HGAA: An Architecture to Support Hierarchical Group and Attribute-Based Access Control

Western
UNIVERSITY · CANADA

Daniel Servos
Western University
London, Ontario
dservos5@uwo.ca

Sylvia L. Osborn
Western University
London, Ontario
sylvia@csd.uwo.ca

March 21st
ABAC 2018

Outline

- **Outline**
- Background
- The Problem and Current Solutions
- HGAA
  - Overview
  - Attribute Authority & Attribute Certificate
  - Policy Authority & HGABAC Name Space
  - User Service Provider
- Implementation & Preliminary Results
- Conclusions

Background

HGABAC

Hierarchical Group and Attribute-Based Access Control

Earlier Work:

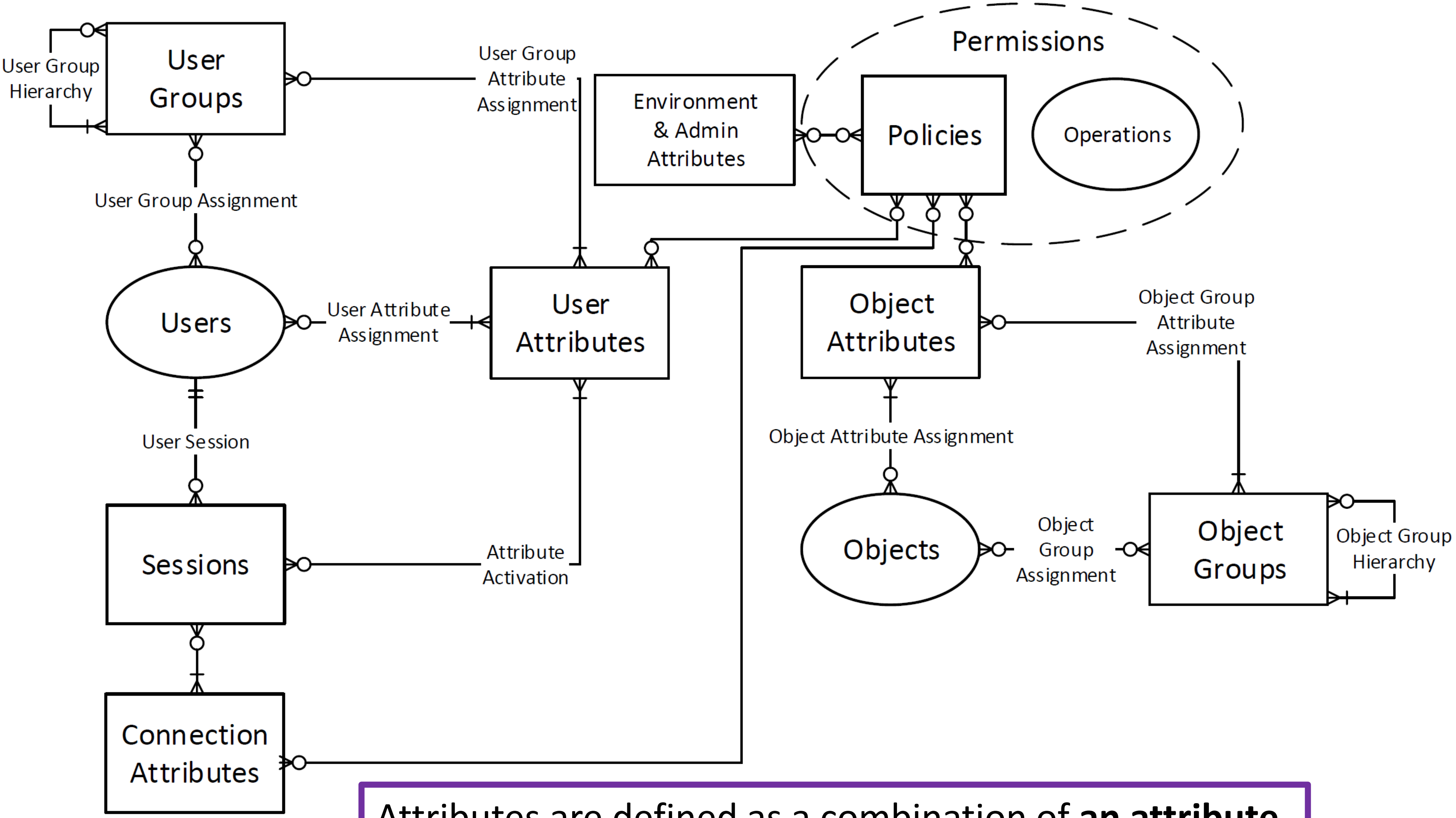
Daniel Servos and Sylvia L. Osborn. "HGABAC: Towards a formal model of hierarchical attribute-based access control." *International Symposium on Foundations and Practice of Security (FPS'2014).* November 5, 2014

HGABAC

Hierarchical Group and Attribute-Based Access Control

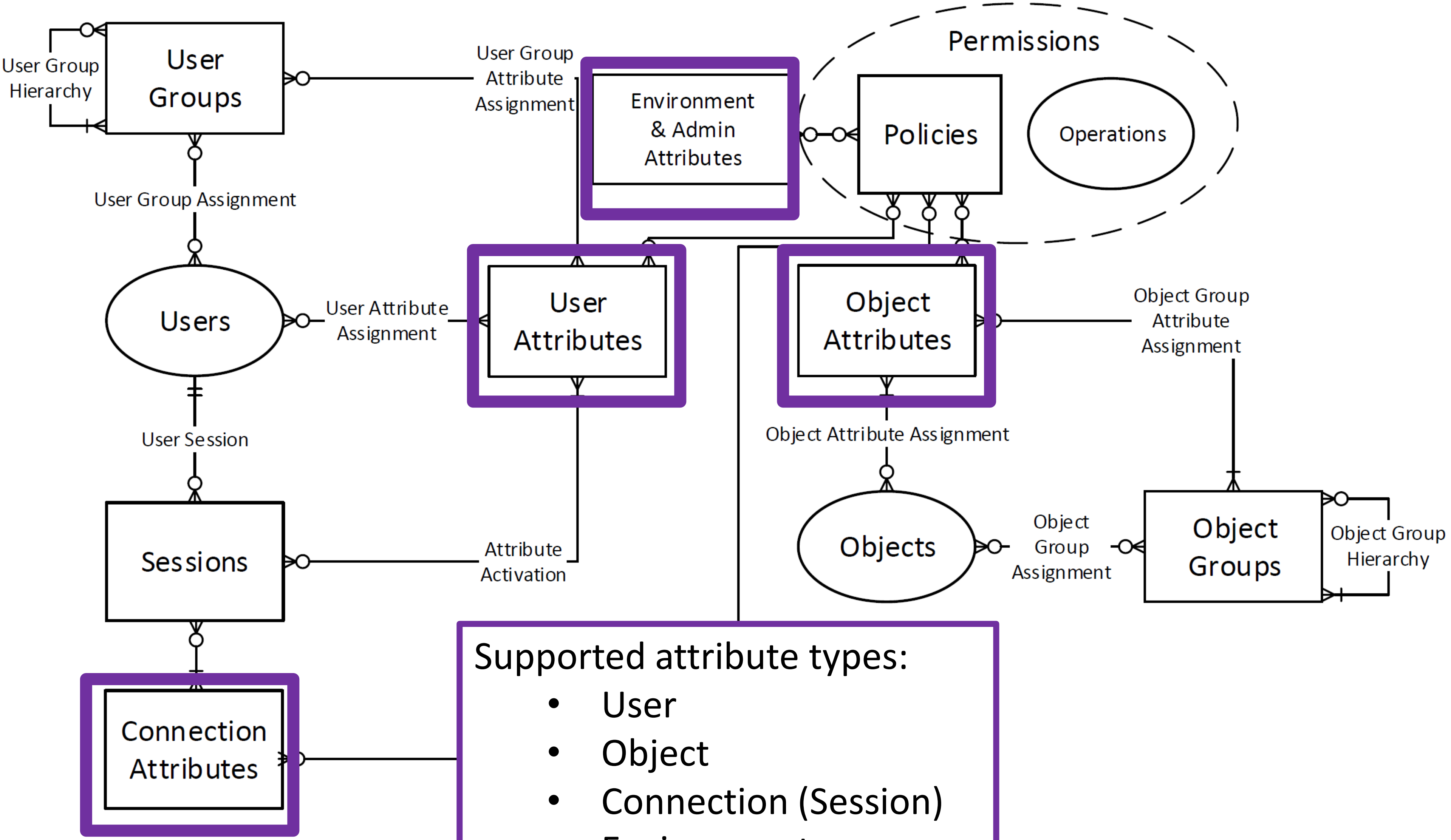
- Formal attribute-based access control model
- Introduces concepts of hierarchical user and object groups.
- Goals:
  - Lightweight
  - Easy to comprehend policies
  - User and object groups to simplify administration
  - Scalable
  - Ability to emulate traditional models (MAC, DAC, RBAC)
- Shown to be capable of emulating MAC, DAC and RBAC (including hierarchical roles).

HGABAC



Attributes are defined as a combination of **an attribute name**, **attribute type** and a **set of values**.

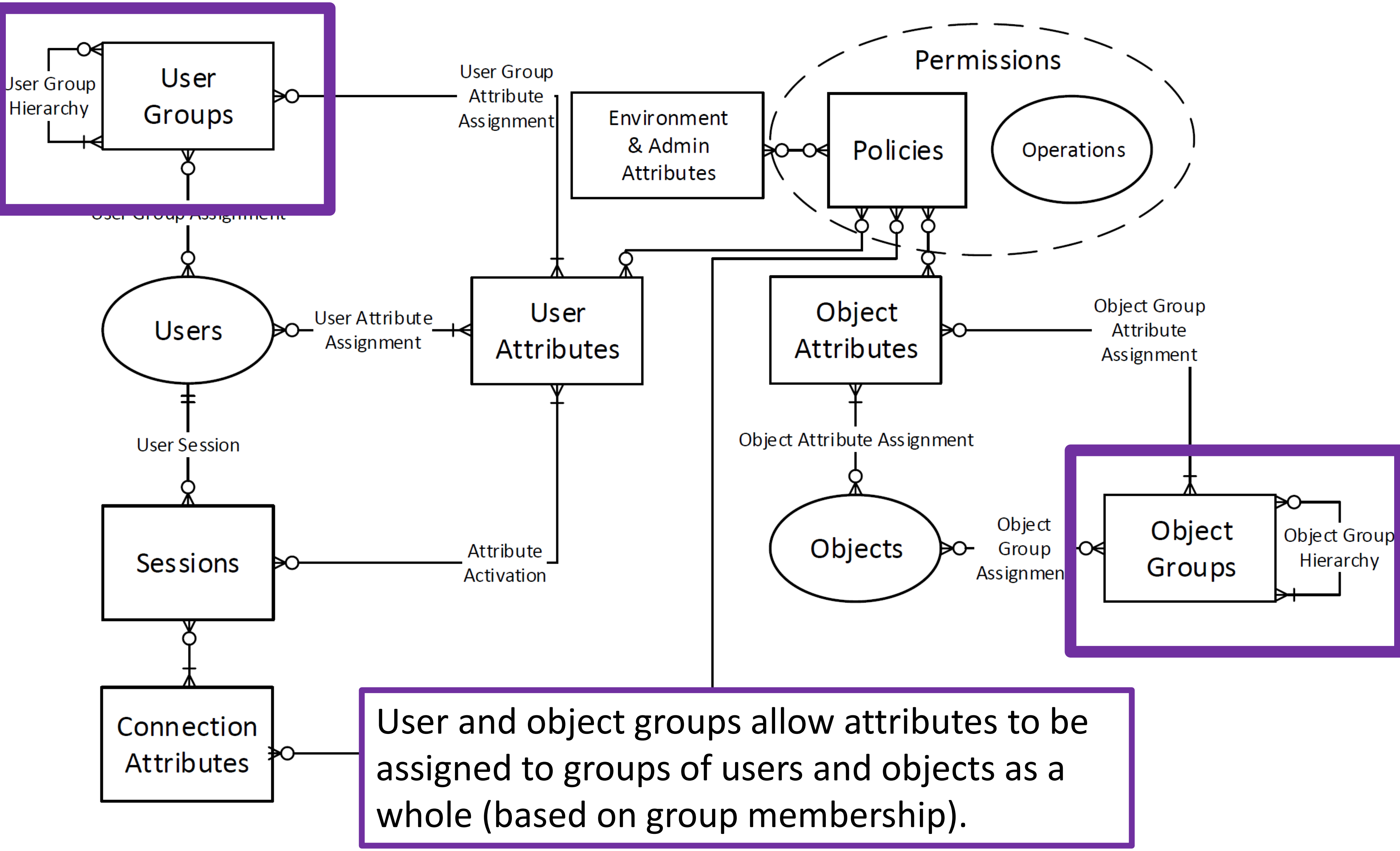
HGABAC



Supported attribute types:

