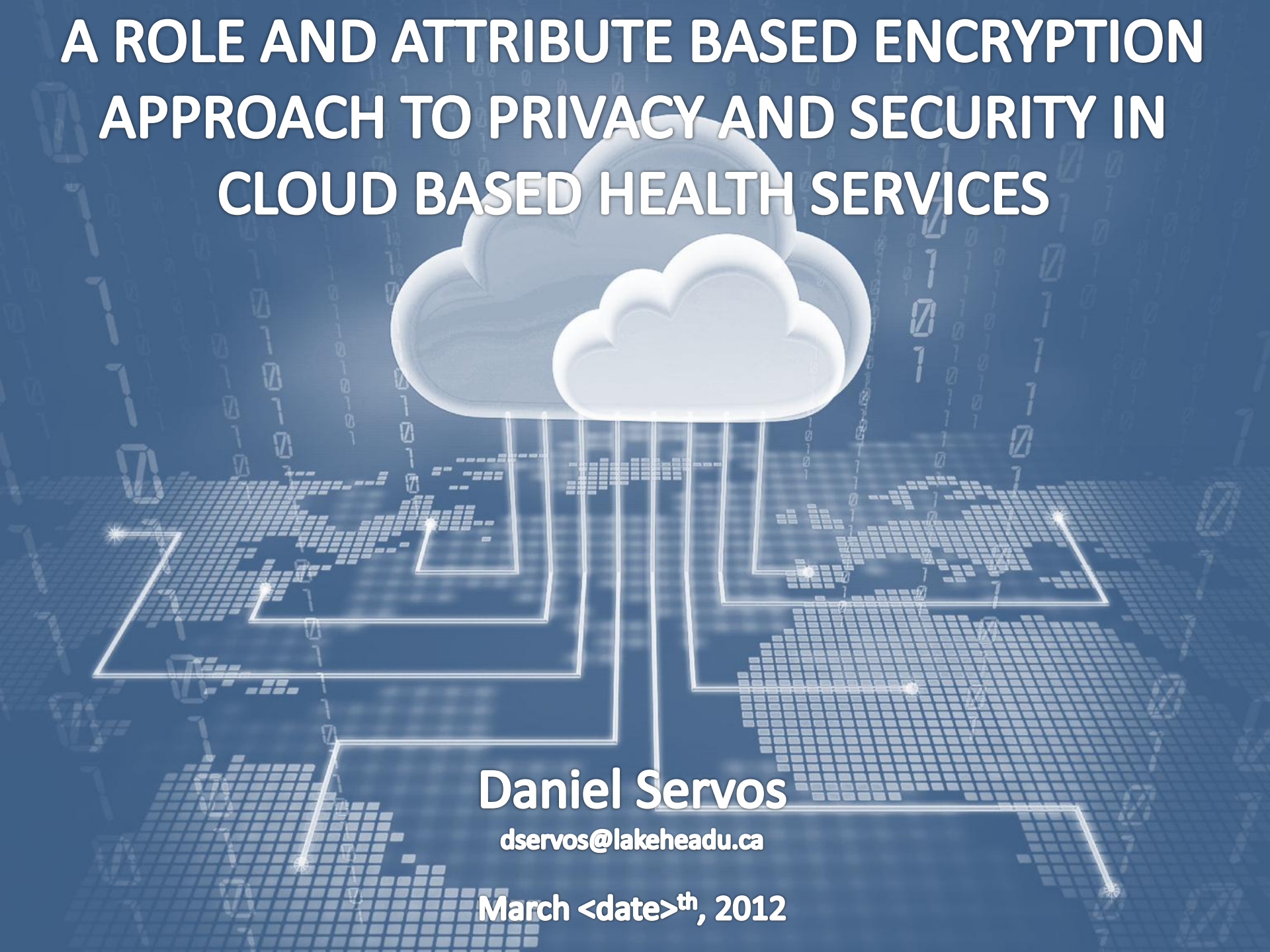
# A ROLE AND ATTRIBUTE BASED ENCRYPTION APPROACH TO PRIVACY AND SECURITY IN CLOUD BASED HEALTH SERVICES

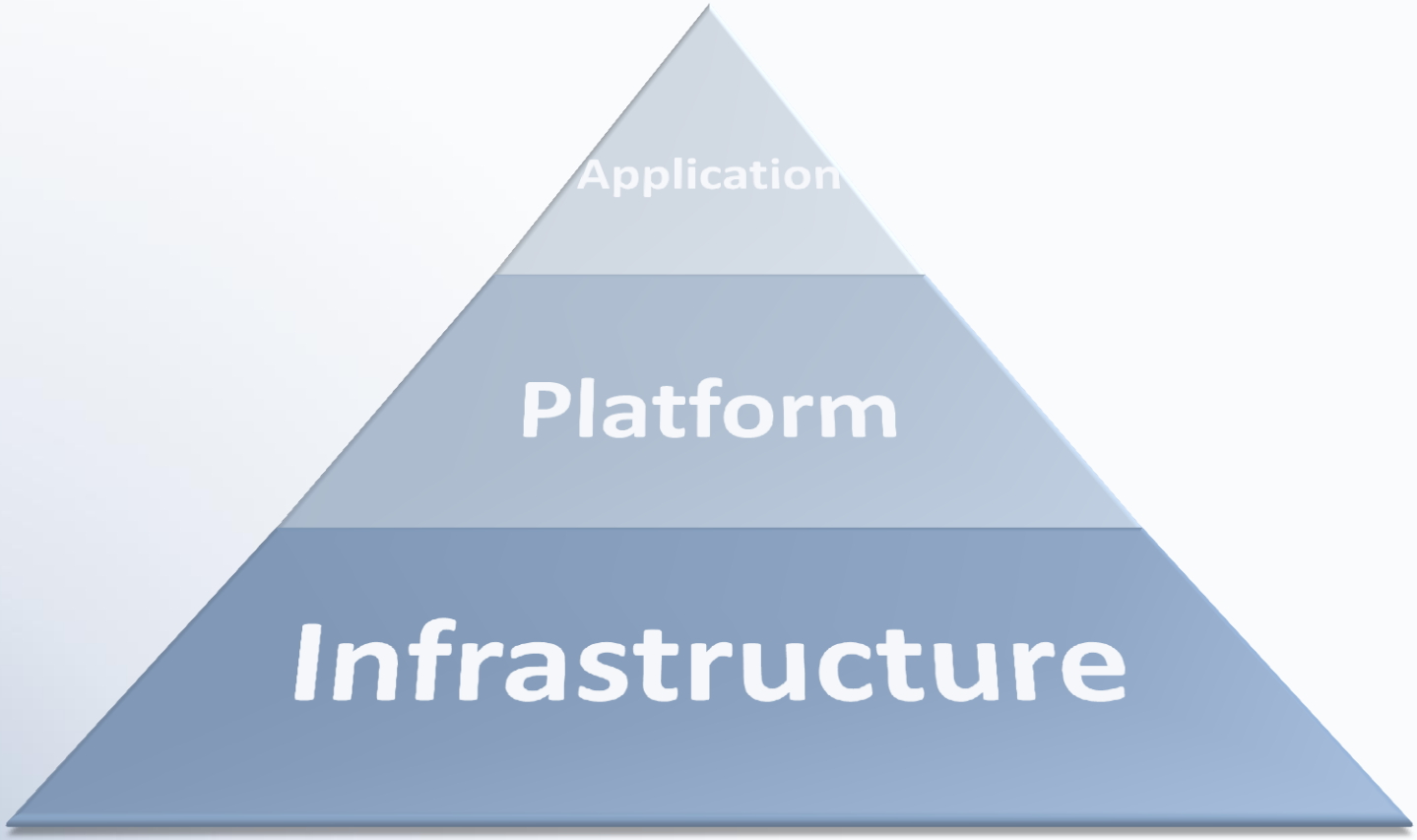



Daniel Servos

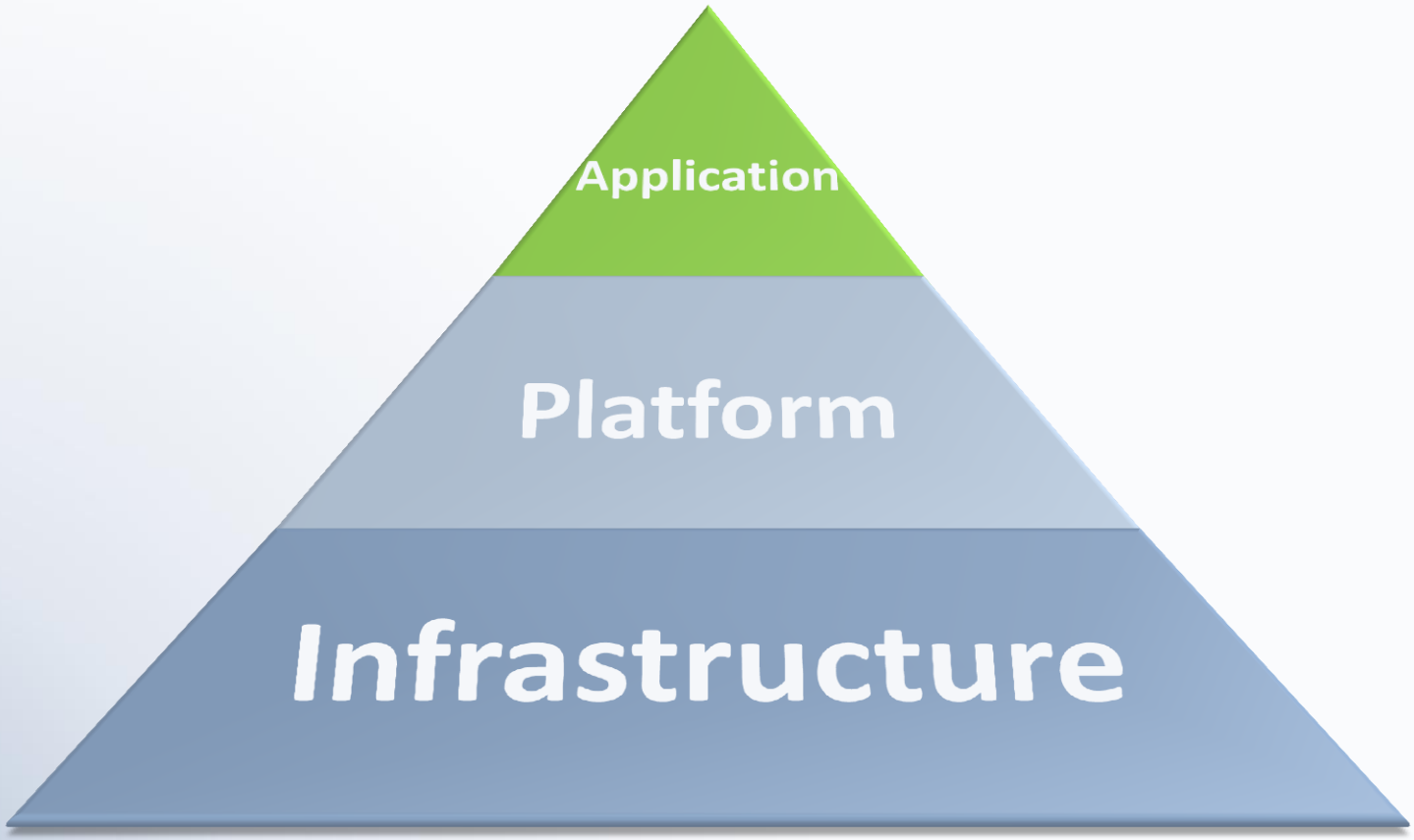
dservos@lakeheadu.ca


March <date>th, 2012

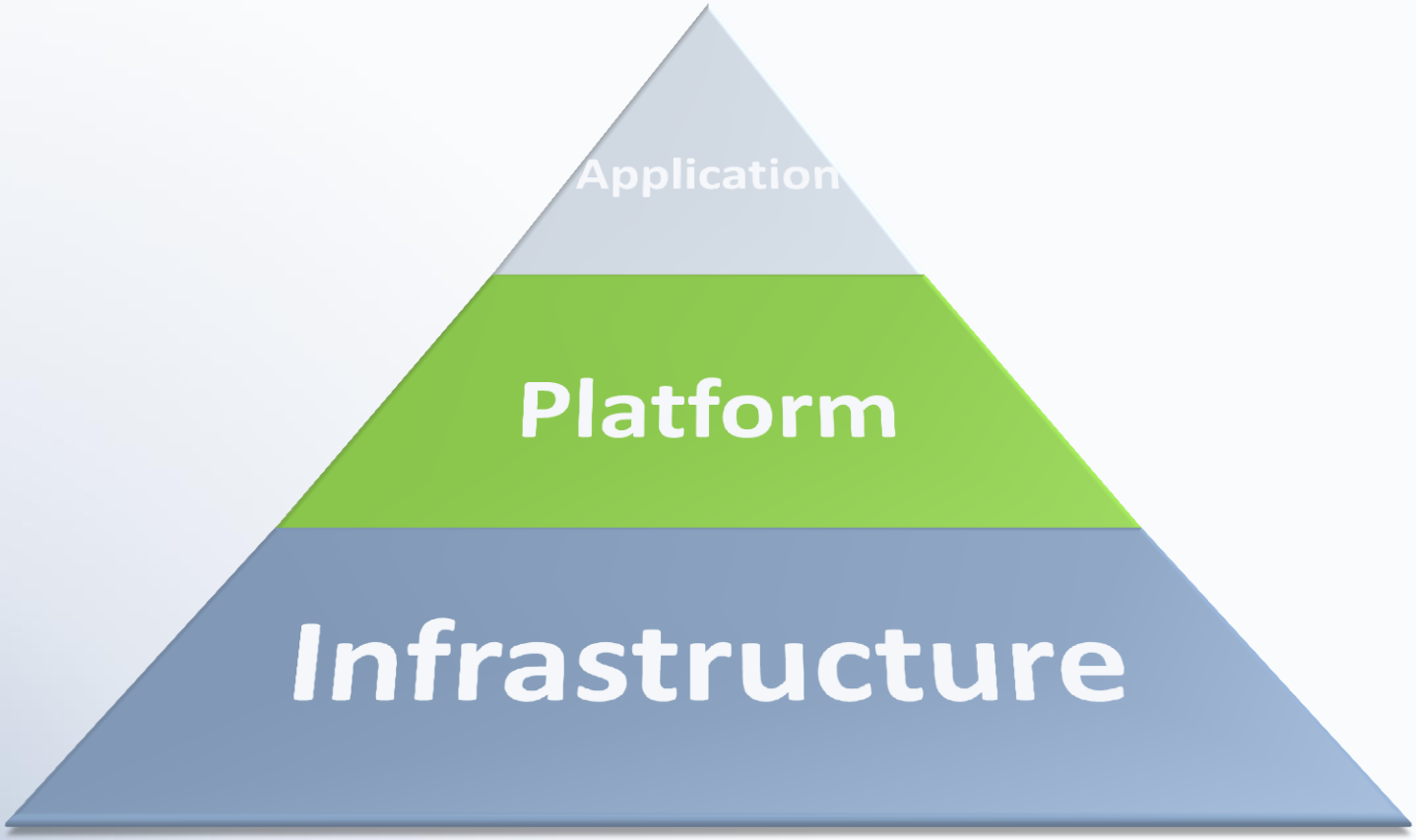
# What is Cloud Computing?



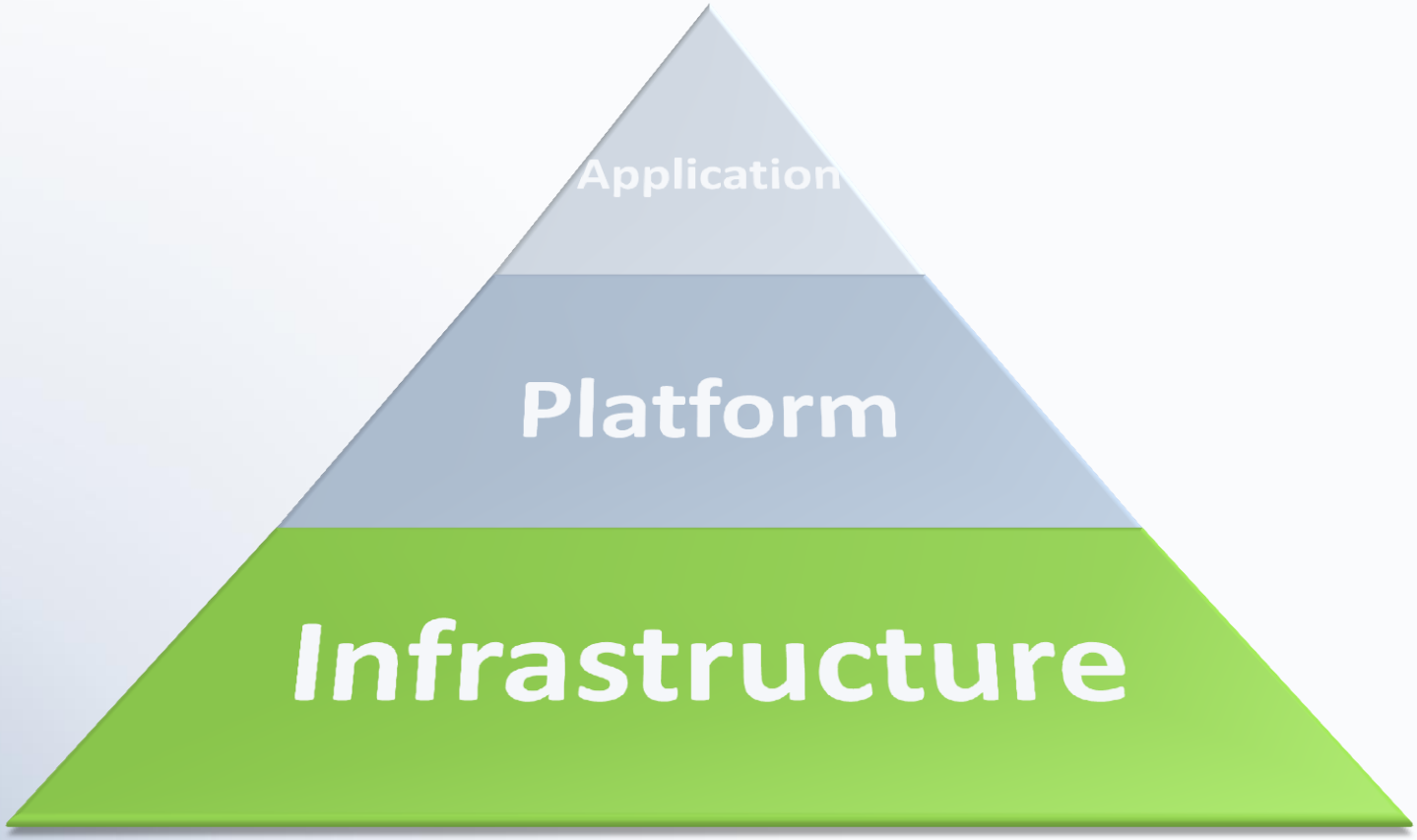
# What is Cloud Computing?



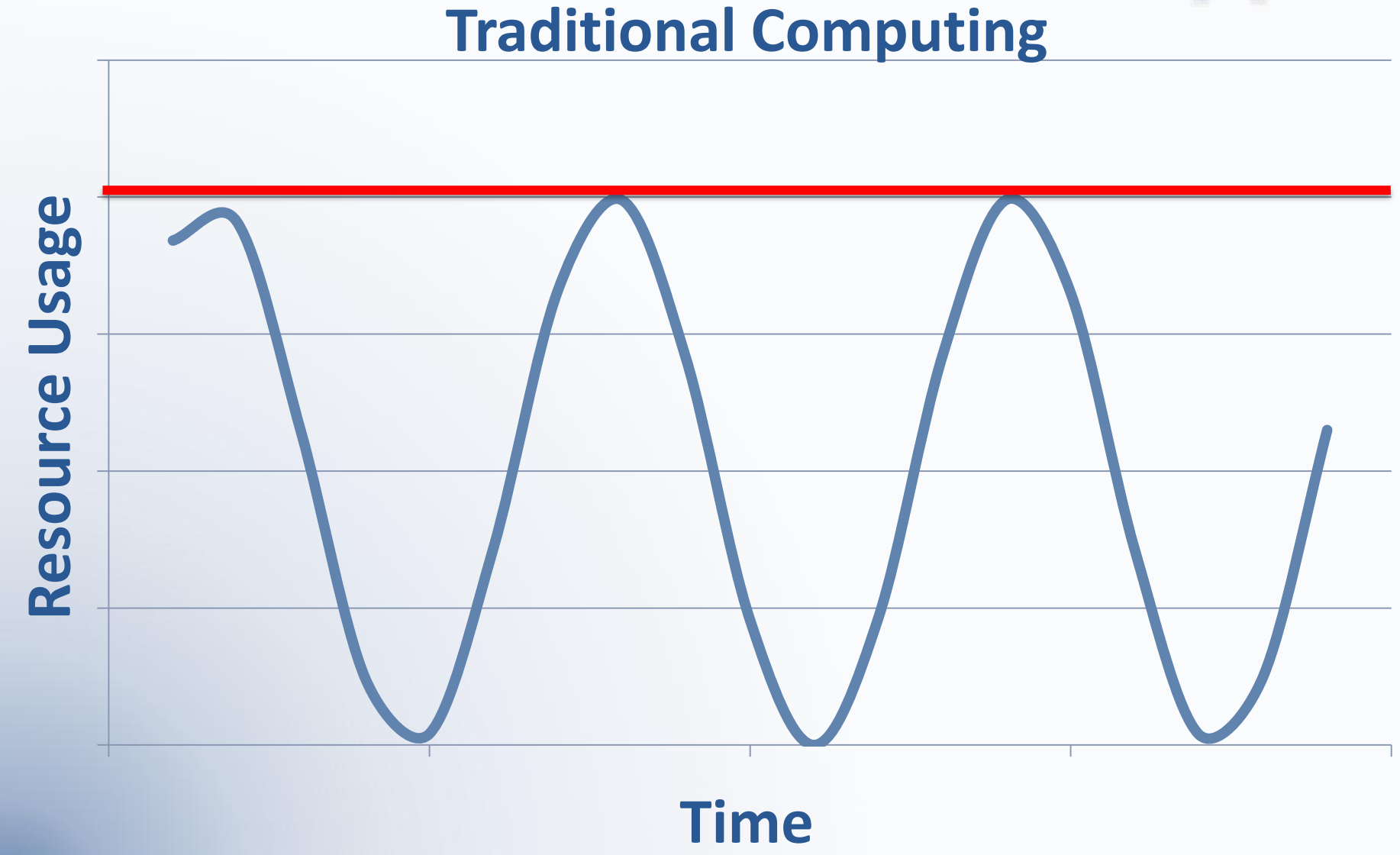
# What is Cloud Computing?



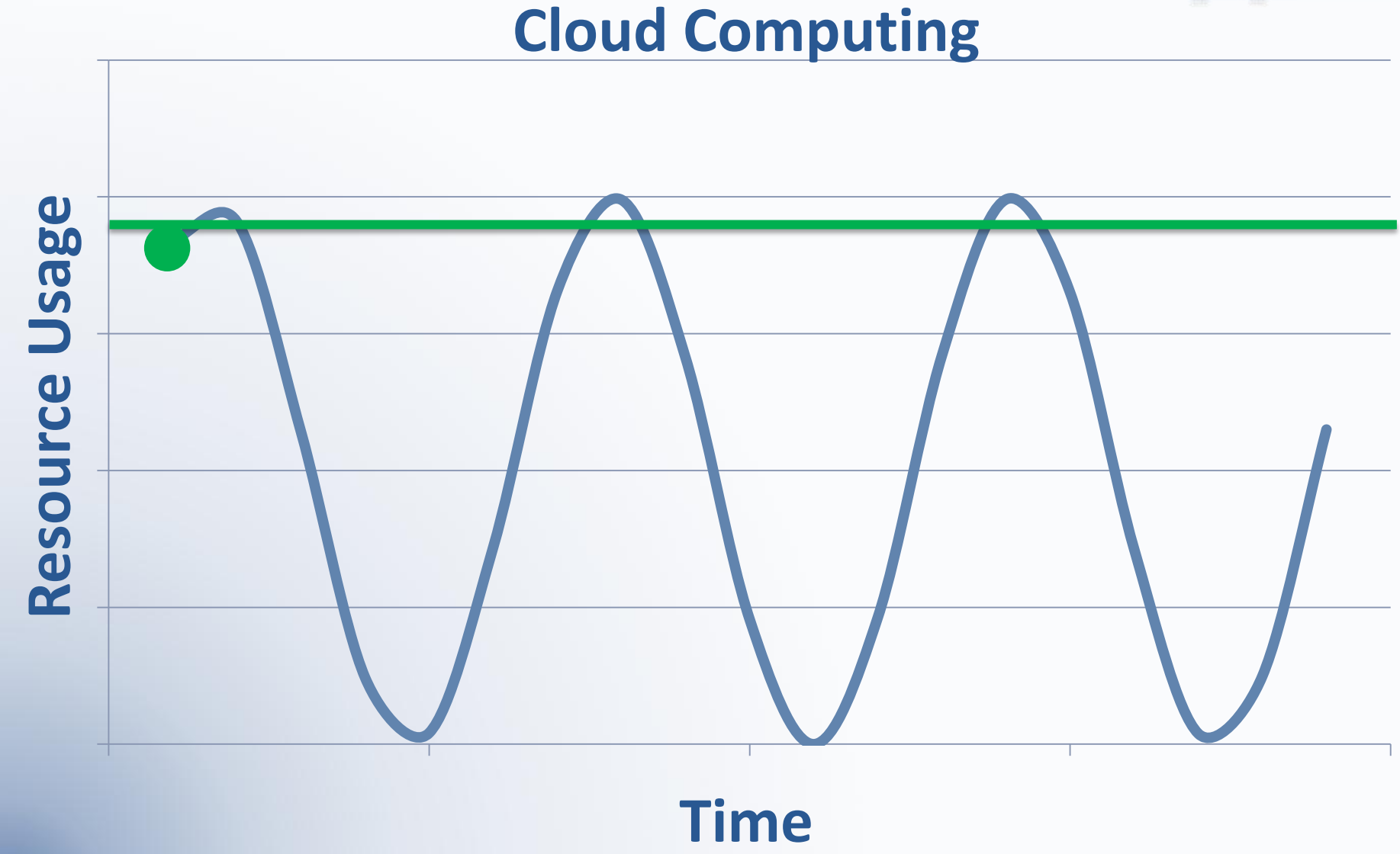
# What is Cloud Computing?



