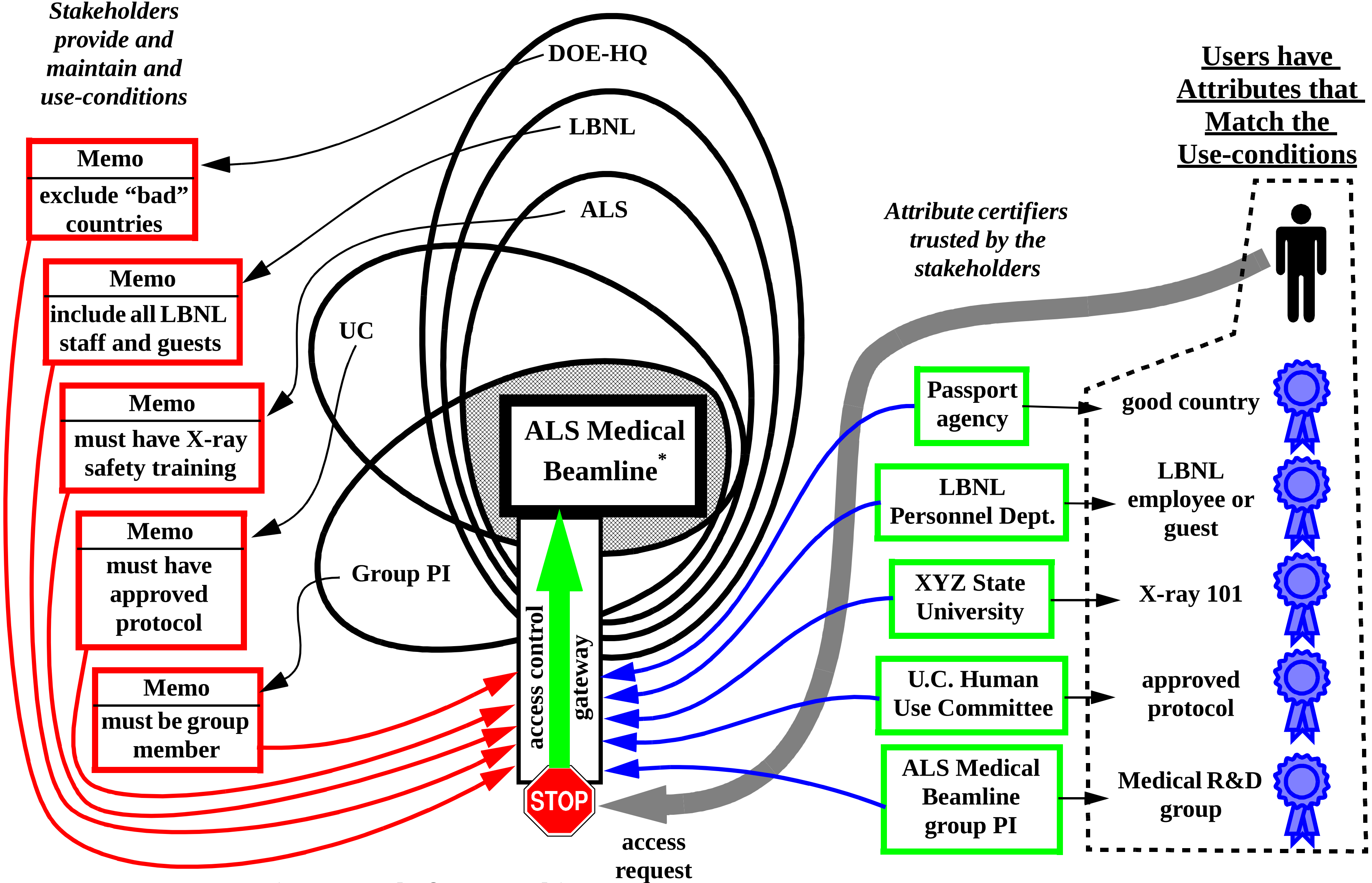
The goal for security in such distributed environments is to reflect, in a computing and communication based working environment, the general principles that have been established in society for policy-based resource access control.

Each responsible entity -- principals and stakeholders -- should be able to make their assertions (as they do now by signing, e.g., a policy statement) without reference to a mediator, and especially without reference to a centralized mediator (e.g. a system administrator) who must act on their behalf. The mechanism must be dynamic and easily used while maintaining strong assurances. Only in this way will computer-based security systems achieve the decentralization and utility needed for the scalability to support large distributed environments.

The computer systems based resource access control mechanisms should be able to collect all of the relevant assertions and make an unambiguous access decision without requiring entity-specific or resource-specific local, static configuration information that must be

Use-conditions are Imposed by Stakeholders

Users have Attributes that Match the Use-conditions



Stakeholders provide and maintain and use-conditions

Memo

exclude “bad” countries

Memo

include all LBNL staff and guests

Memo

must have X-ray safety training

Memo

must have approved protocol

Memo

must be group member

DOE-HQ

LBNL

ALS

UC

Group PI

ALS Medical Beamline*

access control gateway

STOP

access request

Attribute certifiers trusted by the stakeholders

Passport agency

LBNL Personnel Dept.

XYZ State University

U.C. Human Use Committee

ALS Medical Beamline group PI

good country

LBNL employee or guest

X-ray 101

approved protocol

Medical R&D group

Access is Granted after Matching Use-conditions and Attributes

Figure 1. Societal Access Control Model

* hypothetical

centrally administered. (This does not imply that such specific configuration is precluded, only that it should not be required.) The mechanism (Figure 2) should also be based on, and evolve with, the emerging, commercially supplied, public-key certificate infrastructure components.

Expected Benefits

For security to be successful in distributed environments -- providing both protection and policy enforcement -- each principal entity should have no more nor less involvement than they do in the currently established procedure that operates in the absence of computer security. That is, those who have the authority to set access conditions or use-conditions by, e.g., holographically signing statements in a paper environment, will digitally sign functionally equivalent statements in a distributed computing based environment. The use of these credentials should b automatic, and the functions of checking credentials, auditing, etc. are performed by appropriate entities in either circumstance.

The expected advantages of computer-based systems are in maintaining access control policy, but with greatly increased independence from temporal and spatial factors (e.g. time zone differences and geographic separation), together with automation of redundant tasks such as credential checking and auditing.

The intended outcome is that the scientific community will more easily share expensive resources, unique systems, sensitive data, etc.

A further expected benefit is that this sort of a security infrastructure should provide the basis of automated brokering of resources that precede the construction of dynamically, and just-in-time configured systems to support, e.g., scientific experiments with transient computing, communication, or storage requirements.

Authorization Based Distributed Security

An approach that addresses the general goals noted above can be based on authorization and attribute certificates. These digitally signed documents have the characteristic that they assert document validity without physical presence of the signer, or physical possession of holographically signed documents. The result is that the digitally signed documents that provide the assertions of the principals, stakeholders, attribute authorities, etc., may be generated, represented, used, and verified independent of time or location.

Other parts of the approach are implemented through the use of “authorities” that provide delegation mechanisms and assured information as digitally signed documents: identity authorities connect human entities to digital signatures, stakeholder authorities provide use-conditions, attribute authorities attest to user characteristics, etc. Additional components include reliable mechanisms for distributing and verifying the digitally signed documents, mechanisms that match use-conditions and attributes, and resource access control mechanisms that use the resulting credentials to enforce policy for the specific resource.

Architecture for Distributed Management of Fine-grained Access Control

A prototype implementation (see [/1/]) that is addressing the problem of distributed management of access control to limited, valuable, or large-scale resources / data / objects -- e.g. large scientific instruments, distributed supercomputers, sensitive but unclassified databases (e.g.

Internet vulnerability and incident databases) is providing some experience with decentralized security environments. The elements of the prototype include:

1) Fully distributed resource management and access: In our target environment, the resource users, resources owners, and other stakeholders, are remote from the protected resource -- the norm in, among others, large-scale scientific instrument environments.

2) Multiple stakeholders: All significant resources have many stakeholders, all of whom will provide their own use-conditions. These use-conditions are specified in the environment of the stakeholder and then provided to the resource access control mechanism.

3) Attribute-based access policy: Users are permitted access to resources based on their attributes that satisfy the owner and stakeholder use-conditions. These attributes are attested to by trusted third parties.

4) Validation of the right-of-access is typically used to establish the security context for an underlying security system such as SSL (e.g. between Web browser and servers, see /3/) and GSS (secure messaging between components of distributed systems, see /4/).