- User
- Object
- Connection (Session)
- Environment
- Admin (Constant)

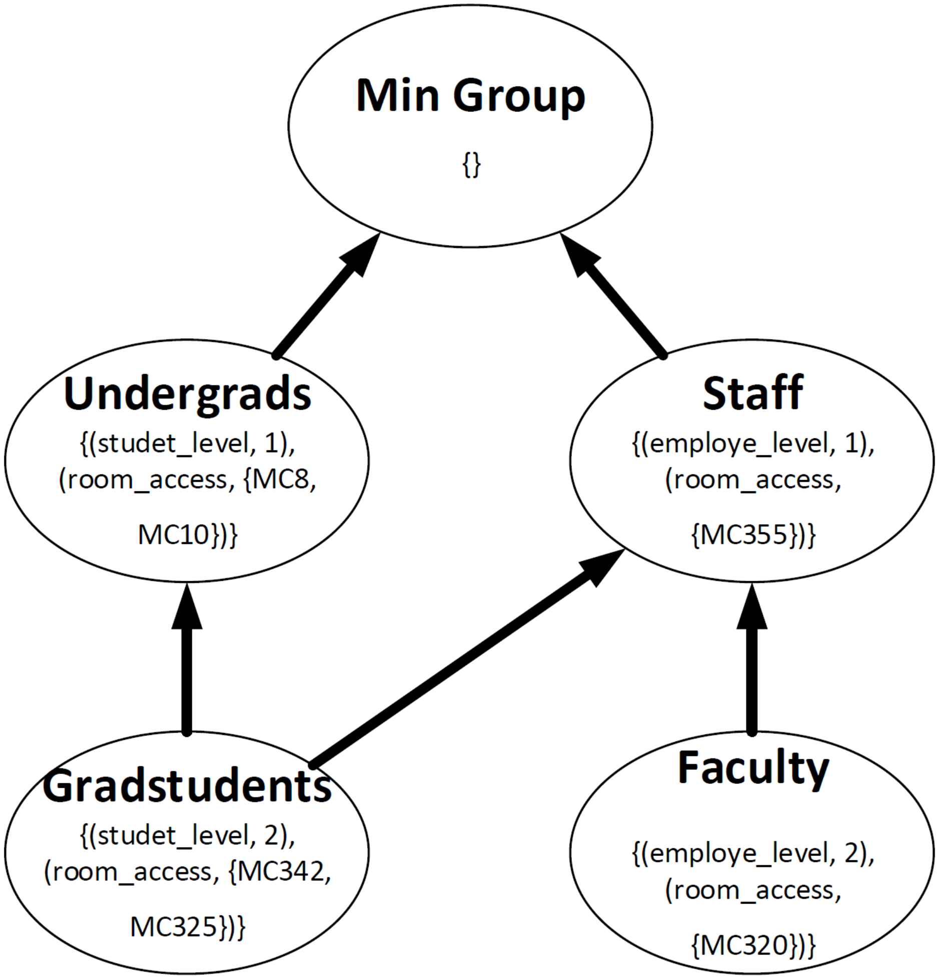
HGABAC



User and object groups allow attributes to be assigned to groups of users and objects as a whole (based on group membership).

HGABAC

User Group Hierarchy Example

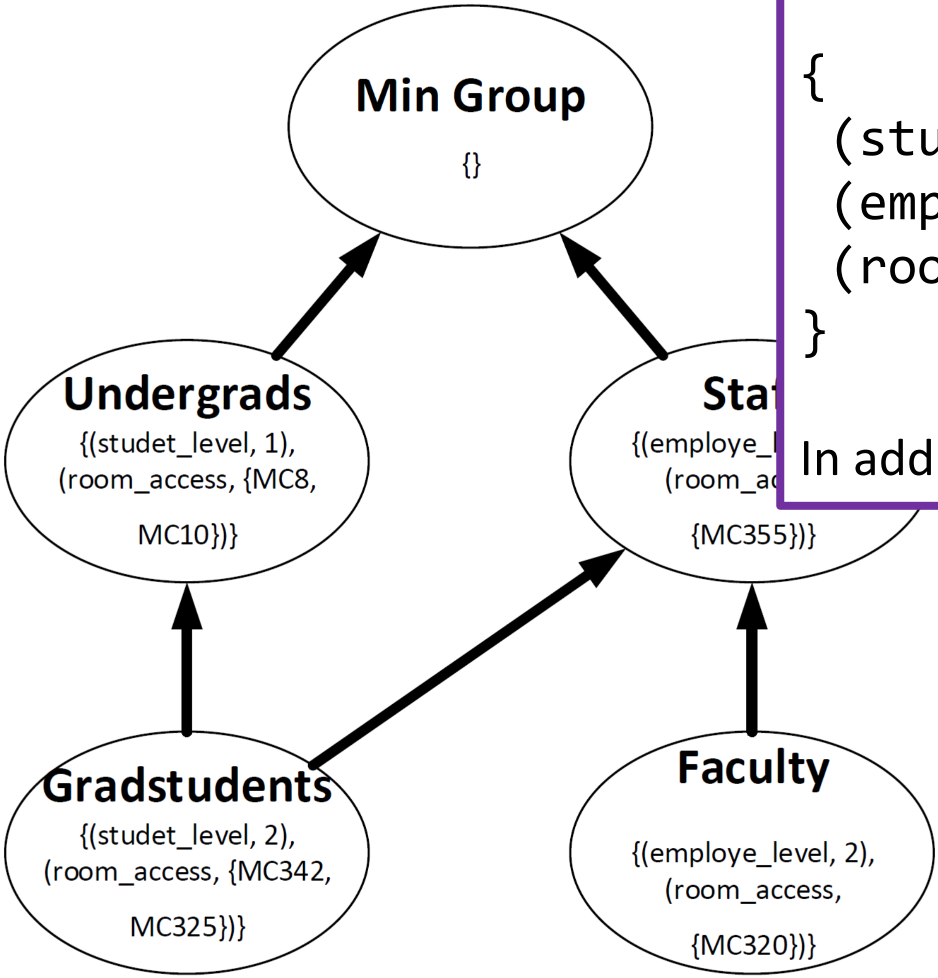

Min Group
{}
Undergrads
{(studet_level, 1), (room_access, {MC8, MC10})}
Staff
{(employe_level, 1), (room_access, {MC355})}
Gradstudents
{(studet_level, 2), (room_access, {MC342, MC325})}
Faculty
{(employe_level, 2), (room_access, {MC320})}

- Group hierarchies are directed acyclic graphs in which all possible paths end in Min Group, a group with no attributes assigned.

- A member of a group is assigned the attributes of the group they are a member as well as all groups on the path to Min Group

HGABAC

User Group Hierarchy Exa



Example:

User assigned to `Gradstudents` groups would have an effective attribute set of:

```
{
 (student_level, {1, 2}),
 (employe_level, 1),
 (room_access, {MC8, MC10, MC355, MC325})
}
```

In addition to any directly assigned attributes.

HGABAC

Policy Language

- Original HGABAC work introduces HGPLv1
- Attribute-based policy language designed for HGABAC
- Aims to be simple and support C-like syntax
- Trinary logic: TRUE, FALSE, UNDEF

HGABAC

Policy Language

Examples:

```
P1:     user.age >= 18 AND object.title = "Adult Only Book"


P2:     user.id = object.author


P3:     user.role IN {"doctor", "intern", "staff"} AND
            user.id != object.patient


P4:     object.type = "program" AND object.required_certifications
            SUBSET user.certifications


P5:     env.time_of_day_hour >= 9 AND env.time_of_day_hour <= 17
```

The Problem
& Current Solutions

The Problem

- Many ABAC models exist but few full solutions.
- Need architecture to fill in the gaps.
- Need to address questions like:
  - Who assigns the attributes and how?
  - How are attributes shared with each party?
  - How does the user provide proof of attribute ownership?
  - Where and how are policies evaluated?
  - How will the model scale in real-world use?

Current Solutions

- AAA Authorization Framework (RFC 2904)
- XACML: eXtensible Access Control Markup Language
- SAML: Security Assertion Markup Language

- NIST Policy Machine, of particular note:

  Smriti Bhatt, Farhan Patwa, and Ravi Sandhu. "ABAC with Group Attributes and Attribute Hierarchies Utilizing the Policy Machine". *ABAC 2017.* March 24.