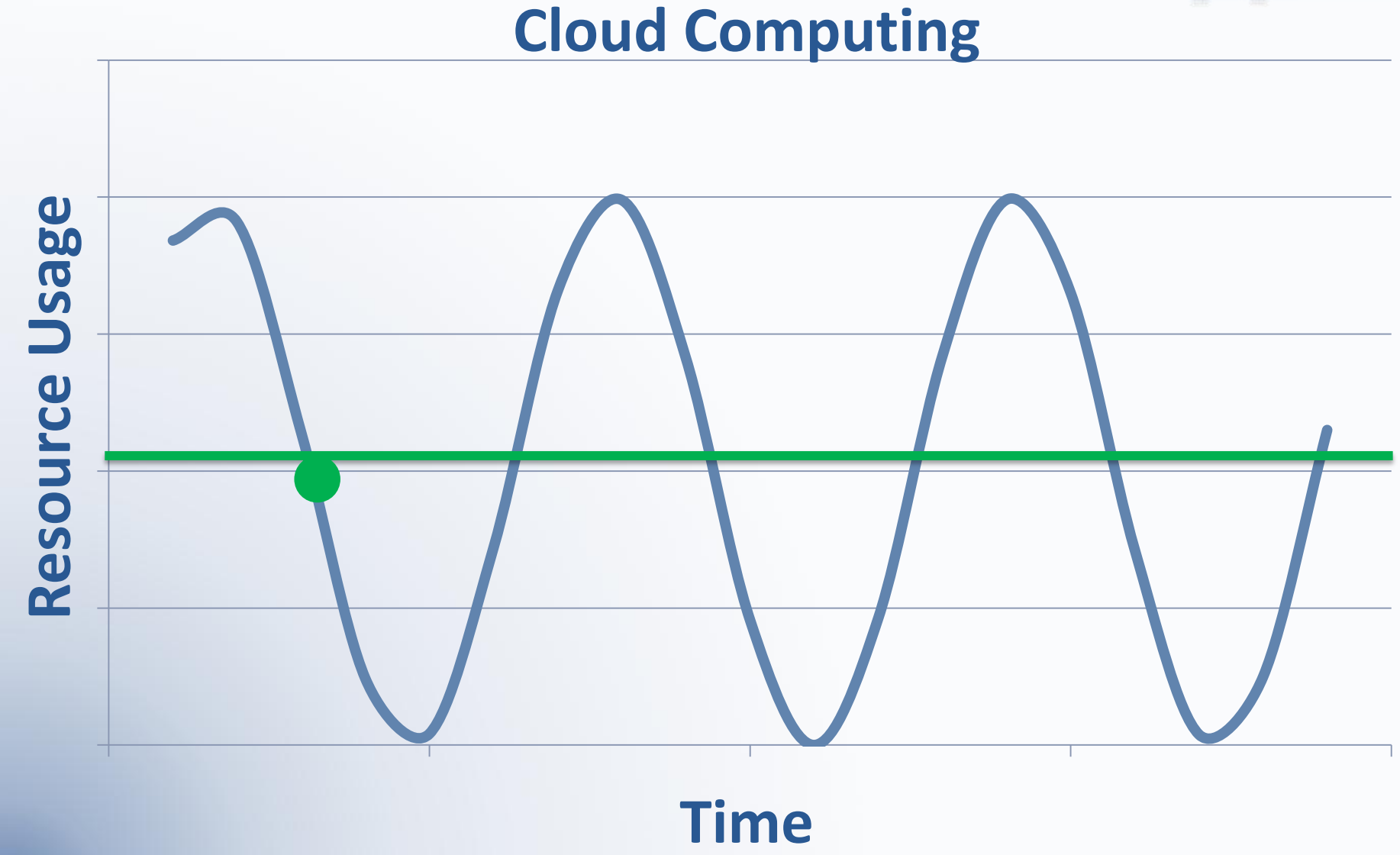
# What is Cloud Computing?



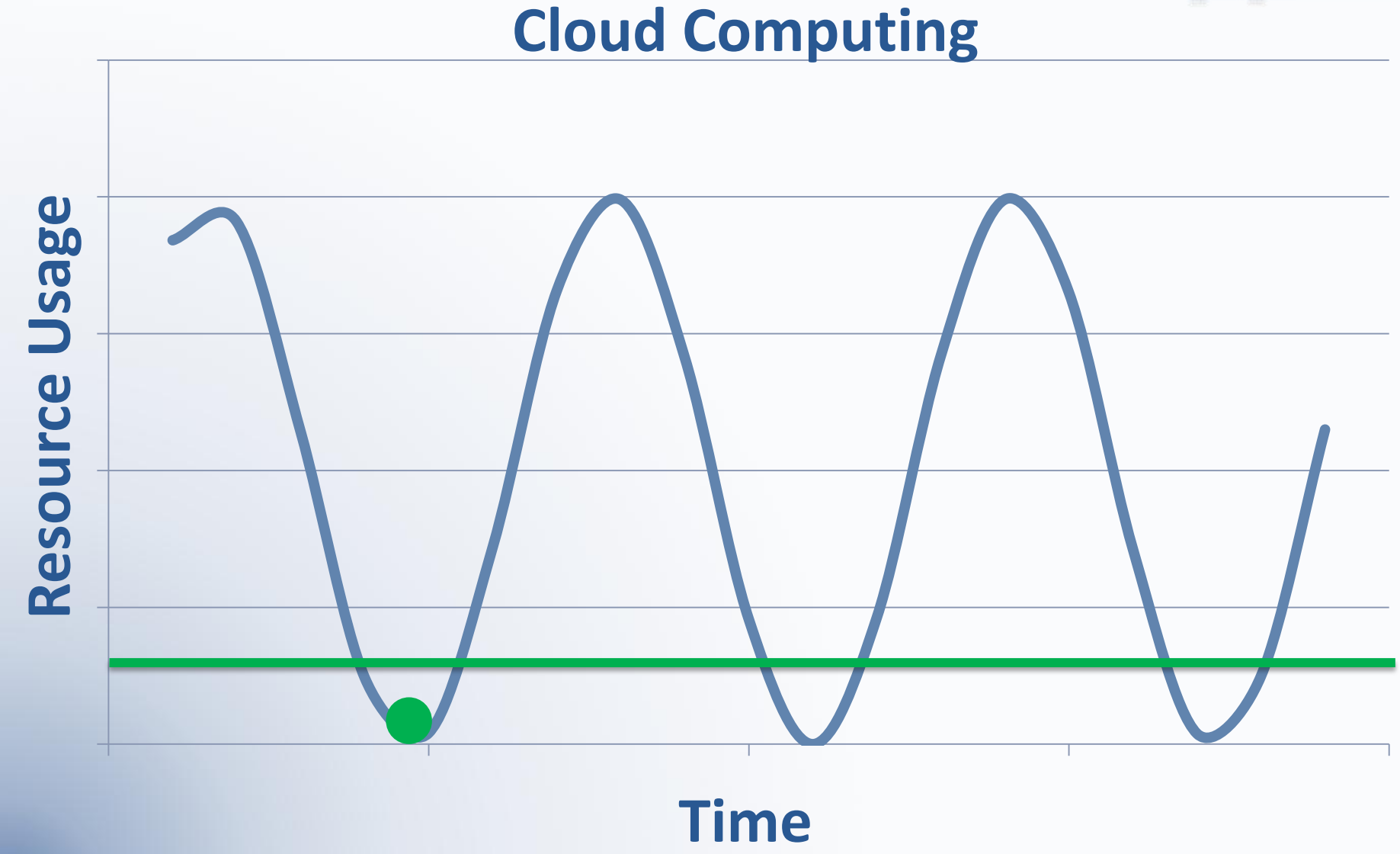
# What is Cloud Computing?



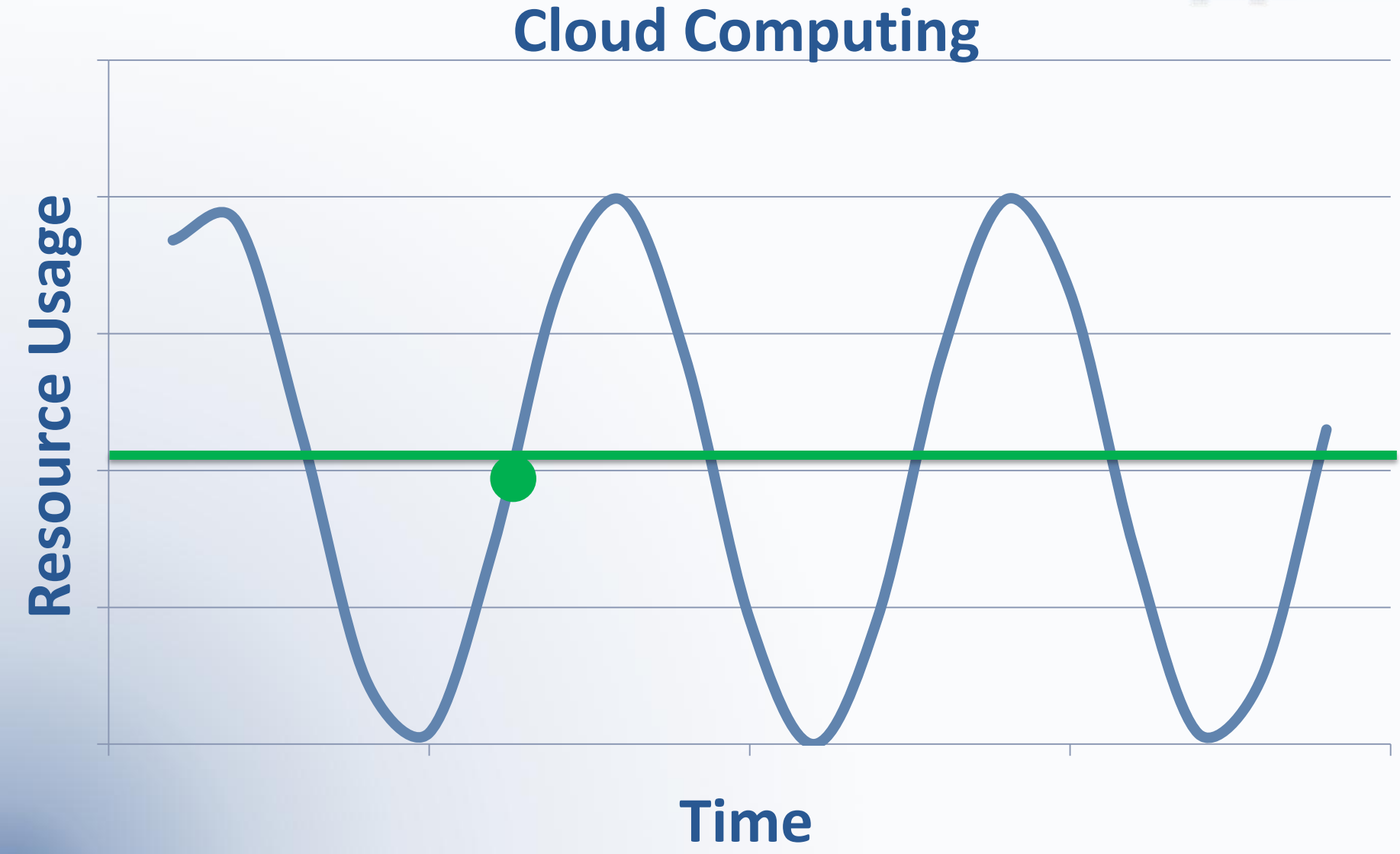
# What is Cloud Computing?



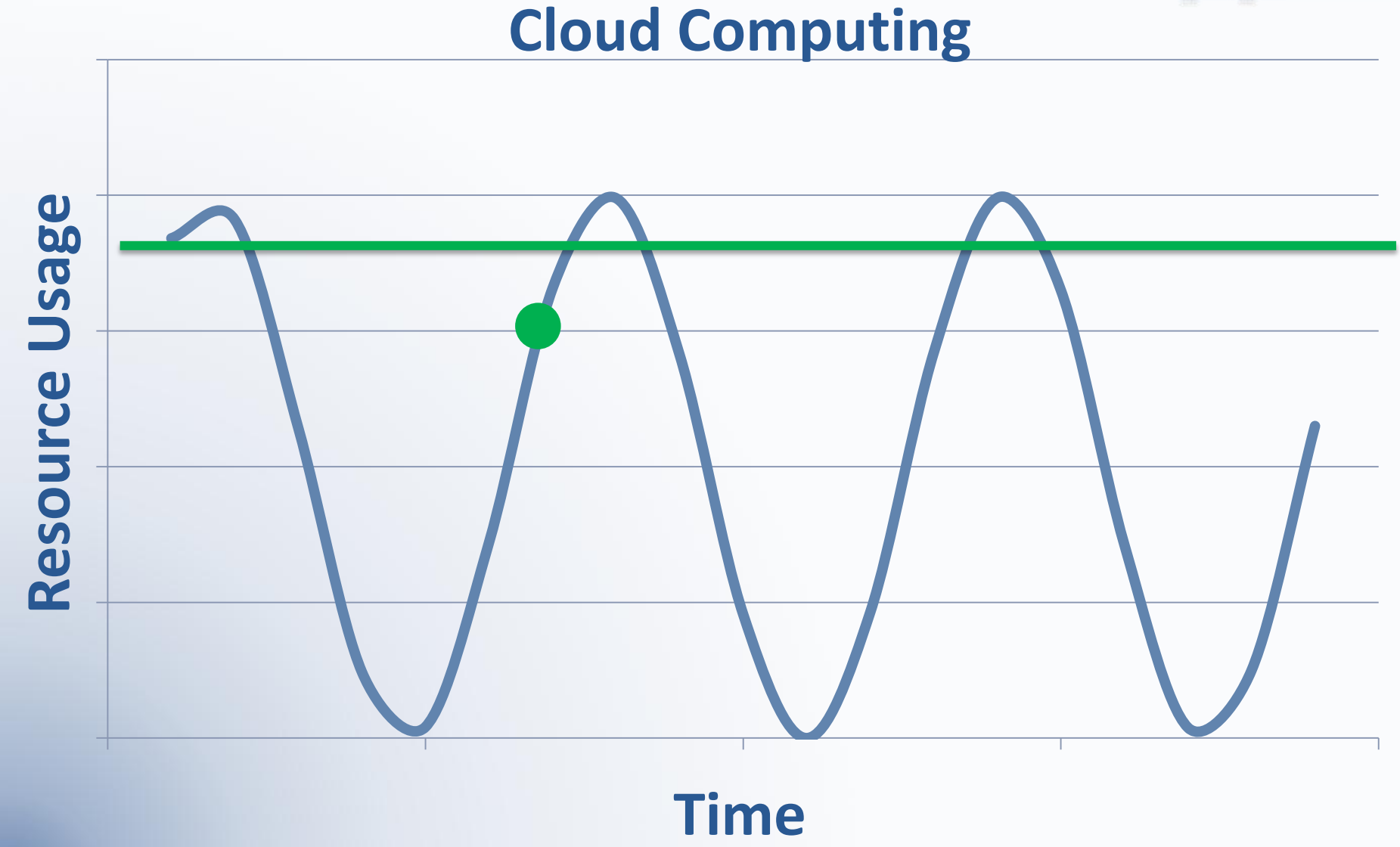
# What is Cloud Computing?



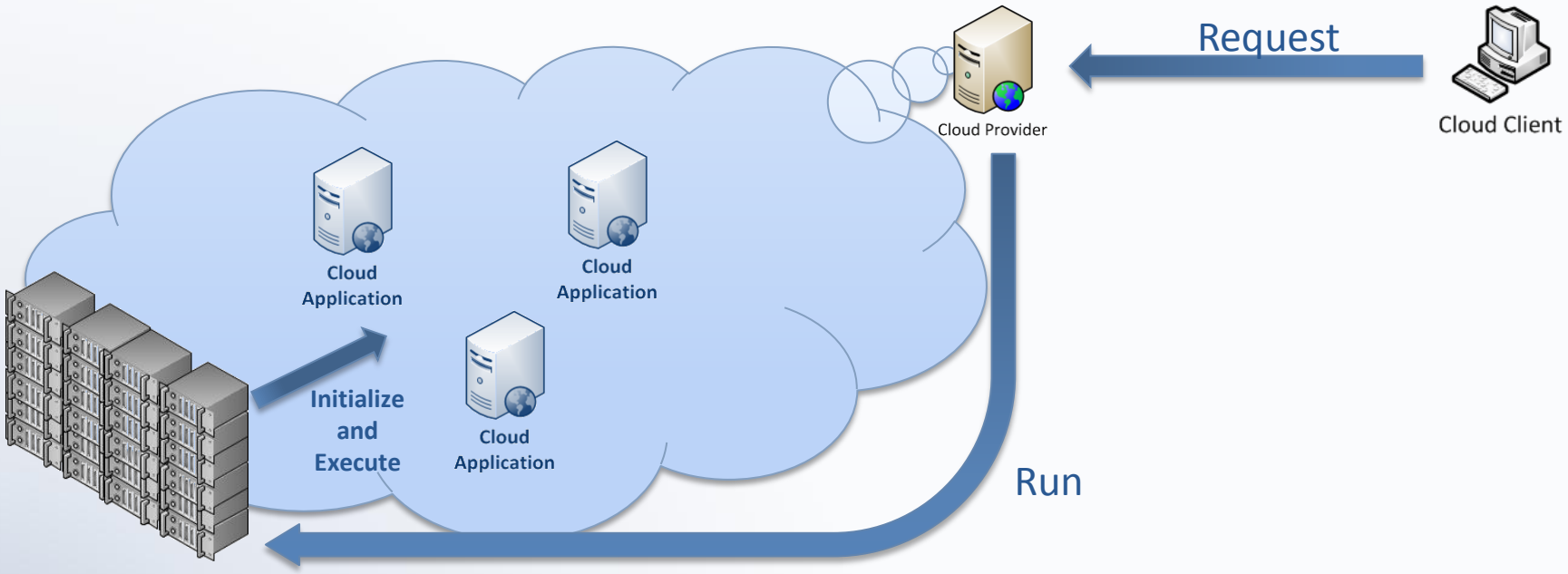
# What is Cloud Computing?



# What is Cloud Computing?



# What is Cloud Computing?



# What is Cloud Computing?

*"...a type of parallel and distributed system consisting of a collection of interconnected and virtualised computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers"*
(Buyya, Yeo, & Venugopal, 2008)

# What is Cloud Computing?

*"Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs."* (Vaquero & al., 2008)

# The Cloud Problem

**Roadblocks to Cloud Adoption:**

- Confidentiality
- Auditability
- Security
- Legal

**Challenges for Cloud Developers:**

- Bottlenecks
- Distributed
- Volatile Storage
- Dynamic IP

# Current Approaches and Techniques

# Current Approaches and Techniques

**Traditional Encryption Falls Short:**

- Cloud provider still has the key
- All users need the key
- Scalability?
- Access control?
- Off the cloud?

# Hardware Based Solution?

# Current Approaches and Techniques

## Cryptographic Coprocessors:

- Computer on a chip
- Dedicated to cryptographic operations
- Tamper-resistant

# Current Approaches and Techniques

**Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures** (Itani, Kayssi, & Chehab, 2009)

- Set of security protocols (PaaS) using cryptologic coprocessors
- Trusted third party (TTP) organization:
  - Configures
  - Installs
  - Inspects
  - Key distribution
- Cloud users:
  - Obtain keys from TTP
  - Create cloud applications with coprocessor in mind
  - Protected components are encrypted

# Current Approaches and Techniques

**Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures** (Itani, Kayssi, & Chehab, 2009)

- Cloud Provider:
  - Stores data under a given profile:
    - No Privacy
    - Privacy with Trusted Provider
    - Privacy with Non-Trusted Provider
  - Provides privacy feedback:
    - Via privacy daemon on coprocessor
    - Encrypted audit log of privacy-related operations
    - Hash chain of log (Schneier & Kelsey, 1999) (Itani, Kayssi, & Chehab, 2005)

# Current Approaches and Techniques

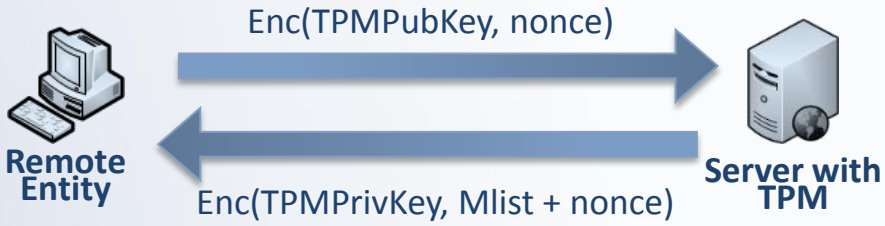
**Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures** (Itani, Kayssi, & Chehab, 2009)

- **Criticisms:**
  - Lack of support
  - Scalability
    - Moves applications from cloud infrastructure to coprocessors
    - Limited coprocessor hardware
    - Limited number of coprocessors
  - Sharing
  - Heavy use of TTP

# Current Approaches and Techniques

**Towards Trusted Cloud Computing** (Santos, Gummadi, & Rodrigues, 2009)

- Propose design for a Trusted Cloud Computing Platform (TCCP)
- Trusted platform module (TPM) based
  - Type of cryptographic coprocessor
  - Ability to provide “remote attestation”:
    - Assigned key pair by manufacture
    - Creates a measurement list at boot
    - Remote entity may request attestation:



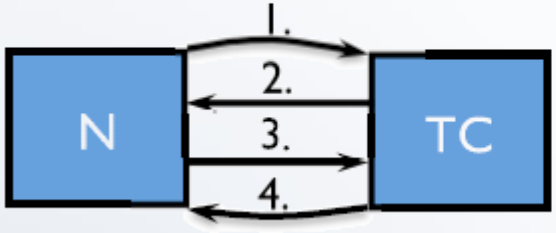
# Current Approaches and Techniques

**Towards Trusted Cloud Computing** (Santos, Gummadi, & Rodrigues, 2009)

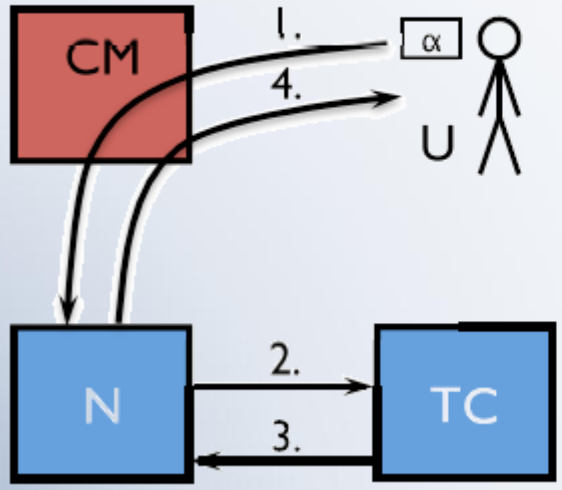
- TPM not enough to secure virtual machines
- TCCP:
  - Trusted virtual machine monitor (TVMM)
    - Limits privileged users
  - Trusted coordinator (TC)
    - Ran by trusted third party
    - Manages TVMM nodes
    - Records TVMM's TPM's public key and expected measurement list
    - Publishes its own TPM's public key, measurement list and trust key

# Current Approaches and Techniques

**Towards Trusted Cloud Computing** (Santos, Gummadi, & Rodrigues, 2009)




1. $n_N$
2. $\{ML_{TC}, n_N\}_{EK^p_{TC}}, n_{TC}$
3. $\{\{ML_N, n_{TC}\}_{EK^p_N}, TK^P_N\}_{TK^P_{TC}}$
4. $\{accepted\}_{TK^P_N}$



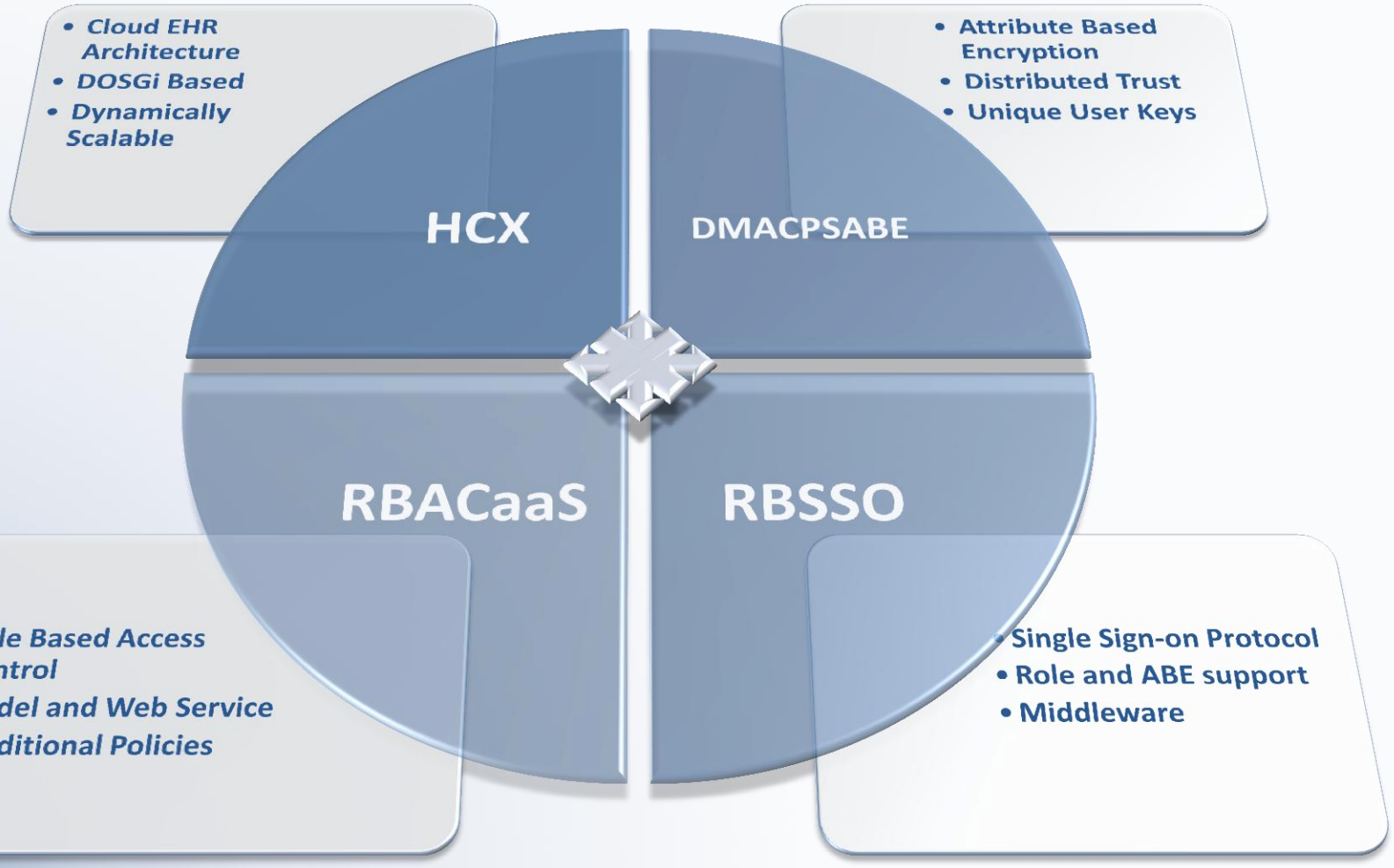

1. $\{\alpha, \#\alpha\}_{K_{VM}}\{n_U, K_{VM}\}_{TK^P_{TC}}$
2. $\{\{\{n_U, K_{VM}\}_{TK^P_{TC}}, n_N\}_{TK^P_N}, N\}_{TK^P_{TC}}$
3. $\{\{n_N, n_U, K_{VM}\}_{TK^P_N}\}_{TK^P_{TC}}$
4. $\{n_U, N\}_{K_{VM}}$