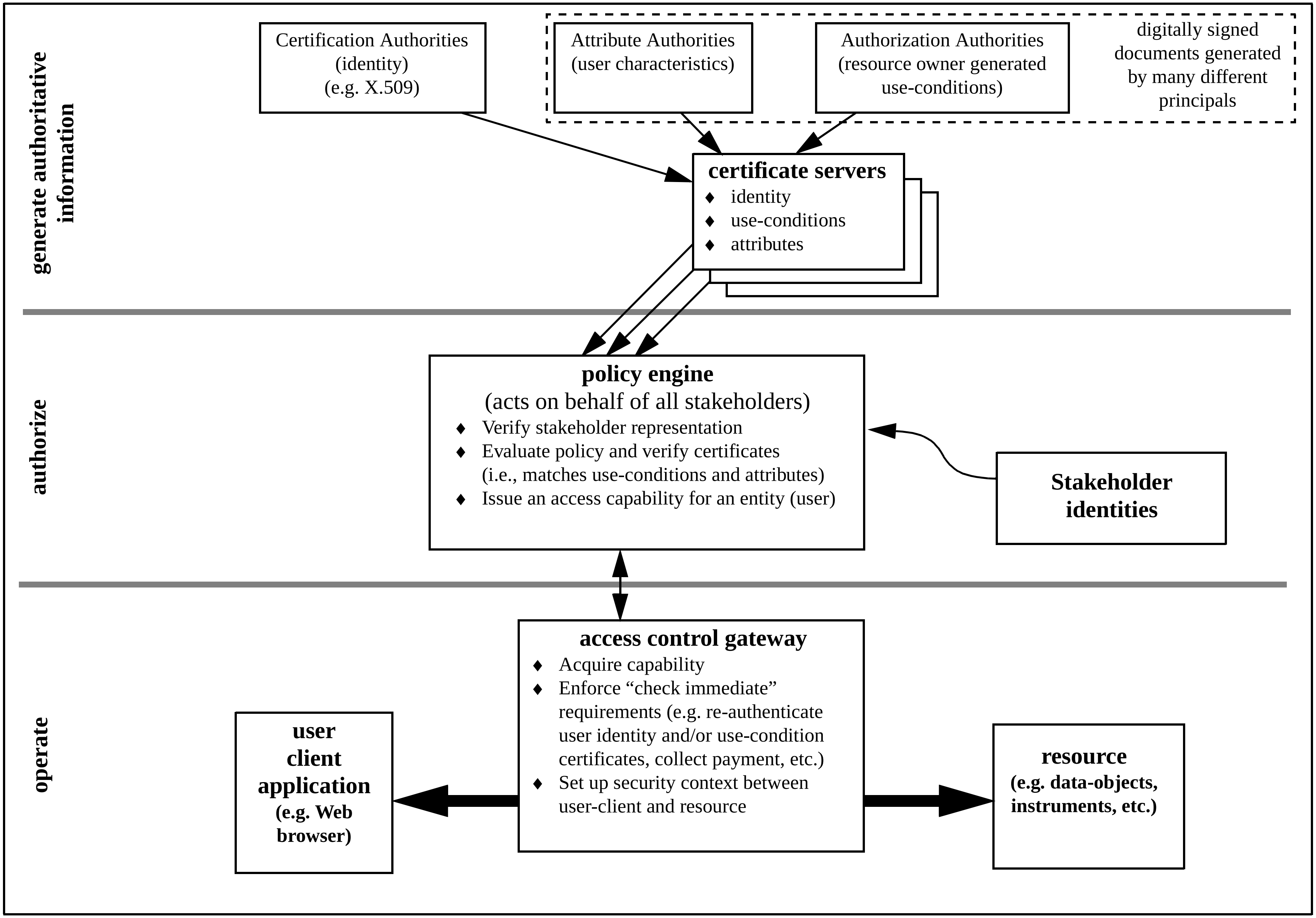
The prototype provides for objects / data / resource owners and other stakeholders to be able to remotely exercise control over access to the resource, for legitimate users (those that satisfy the use-conditions of the resource stakeholders) to obtain easy access, and for unqualified / un-authorized users to be strongly denied access. The architecture is illustrated in the figure below.

In addition to the technology issues of integrity and management of the security system and associated computing platforms, useful security is as much (or more) a deployment and user-ergonomics issue. That is, the problem is as much trying to find out how to integrate good security into the end-user (e.g. scientific) environment so that it will be used, trusted to provide the protection that it claims, easily administered, and genuinely useful in the sense of “providing distributed enterprise capabilities” (that is, providing new functionality that supports distributed organizations and operation), as it is trying to address the more traditional security issues.

While the security architecture provides the basic technology, in order to accomplish a useful service the architecture must be applied in such a way that the resources are protected as intended by the principals. This involves understanding the information / resource use and structure model, and developing a policy model that will support the intended access control. These must be supported by a security model that specifies how the elements of the security architecture and infrastructure will implement the policy model.

The current implementation of this architecture (see /5/) provides a policy engine that:

- implements both flat and hierarchical, multiple use-condition policy models
- uses X.509 identity certificates, and ad-hoc attribute and use-condition certificates obtained from Web and LDAP servers
- provides a policy evaluation service to the Apache Web server and an implementation of SPKM/GSS



generate authoritative information

Certification Authorities (identity) (e.g. X.509)

Attribute Authorities (user characteristics)

Authorization Authorities (resource owner generated use-conditions)

digitally signed documents generated by many different principals

certificate servers

- identity
- use-conditions
- attributes

authorize

policy engine

(acts on behalf of all stakeholders)

- Verify stakeholder representation
- Evaluate policy and verify certificates (i.e., matches use-conditions and attributes)
- Issue an access capability for an entity (user)

Stakeholder identities

operate

user client application (e.g. Web browser)

access control gateway

- Acquire capability
- Enforce "check immediate" requirements (e.g. re-authenticate user identity and/or use-condition certificates, collect payment, etc.)
- Set up security context between user-client and resource

resource (e.g. data-objects, instruments, etc.)

Figure 2. The Overall Architecture of the Authorization Certificate Approach

/1/ “A Use-Condition Centered Approach to Authenticated Global Capabilities: Security Architectures for Large-Scale Distributed Collaboratory Environments”, William Johnston and Case Larsen, January 1997. Available at: http://www-itg.lbl.gov/security/publications.html

/2/ “A Public Key Infrastructure for DOE Security Research” Findings from U. S. Department of Energy, Joint Energy Research / Defense Programs Computing-related Security Research Requirements - Workshop-II Dec 11-13, 1996, Albuquerque, New Mexico. Available at http://www-itg.lbl.gov/security

/3/ “The SSL Protocol” http://live.netscape.com/newsref/std/SSL.html

/4/ “Generic Security Service Application Program Interface”, John Linn, Sep 1993. Available at http://ds.internic.net/rfc/rfc1508.txt. Also see more recent and related drafts at the IETF Common Authentication Technology home page (http://www.ietf.cnri.reston.va.us/html.charters/cat-charter.html) and at http://www.ietf.cnri.reston.va.us/ids.by.wg/cat.html.

/5/ See http://www-itg.lbl.gov/security

The overall security approach is illustrated by its use in a hierarchically structured, Web-based object repository.

The Prototype Environment

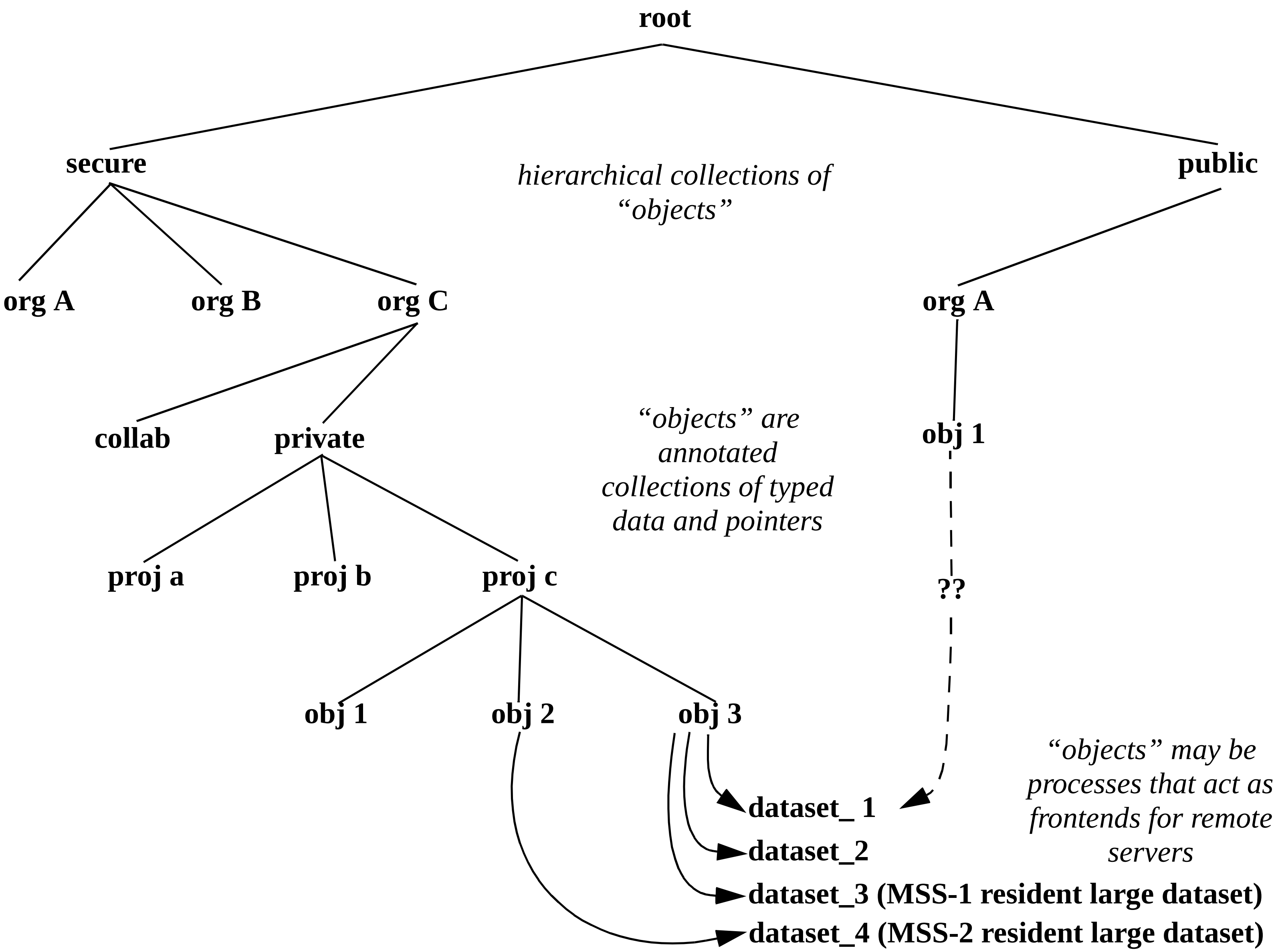
2.0 The WALDO[3] Digital Library Environment

2.1) WALDO -- the Wide Area, Large-Data-Object system -- has a digital library / repository component ("DL") that is used to store and manage federated objects, and which is a service offered to multiple geographically dispersed groups

- 2.1.1) The resources are general Web-based "objects" (or collections thereof)
- 2.1.2) Access is to be controlled on a per-object basis
- 2.1.3) Both public and restricted information will be managed
- 2.1.4) The membership of access groups is assumed to cut across organizational boundaries

[3] See "Real-Time Digital Libraries based on Widely Distributed, High Performance Management of Large-Data-Objects" at http://www-itg.lbl.gov/WALDO