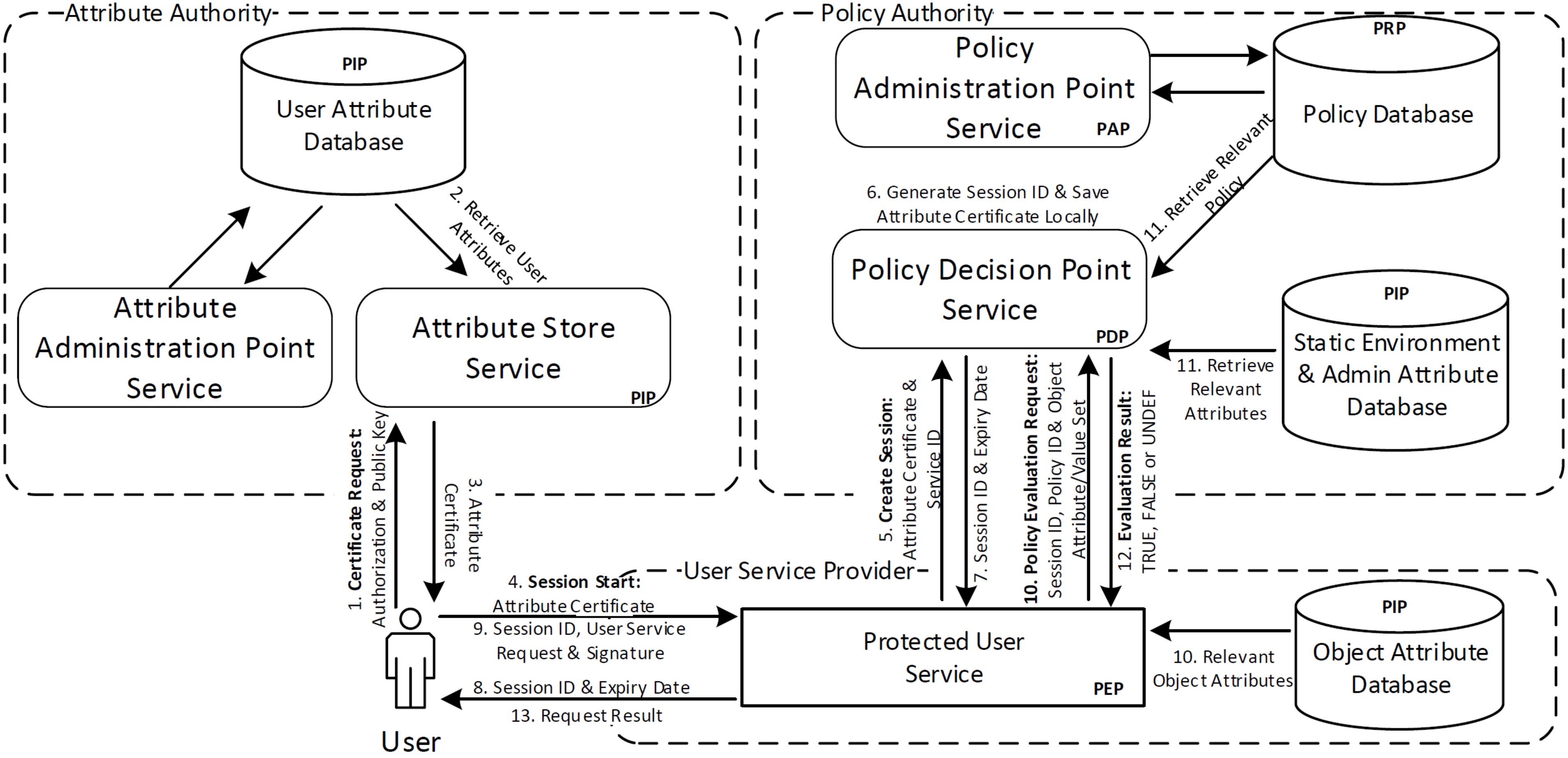
Limitations of Current Efforts

- Offline Policy Information Point (Attribute Stores/Authorities)
- Public Key Infrastructure Overhead
- Future Support for Delegation Concepts
- HGABAC Support
  - Attributes as name value pairs
  - Groups
  - Hierarchy
- Lightweight Approach

HGAA:
Hierarchical Group Attribute Architecture

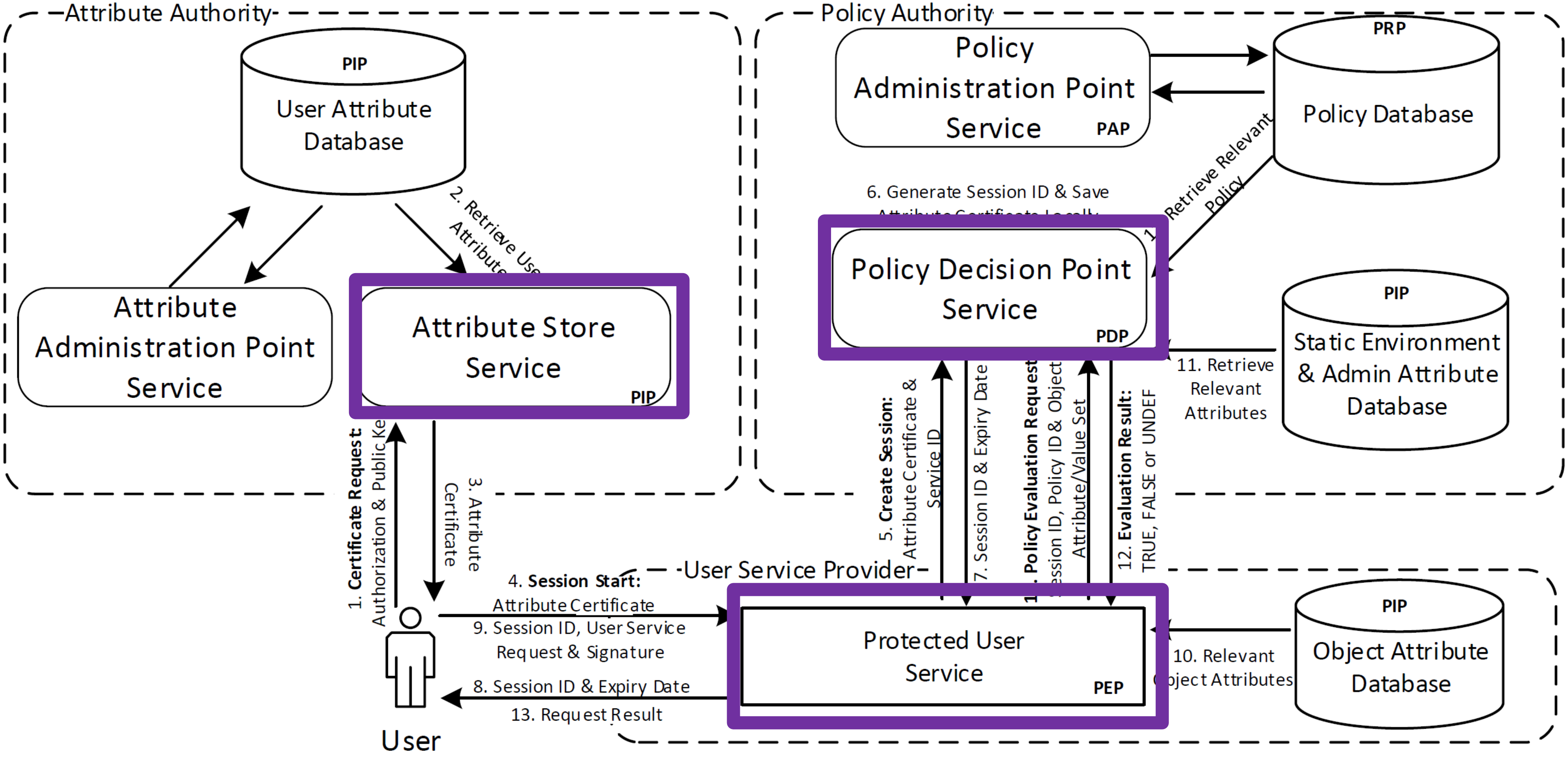
HGAA

Overview


Attribute Authority
PIP
User Attribute Database
2. Retrieve User Attributes
Attribute Administration Point Service
Attribute Store Service
PIP
Policy Authority
Policy Administration Point Service
PAP
PRP
Policy Database
11. Retrieve Relevant Policy
6. Generate Session ID & Save Attribute Certificate Locally
Policy Decision Point Service
PDP
PIP
Static Environment & Admin Attribute Database
11. Retrieve Relevant Attributes
1. Certificate Request
Authorization & Public Key
3. Attribute Certificate
5. Create Session:
Attribute Certificate & Service ID
7. Session ID & Expiry Date
10. Policy Evaluation Request:
Session ID, Policy ID & Object Attribute/Value Set
12. Evaluation Result:
TRUE, FALSE or UNDEF
User Service Provider
4. Session Start:
Attribute Certificate
9. Session ID, User Service Request & Signature
8. Session ID & Expiry Date
13. Request Result
Protected User Service
PEP
10. Relevant Object Attributes
PIP
Object Attribute Database
User

HGAA
Overview

Comprised of three core service types: Attribute Store Services, User Services, and Policy Decision Point Services


Attribute Authority
PIP
User Attribute Database
2. Retrieve User Attribute
Attribute Administration Point Service
Attribute Store Service
PIP
1. Certificate Request
Authorization & Public Key
3. Attribute Certificate
User
4. Session Start:
Attribute Certificate
9. Session ID, User Service Request & Signature
8. Session ID & Expiry Date
13. Request Result
Policy Authority
Policy Administration Point Service
PAP
PRP
Policy Database
Retrieve Relevant Policy
6. Generate Session ID & Save Attribute Certificate Locally
Policy Decision Point Service
PDP
PIP
Static Environment & Admin Attribute Database
11. Retrieve Relevant Attributes
5. Create Session:
Attribute Certificate & Service ID
7. Session ID & Expiry Date
Policy Evaluation Request
Session ID, Policy ID & Object Attribute/Value Set
12. Evaluation Result:
TRUE, FALSE or UNDEF
User Service Provider
Protected User Service
PEP
10. Relevant Object Attributes
PIP
Object Attribute Database

HGAA

Namespace

- Require a way of uniquely identifying attributes and users from different authorities.
- URI based namespace similar to one used in XACML.

H

Na

- utes and

- n XACML.

```
Absolute URI:
    hgabac://<authority>[[/<type>]/<element_name>]

Relative URI:
    [/]<type>/<element_name>
    | [/]<element_name>

type:
    user
    | group[/user | /object]
    | attribute[/<att_sub_type>]
    | object[/<obj_sub_type>]
    | session
    | operation
    | permission
    | policy
    | service

att_sub_types:
    user
    | object
    | environment
    | admin
    | connection
    | unknown
```

HGAA

Namespace

- Require a way of uniquely identifying attributes and users from different authorities.
- URI based namespace similar to one used in XACML.

Examples:

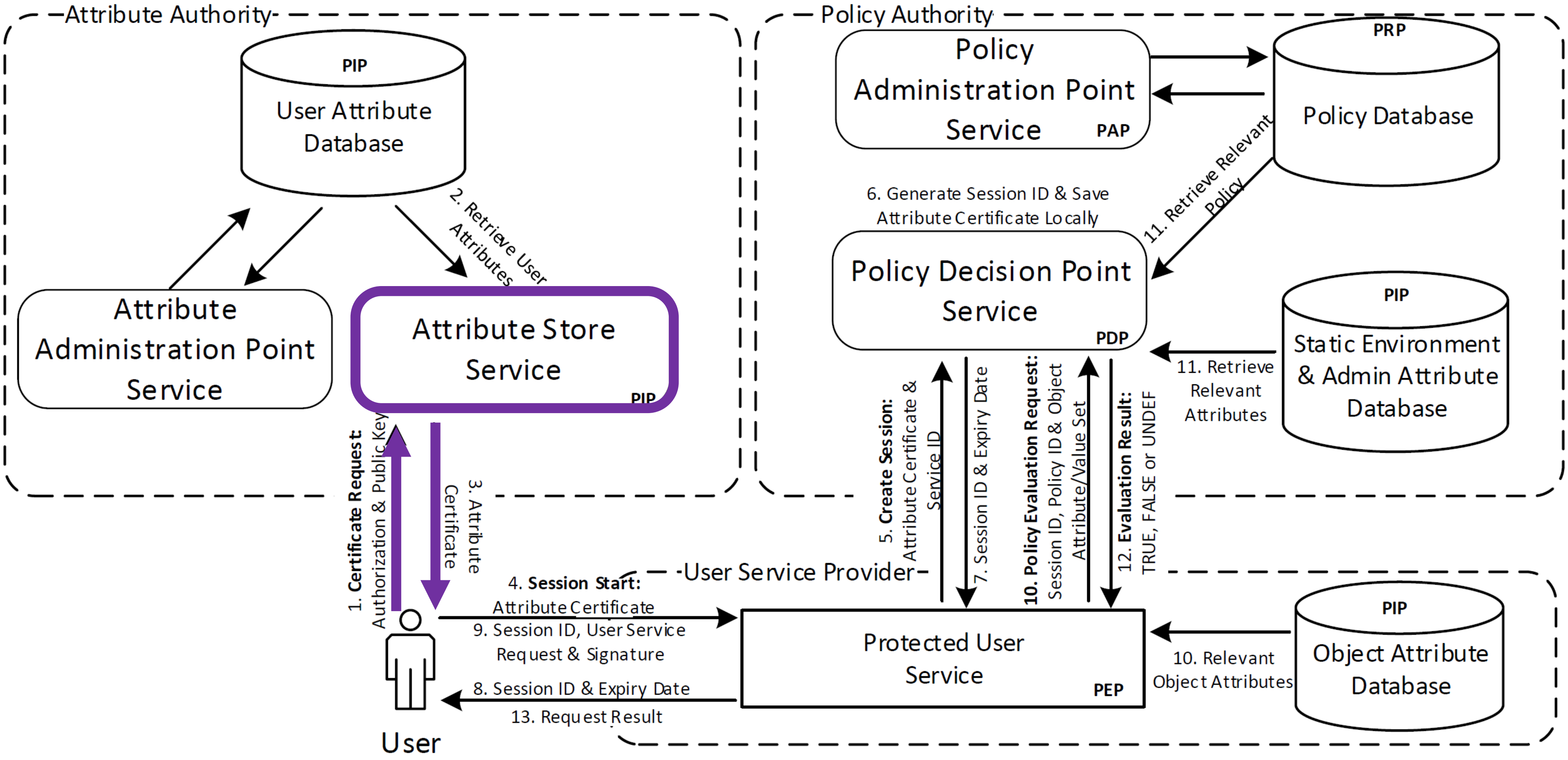
```
hgabac://cs1.ca/attribute/user/age

/attribute/user/age

/attribute/age
```