# Current Approaches and Techniques

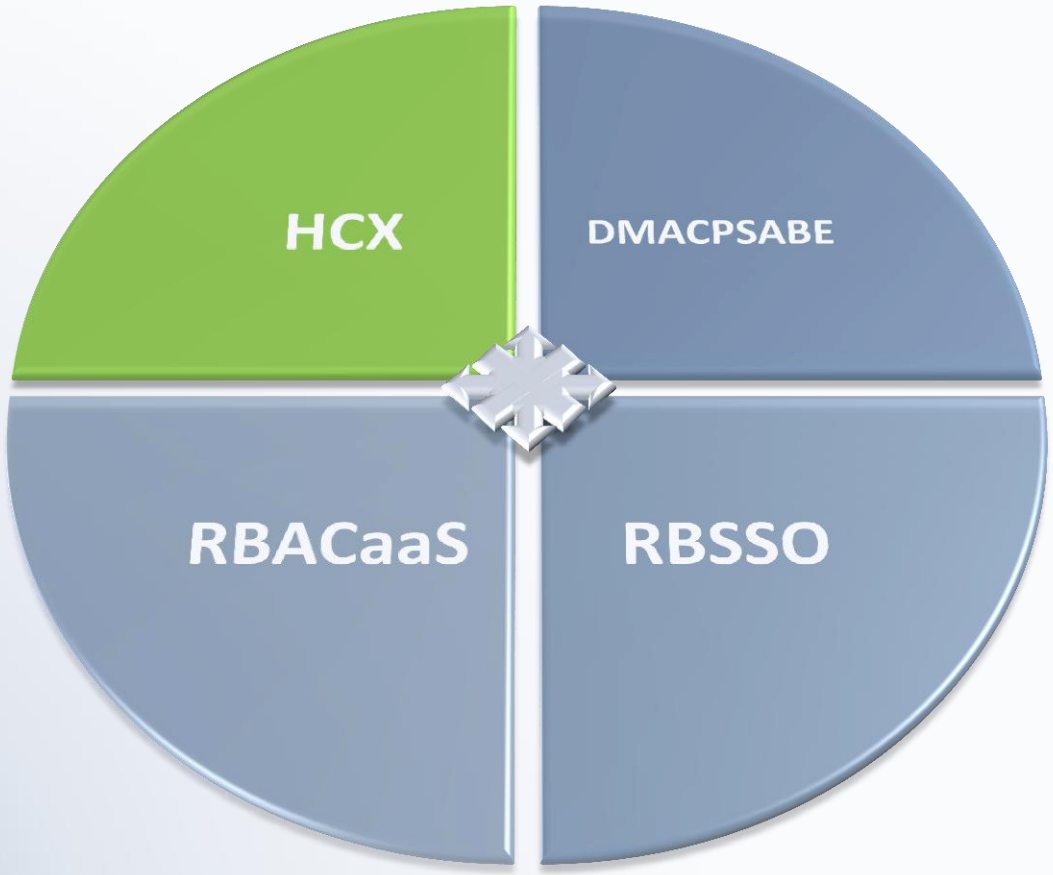
**Towards Trusted Cloud Computing** (Santos, Gummadi, & Rodrigues, 2009)

- **Criticisms:**
  - Lack of support
  - Heavy use of trusted third party
  - Vulnerable to system admin with hardware access
    - Read the contents of RAM
      - In active system (Samyde, Skorobogatov, Anderson, & Quisquater, 2003)
      - Cold boot attack (Halderman, et al., 2009).
    - Contents of hard drive

# Towards Cloud Security and Privacy



# Health Cloud eXchange (HCX)



# The Health Care Use Case

## Why Electronic Health Records?

- Demand for low cost, low maintenance EHRs (Urowitz, et al., 2008)
- Regulatory compliance
- XML based document formats (CCR, CCD, etc)
- Open problem (Zhang, Cheng, & Boutaba, 2010) , (Armbrust, et al., 2009)

# The Health Care Use Case

## Privacy Laws:

**USA**:

- Health Insurance Portability and Accountability Act (HIPAA)
- Gramm-Leach-Bliley Act
- Video Privacy Protection Act
- Cable Communications Policy Act
- Tax preparation laws (e.g. 26 U.S.C. §§ 6713, 7216; 26 C.F.R. § 301.7216)
- Privacy Act of 1973
- Electronic Communications Privacy Act of 1986 (ECPA)

# The Health Care Use Case

## Privacy Laws:

**Canada:**

- Personal Information Protection and Electronic Documents Act (PIPEDA)
- An Act Respecting the Protection of Personal Information in the Private Sector (Quebec)
- The Personal Information Protection Act (Alberta)
- The Personal Information Protection Act (British Columbia)
- The Personal Health Information Protection Act (Ontario)

# EHR Formats

## Continuity of Care Record (CCR)

```
<ContinuityOfCareRecord xmlns='urn:astm-org:CCR'>
    <CCRDocumentObjectID>Doc</CCRDocumentObjectID>
    <Language>
        <Text>English</Text>
    </Language>
    <Version>V1.0</Version>
    <DateTime>
        <ExactDateTime>2008</ExactDateTime>
    </DateTime>
    <Patient>
        <ActorID>Patient</ActorID>
    </Patient>
    <Body>
        <VitalSigns>
            <Result>
              <CCRDataObjectID>0001</CCRDataObjectID>
              <Description>
                  <Text>Blood Pressure</Text>
              </Description>
              <Test>
                  <CCRDataObjectID>0002</CCRDataObjectID>
                  <Description>
                      <Text>Systolic</Text>
                      <Code>
                          <Value>163030003</Value>
                          <CodingSystem>SNOMEDCT</CodingSystem>
                      </Code>
                  </Description>
                  <TestResult>
                      <Value>120</Value>
                      <Units>
                          <Unit>mmHg</Unit>
                      </Units>
                  </TestResult>
                </Test>
            </Result>
        </VitalSigns>
    </Body>
    <Actors>
        <Actor>
            <ActorObjectID>Patient</ActorObjectID>
                <Person>
                    <Name>
                        <CurrentName>
                            <Given>John</Given>
                            <Family>Doe</Family>
                        </CurrentName>
                    </Name>
                </Person>
        </Actor>
    </Actors>
</ContinuityOfCareRecord>
```

# EHR Formats

## Continuity of Care Record (CCR)

### Patient Demographics

| Name | Date of Birth | Gender | Identification Numbers | Address / Phone |
|---|---|---|---|---|
| | 01, 1919 | Female | | |

### Alerts

| Type | Date | Code | Description | Reaction | Source |
|---|---|---|---|---|---|
| Allergy | Start date: 04, 2007 | 68387043040 (NDC) | **Amoxicillin** | -Severe | |

### Functional Status

| Type | Date | Code | Description | Status | Source |
|---|---|---|---|---|---|
| Pregnancy status | | 255409004 (SNOMEDCT) | **Pregnant** | | |
| Breastfeeding status | | 413712001 (SNOMEDCT) | **Breastfeeding** | Active | |

### Problems

| Type | Date | Code | Description | Status | Source |
|---|---|---|---|---|---|
| | Start date: 04, 2007 | 410.10 (ICD9) | **Aortic valve disorders** | Active | |

### Procedures

| Type | Date | Code | Description | Location | Substance | Method | Position | Site | Status | Source |
|---|---|---|---|---|---|---|---|---|---|---|
| | Start date: 04, 2007 | 144950 (CPT) | **Appendectomy** | | | | | | | |

### Medications

| Medication | Date | Status | Form | Strength | Quantity | SIG | Indications | Instruction | Refills | Source |
|---|---|---|---|---|---|---|---|---|---|---|
| **Ibuprofen** | Prescription date: 01, 2007 | Active | Tablet | 100 MG | | 1 tablet Oral 1 time per day | | | | |

### Immunizations

| Code | Vaccine | Date | Route | Site | Source |
|---|---|---|---|---|---|
| Diphtheria antitoxin (CPT) | **Ibuprofen** | Start date: 04, 2007 | | | |

### Vital Signs

# A Cloud Based Infrastructure for Sharing Health Records

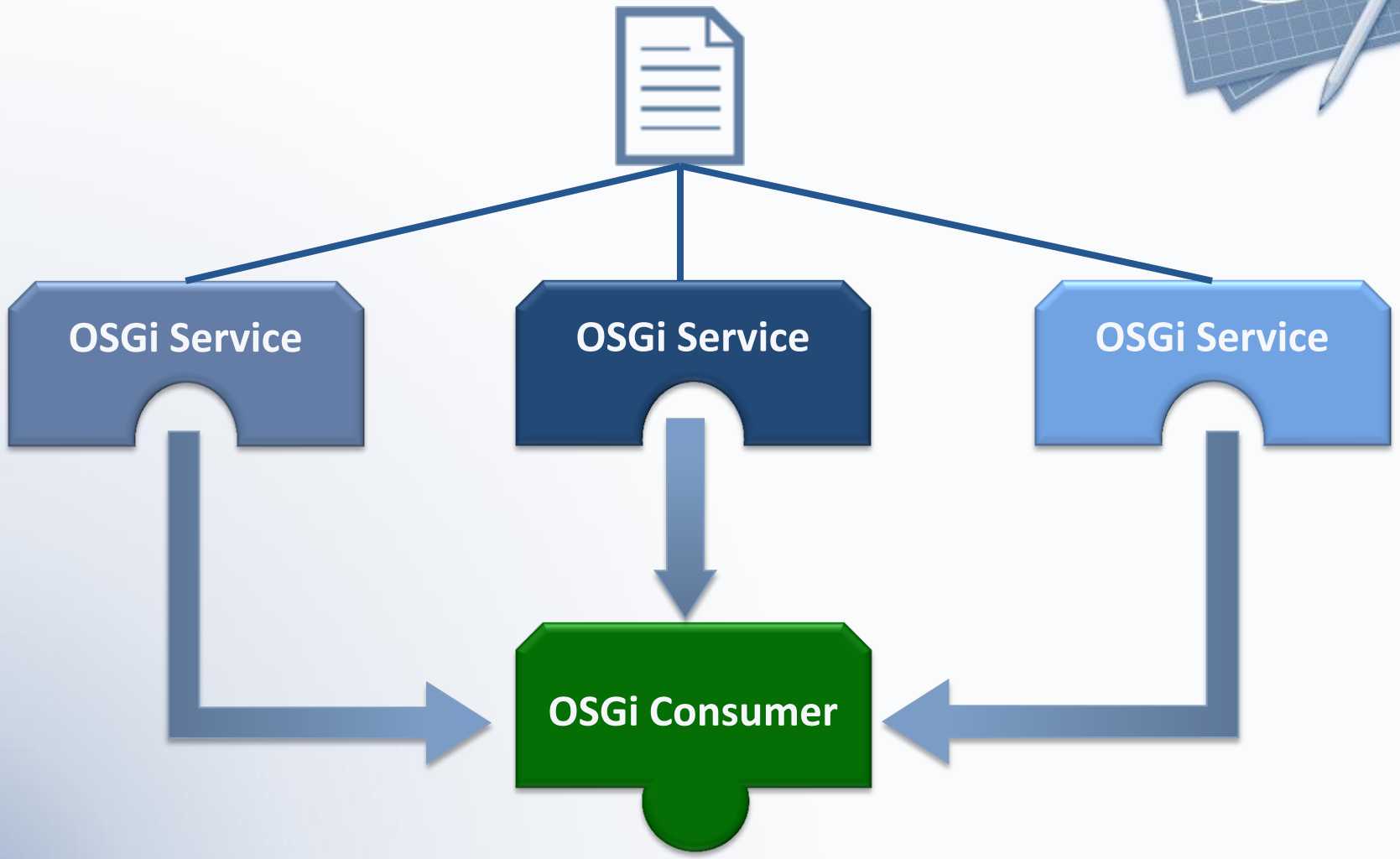
## Health Cloud eXchange (HCX)

**Goals:**

- Modularity
- Interoperability
- Loosely Coupled
- Simplicity
- Leverage Cloud Infrastructure
- Distributed
- Extendibility

# OSGi?

# OSGi



# OSGi

```
interface Hello {
   public String HelloWorld(String name);
}
```

OSGi Service

```
public String HelloWorld(
    return “Hello World!”;
}
```

OSGi Service

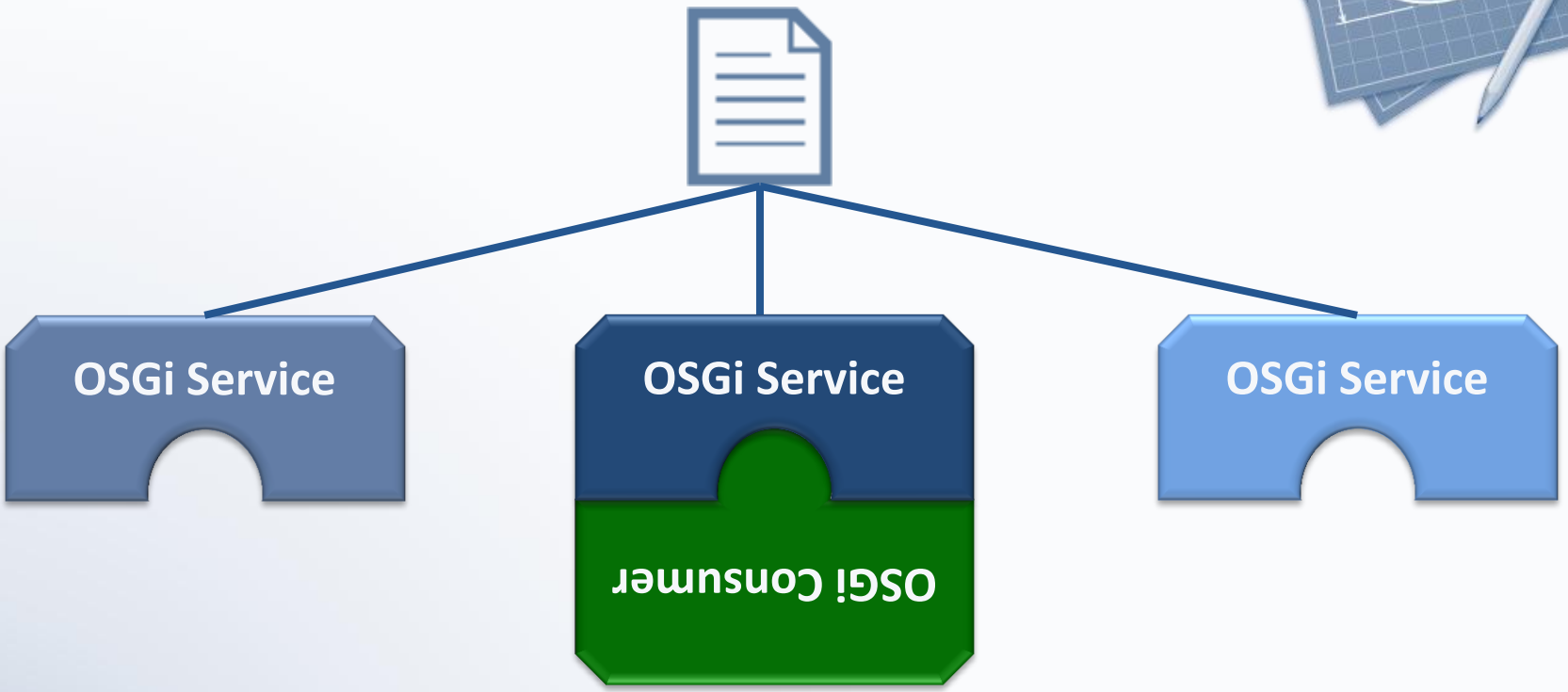
```
public String HelloWorld(String name) {
    return “Good Bye!”;
}
```

OSGi Service

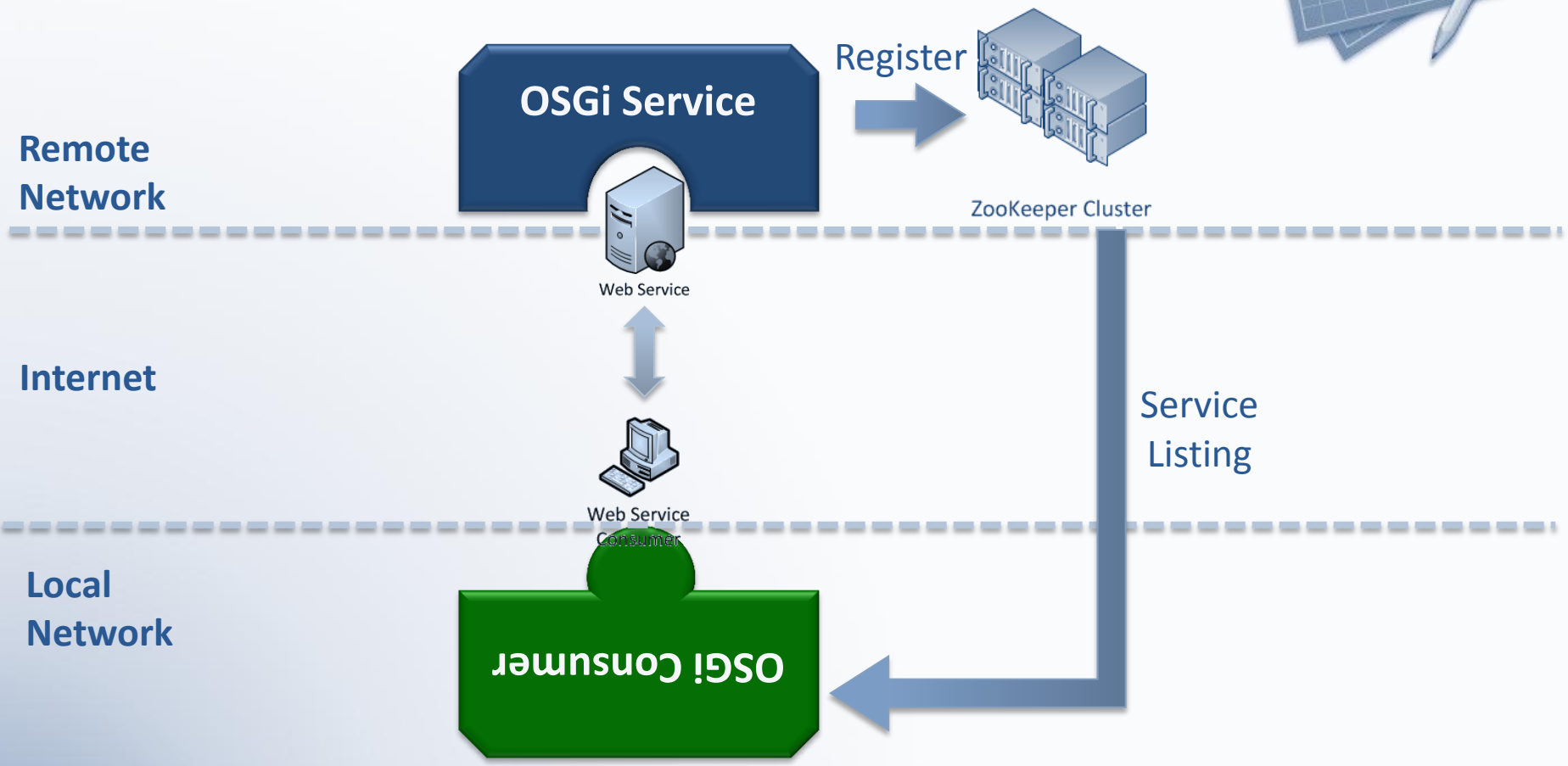
```
public String HelloWorld(String name) {
    return “Hello ” + name + “!”;
}
```

OSGi Consumer

# OSGi



# Distributed OSGi



# Adopting DOSGi for the Cloud

## Virtual Machine Images:

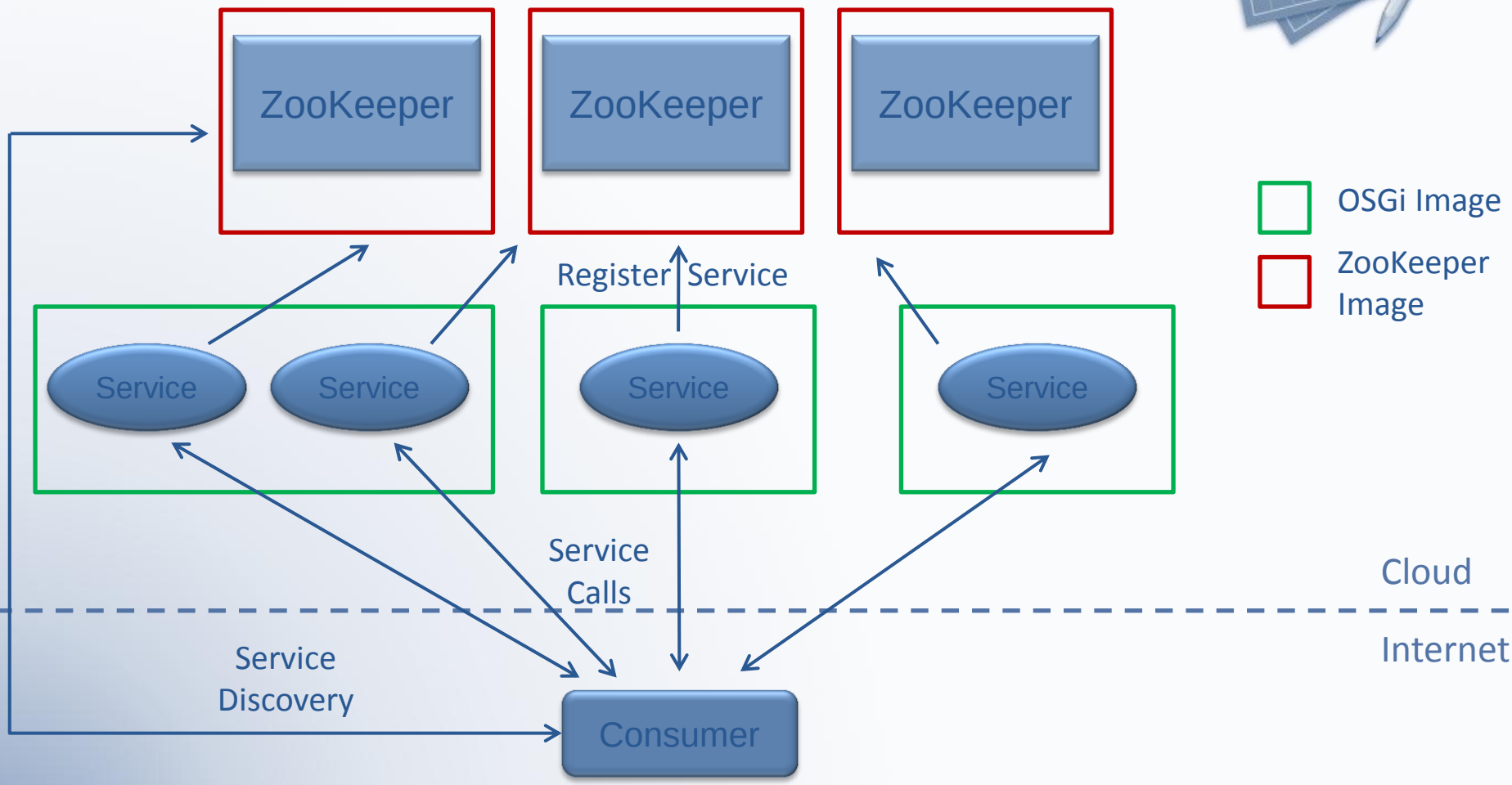
### OSGi Machine Image

- Setup:
  - Linux OS
  - OSGi implementation installed
  - Pax Runner installed (optional)
  - Compendium interfaces bundle
  - Apache CXF DOSGi bundle
- Will Run:
  - OSGi services