The WALDO Digital Library Environment



root

secure

public

hierarchical collections of “objects”

org A

org B

org C

org A

collab

private

obj 1

“objects” are annotated collections of typed data and pointers

proj a

proj b

proj c

??

obj 1

obj 2

obj 3

“objects” may be processes that act as frontends for remote servers

dataset_ 1

dataset_2

dataset_3 (MSS-1 resident large dataset)

dataset_4 (MSS-2 resident large dataset)

Figure 3. The WALDO Digital Library Information Model

The WALDO Digital Library Environment

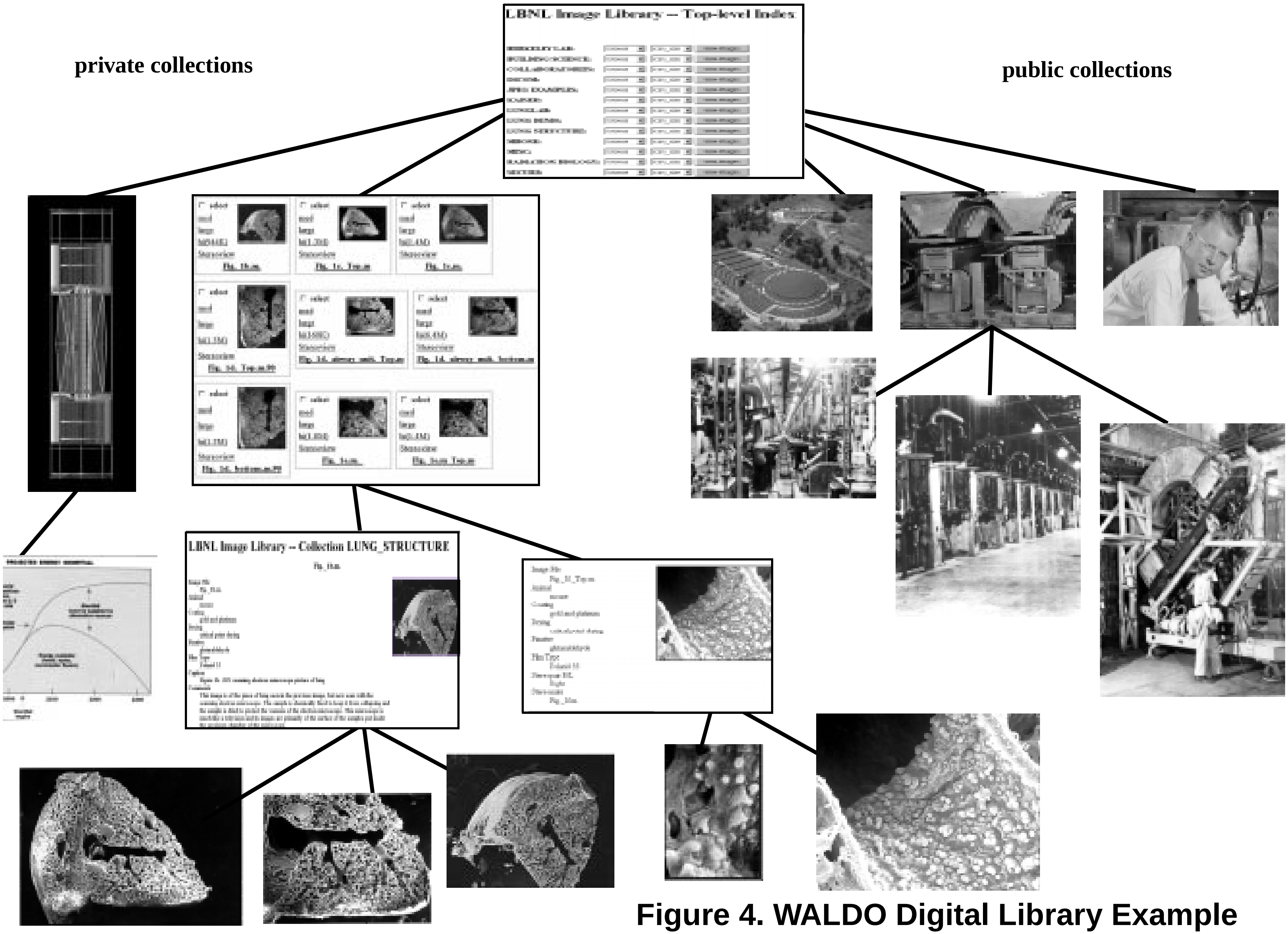

private collections
public collections
LBNL Image Library -- Top-level Index
LBNL Image Library -- Collection LUNG_STRUCTURE

Figure 4. WALDO Digital Library Example

The WALDO Digital Library Environment

2.2) There are multiple stakeholders:

2.2.1) - policy makers establish high-level access requirements

- resource owners / principals establish collaboration requirements

2.2.2) Webmasters allocate server resources (e.g. storage space)

2.2.3) DL data owners create objects by depositing data and maintaining metadata in collections defined by curators

2.2.4) DL curators manage object collections and establish use-conditions for those collections

- curators may delegate read/write/create and curation functions

The WALDO Digital Library Environment

2.3) The Web server is assumed to be a <u>secure computing platform</u>

(The access control mechanism does not protect against malicious system intrusion, though confidentiality in the face of such attack is feasible. E.g., the data owners can encrypt data prior to storing in the DL and distributed the decryption keys independently of the DL.)

2.4) The Webmaster is a <u>trusted third-party</u> who encodes the ownership of resources (binds resources and owners)

3.0The Digital Library Information Model

To summarize, the information model consists of:

3.1) hierarchically organized object collections managed by curators

3.2) the object definitions are a mix of typed data, textual metadata, and URLs referencing local and remote independent data sets and servers

3.3) the objects components are accessed only through the object definitions, i.e. access control is applied to objects
(this is not true in the general WALDO environment)

4.0 The Digital Library Policy Model

A *policy model* is built on a general security model in a way that will support the access policies needed in a particular resource domain.

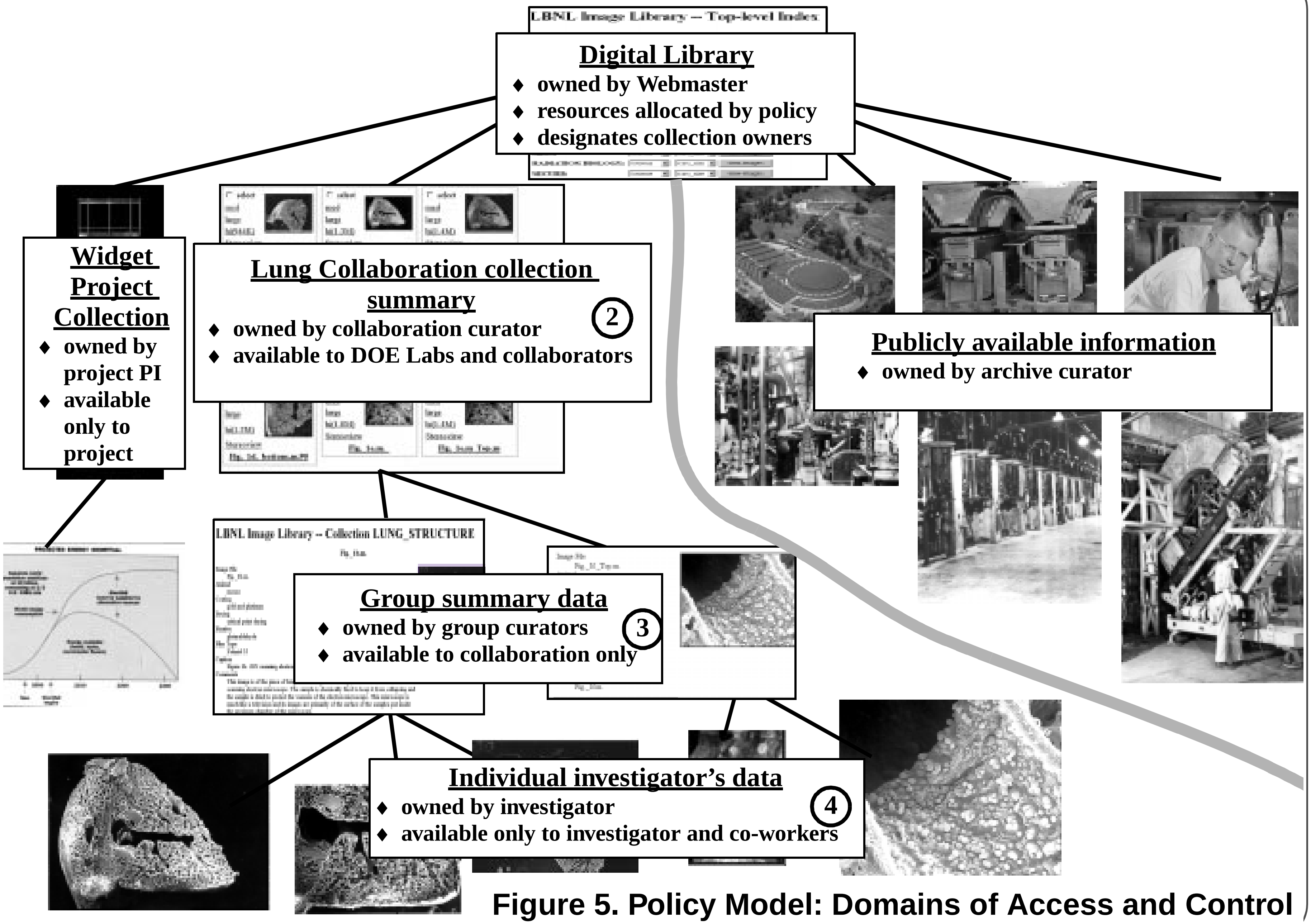
The characteristics of a particular policy model - e.g. hierarchical authority with delegation - is a function of the resource / application domain.