Authority

Attribute Type

Attribute Name

Attribute Store Service


Attribute Authority
PIP
User Attribute Database
Attribute Administration Point Service
Attribute Store Service
PIP
2. Retrieve User Attributes
1. Certificate Request
Authorization & Public Key
3. Attribute Certificate
User
4. Session Start:
Attribute Certificate
9. Session ID, User Service Request & Signature
8. Session ID & Expiry Date
13. Request Result
User Service Provider
Protected User Service
PEP
Policy Authority
Policy Administration Point Service
PAP
PRP
Policy Database
11. Retrieve Relevant Policy
6. Generate Session ID & Save Attribute Certificate Locally
Policy Decision Point Service
PDP
PIP
Static Environment & Admin Attribute Database
11. Retrieve Relevant Attributes
5. Create Session:
Attribute Certificate & Service ID
7. Session ID & Expiry Date
10. Policy Evaluation Request:
Session ID, Policy ID & Object Attribute/Value Set
12. Evaluation Result:
TRUE, FALSE or UNDEF
PIP
Object Attribute Database
10. Relevant Object Attributes

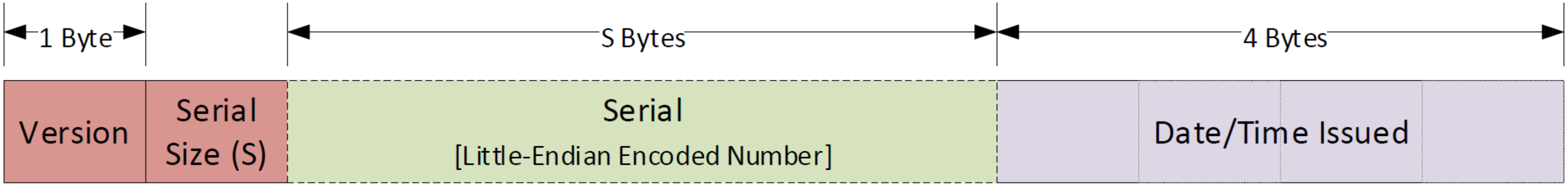
Users request an attribute certificate from their home attribute authority containing a subset of their assigned attributes.

Attribute Certificate

- Loosely based on X.509 Attribute Certificates but do not require X.509 infrastructure
- Contains information about issuer (attribute authority), holder (the user), their activated attribute set and a number of other properties.
- Includes User and Connection attributes.
- Cryptographically signed by attribute authority.
- Offer proof of attribute ownership.

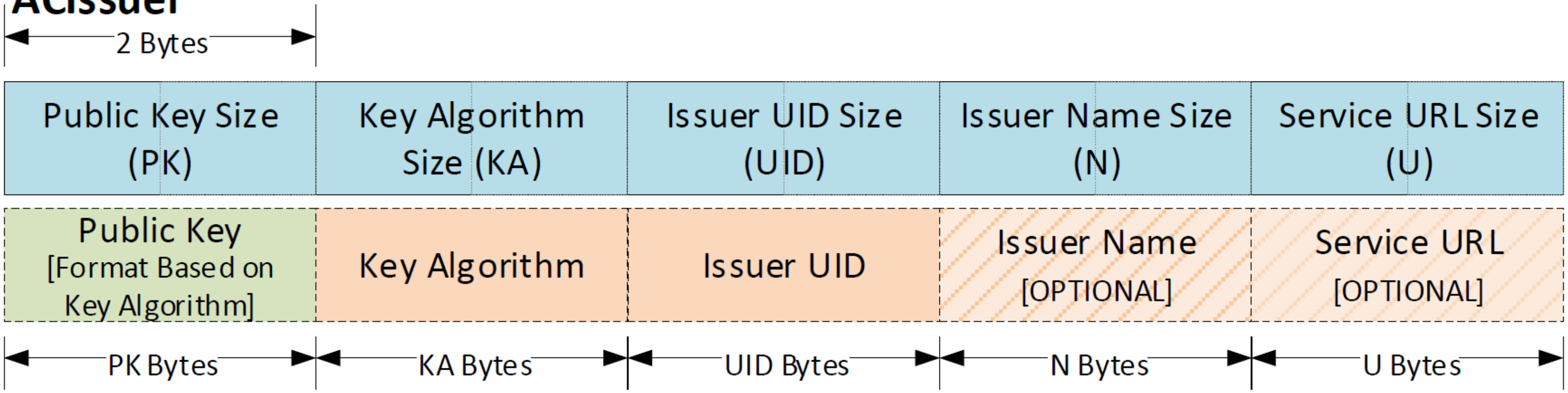
Attribute Certificate

ACInformation



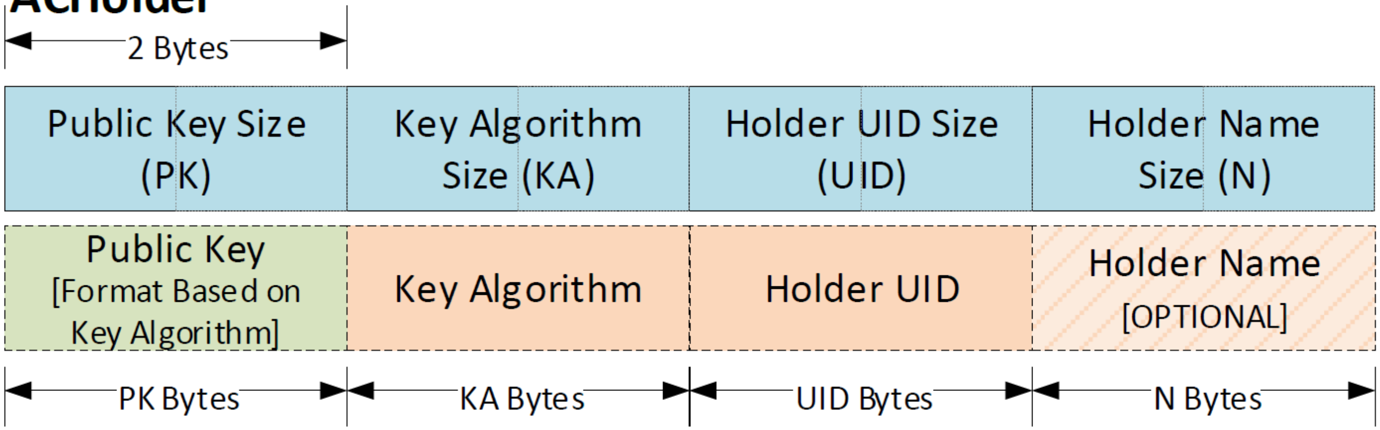
1 Byte		S Bytes	4 Bytes
Version	Serial Size (S)	Serial [Little-Endian Encoded Number]	Date/Time Issued

ACIssuer



2 Bytes				
Public Key Size (PK)	Key Algorithm Size (KA)	Issuer UID Size (UID)	Issuer Name Size (N)	Service URL Size (U)
Public Key [Format Based on Key Algorithm]	Key Algorithm	Issuer UID	Issuer Name [OPTIONAL]	Service URL [OPTIONAL]
PK Bytes	KA Bytes	UID Bytes	N Bytes	U Bytes

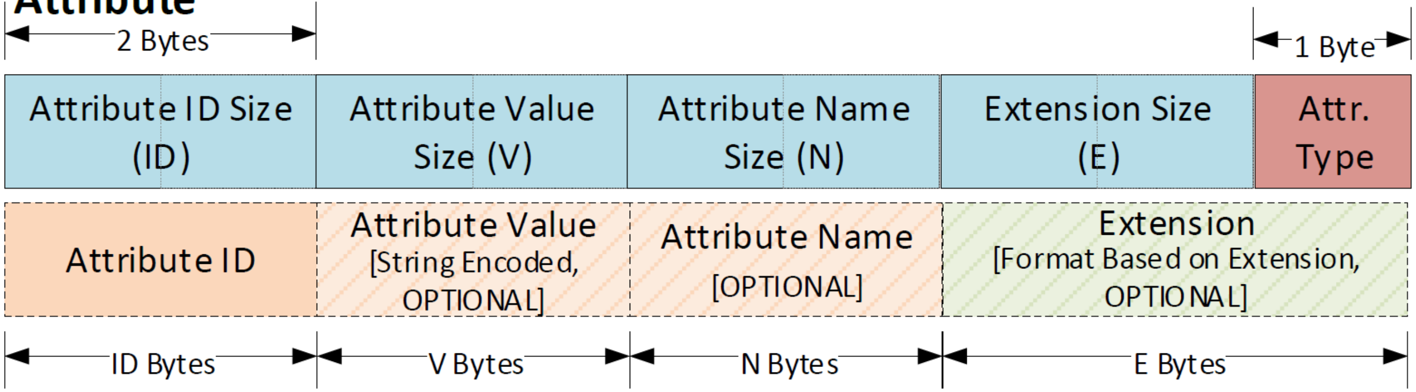
ACHolder



2 Bytes			
Public Key Size (PK)	Key Algorithm Size (KA)	Holder UID Size (UID)	Holder Name Size (N)
Public Key [Format Based on Key Algorithm]	Key Algorithm	Holder UID	Holder Name [OPTIONAL]
PK Bytes	KA Bytes	UID Bytes	N Bytes

Attribute Certificate

Attribute



2 Bytes

1 Byte

Attribute ID Size (ID)

Attribute Value Size (V)

Attribute Name Size (N)

Extension Size (E)

Attr. Type

Attribute ID

Attribute Value [String Encoded, OPTIONAL]

Attribute Name [OPTIONAL]

Extension [Format Based on Extension, OPTIONAL]

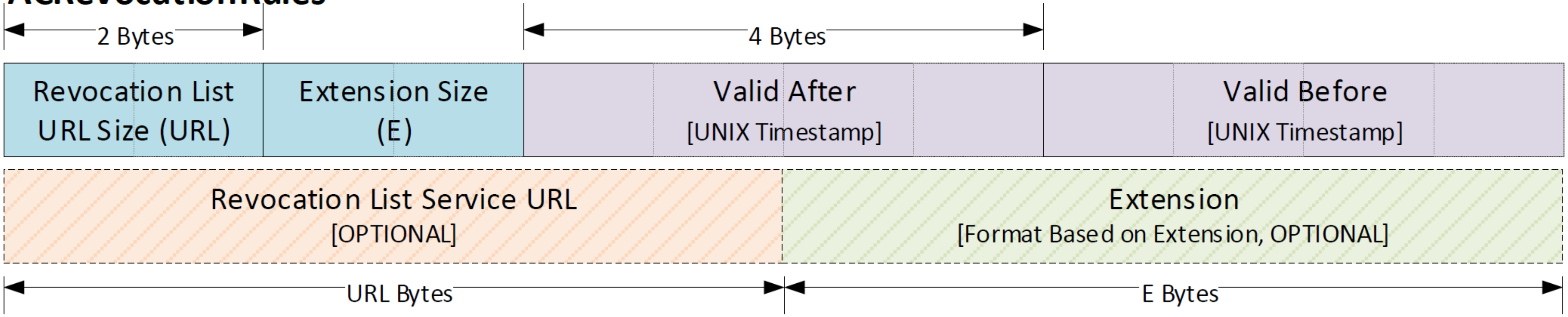
ID Bytes

V Bytes

N Bytes

E Bytes

ACRevocationRules



2 Bytes

4 Bytes

Revocation List URL Size (URL)