### ZooKeeper Machine Image

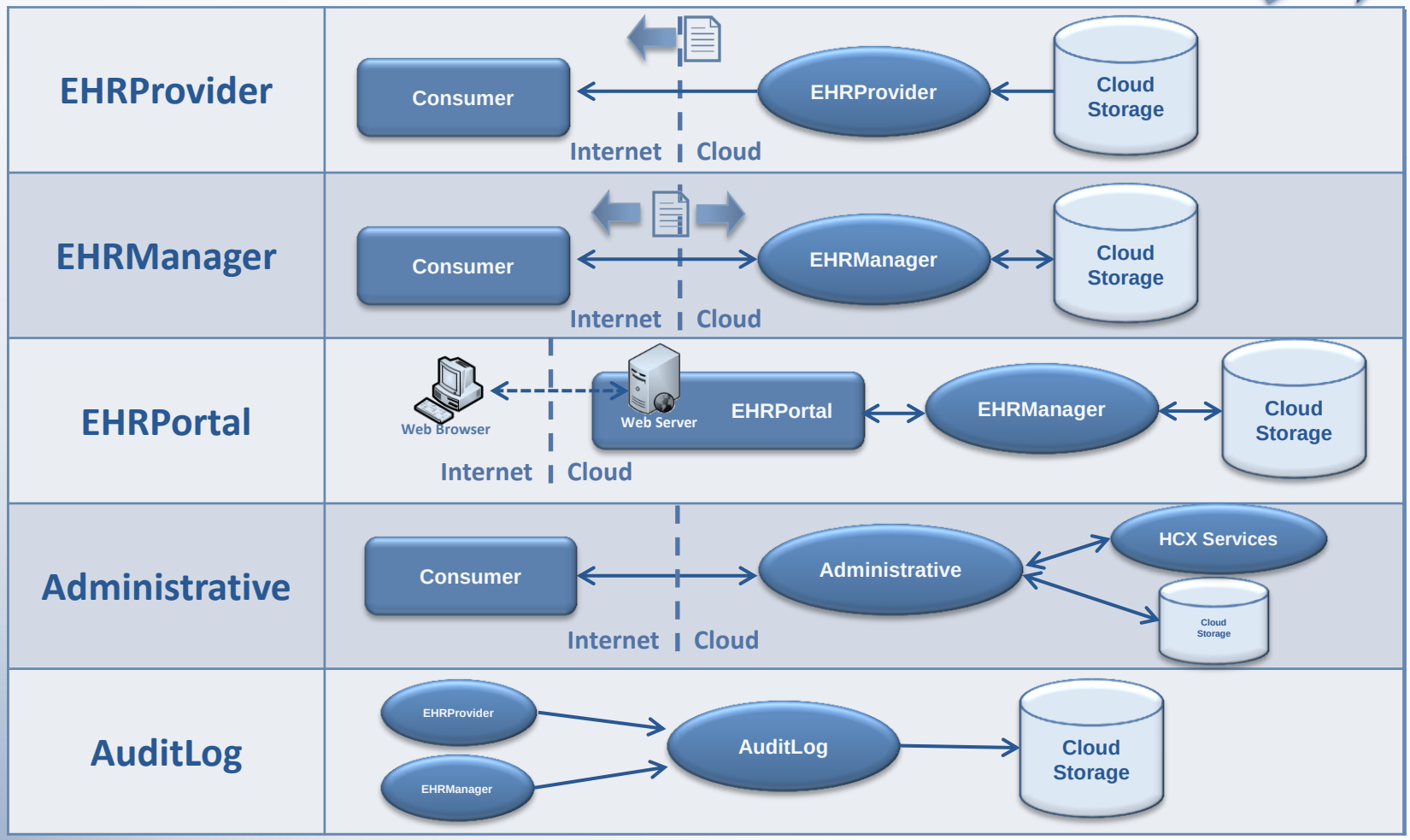
- Setup:
  - Linux OS
  - Apache ZooKeeper installed
- Will Run:
  - ZooKeeper Server
- Note:
  - Premade ZooKeeper images for EC2

# Adopting DOSGi for the Cloud

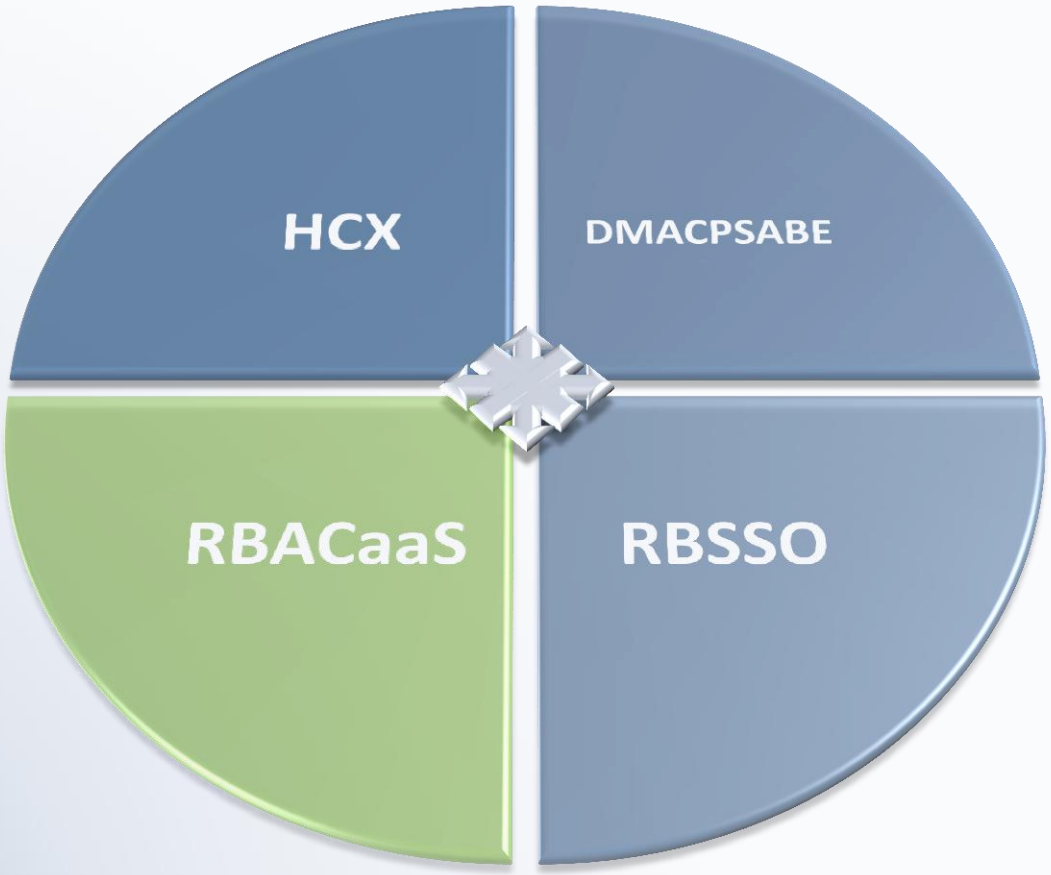


# The HCX Architecture

## HCX Services:



# Role Based Access Control as a Service (RBACaaS)



# Access Control for the Cloud

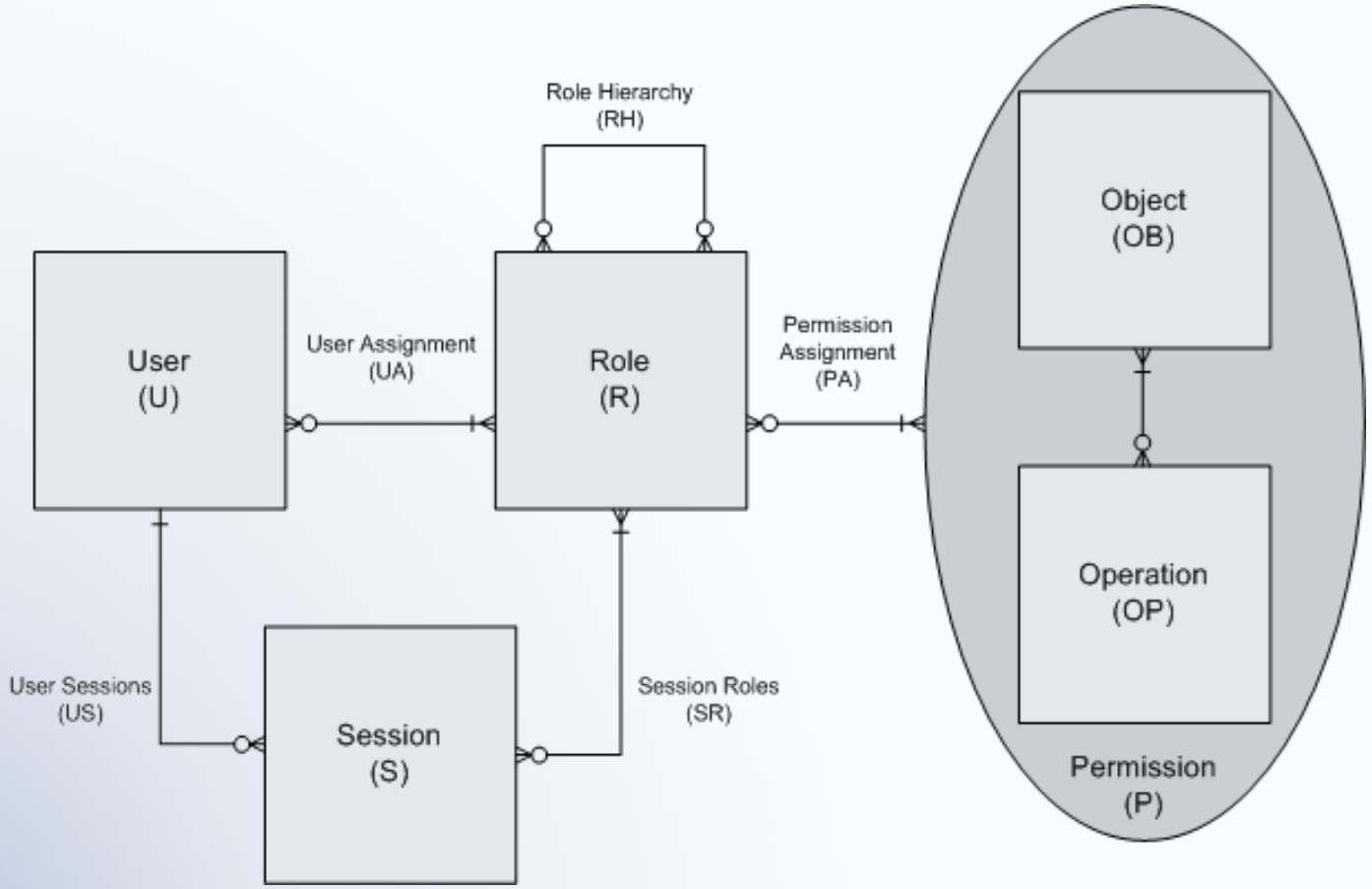
## Qualities of Cloud Access Control:

- Scalable
- Distributed
- Confidential
- Simple Administration
- Reliable

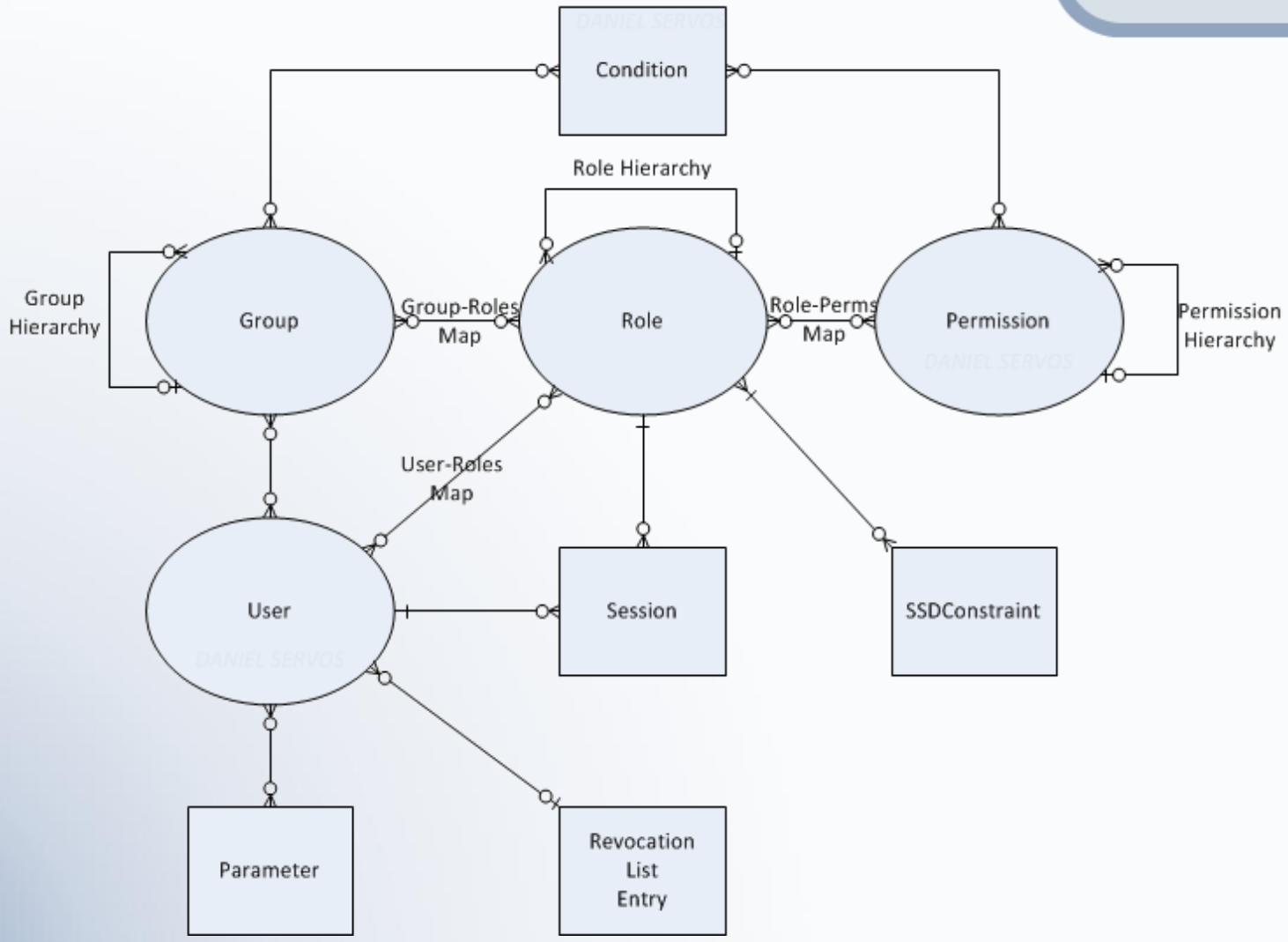
## Traditional Access Control:

- Discretionary access control (DAC)
- Mandatory access control (MAC)

# Role Based Access Control



# A New Take on RBAC (RBAC as a Service)

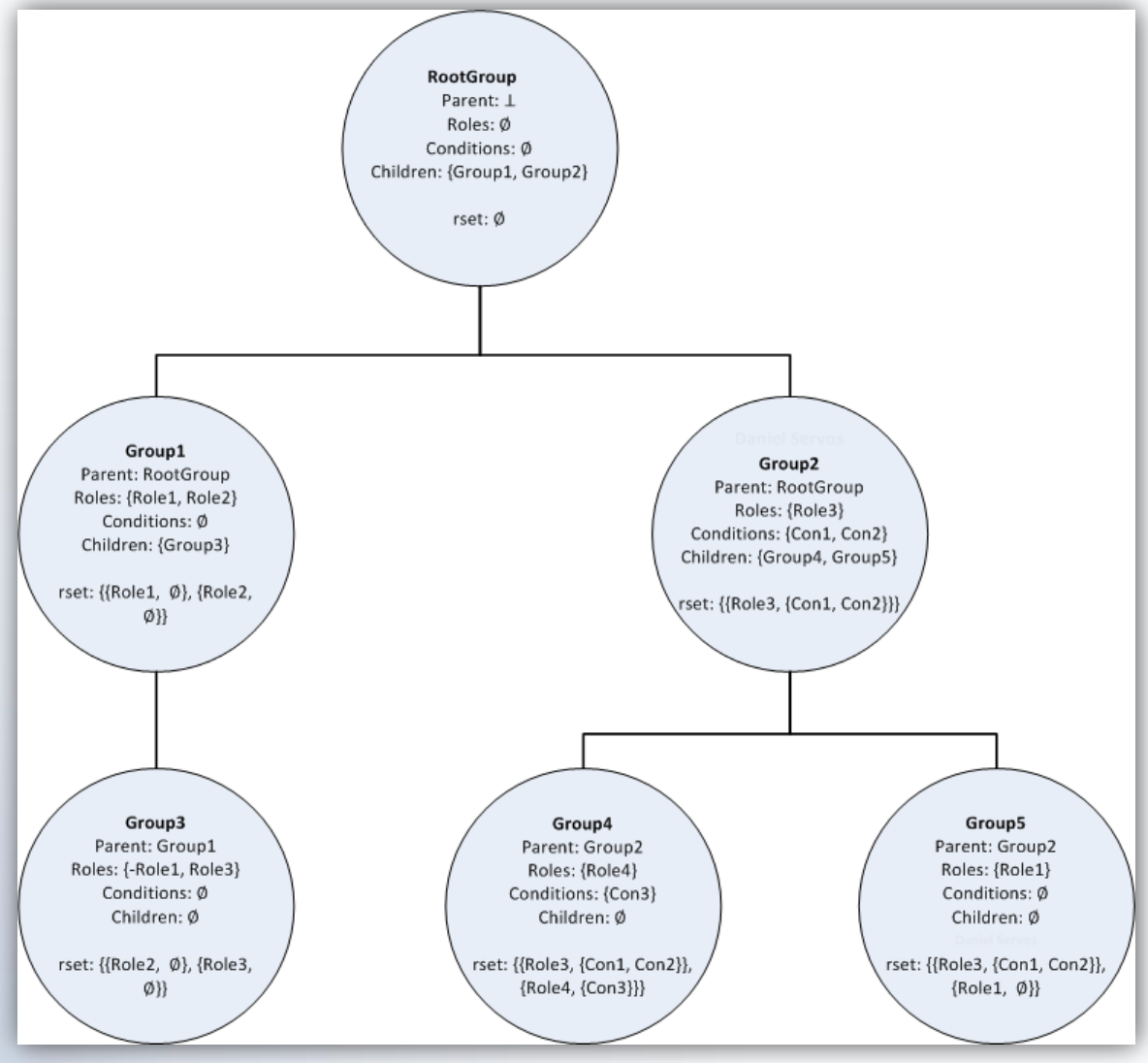


# A New Take on RBAC (RBAC as a Service)

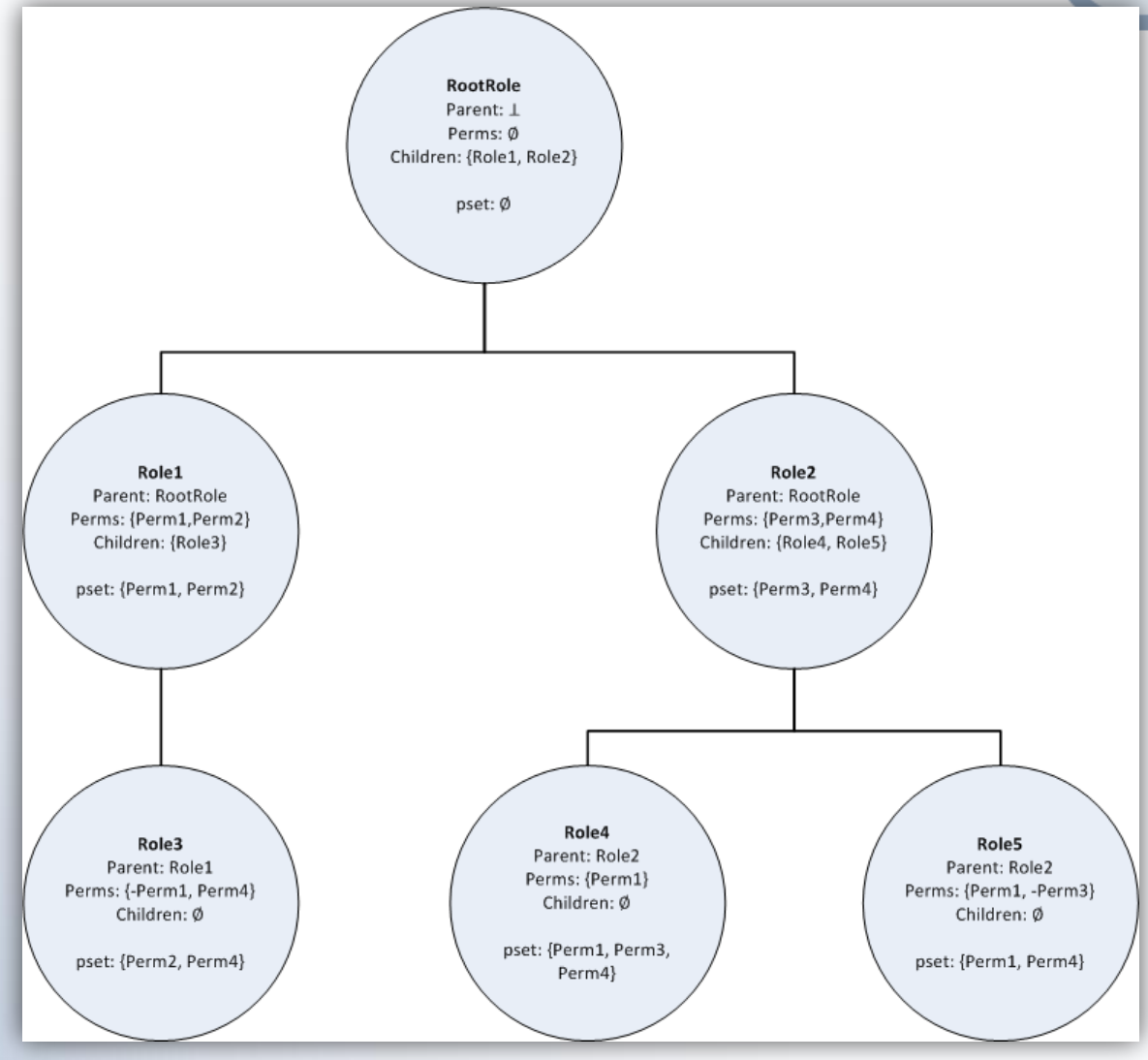
```
rbac_uri                = “RBAC:” element_types “:” id
                        / “RBAC:perm:” perm_id

element_types           = “user”
                        / “group”
                        / “role”
                        / “cond”
                        / “const”
                        / “param”

id                      = domain “:” sid

perm_id                 = domain “:” perm_sid

domain                  = (ALPHA / DIGIT) *( ALPHA / DIGIT / “-“ / “.”) [“_” port]

port                    = ( 1-9 ) *( DIGIT )

sid                     = +( ALPHA / DIGIT / “-“ / “.” / “*” / “_”)

perm_sid                = “*” / ( +( ALPHA / DIGIT / “-“ / “_”) “.” perm_id )
```

```
RBAC:user:clutch.lakeheadu.ca_1337:daniel.servos
RBAC:role:localhost:doctor
RBAC:perm:clutch:EHRManager.view.*
```

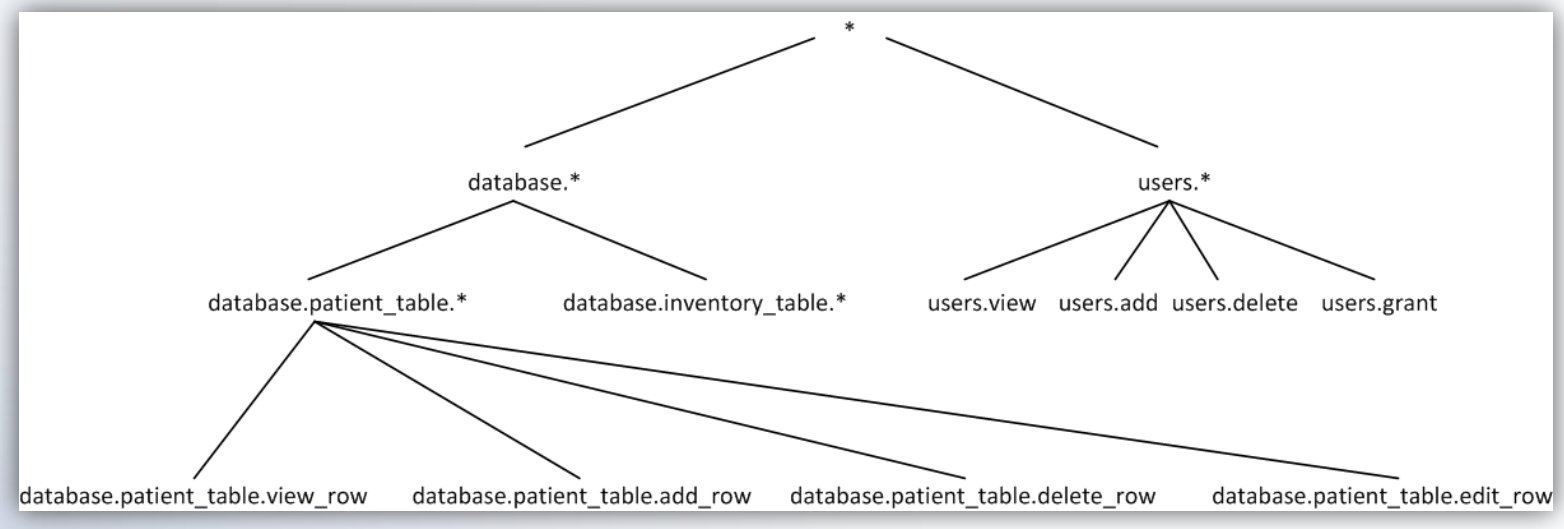
# A New Take on RBAC (RBAC as a Service)



# A New Take on RBAC (RBAC as a Service)



# A New Take on RBAC (RBAC as a Service)



# A New Take on RBAC (RBAC as a Service)

```
condition    =  exp [ bool_op condition ]

exp          =  var op var
             /  [“!”] bool_var
             /  [“!”] “(“ condition “)”

var          =  const
             /  user_param
             /  system_param

bool_var     = boolean
             / user_param
             / system_param

op           =  “>” / “<” / “=” / “>=” / “<=” / “!=”

boolvar      = “AND” / “OR”

user_param   =  id

system_param=  “SYSTEM:” sid

const        =  int
             /  float
             /  string

int          =   [“-”] ( 1-9 ) *( DIGIT )
             /   “0”

float        =   int “.” +( DIGIT )

string       =  “\”” *( ALPHA / DIGIT / “-“ / “.” / “*” / “_” / “:”) “\””

boolean      =   “TRUE” / “FALSE”
```