4.1) Access rights are specified by “many” independent stakeholders:

- **data owners**
- **collection curators**
- **policy makers**

4.2) Access “groups” are defined by required attributes, and consist of the collection of entities that possess those attributes

Policy Model


LBNL Image Library -- Top-level Index
LBNL Image Library -- Collection LUNG_STRUCTURE

Digital Library

- owned by Webmaster
- resources allocated by policy
- designates collection owners

Widget Project Collection

- owned by project PI
- available only to project

Lung Collaboration collection summary 2

- owned by collaboration curator
- available to DOE Labs and collaborators

Publicly available information

- owned by archive curator

Group summary data 3

- owned by group curators
- available to collaboration only

Individual investigator's data 4

- owned by investigator
- available only to investigator and co-workers

Figure 5. Policy Model: Domains of Access and Control

Policy Model

4.3) Access rights are effectively hierarchical, with increasing restriction “down” from the root (top) of the hierarchy (access restrictions are inherited)

(This provides for overall policy in the form of top-level use-conditions.)

4.4) “Action” on resources is controlled independently of “access” (e.g. for the various resource actions - read, write (object creation), and DL curation (management of collections))

4.5) DL curators may delegate the authority to manage subcollections, and therefore curators exist in a hierarchical relationship with respect to access restrictions

4.6) Authority delegation (i.e., restricted delegation) is provided by allowing subordinate curators to specify access rights that are not dis-allowed at a higher level

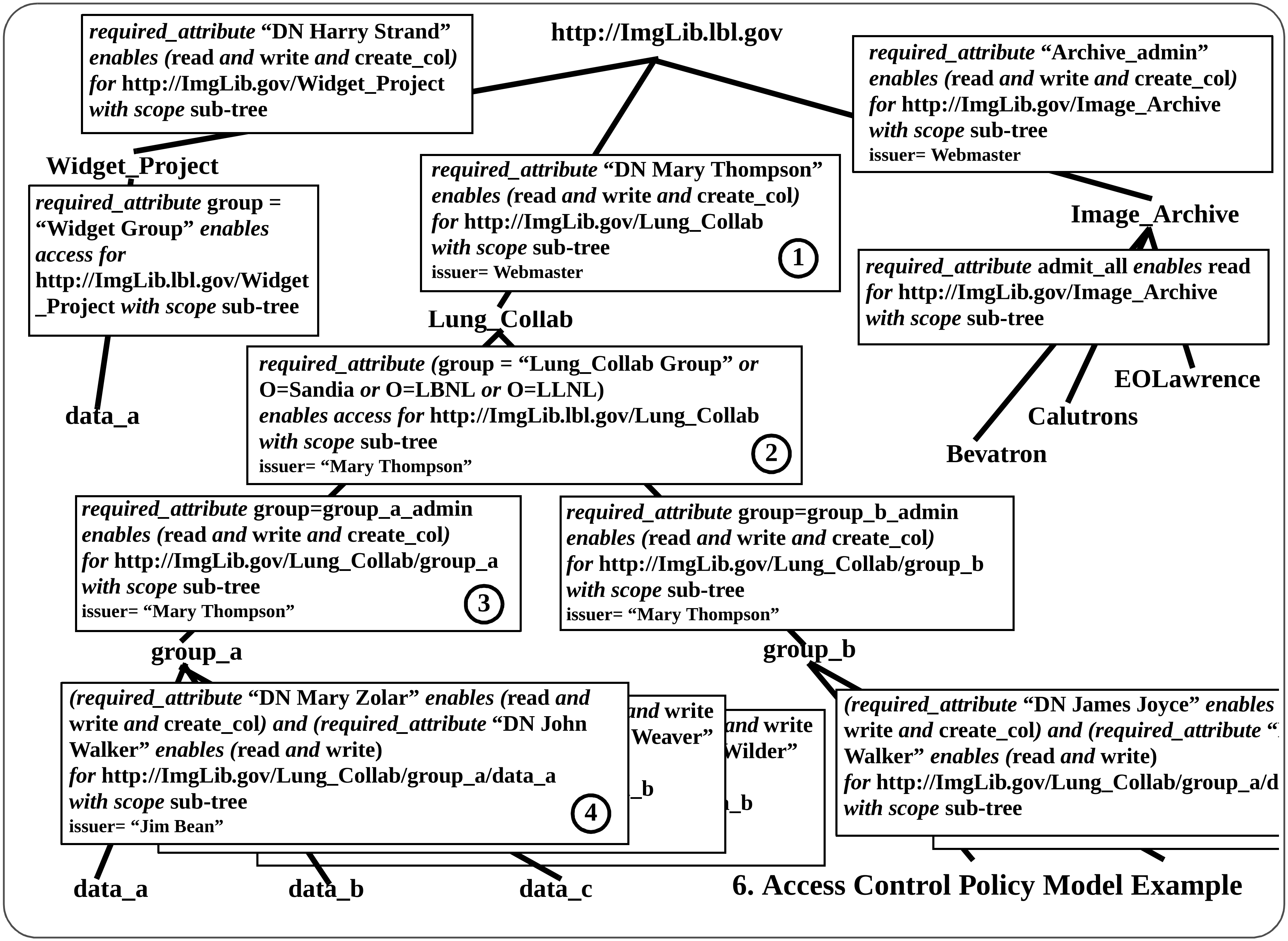
Policy Model

The general access control mechanism is that stakeholders post use-conditions (as certificates) that define access groups in terms of required attributes. The attributes are attested to be trusted third-parties that also post these certificates. (“group” is a general term unrelated to any organizational unit.)

In this policy model the authority of the stakeholder to require use-conditions is maintained by where in the object hierarchy users are permitted to write data.

However, at any point the hierarchy can be flattened in order to allow users complete flexibility to determine their own access policy (there might be only one or two top-level requirements, after which everyone defines their own access requirements)

Policy Model



http://ImgLib.lbl.gov

required_attribute “DN Harry Strand”
*enables (*read *and* write *and* create_col*)*
for http://ImgLib.gov/Widget_Project
with scope sub-tree

Widget_Project

required_attribute group = “Widget Group” *enables access for* http://ImgLib.lbl.gov/Widget_Project *with scope* sub-tree

data_a

required_attribute “DN Mary Thompson”
*enables (*read *and* write *and* create_col*)*
for http://ImgLib.gov/Lung_Collab
with scope sub-tree
issuer= Webmaster
1

Lung_Collab

*required_attribute (*group = “Lung_Collab Group” *or* O=Sandia *or* O=LBNL *or* O=LLNL)
enables access for http://ImgLib.lbl.gov/Lung_Collab
with scope sub-tree
issuer= “Mary Thompson”
2

required_attribute “Archive_admin”
*enables (*read *and* write *and* create_col*)*
for http://ImgLib.gov/Image_Archive
with scope sub-tree
issuer= Webmaster

Image_Archive

required_attribute admit_all *enables* read
for http://ImgLib.gov/Image_Archive
with scope sub-tree

EOLawrence

Calutrons

Bevatron

required_attribute group=group_a_admin
*enables (*read *and* write *and* create_col*)*
for http://ImgLib.gov/Lung_Collab/group_a
with scope sub-tree
issuer= “Mary Thompson”
3

required_attribute group=group_b_admin
*enables (*read *and* write *and* create_col*)*
for http://ImgLib.gov/Lung_Collab/group_b
with scope sub-tree
issuer= “Mary Thompson”

group_a

group_b

(required_attribute “DN Mary Zolar” *enables (*read *and* write *and* create_col*) and (required_attribute* “DN John Walker” *enables (*read *and* write)
for http://ImgLib.gov/Lung_Collab/group_a/data_a
with scope sub-tree
issuer= “Jim Bean”
4

nd write
Weaver”
_b

and write
Wilder”
_b

(required_attribute “DN James Joyce” *enables*
write *and* create_col*) and (required_attribute* “
Walker” *enables (*read *and* write)
for http://ImgLib.gov/Lung_Collab/group_a/d
with scope sub-tree