Extension Size (E)

Valid After [UNIX Timestamp]

Valid Before [UNIX Timestamp]

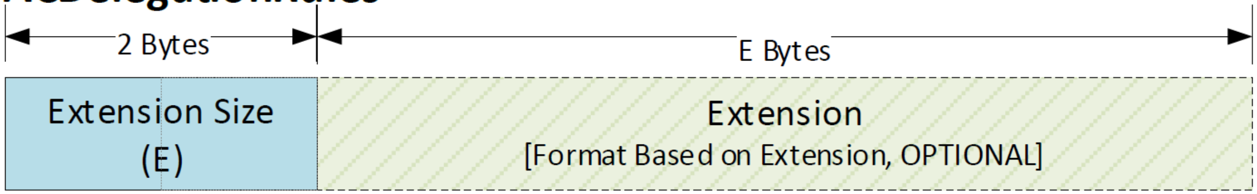
Revocation List Service URL [OPTIONAL]

Extension [Format Based on Extension, OPTIONAL]

URL Bytes

E Bytes

ACDelegationRules



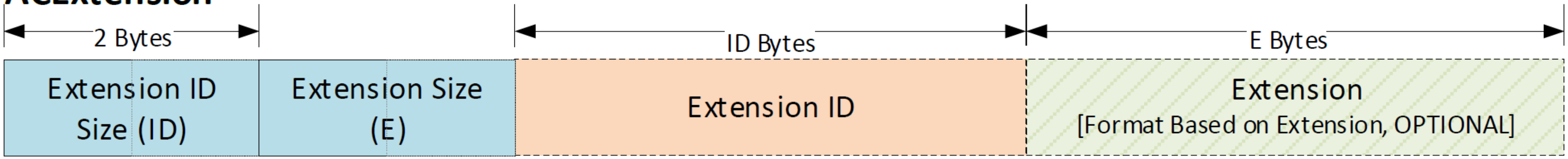
2 Bytes

E Bytes

Extension Size (E)

Extension [Format Based on Extension, OPTIONAL]

ACExtension



2 Bytes

ID Bytes

E Bytes

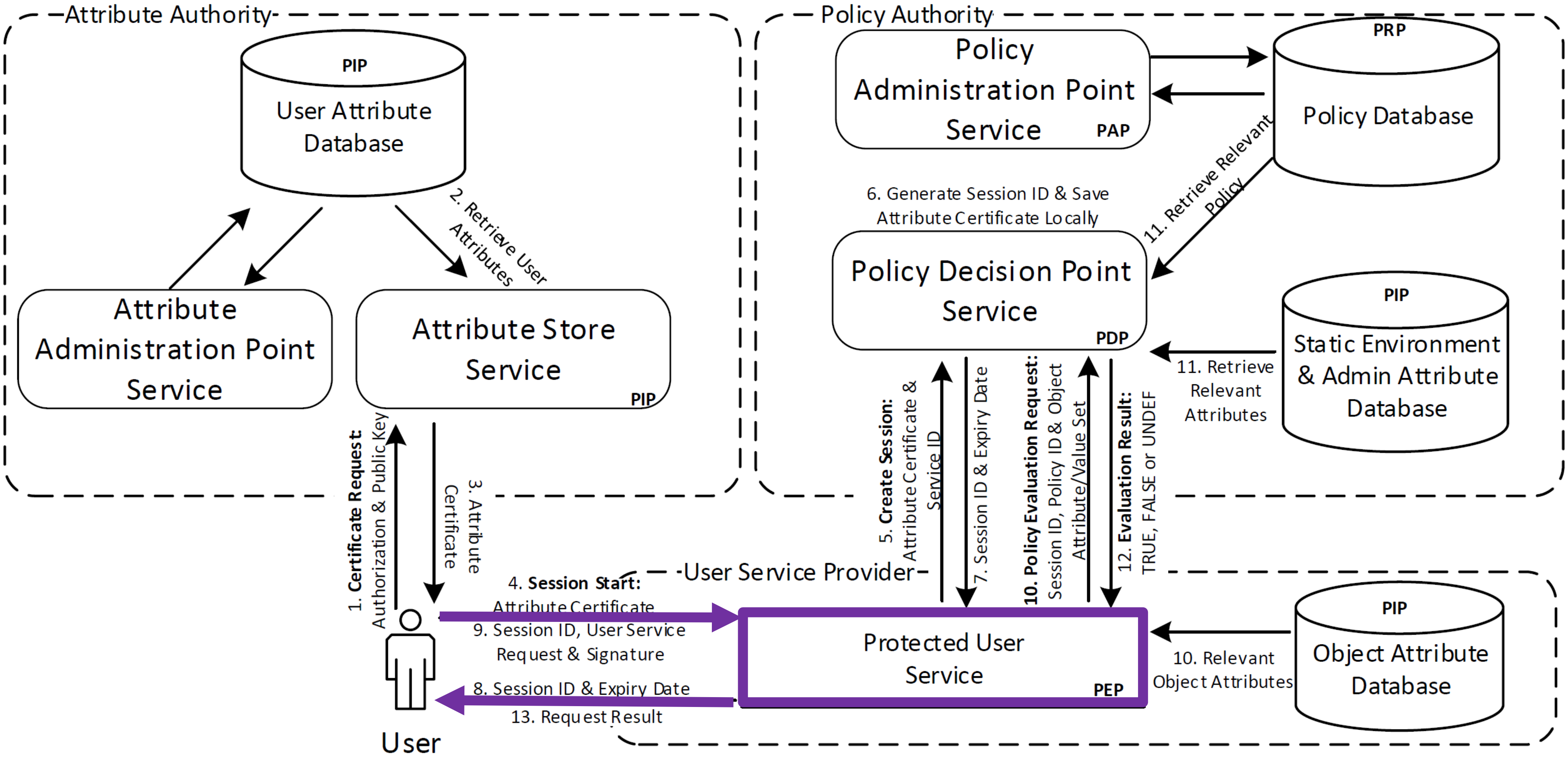
Extension ID Size (ID)

Extension Size (E)

Extension ID

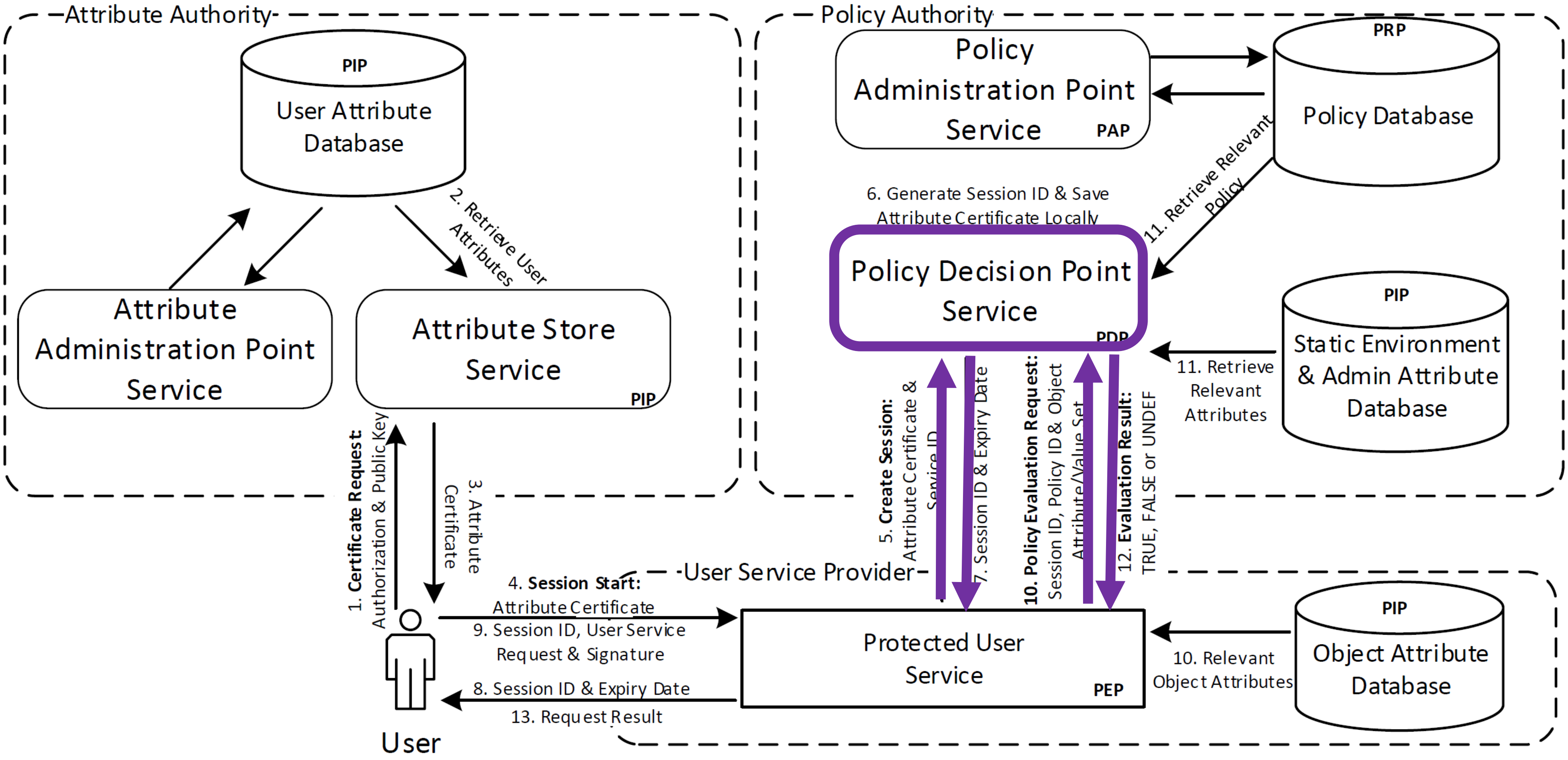
Extension [Format Based on Extension, OPTIONAL]

User Services


Attribute Authority
PIP
User Attribute Database
2. Retrieve User Attributes
Attribute Administration Point Service
Attribute Store Service
PIP
1. Certificate Request: Authorization & Public Key
3. Attribute Certificate
Policy Authority
Policy Administration Point Service
PAP
PRP
Policy Database
11. Retrieve Relevant Policy
6. Generate Session ID & Save Attribute Certificate Locally
Policy Decision Point Service
PDP
PIP
Static Environment & Admin Attribute Database
11. Retrieve Relevant Attributes
5. Create Session: Attribute Certificate & Service ID
7. Session ID & Expiry Date
10. Policy Evaluation Request: Session ID, Policy ID & Object Attribute/Value Set
12. Evaluation Result: TRUE, FALSE or UNDEF
User Service Provider
4. Session Start: Attribute Certificate
9. Session ID, User Service Request & Signature
8. Session ID & Expiry Date
13. Request Result
User
Protected User Service
PEP
PIP
Object Attribute Database
10. Relevant Object Attributes

User authenticates with and makes requests upon services by providing their signed attribute certificate as part of the request.