# A New Take on RBAC (RBAC as a Service)

HOSPITAL_DOMAIN:WHMIS_SAFETY AND HOSPITAL_DOMAIN:AGE >= 18

SYSTEM:TIME_HOUR >= 9 AND SYSTEM:TIME_HOUR <= 17

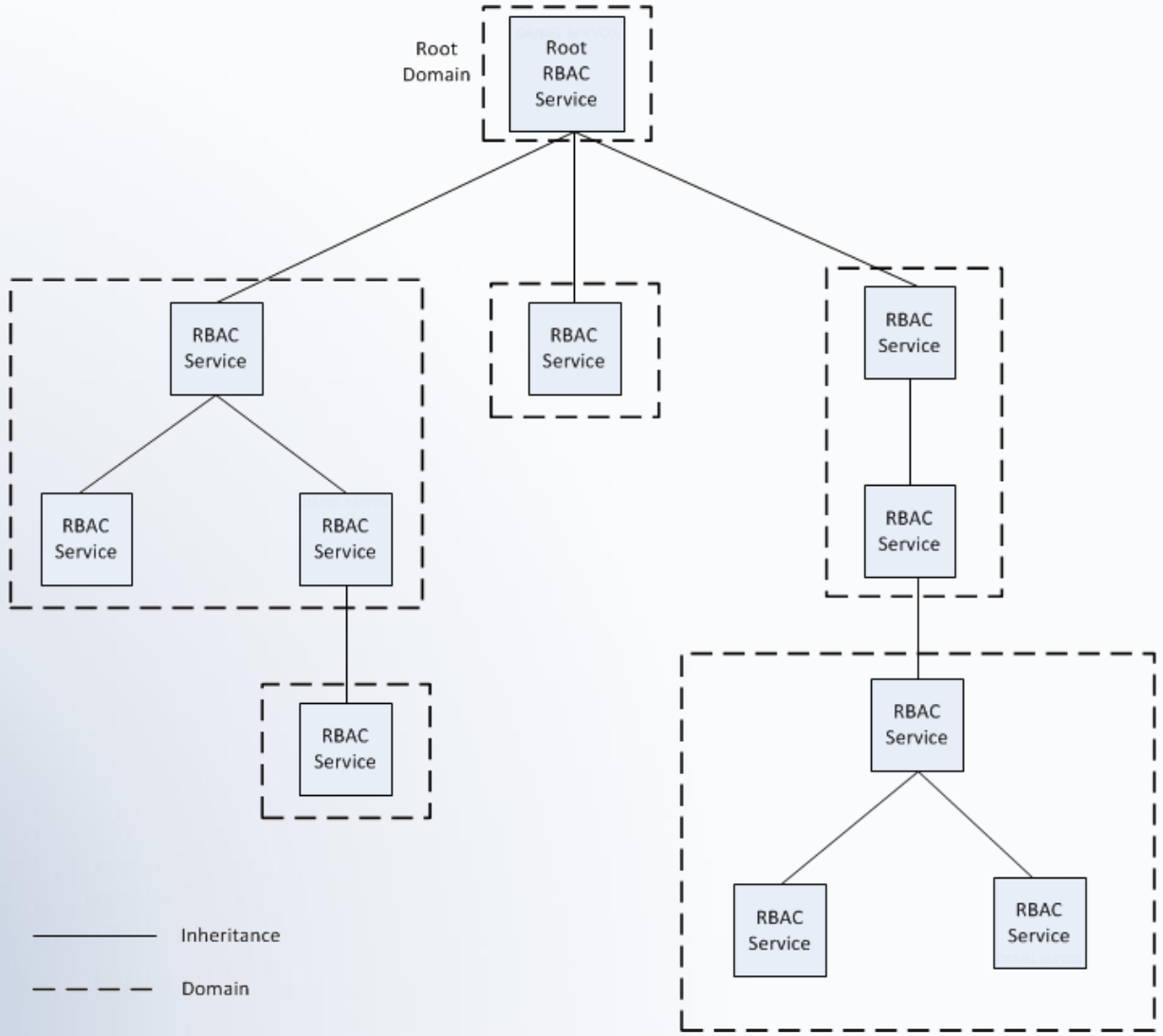
SYSTEM:USER_IP_1 == 192 AND SYSTEM:USER_IP_2 == 168 AND (SYSTEM:USER_IP_3 == 100 OR SYSTEM:USER_IP_3 == 110)

UNIVERSITY_DOMAIN:STUDENT_YEAR >= 4 AND UNIVERSITY_DOMAIN:DEPARTMENT = “COMP”

| Parameter Name | Type | Perm | Description |
|---|---|---|---|
| SYSTEM:TIME_STAMP | Integer | ✓ | The current date and time as a Unix time stamp. |
| SYSTEM:TIME_DAY | Integer | ✓ | A number [1, 31] representing the current day in the current month. Based on gregorian calendar and UTC. |
| SYSTEM:TIME_HOUR | Integer | ✓ | A number [0, 23] representing the current hour in UTC. |
| SYSTEM:TIME_MINUTE | Integer | ✓ | A number [0, 59] representing the current minute in UTC. |
| SYSTEM:TIME_SECOND | Integer | ✓ | A number [0, 59] representing the current second in UTC. |
| SYSTEM:TIME_WEEK_DAY | Integer | ✓ | The current week day represented by a number starting at 0 for Sunday and ending at 6 for Saturday. Based on UTC. |
| SYSTEM:TIME_MONTH | Integer | ✓ | A number [1, 12] representing the current UTC gregorian calendar month |
| SYSTEM:TIME_YEAR | Integer | ✓ | A number representing the current gregorian calendar year in UTC. |
| SYSTEM:USER_IP | Integer | ✓ | An integer representation of the user's version 4 IP at the time they authenticated with the server. |
| SYSTEM:USER_IP_1 | Integer | ✓ | An integer representation of the first byte of a user's version 4 IP at the time they authenticated with the server. |
| SYSTEM:USER_IP_2 | Integer | ✓ | An integer representation of the second byte of a user's version 4 IP at the time they authenticated with the server. |
| SYSTEM:USER_IP_3 | Integer | ✓ | An integer representation of the third byte of a user's version 4 IP at the time they authenticated with the server. |
| SYSTEM:USER_IP_4 | Integer | ✓ | An integer representation of the fourth byte of a user's version 4 IP at the time they authenticated with the server. |
| SYSTEM:USER_HOST | String | | A string containing the user's hostname at the time they authenticated with the server. |
| SYSTEM:USER_HOST_DOMAIN | String | | A string containing the domain part of a user's hostname at the time they authenticated with the server. |
| SYSTEM:USER_DOMAIN | String | | A string containing the server's RBACaaS domain name. |
| SYSTEM:USER_DOMAIN_ID | Integer | ✓ | The ID assigned to the server's RBACaaS domain. |
| SYSTEM:USER_ID | String | | A string containing the user's RBACaaS ID. |
| SYSTEM:USER_SID | String | | A string containing the user's RBACaaS SID. |
| SYSTEM:USER_GID | Integer | ✓ | The user's RBACaaS GID. |
| SYSTEM:USER_START_DATE | Integer | ✓ | A unix time stamp containing the date the user's account was activated. |
| SYSTEM:USER_END_DATE | Integer | ✓ | A unix time stamp containing the date the user's account will be or was deactivated or "0" if no such date is set. |
| SYSTEM:SESSION_START | Integer | ✓ | A unix time stamp containing the date and time the user's session was started. |
| SYSTEM:SESSION_EXPIRE | Integer | ✓ | A unix time stamp containing the date and time the user's session will expire. |
| SYSTEM:CLIENT_VERSION | Integer | ✓ | An integer representation of the version number of the client software the user used to authenticate with the server. |
| SYSTEM:SERVER_VERSION | Integer | ✓ | An integer representation of the version number of the server software being used. |
| SYSTEM:AUTH_METHOD | Integer | ✓ | An integer representing the authentication method used to authorize the user. |

# Distributed Function and Scalability

## Distributed Function:



# Distributed Function and Scalability

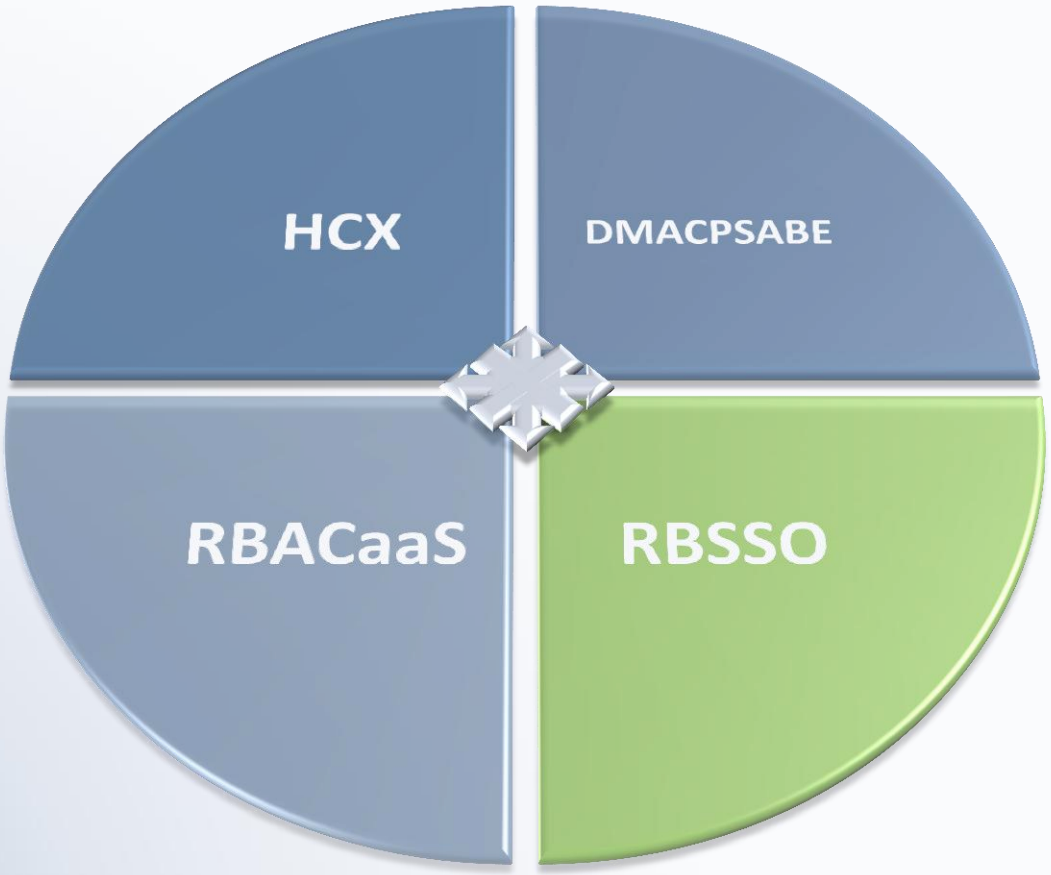
## Scalability:

- Permissions Set and Role List Caching
- Load Balancing
- Sharding
- Limited Contact/Traffic

## Authentication:

- Each domain runs it's own AuthServer
- Returns set of permissions for active role
- Returns parameter/value map for user
- Confirms if user may use active role

# Role Based Single Sign-On (RBSSO)



# Role Based Single Sign On

## Design Goals:

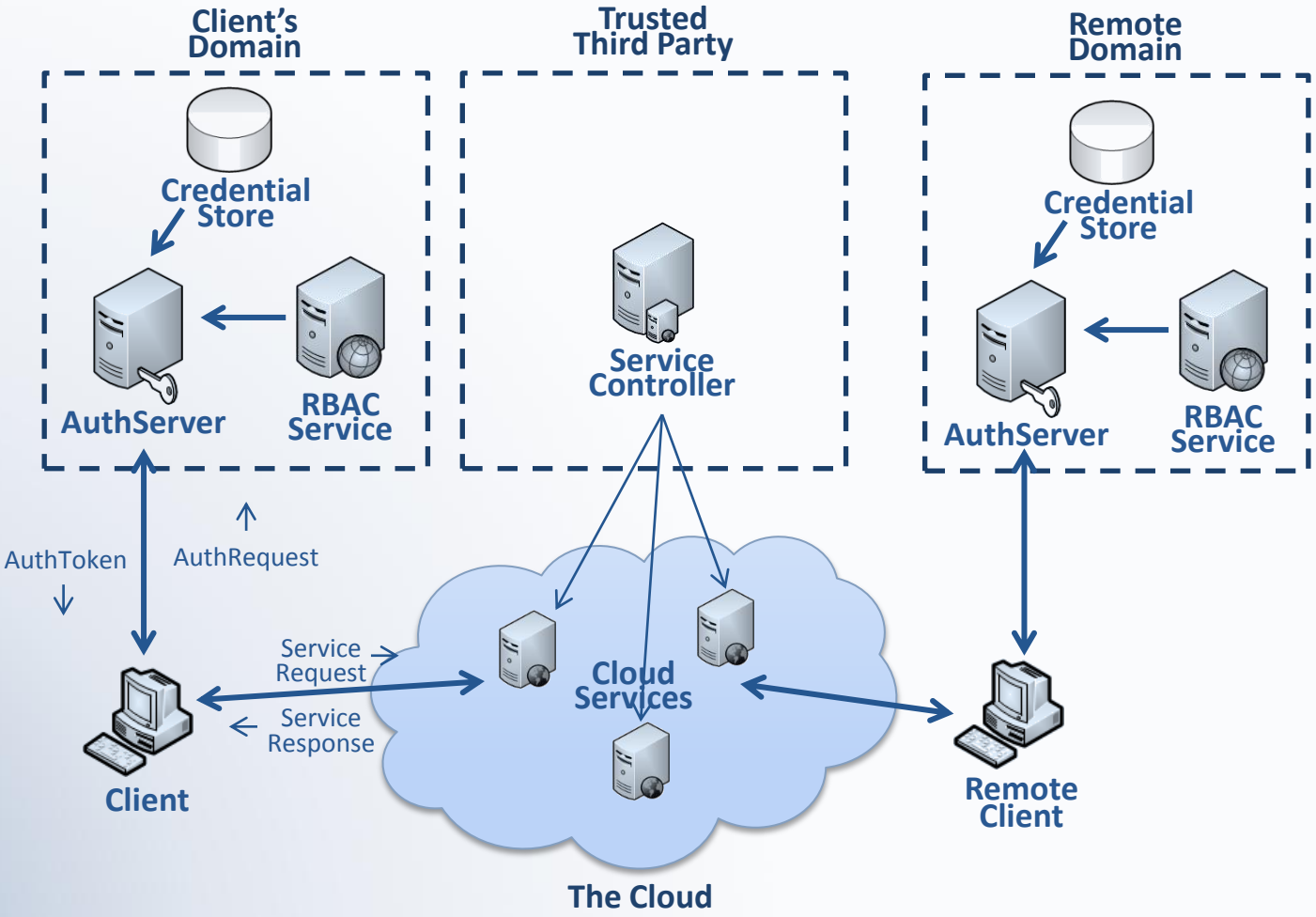
- Distributed
- Isolated
- Scalable
- Secure
- Convenient
- Compatibility

# Role Based Single Sign On

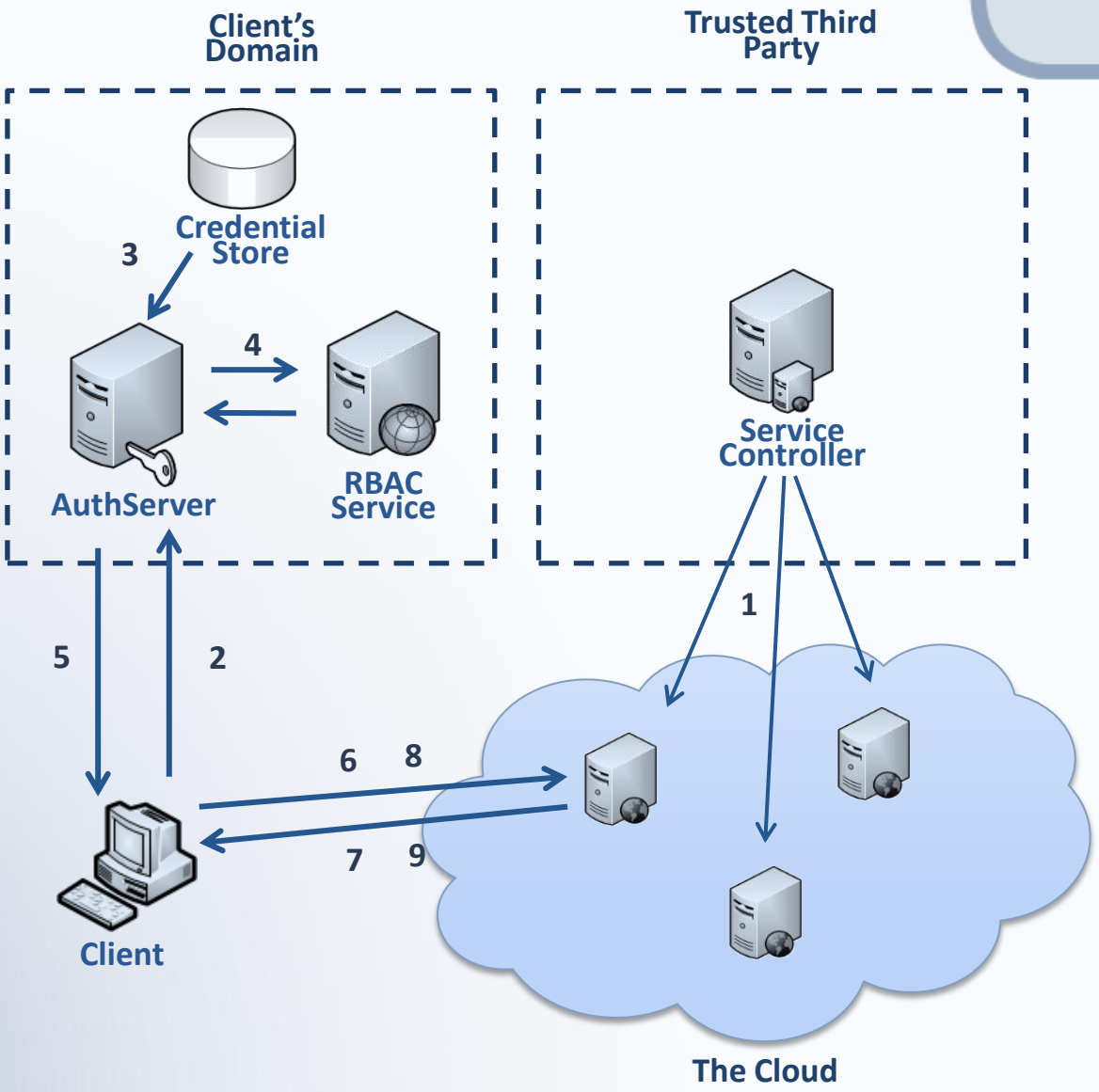
## Major Components:

- AuthServers
- RBAC Service (RBACaaS)
- Cloud Based Services (HCX)
- Service Controllers
- End User Clients

# Role Based Single Sign On



# Protocol


Client's Domain
Trusted Third Party
Credential Store
3
4
AuthServer
RBAC Service
Service Controller
1
5
2
6
8
7
9
Client
The Cloud

# Protocol

## ServiceToken:

| | Byte | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 |
|---|---|---|---|---|---|---|---|---|---|
| Header (5 bytes) | 1..5 | Version | Message Length | | | | | | |
| Body | 6 .. 13 | Time Stamp | | | | | | | |
| | 14 .. 17 | Service List Length (sl) | | | | | | | |
| | 18.. sl+17 | Service List | | | | | | | |
| | sl+18 .. sl+21 | SKpub Length (skp) | | | | | | | |
| | sl+22 .. sl+21+skp | ANS.1 Encoded SKpub | | | | | | | |
| | sl+22+skp .. sl+25+skp | Text Length (tx) | | | | | | | |
| | sl+26+skp .. sl+25+skp+tx | String(Public IP or Host + Delimiter + Instance ID + Delimiter + Service Controller ID + Delimiter + Service ID) | | | | | | | |
| Tail | sl+26+skp+tx .. end of token | Message Signature<br>Signature(SCpri, Version + Body) | | | | | | | |

# Protocol

## AuthRequest:

| Byte | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 |
|---|---|---|---|---|---|---|---|---|---|
| **Header** (9 bytes) | | | | | | | | | |
| 1..9 | Version | Message Length | | | | Body Length | | | |
| **Body** (Encrypted with CKsec) | | | | | | | | | |
| 1..8 | Time Stamp | | | | | | | | |
| 9..16 | Request ID | | | | | | | | |
| 17..20 | CKpub Length (ckp) | | | | | | | | |
| 21 .. ckp+20 | ANS.1 Encoded CKpub | | | | | | | | |
| ckp+21 .. ckp+24 | Text Length (tx) | | | | | | | | |
| ckp+25 .. ckp+24+tx | String(AuthServer ID + Delimiter + User ID + Delimiter + Role ID) | | | | | | | | |
| ckp+25+tx .. end of body | User Credentials | | | | | | | | |
| **Tail** (Encrypted with AKencpub) | | | | | | | | | |
| 1 .. 4 | CKsec Length (cks) | | | | | | | | |
| 5 .. cks + 4 | ANS.1 Encoded CKsec | | | | | | | | |
| cks + 5 .. end of tail | Message Hash<br>Hash(Version + Body + CKsec Length + CKsec) | | | | | | | | |

# Protocol

Encrypted with CKsec

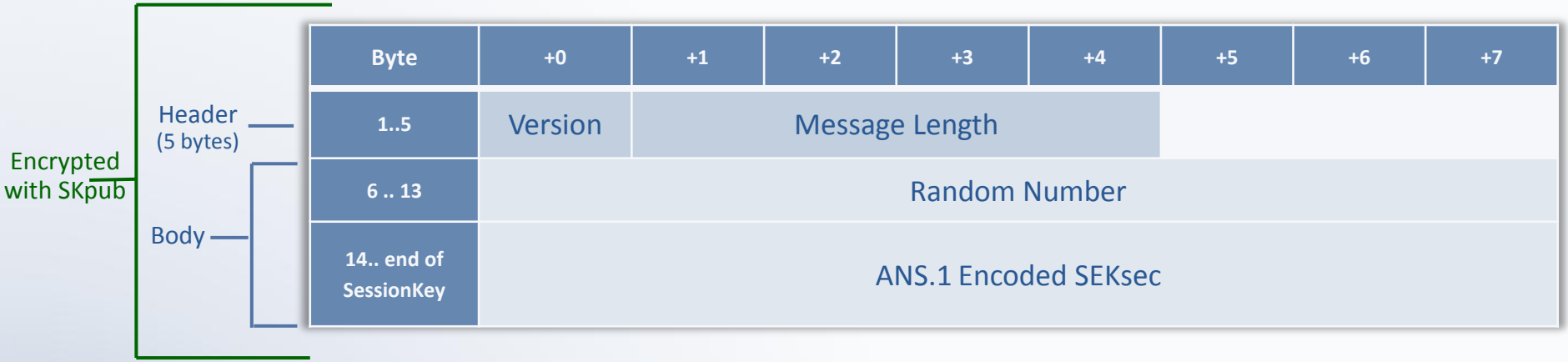
## AuthToken:

| | Byte | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 |
|---|---|---|---|---|---|---|---|---|---|
| Header (5 bytes) | 1..5 | Version | Message Length | | | | | | |
| Body | 6..13 | Time Stamp | | | | | | | |
| | 14..21 | Expiry Date | | | | | | | |
| | 22.. 29 | Session ID | | | | | | | |
| | 30..33 | CKpub Length (ckp) | | | | | | | |
| | 34 .. ckp+33 | ANS.1 Encoded CKpub | | | | | | | |
| | ckp+34 .. ckp+37 | Text Length (tx) | | | | | | | |
| | ckp+38 .. ckp+37+tx | String(AuthServer ID + delim + AuthDomain + delim + User ID + delim + Role ID + delim + Permission/Condition Pair Set + delim + Parameter/Value set) | | | | | | | |
| Tail | ckp+38+tx .. end of token | Message Signature<br>Signature(AKsigpri, Version + Body) | | | | | | | |

| | | |
|---|---|---|
| Key Attachment (Not part of token) | 1..4 | DMACPSABE Key Length (kx) |
| | 5 .. kx+6 | DMACPSABE User Key |

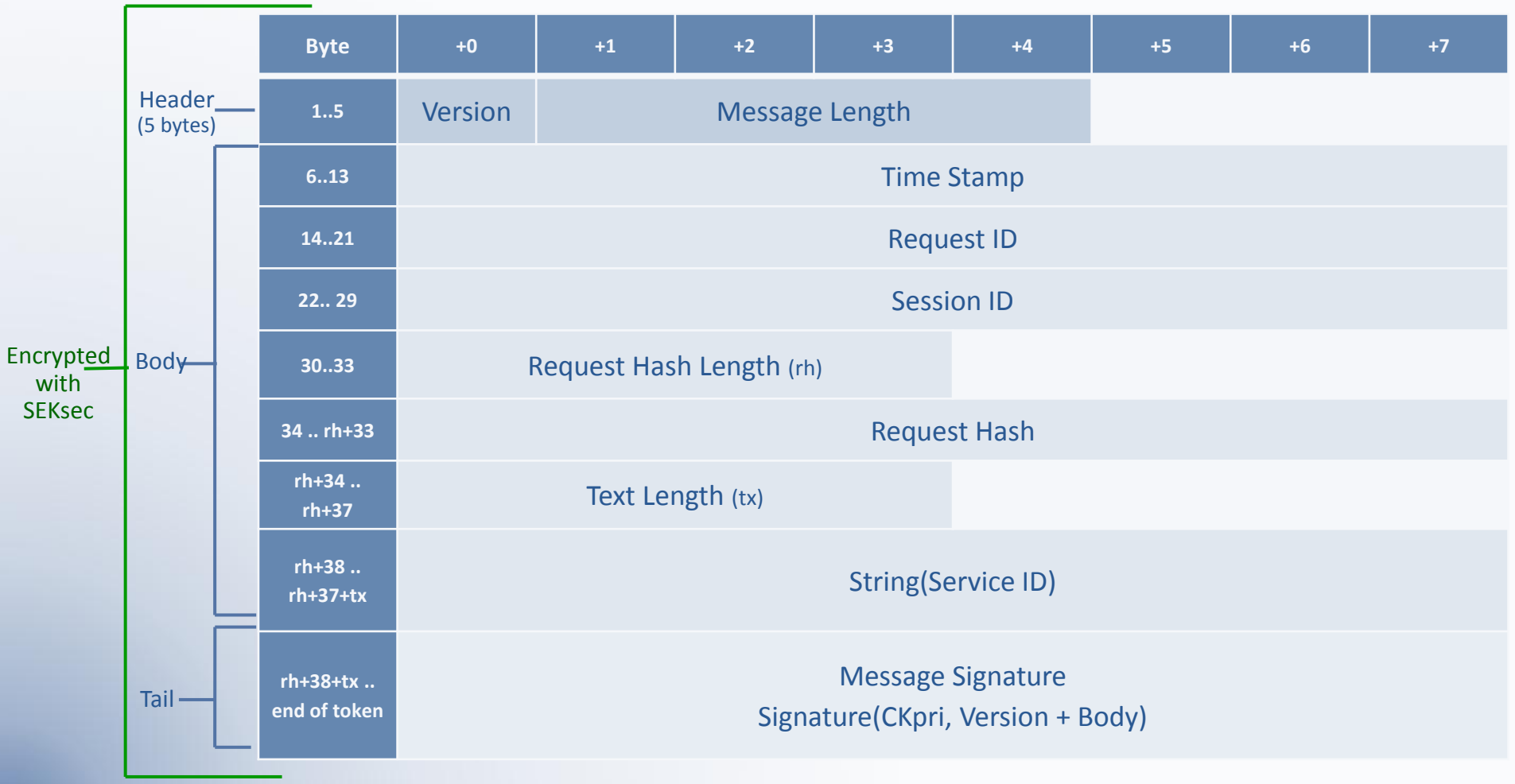
# Protocol

## SessionKey:



# Protocol

## RequestToken:



# Performance Evaluation

## Implementation and Evaluation:

- Implementation of AuthSrever and Client created using Java TCP sockets.
- Authentication performance evaluated against a SSL connection and Kerberos.
- Performance measured in average time per request on low latency local network and higher latency, nosier wide area network.
- Each protocol was tested with 10,000 authentication requests for each network.