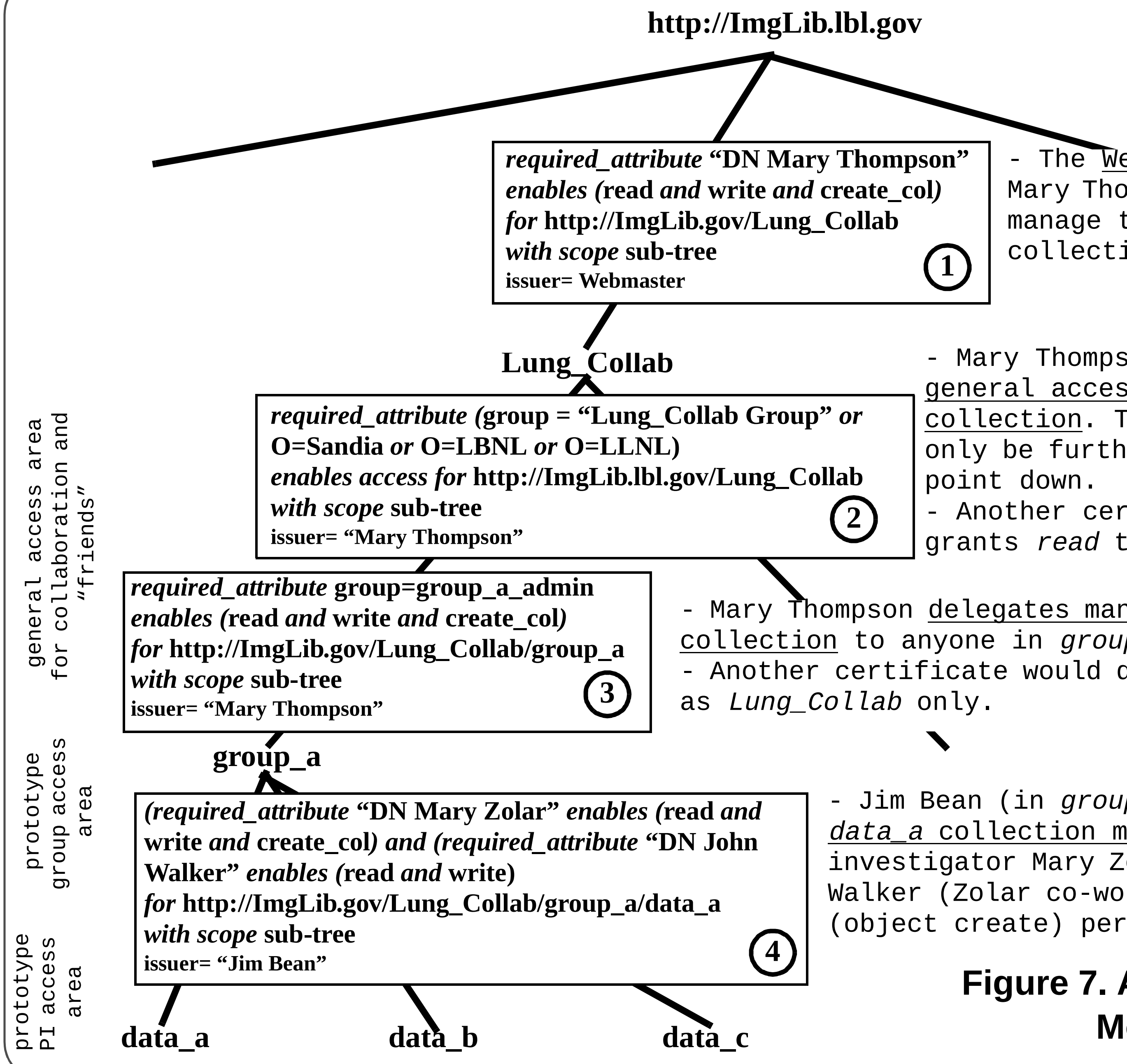
data_a

data_b

data_c

6. Access Control Policy Model Example

Policy Model



http://ImgLib.lbl.gov

required_attribute **"DN Mary Thompson"**
*enables (***read** *and* **write** *and* **create_col***)*
for **http://ImgLib.gov/Lung_Collab**
with scope **sub-tree**
issuer= Webmaster
1

- The Webmaster delegates to Mary Thompson the authority to manage the *Lung_Collab* collection.

Lung_Collab

required_attribute **(group = "Lung_Collab Group"** *or* **O=Sandia** *or* **O=LBNL** *or* **O=LLNL)**
enables access for **http://ImgLib.lbl.gov/Lung_Collab**
with scope **sub-tree**
issuer= "Mary Thompson"
2

- Mary Thompson establishes the general access group for the collection. This access group may only be further restricted from this point down.
- Another certificate at this level grants *read* to this same group.

required_attribute **group=group_a_admin**
*enables (***read** *and* **write** *and* **create_col***)*
for **http://ImgLib.gov/Lung_Collab/group_a**
with scope **sub-tree**
issuer= "Mary Thompson"
3

- Mary Thompson delegates management of the *group_a* collection to anyone in *group_a_admin*.
- Another certificate would define the access group as *Lung_Collab* only.

group_a

(required_attribute **"DN Mary Zolar"** *enables (***read** *and* **write** *and* **create_col***) and (required_attribute* **"DN John Walker"** *enables (***read** *and* **write)**
for **http://ImgLib.gov/Lung_Collab/group_a/data_a**
with scope **sub-tree**
issuer= "Jim Bean"
4

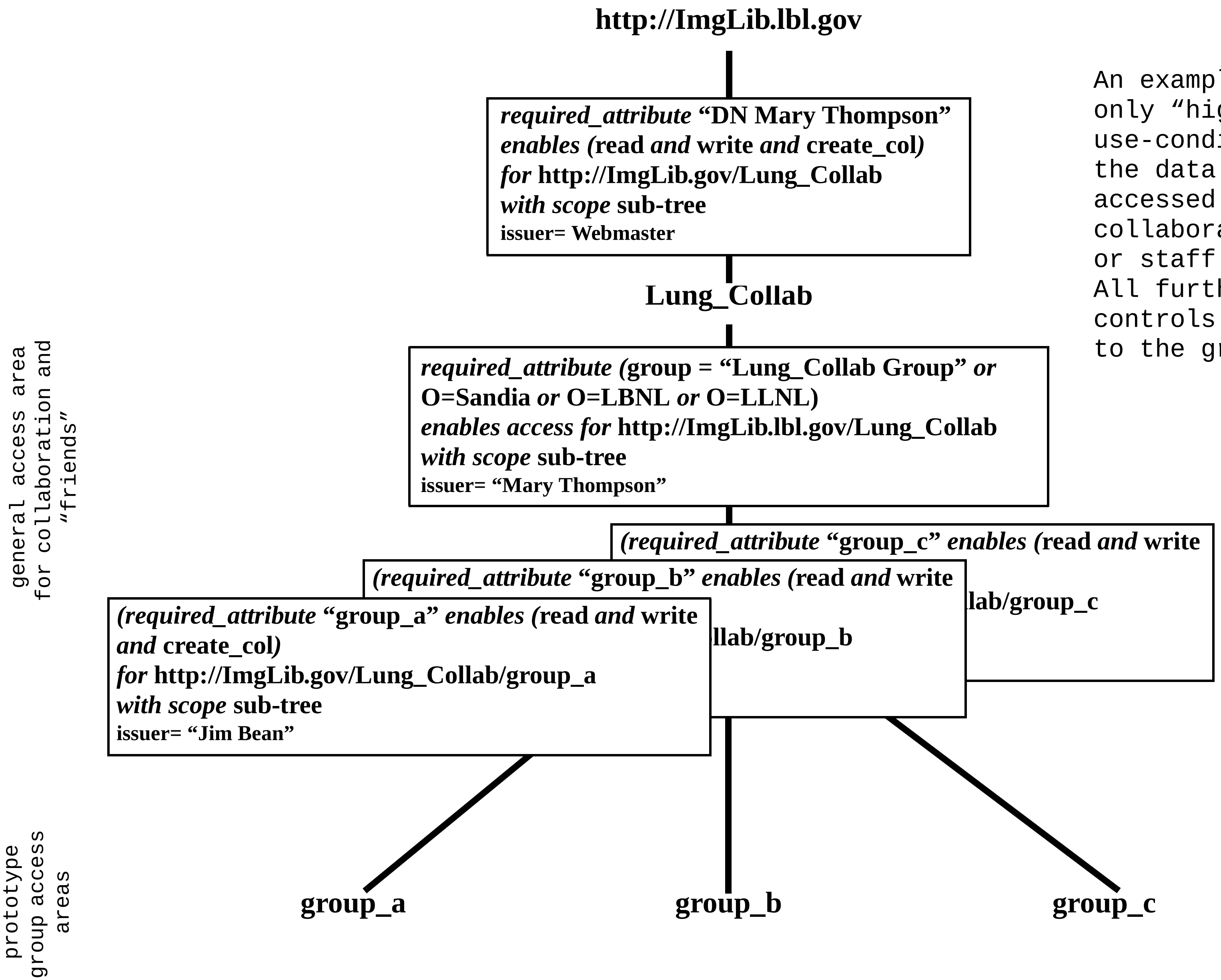
- Jim Bean (in *group_a_admin*) delegates *data_a* collection management to investigator Mary Zolar and grants John Walker (Zolar co-worker) read and write (object create) permission.

data_a **data_b** **data_c**

general access area for collaboration and "friends"
prototype group access area
prototype PI access area

Figure 7. Access Control Policy Model Example

Policy Model



http://ImgLib.lbl.gov

required_attribute “DN Mary Thompson”
enables (read *and* write *and* create_col*)*
for http://ImgLib.gov/Lung_Collab
with scope sub-tree
issuer= Webmaster

Lung_Collab

required_attribute (group = “Lung_Collab Group” *or*
O=Sandia *or* O=LBNL *or* O=LLNL)
enables access for http://ImgLib.lbl.gov/Lung_Collab
with scope sub-tree
issuer= “Mary Thompson”

(required_attribute “group_c” *enables* (read *and* write
llab/group_c

(required_attribute “group_b” *enables* (read *and* write
llab/group_b

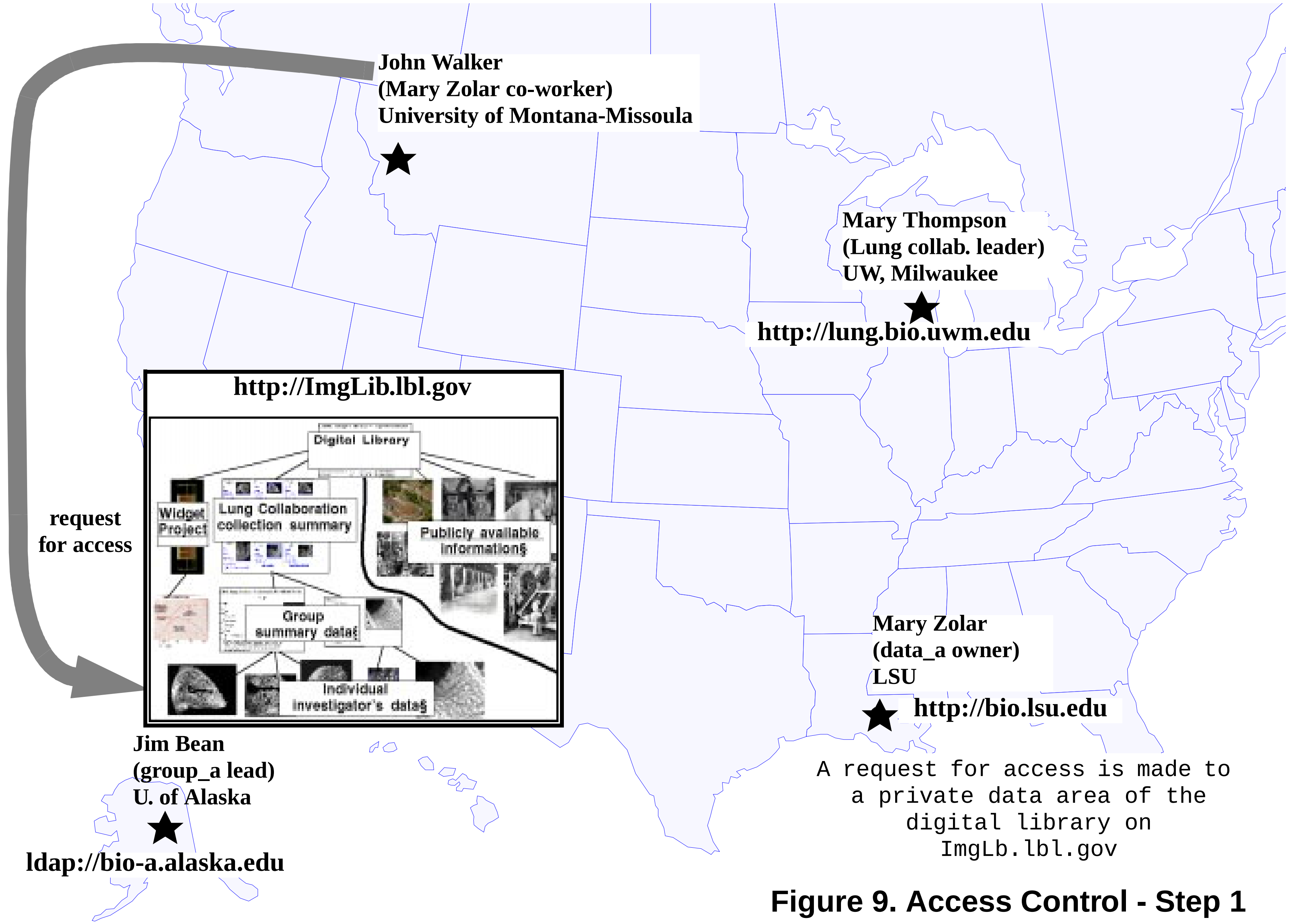
(required_attribute “group_a” *enables* (read *and* write
and create_col*)*
for http://ImgLib.gov/Lung_Collab/group_a
with scope sub-tree
issuer= “Jim Bean”