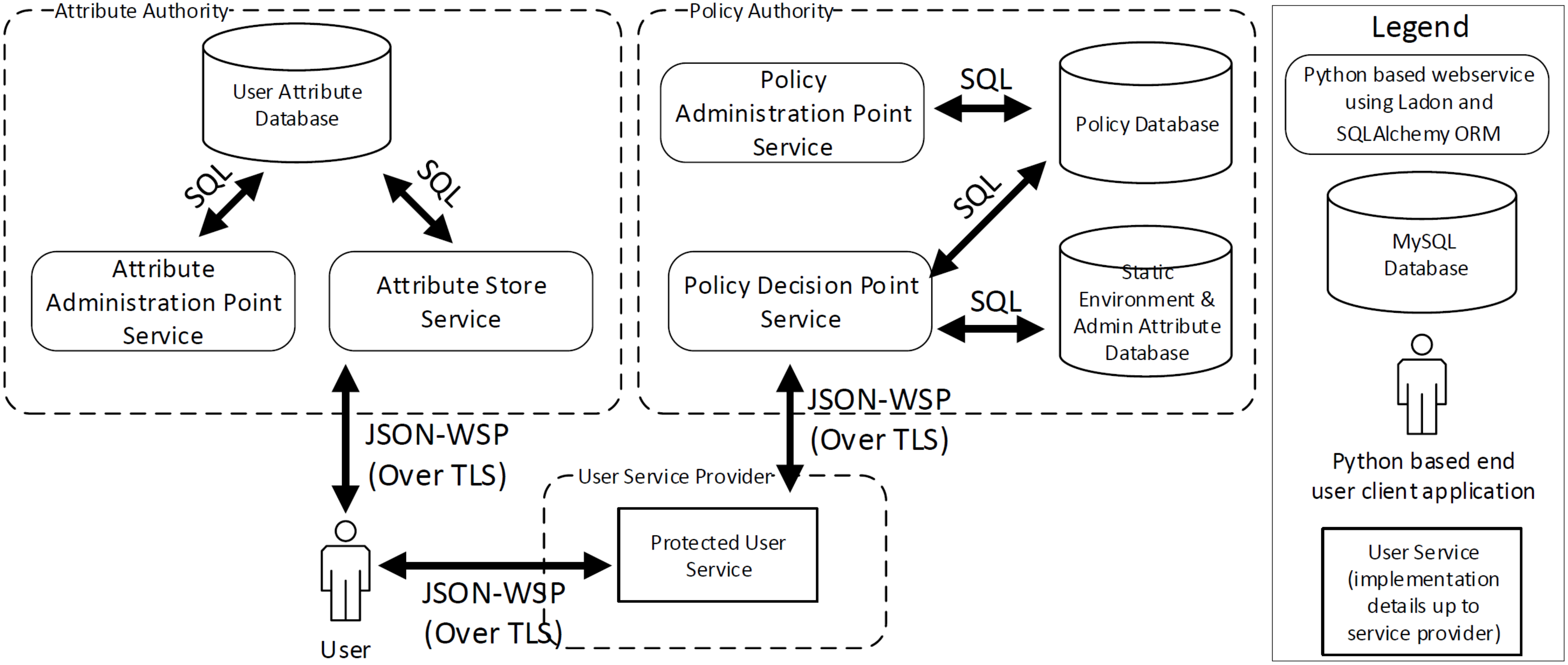
Policy Decision Point Service


Attribute Authority
PIP
User Attribute Database
2. Retrieve User Attributes
Attribute Administration Point Service
Attribute Store Service
PIP
1. Certificate Request: Authorization & Public Key
3. Attribute Certificate
User
4. Session Start: Attribute Certificate
9. Session ID, User Service Request & Signature
8. Session ID & Expiry Date
13. Request Result
Policy Authority
Policy Administration Point Service
PAP
PRP
Policy Database
11. Retrieve Relevant Policy
6. Generate Session ID & Save Attribute Certificate Locally
Policy Decision Point Service
PDP
PIP
Static Environment & Admin Attribute Database
11. Retrieve Relevant Attributes
5. Create Session: Attribute Certificate & Service ID
7. Session ID & Expiry Date
10. Policy Evaluation Request: Session ID, Policy ID & Object Attribute/Value Set
12. Evaluation Result: TRUE, FALSE or UNDEF
User Service Provider
Protected User Service
PEP
PIP
Object Attribute Database
10. Relevant Object Attributes

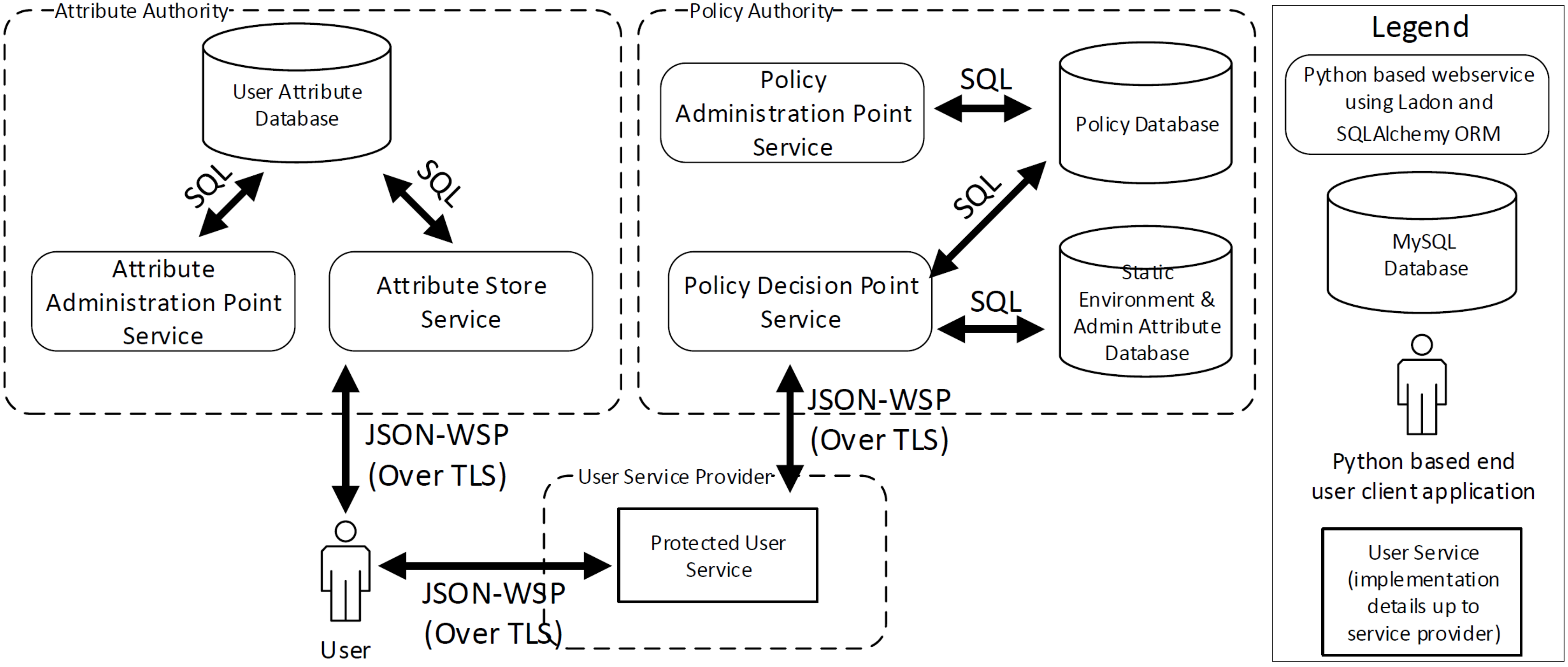
User services evaluate access request by contacting a Policy Decision Point Service with a copy of the user's attribute certificate, relevant object attributes and policy ID.

Implementation & Preliminary Results

Implementation: Services


Attribute Authority
User Attribute Database
SQL
SQL
Attribute Administration Point Service
Attribute Store Service
Policy Authority
Policy Administration Point Service
SQL
Policy Database
SQL
Policy Decision Point Service
SQL
Static Environment & Admin Attribute Database
JSON-WSP (Over TLS)
JSON-WSP (Over TLS)
User Service Provider
Protected User Service
JSON-WSP (Over TLS)
User
Legend
Python based webservice using Ladon and SQLAlchemy ORM
MySQL Database
Python based end user client application
User Service (implementation details up to service provider)

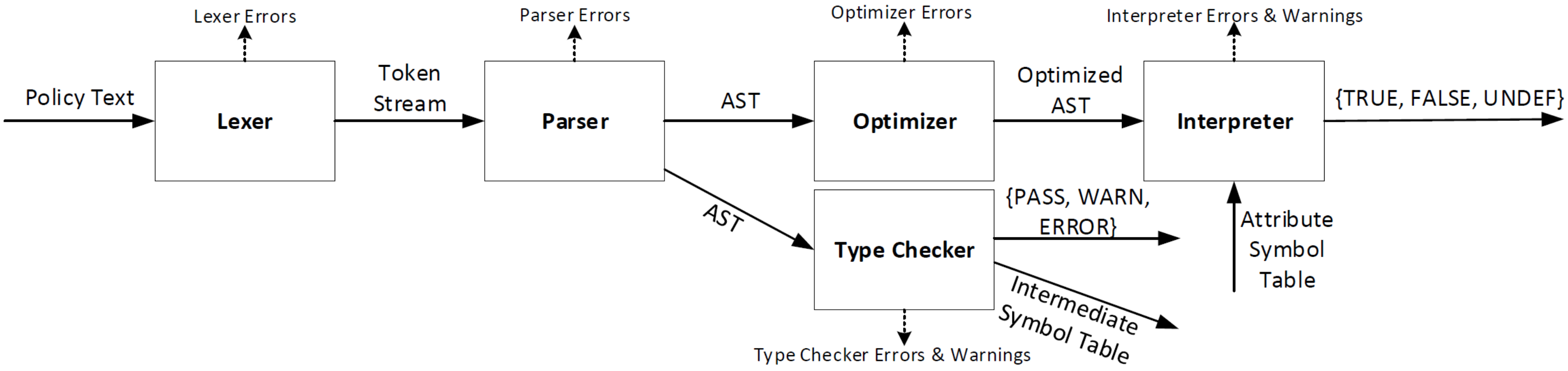
Implementation: Services



Attribute Authority
User Attribute Database
SQL
SQL
Attribute Administration Point Service
Attribute Store Service
Policy Authority
Policy Administration Point Service
SQL
Policy Database
SQL
Policy Decision Point Service
SQL
Static Environment & Admin Attribute Database
JSON-WSP (Over TLS)
JSON-WSP (Over TLS)
User Service Provider
Protected User Service
JSON-WSP (Over TLS)
User
Legend
Python based webservice using Ladon and SQLAlchemy ORM
MySQL Database
Python based end user client application
User Service (implementation details up to service provider)

JSON based webservices implemented in Python using Ladon framework and SQLAlchemy ORM