CLUTCH is Lakehead University's Testbed for Cloud Health (CLUTCH)

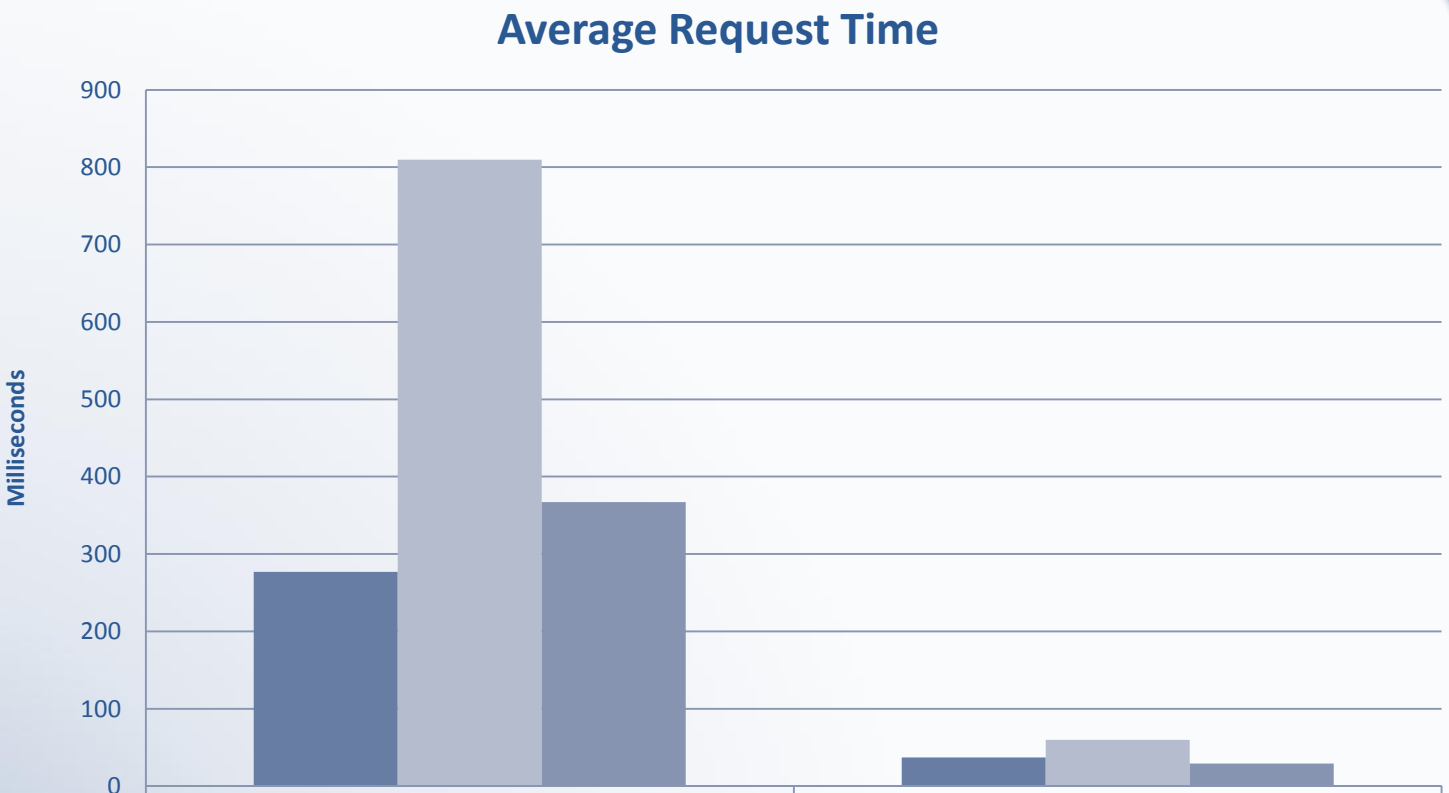
# Performance Evaluation


Based on 1,000 requests per run per protocol.

# Performance Evaluation


Based on 1,000 requests per run per protocol.

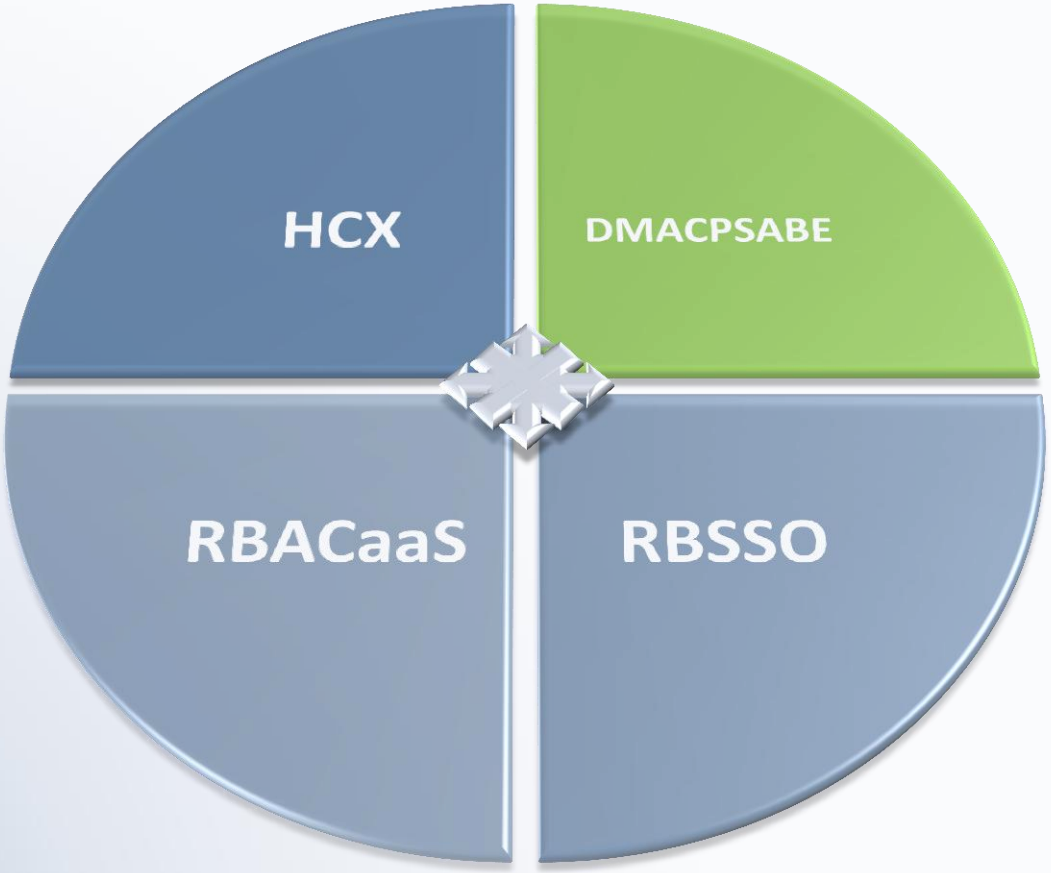
# Performance Evaluation




| | WAN | LAN |
|---|---|---|
| RBSSO | 276.7162 | 36.8437 |
| SSL | 809.5823 | 59.7964 |
| Kerberos | 367.0858 | 29.0673 |

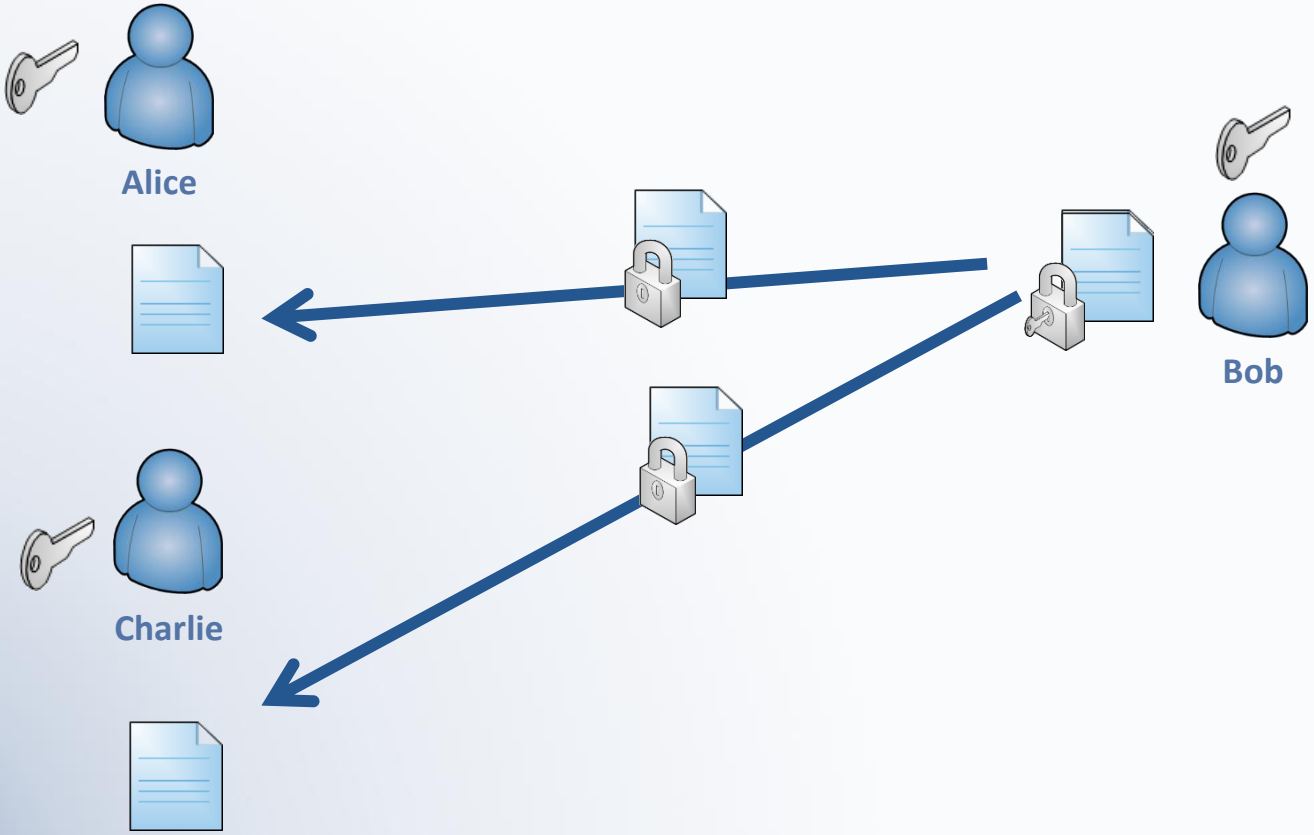
Based on 1,000 requests per run per protocol.

# Distributed Multi-Authority Ciphertext-Policy Shared Attribute-Based Encryption

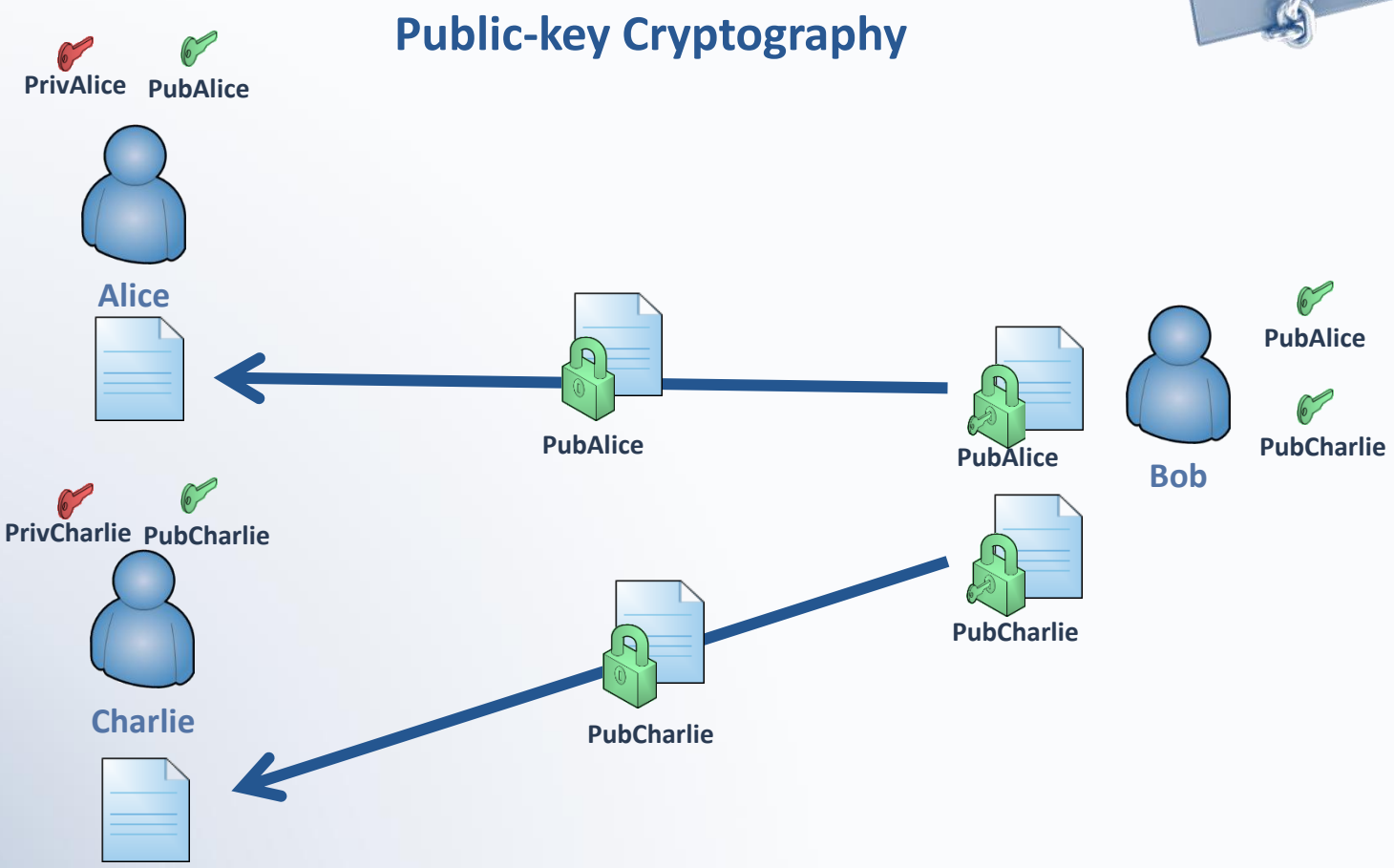


# Cloud Privacy Through Attribute Based Encryption

## Symmetric Encryption

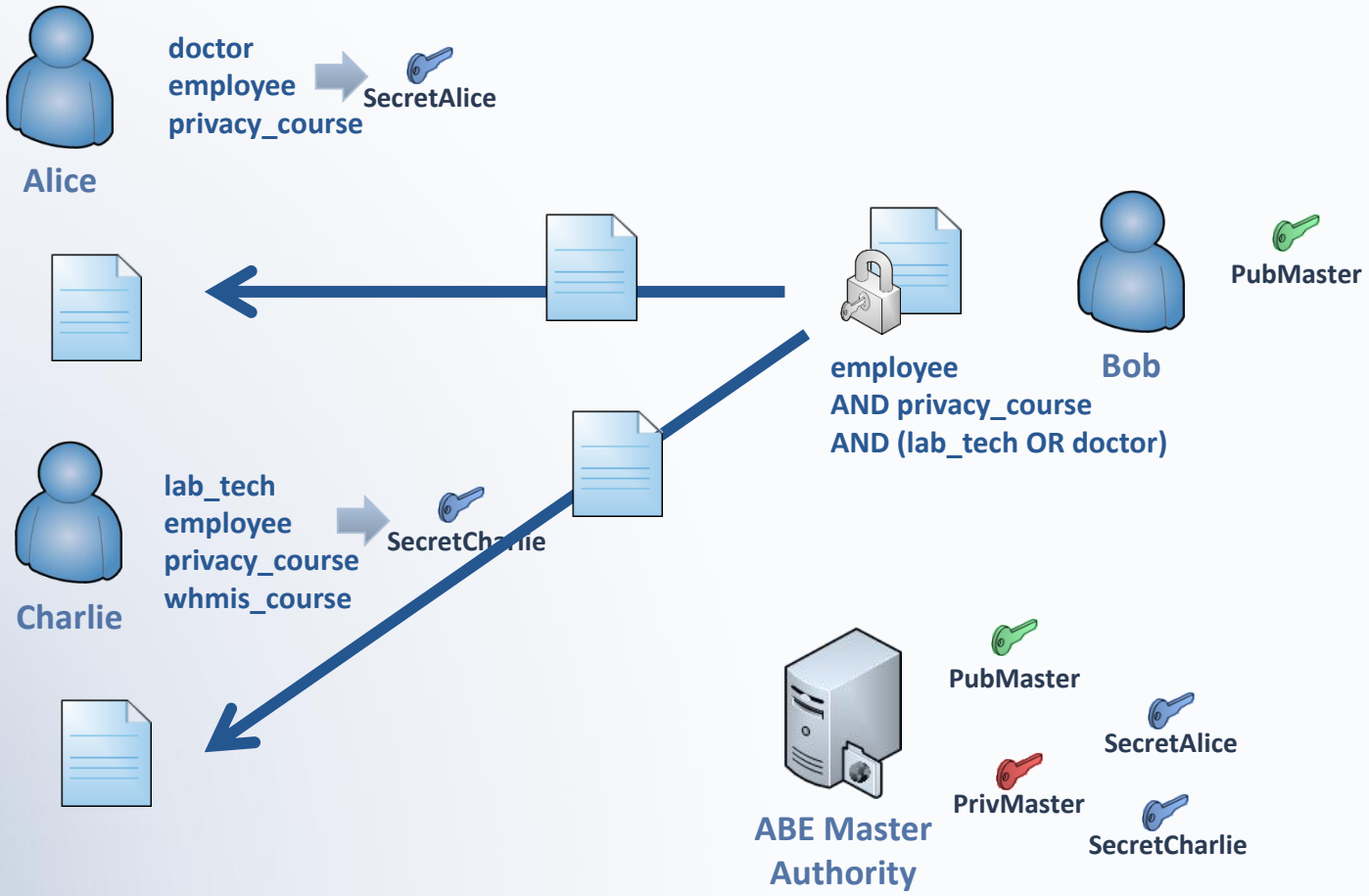


# Cloud Privacy Through Attribute Based Encryption



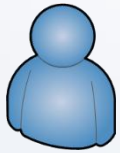
# Cloud Privacy Through Attribute Based Encryption

## Attribute Based Encryption



# Cloud Privacy Through Attribute Based Encryption

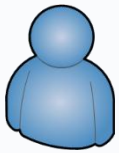
## Ciphertext Policy Attribute Based Encryption



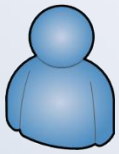
Alice

age = 32
employee_type = 103
privacy_course
year_level = 6

year_level >= 2 AND age > 17
AND (employee_type = 102 OR employee_type = 3)
AND privacy_course



Bob



Charlie

age = 25
employee_type = 3
privacy_course
whmis_course
year_level = 2

ABE Master Authority

# Pairing-Based Cryptography

**Definition 1:** Bilinear Map

A bilinear map from the cyclic groups of the same order $G_1 \times G_2$ to a cyclic group of the same order $G_t$ is the function:

$$e: G_1 \times G_2 \rightarrow G_t$$

Such that:

$$\forall u \in G_1, \forall v \in G_2, \forall a, b \in \mathbb{Z}: e(u^a, v^b) = e(u, v)^{ab}$$

**Definition 2:** Admissible Bilinear Map

A bilinear map, $e$, is considered to be admissible if for two generators $g_1$ and $g_2$ of groups $G_1$ and $G_2$:

$$G_1 \times G_2 \rightarrow G_t \text{ and } e(g_1, g_2) = G_t$$

and $e$ is efficiently computable.

**Definition 3:** Symmetric Pairing

A pairing of two groups $G_1$ and $G_2$ is considered to be symmetric if:

$$G_1 = G_2 = G$$

such that:

$$G \times G = G_t$$

# Pairing-Based Cryptography

## Cytological Problems:

- **Bilinear Diffie-Hellman problem:** Given $g$, $g^a$, $g^b$, $g^c$ compute $e(g,g)^{abc}$
- **Gap Diffie-Hellman problem:** Solve CDH in $G$.
- **k-Bilinear Diffie-Hellman Inversion problem:** Given $g, g^y, g^{y^2}, \dots, g^{y^k}$, compute $e(g,g)^{\frac{1}{y}}$.
- **k-Decisional Bilinear Diffie-Hellman Inversion problem:** Distinguish $g, g^y, g^{y^2}, \dots, g^{y^k}, e(g,g)^{\frac{1}{y}}$ from $g, g^y, g^{y^2}, \dots, g^{y^k}, e(g,g)^z$.

# Fuzzy Identity-Based Encryption

$$\Delta i, S(x) = \prod_{j \in S,\ j \neq i} \frac{x - j}{i - j}$$

$PubKey, PrivKey = Setup(n, d)$:
choose: $g_1 = g^y$ for some $y$
choose: $g_2 \in G$
choose randomly: $t_1, \dots, t_{n+1} \in G$
$PubKey = (n, d, g_1, g_2, t_1, \dots, t_{n+1})$
$PrivKey = y$

**Equation 1:** FIBE Setup function

$SK = KeyGeneration(PubKey, PrivKey, ID)$:
choose randomly: A $d-1$degree polynomial $q$ where $q(0) = y$
$FOR\ i \in ID$:

$$D_i = g_2^{q(i)} \cdot \left( g_2^{i^n} \prod_{k=1}^{n+1} t_k^{\Delta k, N(i)} \right) \quad where\ N\ is\ the\ set\ \{1, \dots, n+1\}$$

$r_i = random\ number\ in\ \mathbb{Z}_p$
$d_i = g^{r_i}$
$SK = (D, d)$

**Equation 2:** FIBE KeyGeneration function

$C = Encryption(PubKey, w', M \in G_t)$:
choose randomly: $s \in \mathbb{Z}_p$
$E' = M \cdot e(g_1, g_2)^s$
$E'' = g^s$
$FOR\ i \in w'$:

$$E_i = \left( g_2^{i^n} \prod_{k=1}^{n+1} t_k^{\Delta k, N(i)} \right)^s$$

$C = (w', E', E'', E)$

**Equation 3:** FIBE Encryption function

$M = Decryption(PubKey, SK, C)$:
choose: an arbitrary $d$ element subset $S$ of $w \cap w'$

$$M = E' \prod_{i \in S} \left( \frac{e(d_i, E_i)}{e(D_i, E'')} \right)^{\Delta i, S(0)}$$

**Equation 4:** FIBE Decryption function

# Ciphertext Policy Attribute Based Encryption

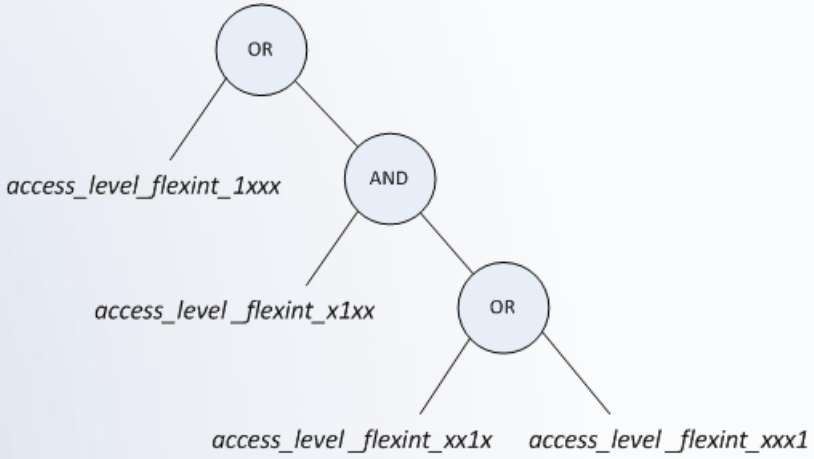
**access_level = 5**

access_level_flexint_0xxx
access_level_flexint_x1xx
access_level_flexint_xx0x
access_level_flexint_xxx1

**> < >= <= =**

**access_level >= 5**

access_level _flexint_1xxx OR (access_level _flexint_x1xx AND (access_level _flexint_xx1x OR access_level _flexint_xxx1))



# Ciphertext Policy Attribute Based Encryption

$PK, MK = SETUP()$:

choose: $G_0$ of prime order $p$ with generator $g$

choose randomly: $\alpha, \beta \in \mathbb{Z}_p$

$$PK = \left(G_0, g, h = g^{\beta}, e(g,g)^{\alpha}, f = g^{\frac{1}{\beta}}\right)$$

$$MK = (\beta, g^{\alpha})$$

**Equation 9:** Setup Function

$CT = ENCRYPT(PK, M, \tau)$:

chose randomly: $s \in \mathbb{Z}_p$

$q = CreatePolynomials(\tau_r, s)$

$Y = \forall leaf\ nodes \in \tau$

$$CT = (\tau, \tilde{C} = Me(g,g)^{\alpha s}, C = h^s,$$
$$\forall y \in Y:\ C_y = g^{q_y(0)}, C'_y = H\left(att(y)\right)^{q_y(0)})$$

**Equation 10:** Encrypt Function

$q = CreatePolynomials(x, s)$:

"Starting with the root node [$\tau_r$] the algorithm sets $q_r(0) = s$. Then, it chooses $d_r$ other points of the polynomial $q_r$ randomly to define it completely. For any other node $x$, it sets $q_x(0) = q_{parent}(x)(index(x))$ and chooses $d_x$ other points randomly to completely define $q_x$."