group_a group_b group_c

An example where the only “high-level” use-condition is that the data may only be accessed by collaboration members or staff of the Labs. All further access controls are delegated to the groups.

Figure 8. Access Control Policy Model - “flat hierarchy” Example

Policy Model


http://ImgLib.lbl.gov
Digital Library
Widget Project
Lung Collaboration collection summary
Publicly available information§
Group summary data§
Individual investigator's data§

John Walker
(Mary Zolar co-worker)
University of Montana-Missoula

Mary Thompson
(Lung collab. leader)
UW, Milwaukee

http://lung.bio.uwm.edu

request
for access

Mary Zolar
(data_a owner)
LSU

http://bio.lsu.edu

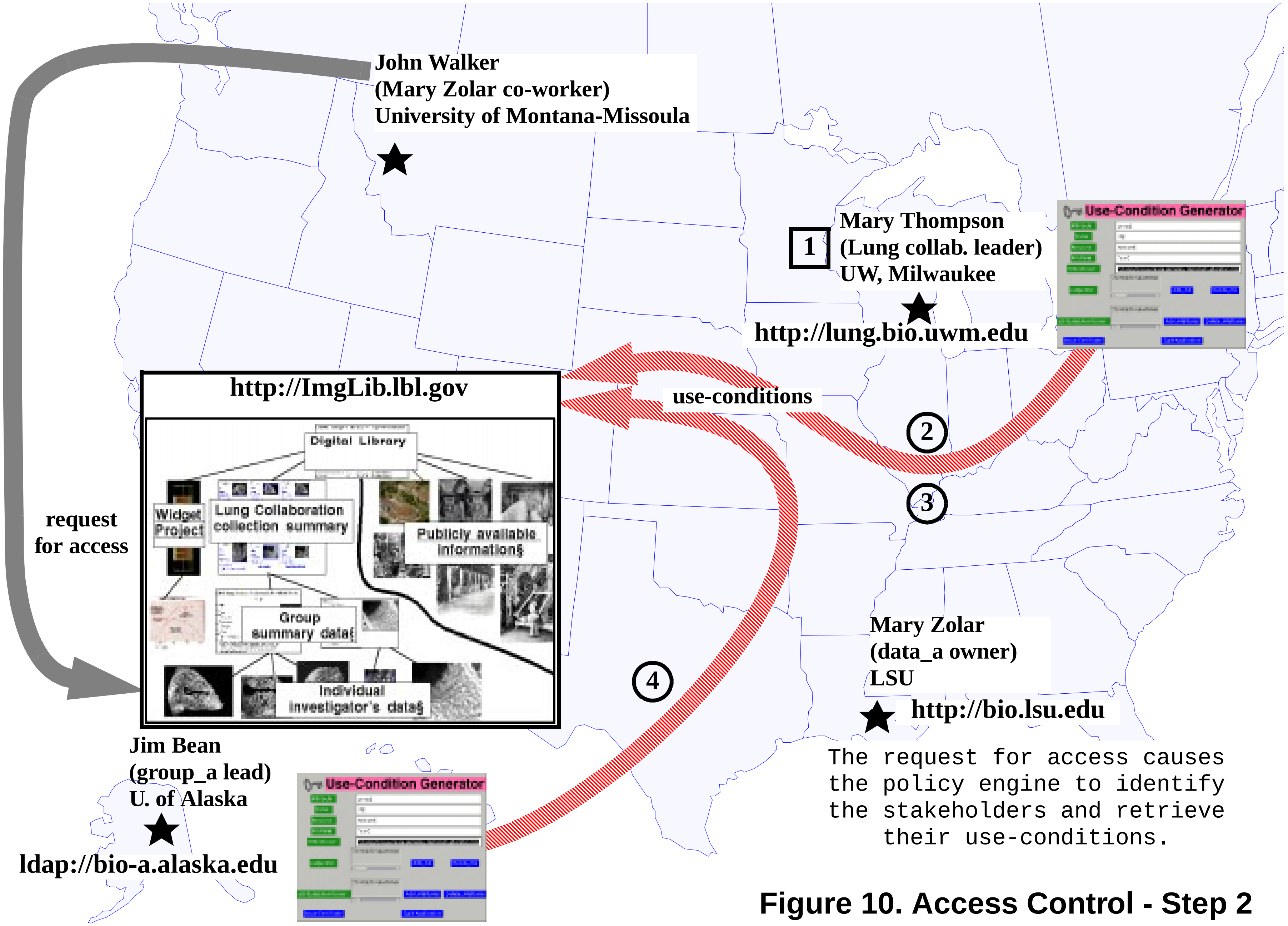
Jim Bean
(group_a lead)
U. of Alaska

ldap://bio-a.alaska.edu

A request for access is made to
a private data area of the
digital library on
ImgLb.lbl.gov

Figure 9. Access Control - Step 1

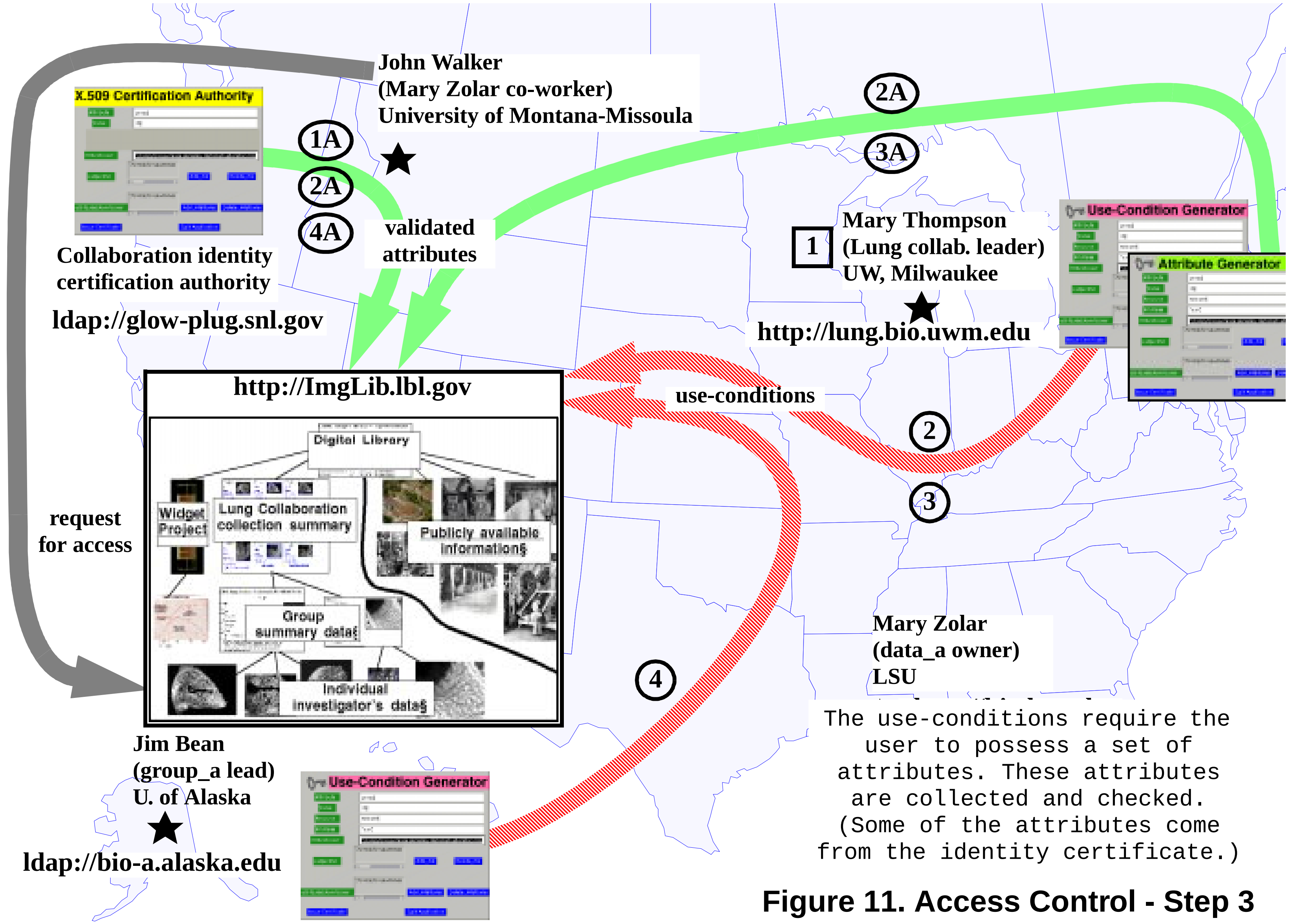
Policy Model



The request for access causes the policy engine to identify the stakeholders and retrieve their use-conditions.