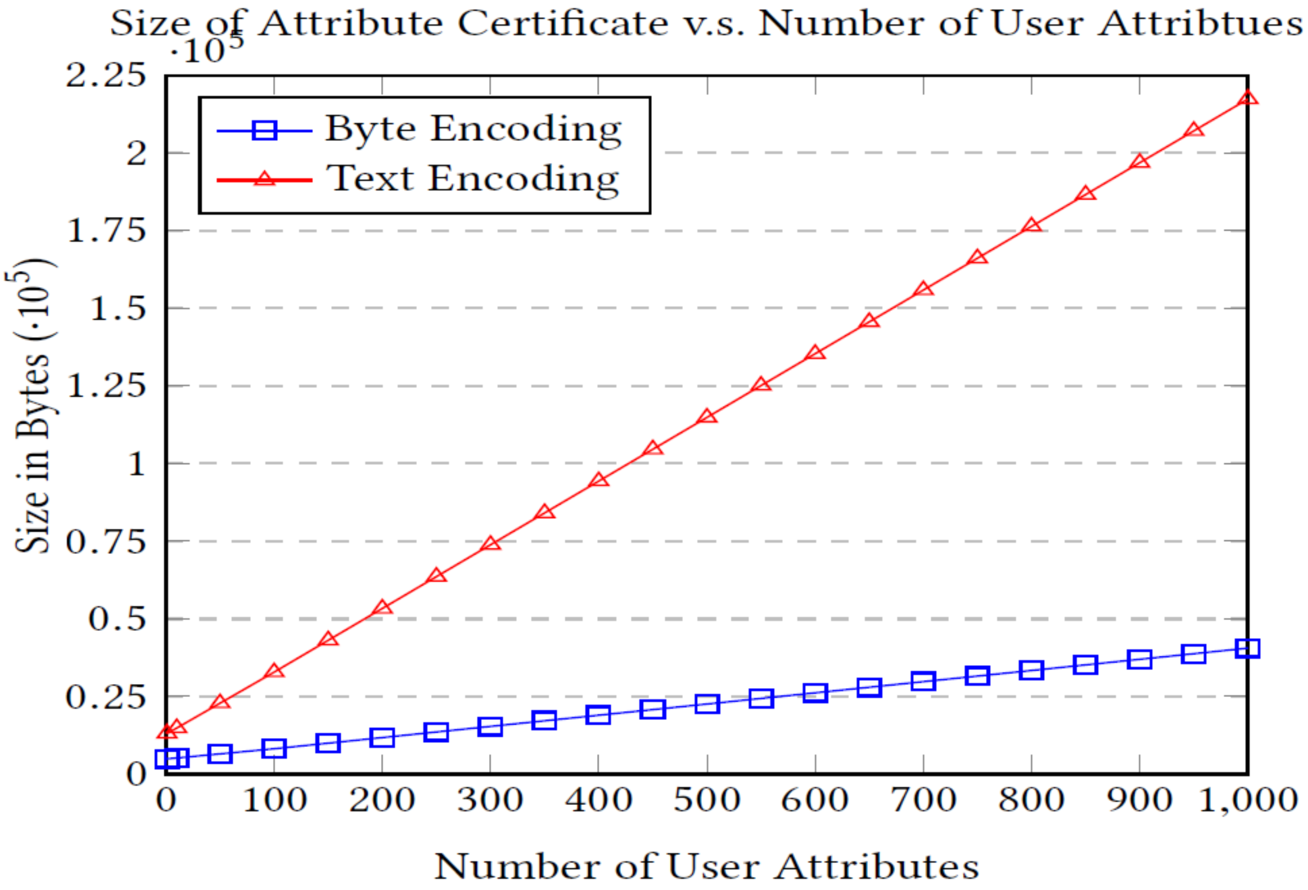
Implementation: HGPL Interpreter



- HGPL interpreter created in Python that utilizes a recursive descent parsing strategy.
- Policies stored as precomputed AST.
- When combined with attributes, result is a TRUE, FALSE or UNDEF decision.