$SK = KEYGEN(MK, S, PK)$:

choose randomly: $r \in \mathbb{Z}_p$

$D = g^{(\alpha + r)/\beta}$

$FOR\ \forall j \in S$:

choose randomly: $r_j \in \mathbb{Z}_p$

$D''_j = g^r \cdot H(j)^{r_j}$

$D'_j = g^{r_j}$

$SK = (D, D'', D')$

**Equation 11:** KeyGen Function

$M = DECRYPT(CT, SK, PK)$:

$A = DECRYPTNODE(CT, SK, PK, root(\tau))$

$IF\ A \neq \perp$:

$$M = \frac{\tilde{C}}{\frac{e(C,D)}{A}}$$

$ELSE$:

$M = \perp$

**Equation 12:** Decryption Function

# Ciphertext Policy Attribute Based Encryption

$A = DECRYPTNODE(CT, SK, PK, x)$:

*IF x is a leaf node:*

$i = att(x)$

*IF* $i \in S$:

$$A = \frac{e(D''_i, C_x)}{e(D'_i, C'_x)}$$

*ELSE*:

$A = \perp$

*ELSE*:

$\forall z$ *child of* $x$: $F_z = DECRYPTNODE(CT, SK, PK, z)$

$S_x = \forall z$ *child of* $x$ *and* $F_z \neq \perp$

*IF* $S_x = \emptyset$:

$A = \perp$

*ELSE*:

$$A = \prod_{z \in S_x} F_z^{\Delta_{i,S'_x}(0)} \quad where \; {}^{i=index(z)}_{S'_x=\{index(z):z\in S_x\}}$$

**Equation 13:** Recursive DecryptNode Function

$\widetilde{SK} = DELEGATE(SK, \tilde{S}, PK)$:

choose randomly: $\tilde{r} \in \mathbb{Z}_p$

$\tilde{D} = Df^{\tilde{r}}$

$FOR \; \forall k \in \tilde{S}$:

choose randomly: $\tilde{r}_k \in \mathbb{Z}_p$

$\tilde{D''}_k = D_k g^{\tilde{r}} H(k)^{\tilde{r}_k}$

$\tilde{D'}_k = D'_k g^{\tilde{r}_k}$

$\widetilde{SK} = (\tilde{D}, \tilde{D''}_k, \tilde{D'}_k)$

**Equation 14:** Delegate Function

# Ciphertext Policy Attribute Based Encryption

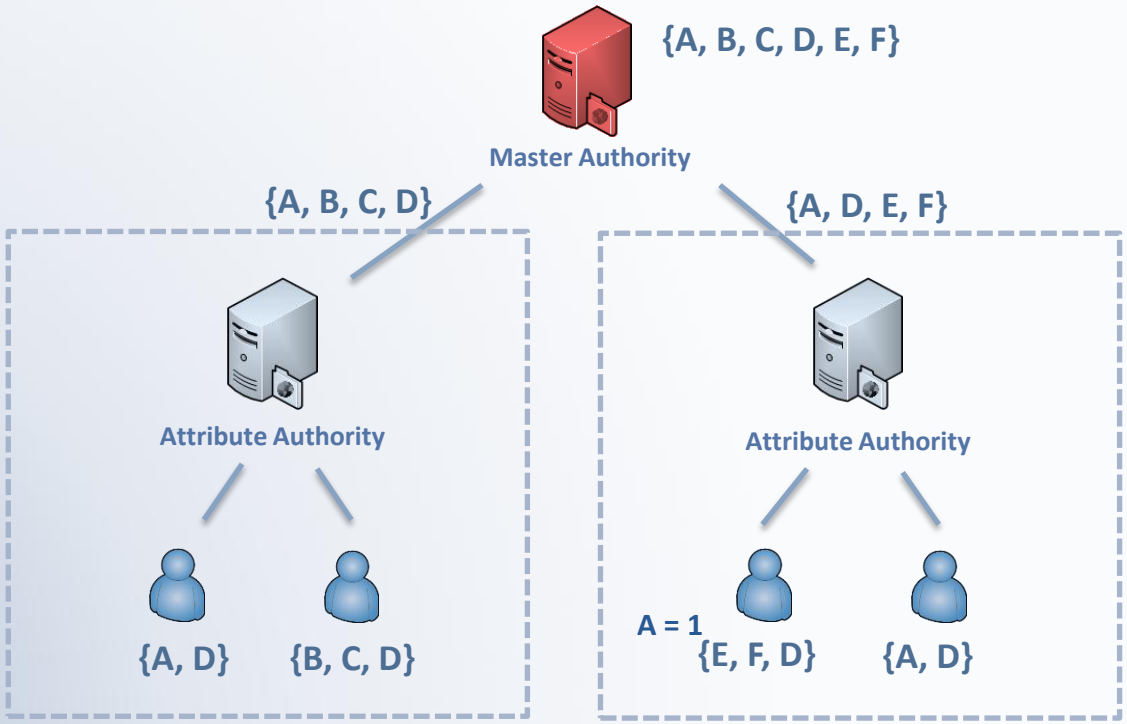
**Criticisms:**

- Single central authority
  - Trust issues
  - Bottleneck
- Expose information about document
- Missing not equals operation

# Distributed Multi-Authority Ciphertext-Policy Shared Attribute-Based Encryption

## DMACPSABE:

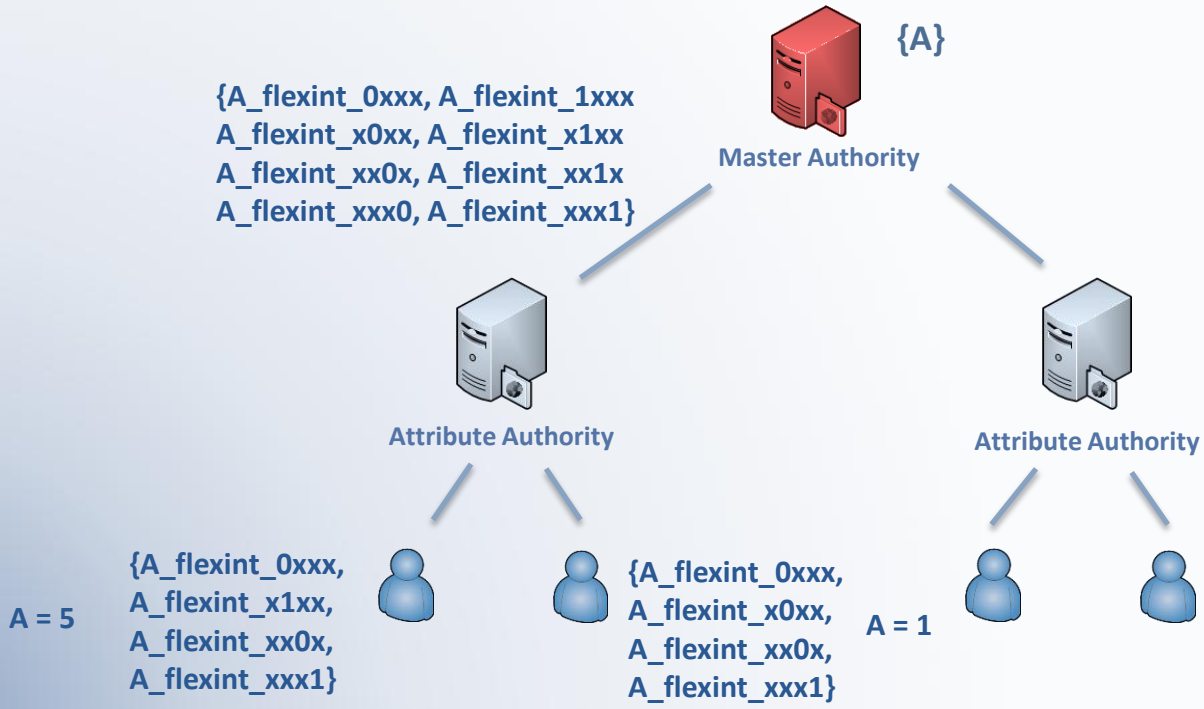
- Extends CP-ABE (Bethencourt & et. al. 2007)
- Each authority delegated subset of attributes
- Limited involvement of master authority



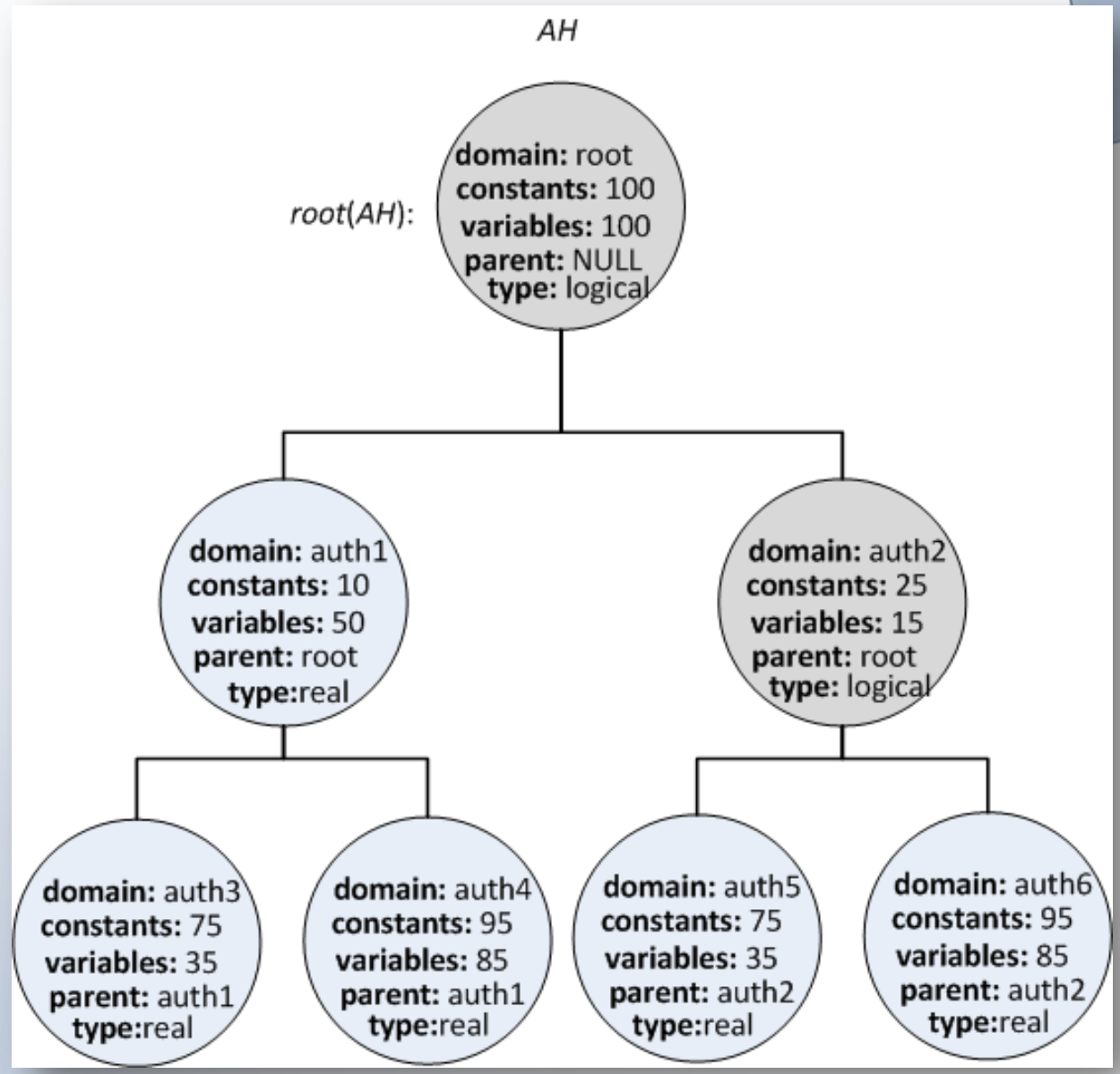
# Distributed Multi-Authority Ciphertext-Policy Shared Attribute-Based Encryption

## DMACPSABE:

- Extends CP-ABE (Bethencourt & et. al. 2007)
- Each authority delegated subset of attributes
- Limited involvement of master authority



# Distributed Multi-Authority Ciphertext-Policy Shared Attribute-Based Encryption



# Distributed Multi-Authority Ciphertext-Policy Shared Attribute-Based Encryption

## Constructions:

### Setup:

$$PK, MK, f, ASK = Setup(AH):$$
choose: $G_0$ *of prime order* $p$ *with generator* $g$
choose randomly: $\alpha, \beta \in \mathbb{Z}_p$
$PK = \left(G_0, g, h = g^{\beta}, e(g,g)^{\alpha}\right)$
$f = g^{\frac{1}{\beta}}$
$MK = (\beta, g^{\alpha})$
$AS = AuthAttSet(root(AH), \emptyset)$
$ASK = \forall S \in AS: \; ASK_s = KEYGEN(MK, S, PK)$

**Equation 15: DMACPSABE Setup Function**

$$AS = AuthAttSet(x, parentset):$$
*FOR* 1 .. $constants(x)$ *as* $i$:
$S_i = string(domain(x) + \text{"\_c"} + i)$

*FOR* 1 .. $(variables(x) * 2)$ *by* 2 *as* $k$:
$S_{constants(x)+k} = string(domain(x) + \text{"\_v"} + i + \text{" = 0"})$
$S_{constants(x)+k+1} = string(domain(x) + \text{"\_v"} + i + \text{" ="} + INT\_MAX)$

$S_{constants(x)+(variables(x)*2)+1} = string(\text{"auth\_key = "} + \text{auth\_index(x)})$
$S = ConvertAtts(S)$
$P = ConvertAtts(\{string(\text{auth\_key = } + \text{ auth\_index(parent(x))})\})$
$S = S \cup (parentset \setminus P)$

*IF* $type(x) = real$:
$AS = \{S\} \cup \forall z$ *child of* $x$: $AuthAttSet(z, S)$
*ELSE*:
$AS = \forall z$ *child of* $x$: $AuthAttSet(z, S)$

**Equation 16: Recursive DMACPSABE AuthAttSet Function**

# Distributed Multi-Authority Ciphertext-Policy Shared Attribute-Based Encryption

## Constructions:

UserKeyGen:

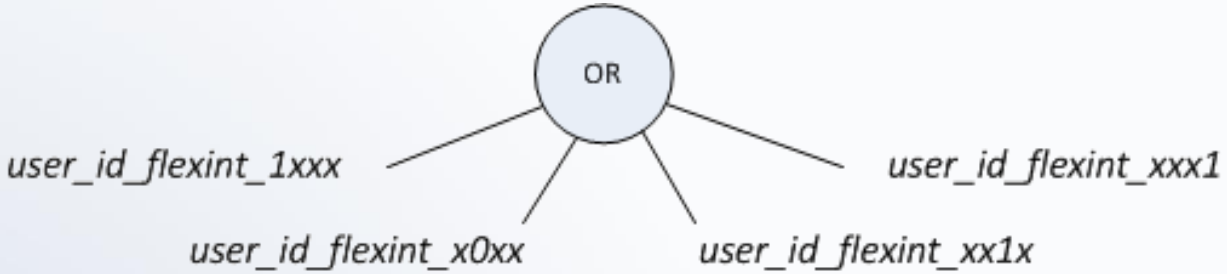
$$
\begin{aligned}
&USK = UserKeyGen(ASK_i, US, PK, f):\\
&\quad \text{choose randomly: } \tilde{r} \in \mathbb{Z}_p\\
&\quad \tilde{D} = Df^{\tilde{r}}\\
&\quad FOR\ \forall k \in US:\\
&\qquad \text{choose randomly: } \tilde{r}_k \in \mathbb{Z}_p\\
&\qquad \widetilde{D''}_k = D_k g^{\tilde{r}} H(k)^{\tilde{r}_k}\\
&\qquad \tilde{D}'_k = D'_k g^{\tilde{r}_k}\\
&\quad USK = (\tilde{D}, \widetilde{D''}_k, \tilde{D}'_k)
\end{aligned}
$$

**Equation 17: DMACPSABE UserKeyGen Function**

# Distributed Multi-Authority Ciphertext-Policy Shared Attribute-Based Encryption

## Not Equals:

**user_id ≠ 4:**




user_id_flexint_1xxx
OR user_id_flexint_x0xx
OR user_id_flexint_xx1x
OR user_id_flexint_xxx1

# Distributed Multi-Authority Ciphertext-Policy Shared Attribute-Based Encryption

## User Origin:

{auth_key}

Master Authority

auth_key = 1

{auth_key_flexint_0xxx,
auth_key_flexint_x0xx,
auth_key_flexint_xx0x,
auth_key_flexint_xxx1}

Attribute Authority

Domain 1

Attribute Authority

Domain 2

auth_key = 2

{auth_key_flexint_0xxx,
auth_key_flexint_x0xx,
auth_key_flexint_xx1x,
auth_key_flexint_xxx0}

# Distributed Multi-Authority Ciphertext-Policy Shared Attribute-Based Encryption

## Revocation and Expiration:

- User Key
  - Revocation:

$$user\_id \neq 1234$$

$$(user\_id \neq 1234 \text{ OR } auth\_key \neq 1)$$

  - Expiration:

$$key\_date \geq \{DATE\ OF\ ENCRYPTION\}$$

$$key\_date \geq \{DATE\ OF\ ENCRYPTION - 1\ Week\}$$

$$key\_expiration \geq \{DATE\ OF\ ENCRYPTION\}$$

- Authority Key
  - Revocation:

$$auth\_key \neq 2$$

  - Expiration:

$$auth\_key\_expiration \geq \{DATE\ OF\ ENCRYPTION\}$$

# Distributed Multi-Authority Ciphertext-Policy Shared Attribute-Based Encryption

## Revocation and Expiration:

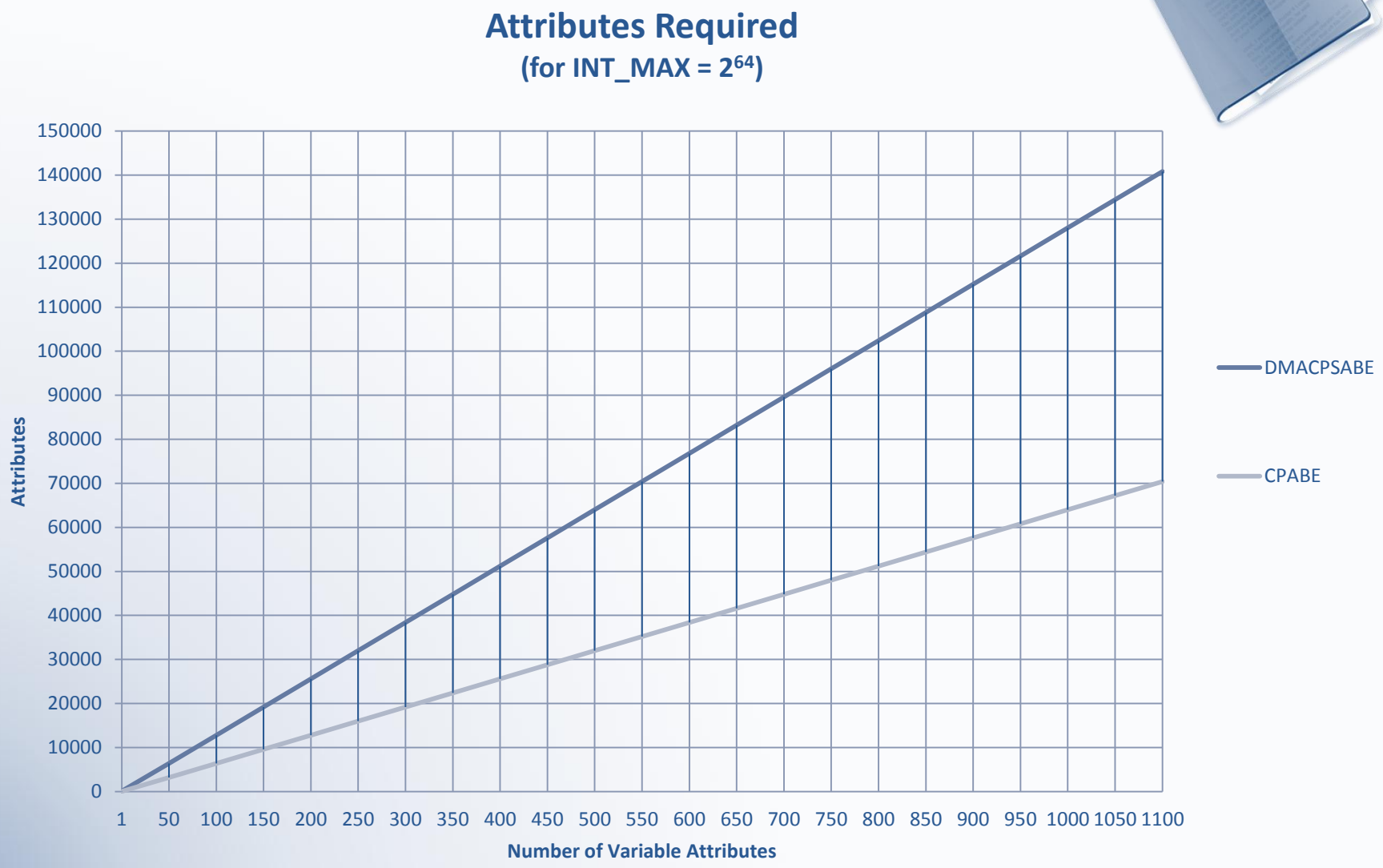
- Authentication Methods
- IP Addresses
- Server Versions
- Client Versions
- Time of Day
- Day of week
- Month
- Year
- GEO IP
- ETC.

# Evaluation

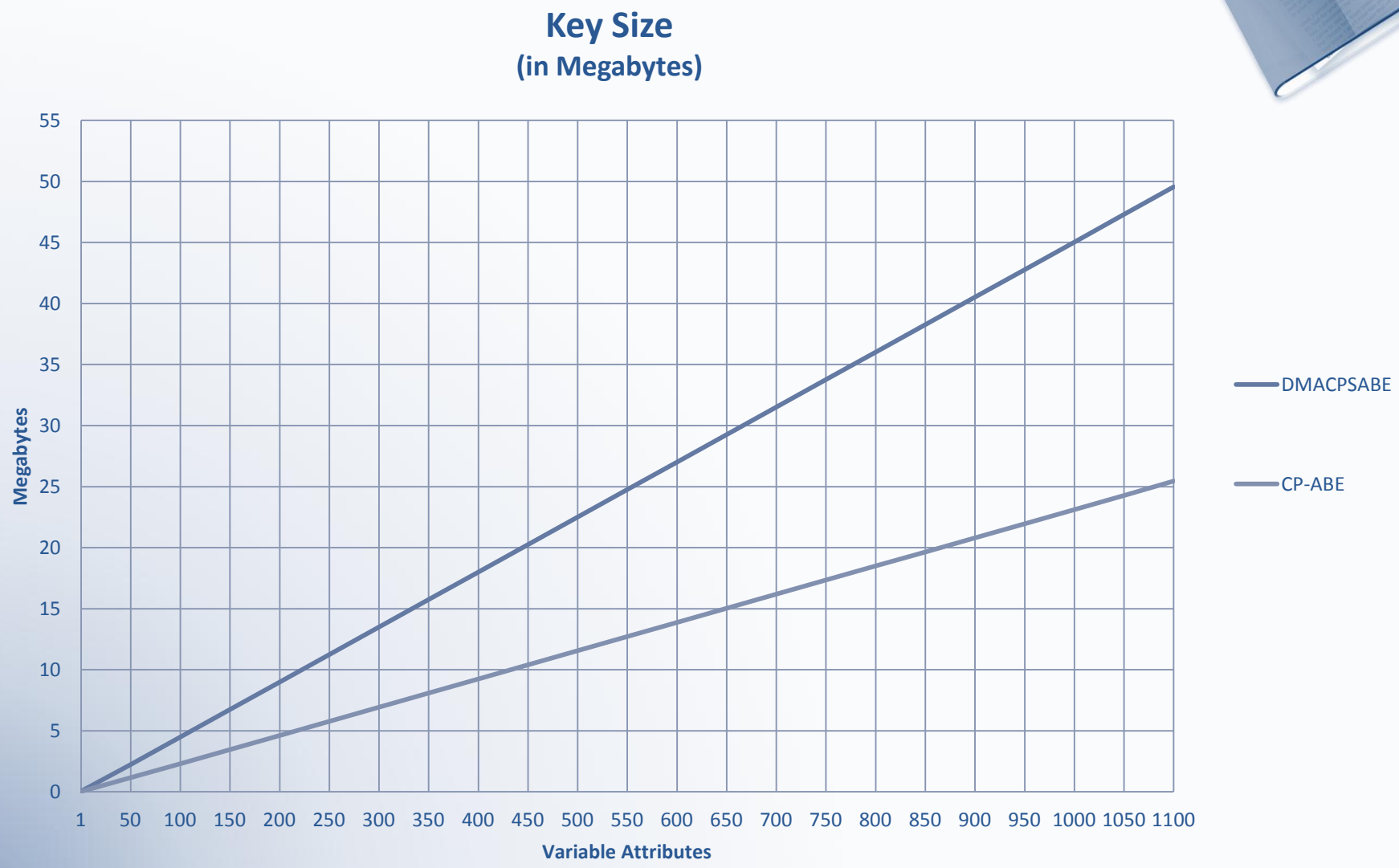
## Implementation and Evaluation Details:

- Based on J. Bethencourt, et al.'s (2006) CP-ABE implementation
- Uses the PBC library for algebraic operations
- Unix/Linux based
- Results compared to Bethencourt, et al.'s (2006) implementation
- System specifications:
  **CPU:** Intel Core2 Quad CPU Q6700 @ 2.66GHz
  **RAM:** 4GB
  **Hard Drive:** 30GB
  **Network:** 10/100/1000Mbps