Figure 10. Access Control - Step 2

Policy Model



X.509 Certification Authority

Collaboration identity certification authority

ldap://glow-plug.snl.gov

John Walker
(Mary Zolar co-worker)
University of Montana-Missoula

1A

2A

4A

validated attributes

2A

3A

1

Mary Thompson
(Lung collab. leader)
UW, Milwaukee

http://lung.bio.uwm.edu

Use-Condition Generator

Attribute Generator

http://ImgLib.lbl.gov

use-conditions

2

3

4

request for access

Mary Zolar
(data_a owner)
LSU

Jim Bean
(group_a lead)
U. of Alaska

ldap://bio-a.alaska.edu

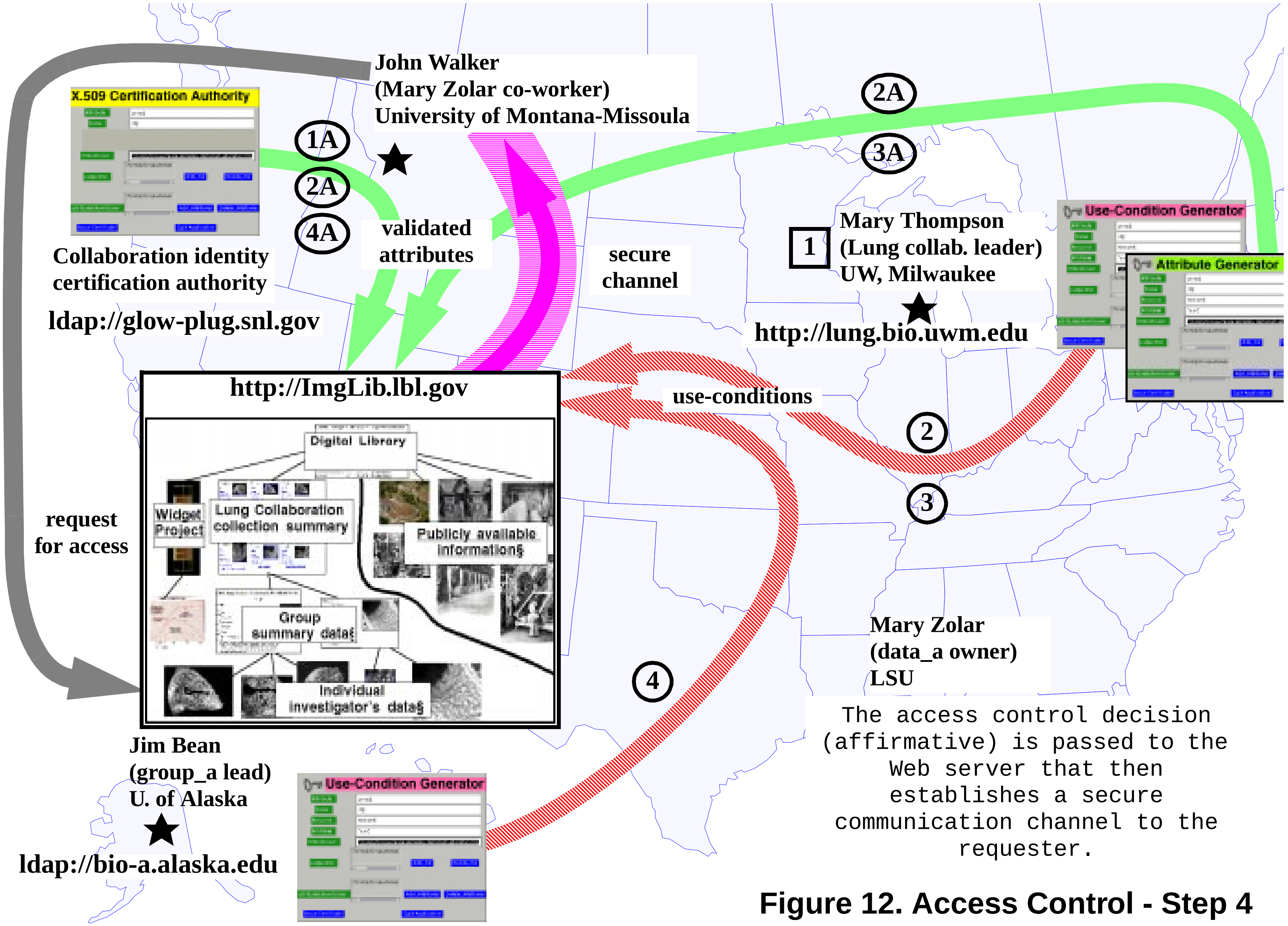
Use-Condition Generator

Digital Library
Widget Project
Lung Collaboration collection summary
Publicly available information§
Group summary data§
Individual investigator's data§

The use-conditions require the user to possess a set of attributes. These attributes are collected and checked. (Some of the attributes come from the identity certificate.)

Figure 11. Access Control - Step 3

Policy Model


X.509 Certification Authority
1A
2A
4A
2A
3A
1
Use-Condition Generator
Attribute Generator
Digital Library
Widget Project
Lung Collaboration collection summary
Publicly available information§
Group summary data§
Individual investigator's data§
2
3
4
Use-Condition Generator

John Walker
(Mary Zolar co-worker)
University of Montana-Missoula

Collaboration identity
certification authority

ldap://glow-plug.snl.gov

validated
attributes

secure
channel

Mary Thompson
(Lung collab. leader)
UW, Milwaukee

http://lung.bio.uwm.edu

http://ImgLib.lbl.gov

use-conditions

request
for access

Mary Zolar
(data_a owner)
LSU

Jim Bean
(group_a lead)
U. of Alaska

ldap://bio-a.alaska.edu

The access control decision (affirmative) is passed to the Web server that then establishes a secure communication channel to the requester.

Figure 12. Access Control - Step 4

Policy Model

A “policy engine” collects the use-conditions and corresponding attributes and validates them. At that point the job of the access control system is essentially complete. The affirmative or negative response is passed to the security system of the application, which then provides access or denies it. Most applications will probably make use of a standard security system to establish secure, end-to-end communication if they are doing access control. The current policy engine is interfaced to

1) the Apache Web server using the Secure Sockets Layer (ssl) security system, and

2) the GSS/API secure messaging system that is used to implement secure distributed applications.

In both cases the access control policy engine and the security system that supports the application are separate and independent.

5.0 The General Security Model for Access Control

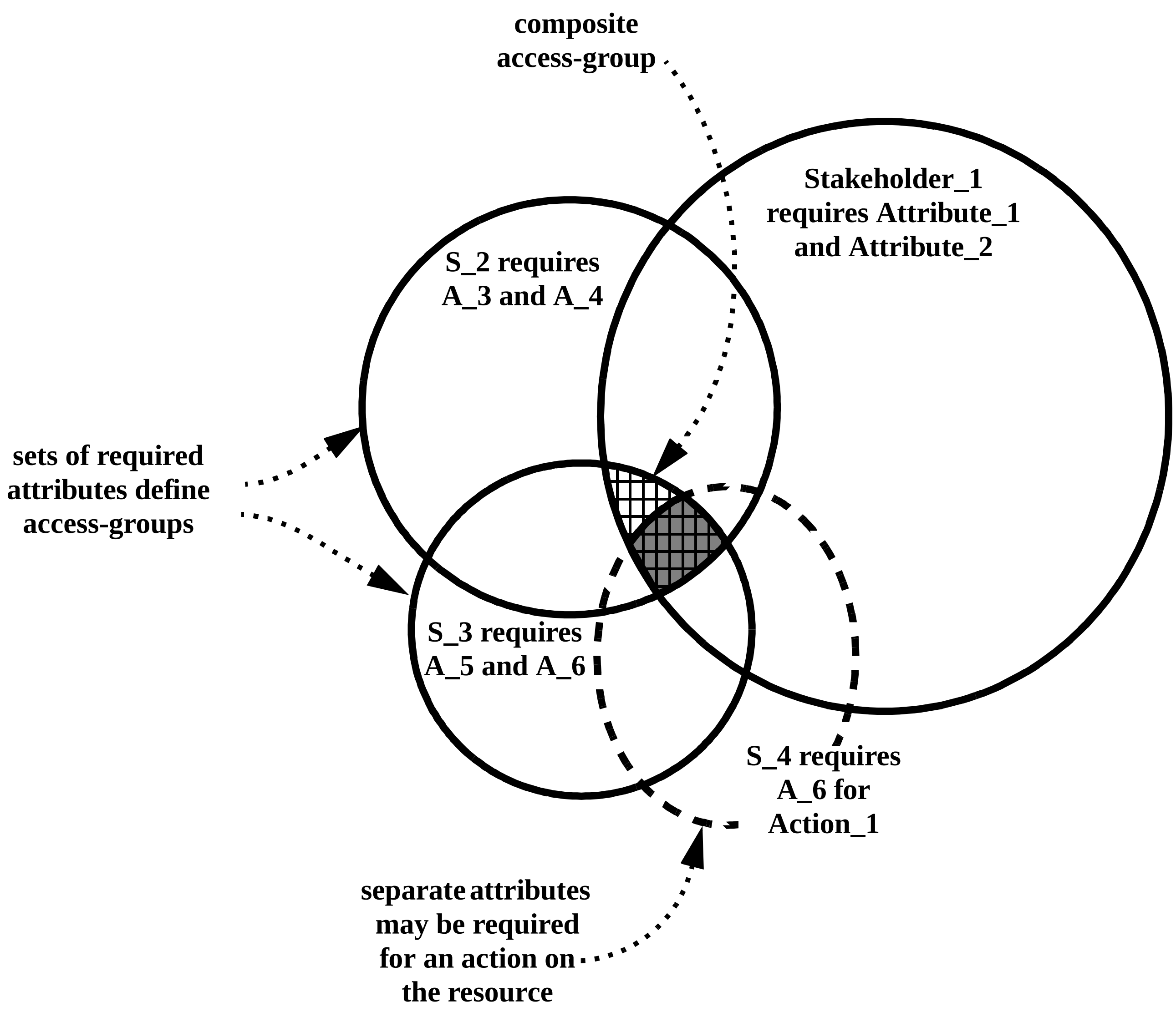
The goal of the security model is to be able to support a variety of policy models.