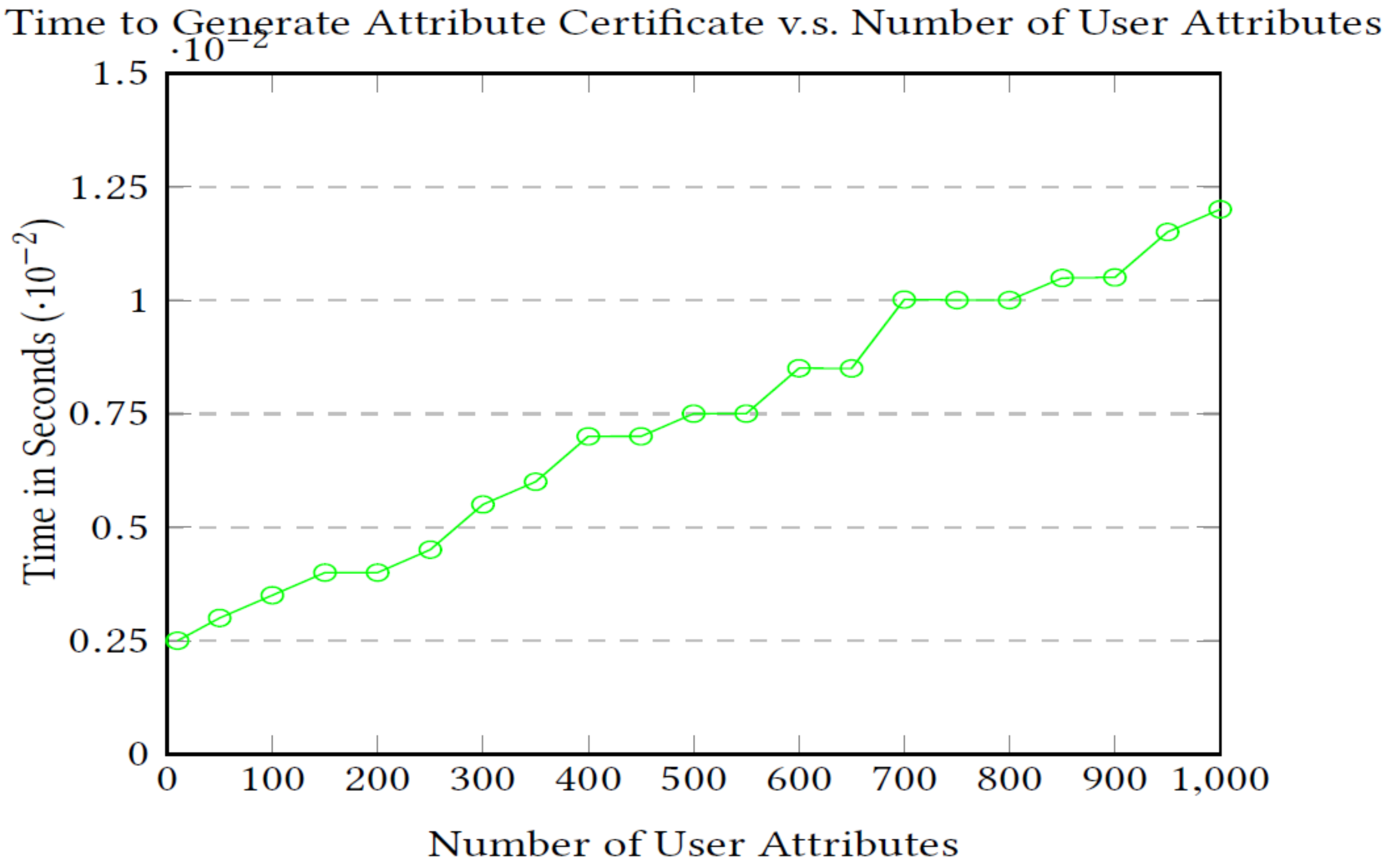
Preliminary Results

Attribute Certificate



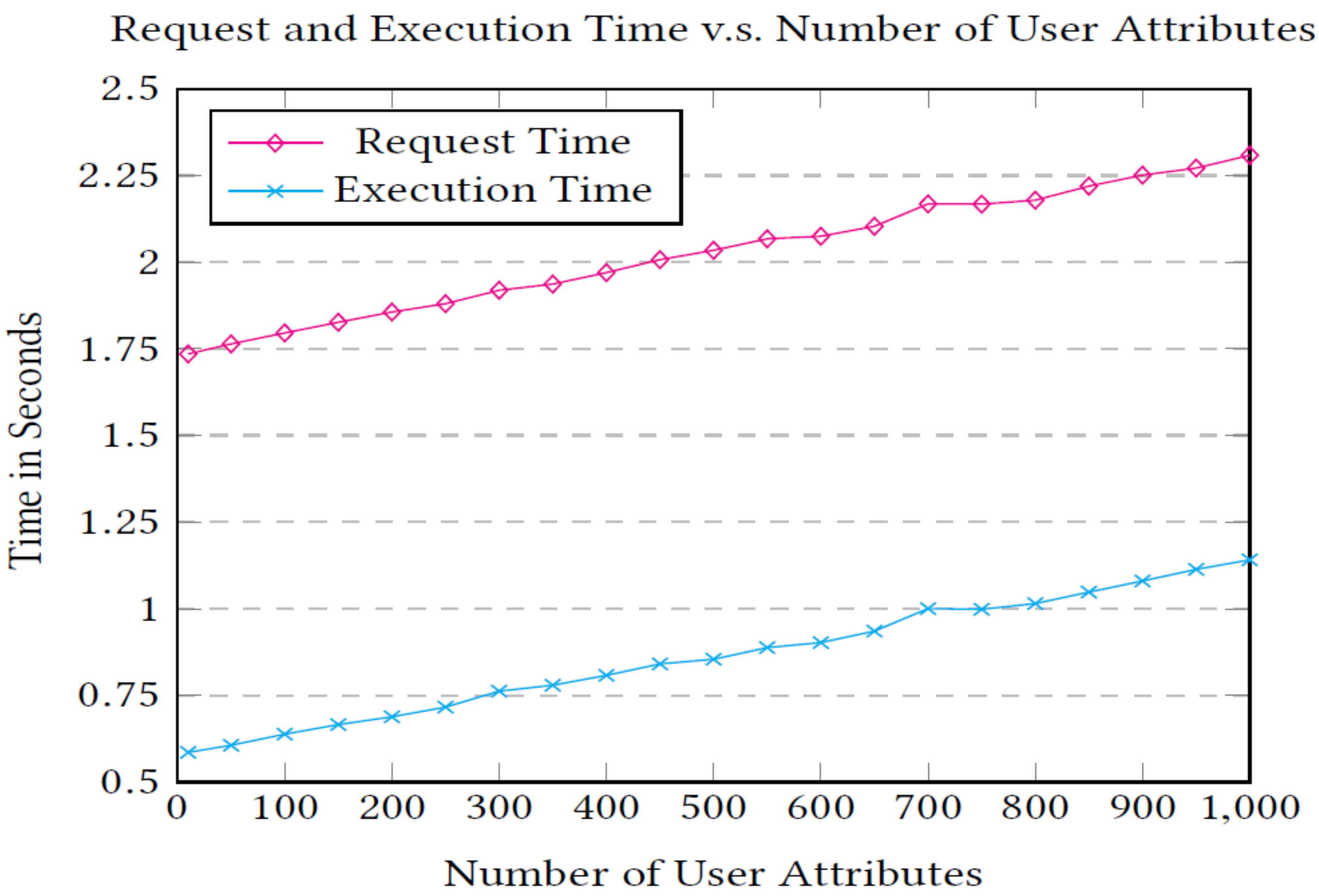
Preliminary Results

Attribute Certificate



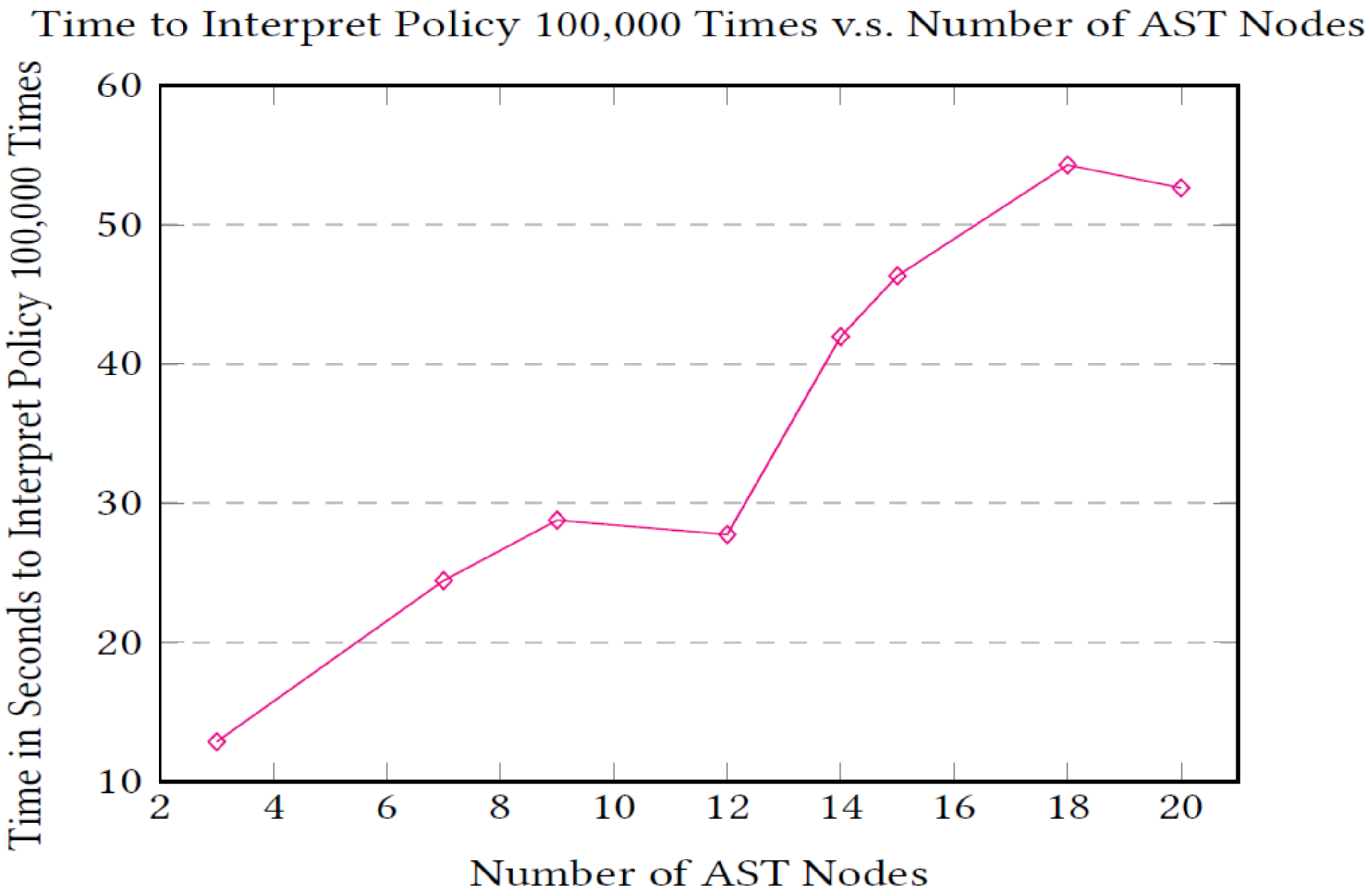
Preliminary Results

Attribute Authority



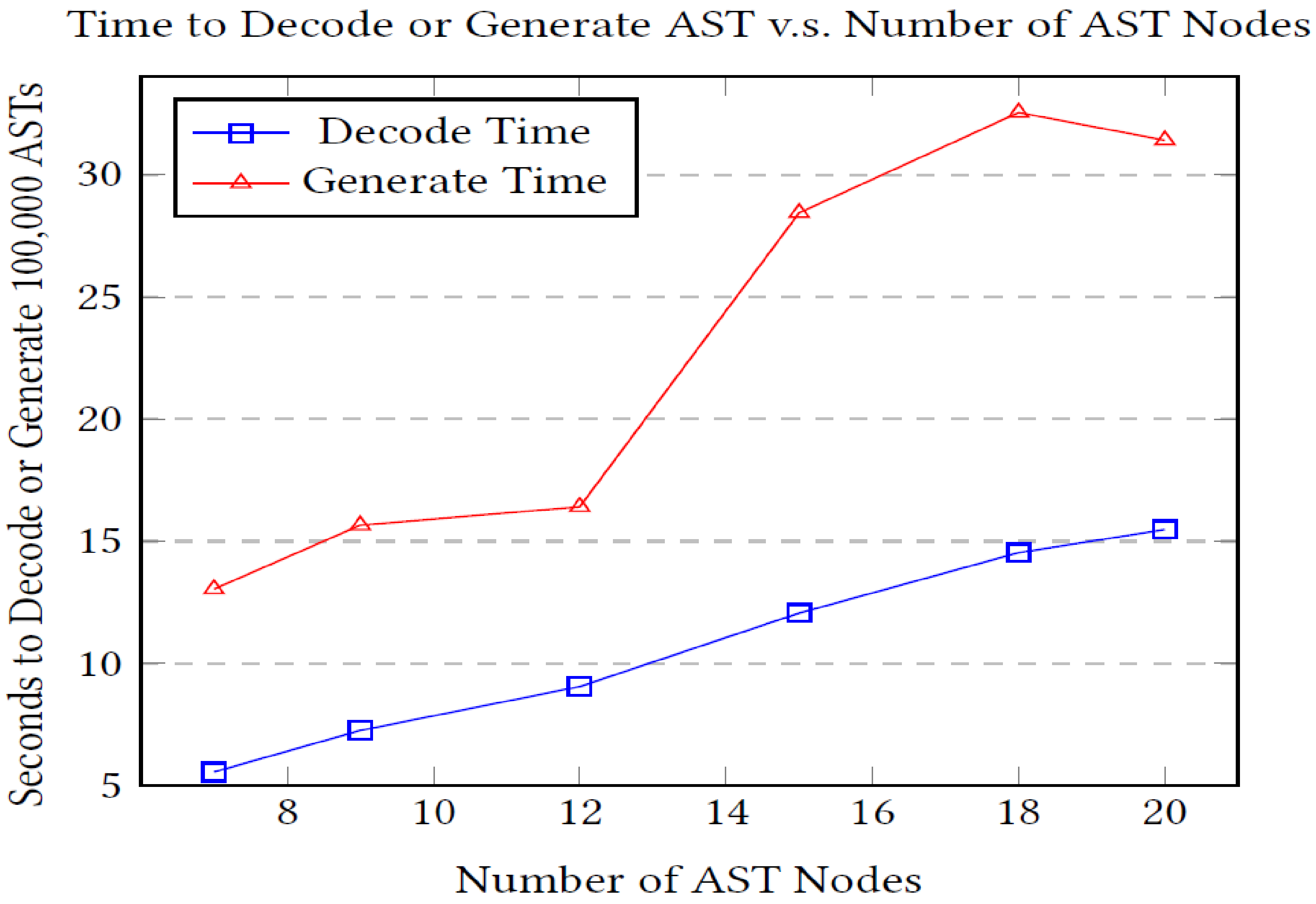
Preliminary Results

HGPL Interpreter



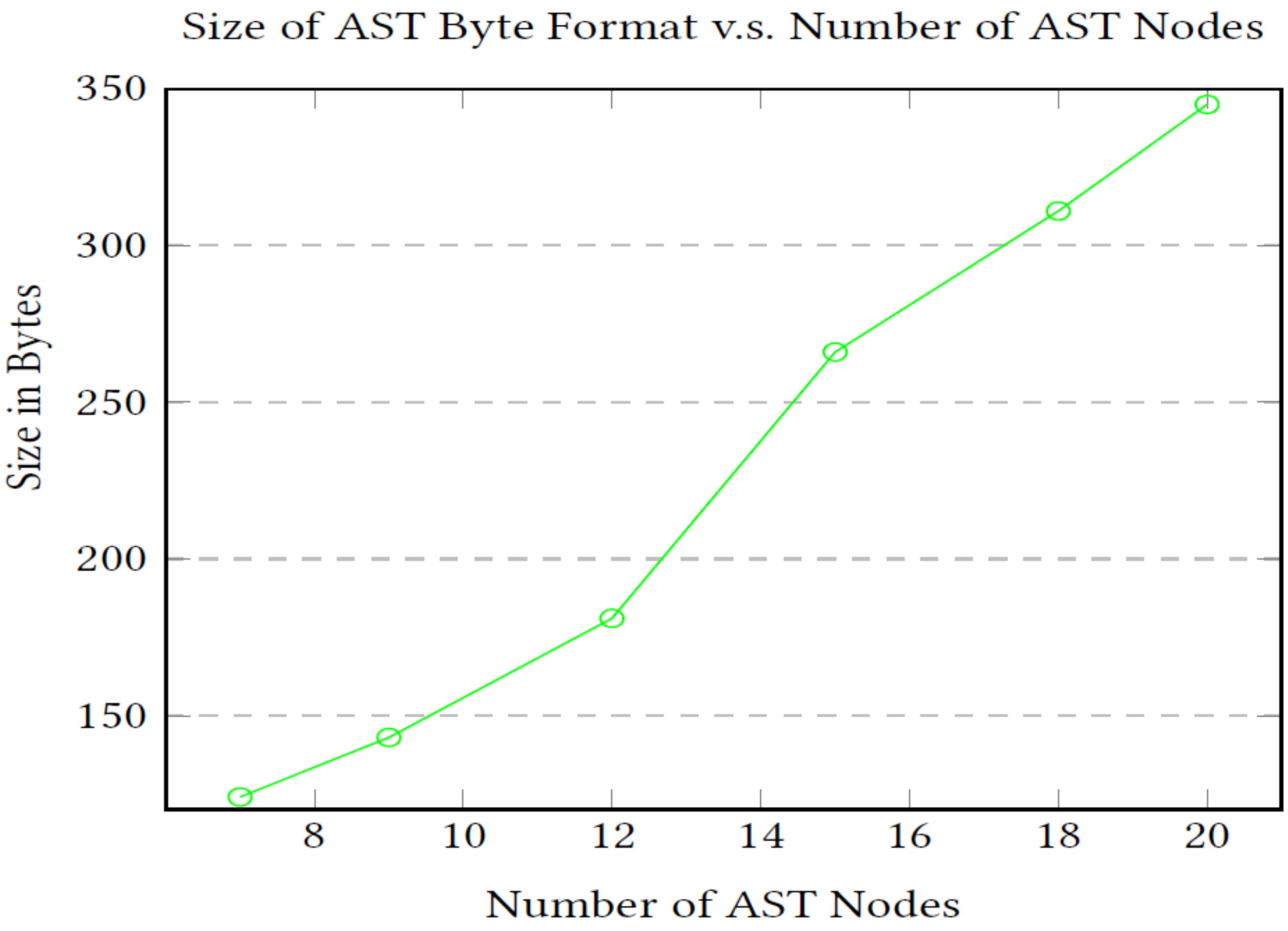
Preliminary Results

HGPL Interpreter



Preliminary Results

HGPL Interpreter



Conclusions & Future Work

Conclusions

- First architecture that supports full HGABAC model.
- Attribute Certificate specification and encoding presented.
- HGABAC namespace introduced.
- HGPL updated and interpreter created.
- Preliminary evaluation suggests linear scalability (with number of attributes and number of AST nodes).

Directions for Future Work

- Explore applicability to other ABAC models.
- Further evaluate architecture under more diverse and real-world scenarios.
- Investigate use of XACML and/or SAML and impact on performance.
- Extending HGABAC and HGAA to support user-to-user temporary delegation.
- Incorporate administration model (use $GURA_G$?).

Questions?