The security model provides for controlling access to resources via restrictions imposed by several types of use-conditions that are defined independently by multiple stakeholders:

- access groups are defined implicitly by requiring a set of attributes
- actions on resources may be further restricted by requiring additional attributes (evaluated independent of access)
- operational requirements (e.g. time-of-day) are defined and satisfied by “data fields” in attribute certificates

These use-conditions are satisfied by (certified) attributes of those entities trying to gain access to resources.

Security Model



composite
access-group

Stakeholder_1
requires Attribute_1
and Attribute_2

S_2 requires
A_3 and A_4

sets of required
attributes define
access-groups

S_3 requires
A_5 and A_6

S_4 requires
A_6 for
Action_1

separate attributes
may be required
for an action on
the resource

11. Intersection of Groups Defined by Required Attributes

Security Model

5.1) The security model establishes the basic structure and interpretation of use-condition and attribute certificates

5.2) The security model establishes the basic functions of the *policy_engine* that processes the use-condition certificates (however the details will be dictated by the policy model)

5.3) User attributes that match those requested by use-conditions are certified by a trusted third-party (in attribute certificates)

5.4) Both use-condition (authorization) and attribute certificates are generated by mechanisms controlled by the principals (those with authority over the resource or trusted to certify an attribute)

Security Model

5.5) Authority to issue certificates is established by naming the issuing stakeholders and trusted CA
(in a “policy file” specific to the resource for use-conditions, and in the use-condition certificates for attributes)

5.6) Inclusion of all relevant certificates is guaranteed by naming trusted certificate servers (agents of CAs)

5.7) In support of a hierarchical policy model, stakeholders impose policy by specifying use-conditions with
global, local, or sub-tree scope - thereby establishing access-groups overall, only at one “level” (resource), or from here “down”

> Is “global” needed? Yes. What if the resource is organized hierarchically, but ownership is not? Then a global scope merely means that the use-condition was placed “here” rather than at some other point. What if resource organization is a mesh? Then there is no preferred location for the use-conditions. Actually, once a “use-condition issuer” is named with “global” scope, then its cert server must be

Security Model

searched for relevant use-conditions at every access attempt.

5.8) The model does not (currently) address:

5.8.1) the issue of a resource owner needing to grant access to subordinate resources (data) to a previously excluded user

(i.e. there is no way to grant a local exception to a higher level restriction)

5.8.2) CAs that provide certificate revocation lists

(Certificate caching at the resource can be inhibited so that each access request results in requesting a certificate from the CA server, thus revoked certificates may not be used in this case.)

5.9) General use-condition semantics:

({ **[boolean combinations of] required_attribute**
[in_access-group]
[admit_all] }

[*combined with* **[boolean combinations of] required_attributes [from X.509]**]

enables { **access**
any_action
action [list] }) **[boolean combinations of (")]**

for **(resource [list])**

[***with scope*** { **global**
sub-tree
local }]

{} any one of
[] optional
() must group
" repeat previous group

Security Model

[required_attribute [*from X.509*]

***is issued by* (attribute_issuer *from* server) [list]]**

{} any one of
[] optional
() must group
" repeat previous group

Security Model

5.9.1) "attributes" are text strings whose meaning is established as part of the resource policy (no semantic analysis)

5.9.2) required attributes may be satisfied with attribute certificates or X.509 identity certificates (and probably SPKI certificates in the future)

5.9.3) access is denied all by default
(a special "attribute" *admit_all* reverses this)

5.9.4) multiple requirements may be specified in a certificate

5.9.5) enabling "access" means that these attributes are used to establish a general access-group

5.9.6) an access-group is used to impose general access requirements from multiple stakeholders

Security Model

5.9.7) actions on resources must be explicitly enabled
(i.e., there is no default action)

5.9.8) the special attribute named “in_access-group” refers to the current composite access-group and is used to enable an action for all of the currently qualified entities
(i.e., no further attributes are needed for this action)

5.9.9) an optional “scope” is intended to support hierarchical policy models

- global implies the entire “space”
- sub-tree implies from this “point” down
- local applies only to this level of the hierarchy

The scope of ownership of resources is left to the policy model, however once that scope is established a global scope use-condition implies that stakeholder must be consulted for use-conditions for every resource accessed. (c.f. 5.7)

Security Model

5.9.10) the *combined_with* (or “*and_corresponding*”) operator means that the attributes must be checked in combination with each other because a single use-condition requires several separate attributes to satisfy

The use-condition requires that a third party certify the second party (issuer of the primary attribute).

For example, the request for an accredited training class will require a training attribute from a trainer, and an accreditation attribute from a designated third party.

The combined checking mechanism is attribute value and data dependent. This evaluation of this operator will have to be done by an application domain specific module in the policy engine.

For example, a use-condition requiring an X-ray training class taken during a time when the issuing institution was accredited by a specified third-party, might be posed as:

“X-ray training” *combined_with* “X-ray training accreditation”

This use-condition requires an attribute certificate issued by a training institution to the user, that is combined with an attribute certificate issued by LBNL to the institution accrediting its training at the time when the user took the training. E.g.:

Attribute #1 = Subject has “X-ray training”, data: “date received = 10/30/1993”, issued by “XYZ State U.”

Attribute #2 = “XYZ State U.” has “X-ray training accreditation”, data: accreditation = “1/1/1990 through present”, issued by LBNL

Security Model

- **Example use-conditions *for the WALDO Digital Library* (which has a hierarchical policy model):**

1) (group = “Group Diesel-collab” *or* O=Sandia *or* O= LBNL *or* O=LLNL

***enables* access *for* http://injector.snl.gov/Diesel-Collab**

***with scope* global**

A general policy statement that the subject <u>must have any one of the required attribute certificates to have any access to any of the Diesel Collab objects</u>.

group *is issued by* DN= “John Groupdefiner” CA=Sandia *from* http://snarfits.sandia.ca.gov

O *is issued by* CA=Sandia *from* ldap://injector.sandia.ca.gov

Security Model

5.9.11) “Group Diesel-collab” *enables* access *for* http://injector.snl.gov/diesel/VGs *with scope* sub-tree

A policy statement that only those with the attribute “Group Diesel-collab” can access the view graph directories (the organization affiliation allowed by the previous example is not sufficient for the VGs).

5.9.12) 0((“Group Diesel-collab” *enables* (read *and* write)) *or* (“O Sandia” *from X.509 or* “O LBNL” *from X.509 or* “O LLNL” *from X.509*)) *enables* read
***for* http://injector.snl.gov/Diesel-Collab/slides**
***with scope* sub-tree**

Subjects in the Diesel-collab group can add to the slides collection. Subjects that are in one of the organizations mentioned (but not in the Diesel-collab group) can only read the slides collection.

5.9.13) “Group Diesel-collab” enables (read *and* write *and* create_col) *for*

Security Model

http://injector.snl.gov/Diesel-Collab/users *with scope* local

Applies only to the “users” collection and allows anyone in the Diesel-collab group to create and populate a subcollection for themselves.

5.9.14) (“DN Mary Thompson” *enables* (read *and* write *and* create_col)) or (“Group Diesel-collab” *enables* read)

***for* http://injector.snl.gov/Diesel-Collab/users/mrt**

***with scope* sub-tree**

Mary can read, write and create collections from here down, other group members may only read.

5.9.15) “DN Bill Johnson” *enables* (read *and* write *and* create_col)

***for* http://injector.snl.gov/Diesel-Collab/users/wej**

***with scope* sub-tree**

Bill can read, etc. No one else can do anything (including read), since there are no action granting use-conditions that apply to this sub-tree. A higher level global “enables access” (as in 1) only means that some other attribute may grant

Security Model

an action.

5.9.16) operating-period *enables* access *for* LBNL-NCEM-EM-2 *with scope* local

An “operating-period” attribute certificate is required for access to this resource.

Security Model

- **Attribute certificate contents:**

  (“subject” will probably have to be generalized.)

```
{subject [list]}

  has {attribute}

     [data_field {name} value {opaque}]
     [data_field {name} value {opaque}] .....
```

- **Example attributes:**
  - **“DN = W. E. Johnston, O = Lawrence Berkeley National Laboratory”** ***has*** **“Group Diesel-collab”**
  - **“DN=W. E. Johnston, O = Lawrence Berkeley National Laboratory”** ***has*** **operating-period**
    data_field **“time-period: weekdays, 17:00-20:00”**

Security Model

- **The security architecture provides a “*policy engine*” to implement the policy model**

- **Our current *policy_engine* has two components: the policy *evaluator* and the certificate *verifier***

- **The *verifier* collects and validates the use-condition certificates and the attribute certificates, and verifies that the subject has an attribute certificate for each use-condition**

- **The *evaluator* implements the particular policy model and thereby establishes the overall conditions for access:**
    - **establishes the relationship among the use-conditions in the object hierarchy**
    - **evaluates the expressions within the complete collection of use-conditions**
    - **provides a “yes” or “no” answer to the security gateway re: access for a specific resource and entity**

Security Model

6.0 Implementation of the Policy Model for the Digital Library (ImgLib)

- ♦ **Who can establish policy (i.e. issue use-conditions for a resource) is expressed through a combination of relatively static policy configuration files at (potentially) each level of the object hierarchy**

  (most of these static files can, and will eventually, be replaced by another type of use-condition certificate)

- ♦ **For Web servers the *policy_engine* is provided by replacing the standard access control module. This is operational in the Apache server and should be possible in future releases of the Netscape server.**