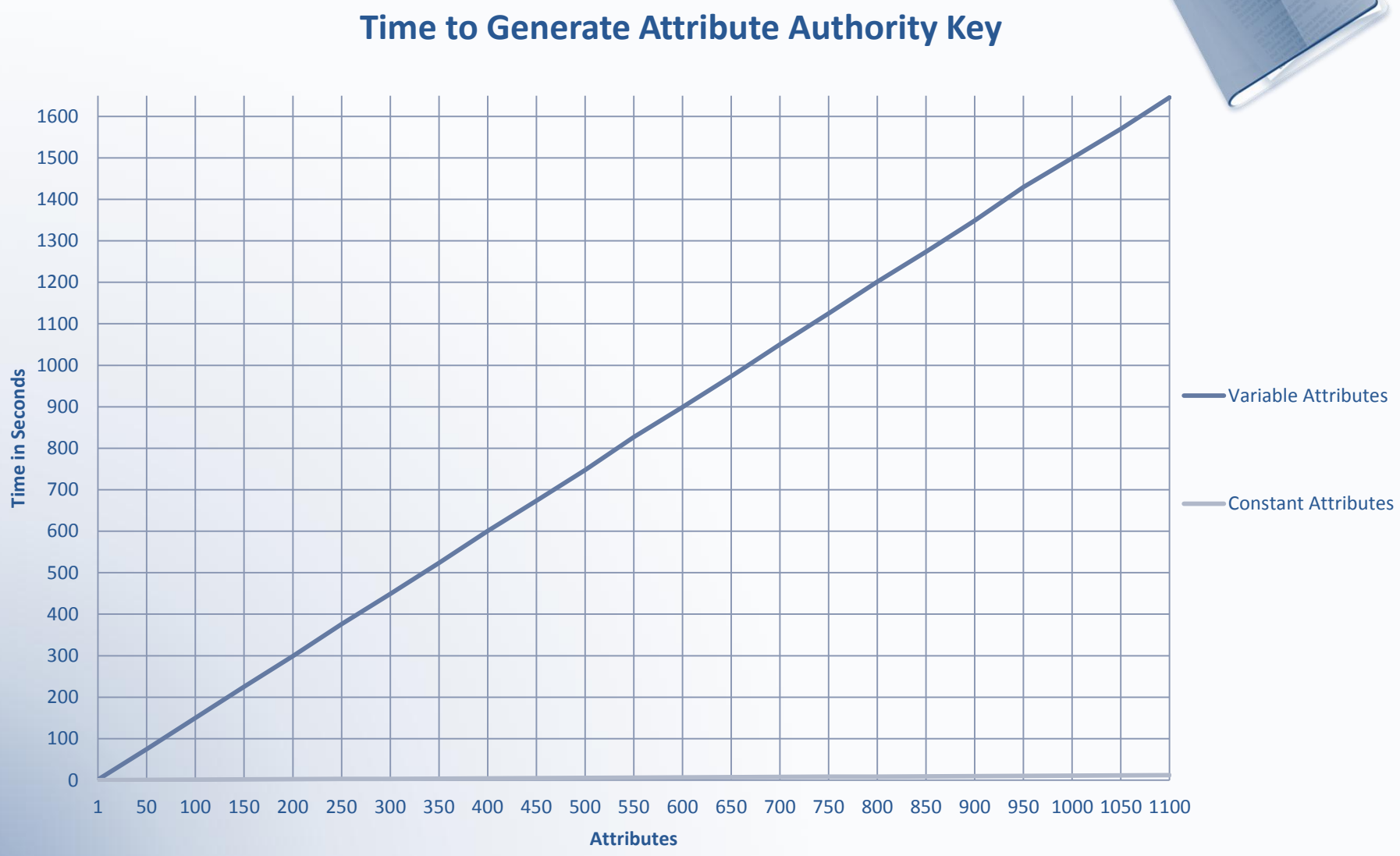
# Evaluation


Attributes Required
(for INT_MAX = 2^64)
Attributes
150000
140000
130000
120000
110000
100000
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
1
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1000
1050
1100
Number of Variable Attributes
DMACPSABE
CPABE

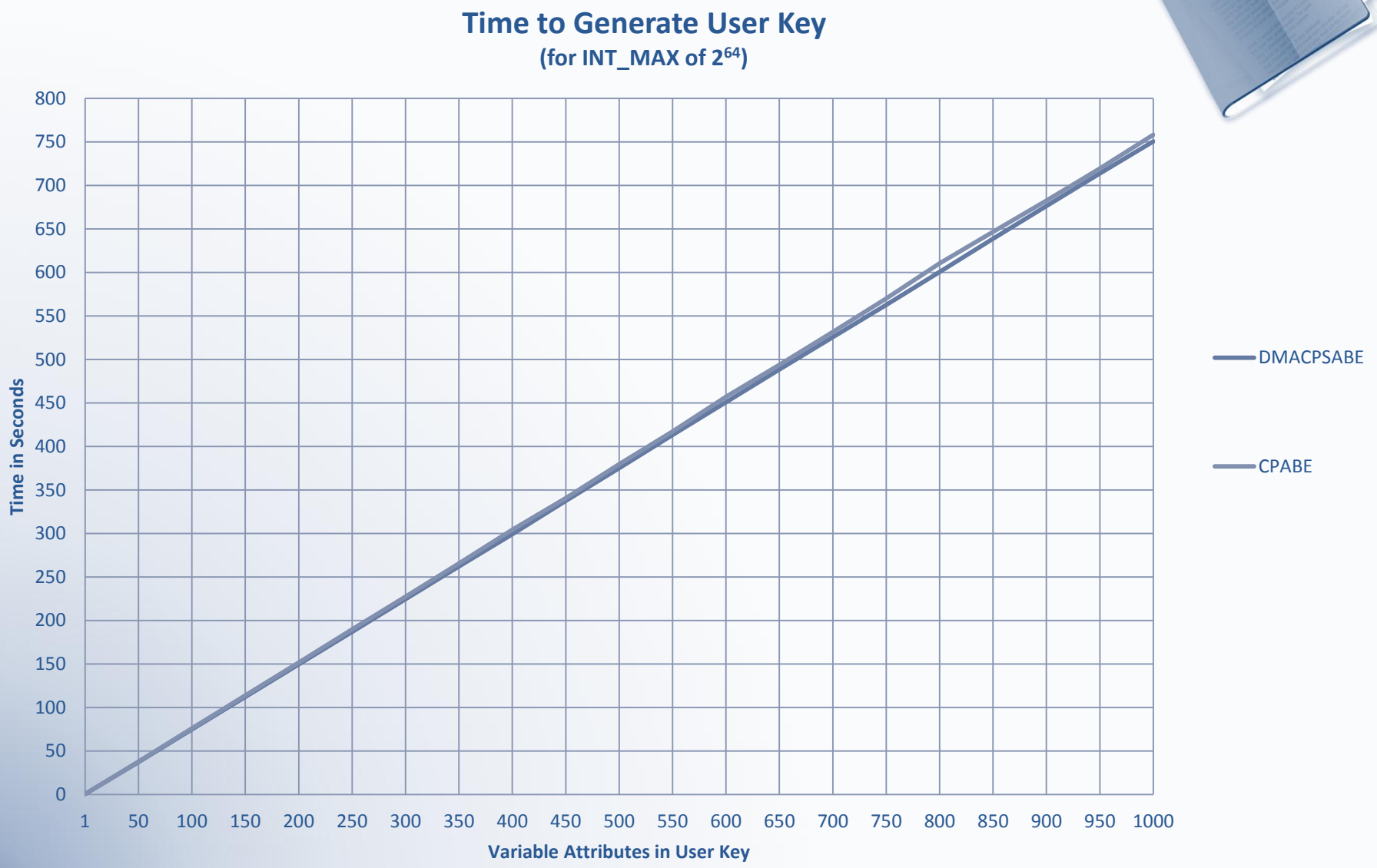
# Evaluation


Key Size
(in Megabytes)
Megabytes
55
50
45
40
35
30
25
20
15
10
5
0
1
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1000
1050
1100
Variable Attributes
DMACPSABE
CP-ABE

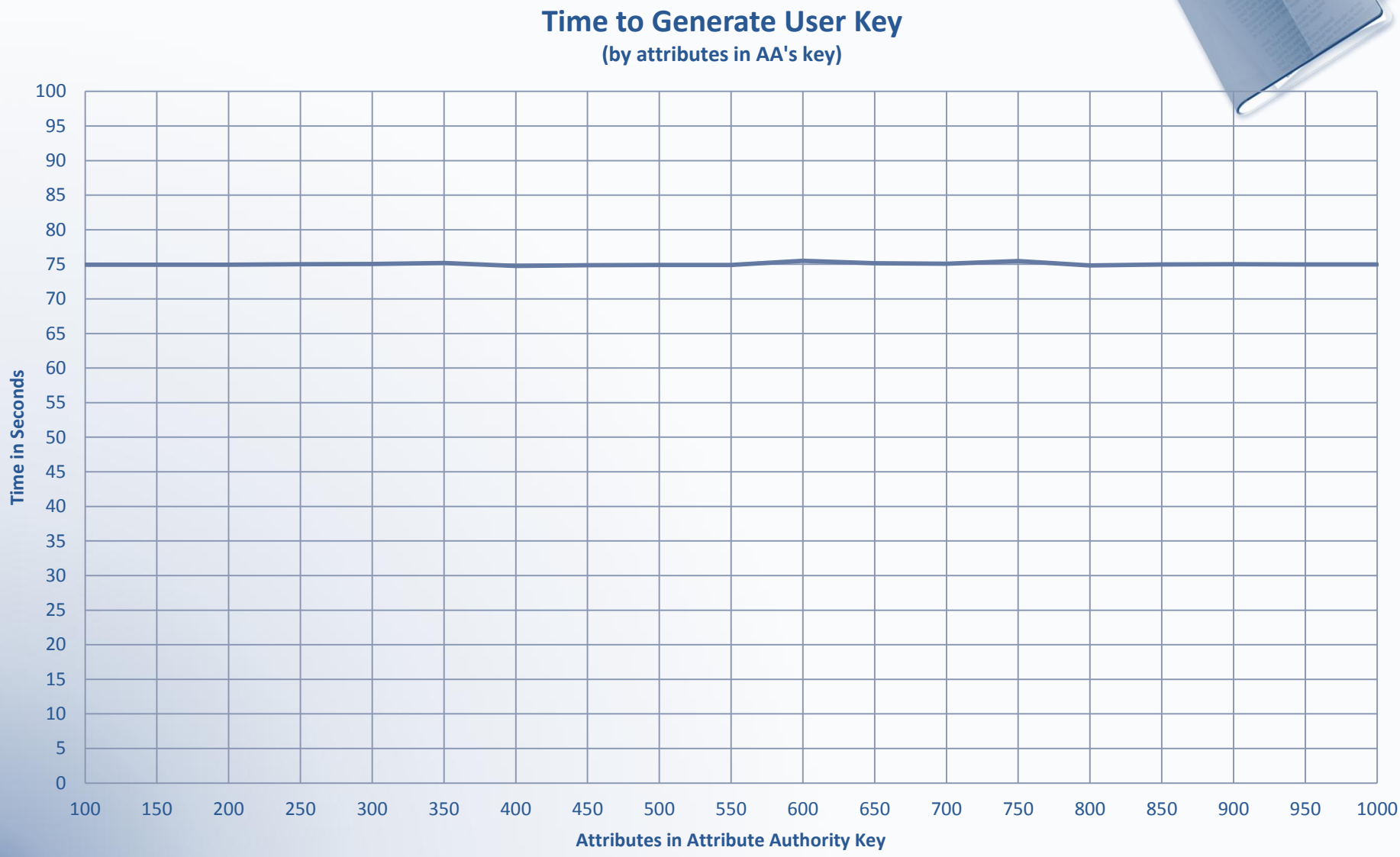
# Evaluation


Time to Generate Attribute Authority Key
Time in Seconds
1600
1500
1400
1300
1200
1100
1000
900
800
700
600
500
400
300
200
100
0
1
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1000
1050
1100
Attributes
Variable Attributes
Constant Attributes

# Evaluation


Time to Generate User Key
(for INT_MAX of 2^64)
Time in Seconds
800
750
700
650
600
550
500
450
400
350
300
250
200
150
100
50
0
1
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1000
Variable Attributes in User Key
DMACPSABE
CPABE

# Evaluation


Time to Generate User Key
(by attributes in AA's key)
Time in Seconds
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1000
Attributes in Attribute Authority Key

# Performance Improvements

$SK = KEYGEN\_PARALLEL(MK, S, PK)$:
$randomize(r)$
$D = g^{(\alpha+r)/\beta}$
$B = Array\ Same\ Size\ as\ S$
$FOR\ \forall i \in S$:
$Start\ Thread\ For\ keygen\_compute(B, r, g, i)$
$WaitForAllThreadsToFinish()$
$SK = (D, B)$

**Equation 18: Parallelized version of the KEYGEN function.**

keygen$_compute(B, r, g, i)$:
$randomize(k)$
$D''_i = g^r \cdot H(i)^k$
$D'_i = g^k$
$B_i = D''_i, D'_i$

**Equation 19: keygen compute function to be run in parallel.**

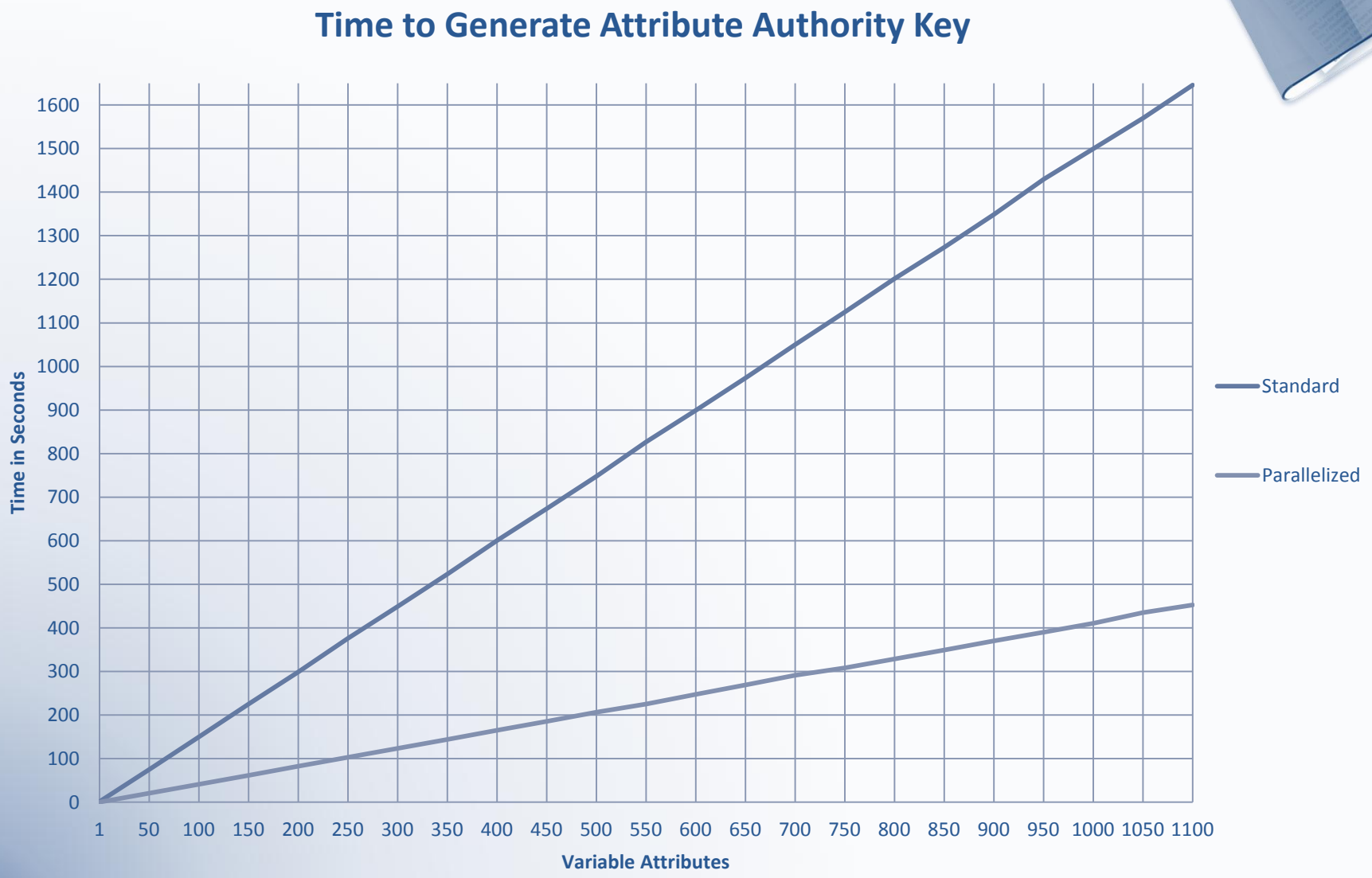
$USK = UserKeyGen\_Parallel(ASK, US, PK, f)$:
$randomize(\tilde{r})$
$\tilde{D} = Df^{\tilde{r}}$
$\tilde{B} = Array\ Same\ Size\ as\ US$
$FOR\ \forall i \in US$:
$Start\ Thread\ For\ userkeygen\_compute(\tilde{B}, \tilde{r}, g, i, D_i)$
$WaitForAllThreadsToFinish()$
$USK = (D, \tilde{B})$

**Equation 20: Parallelized version of the UserKeyGen function.**

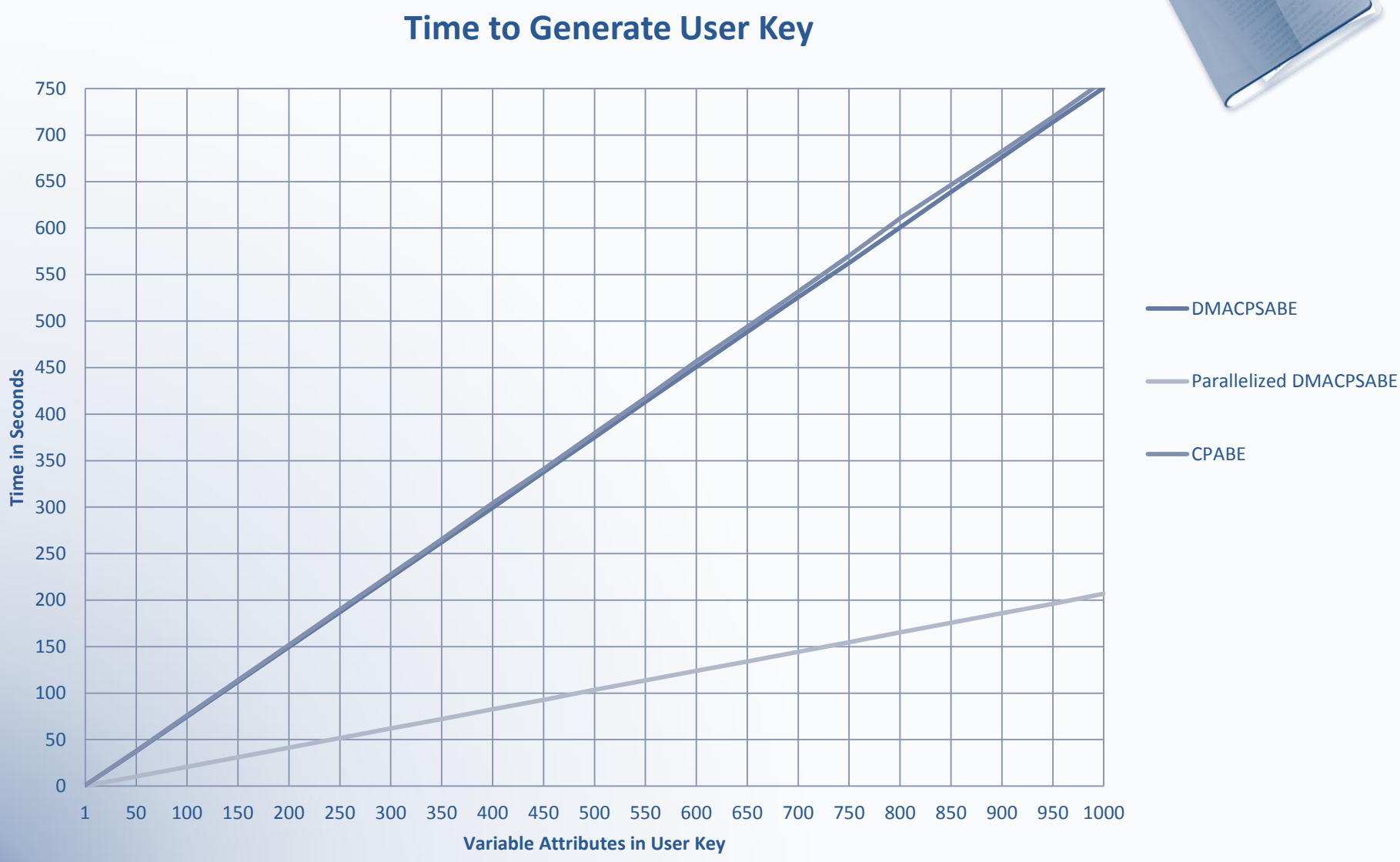
userkeygen$_compute(\tilde{B}, \tilde{r}, g, i, D_i)$:
$randomize(k)$
$\widetilde{D''}_i = D_i \cdot g^{\tilde{r}} \cdot H(i)^k$
$\tilde{D}'_i = D'_i \cdot g^k$
$\tilde{B}_i = \widetilde{D''}_i, \tilde{D}'_i$

**Equation 21: userkeygen compute function to be run in parallel.**

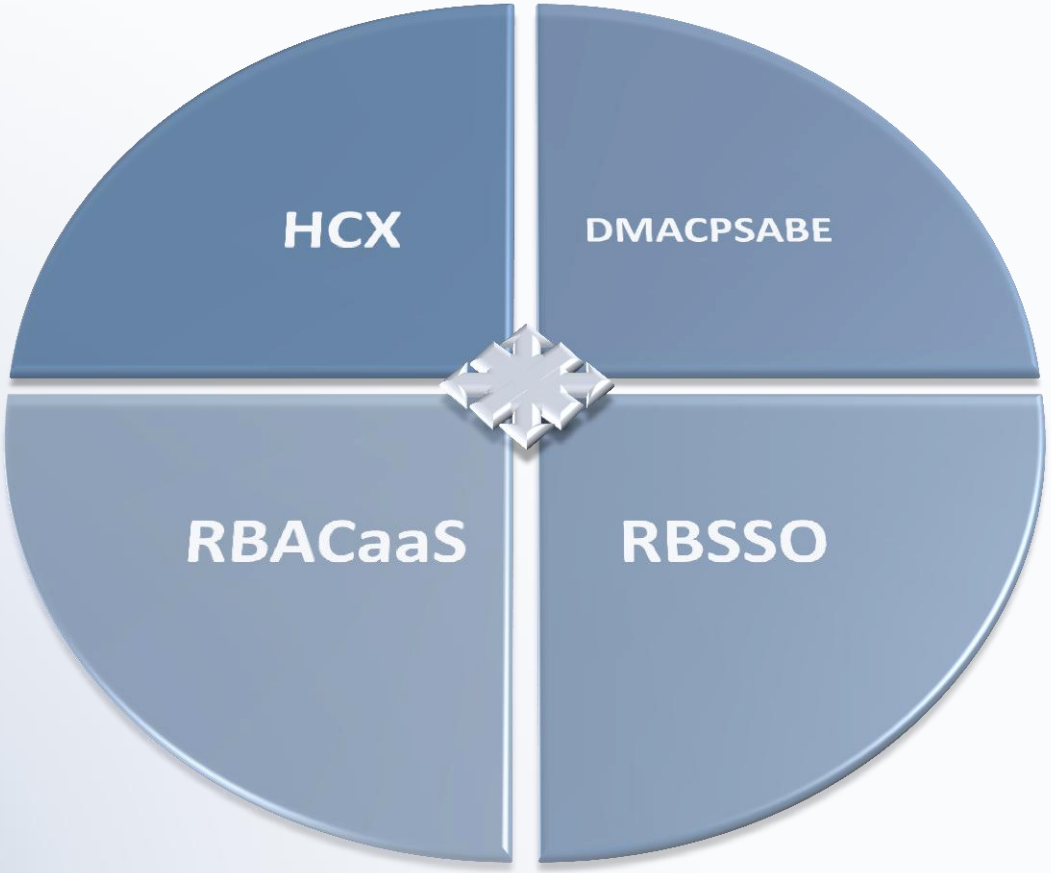
# Performance Improvements



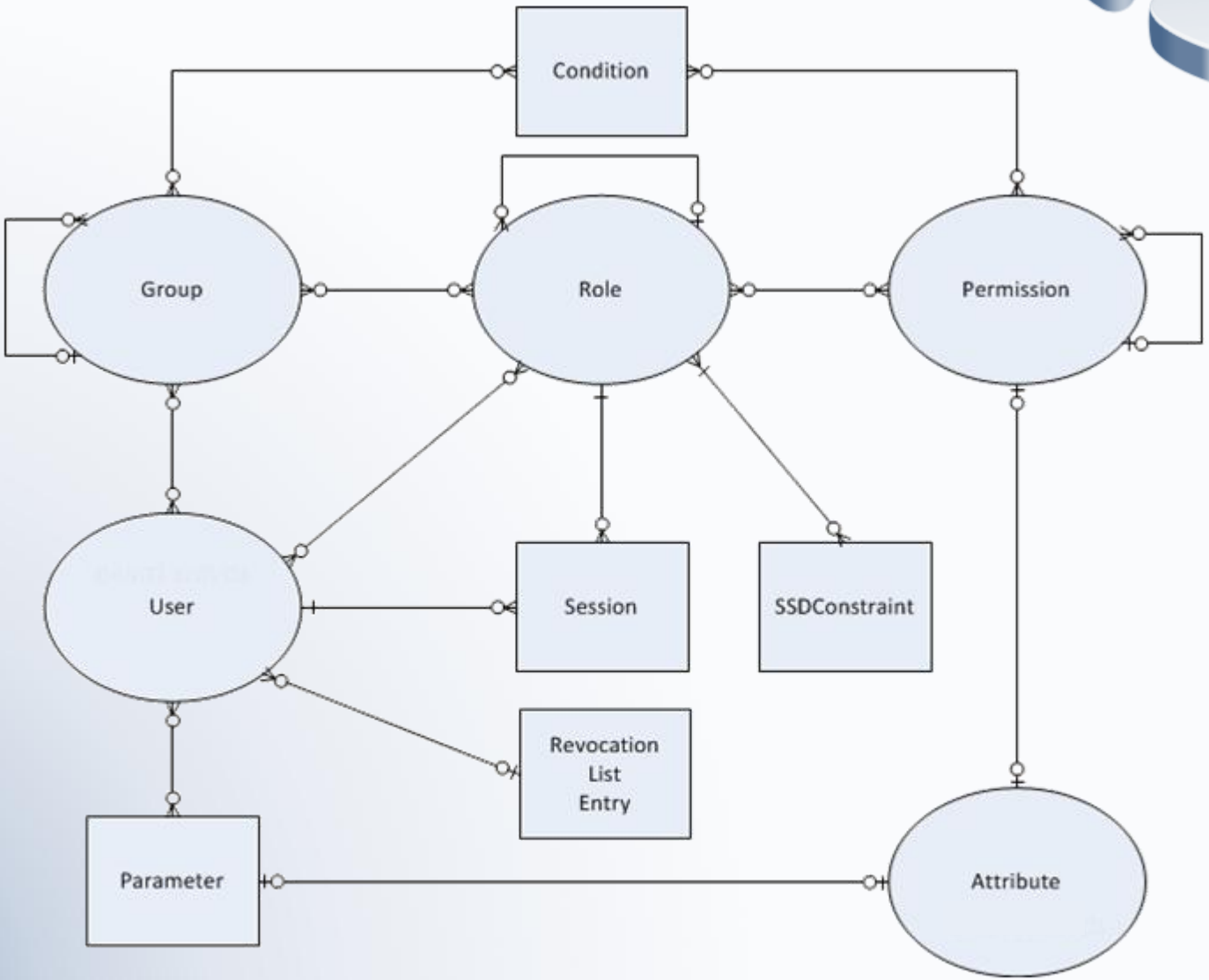
# Performance Improvements



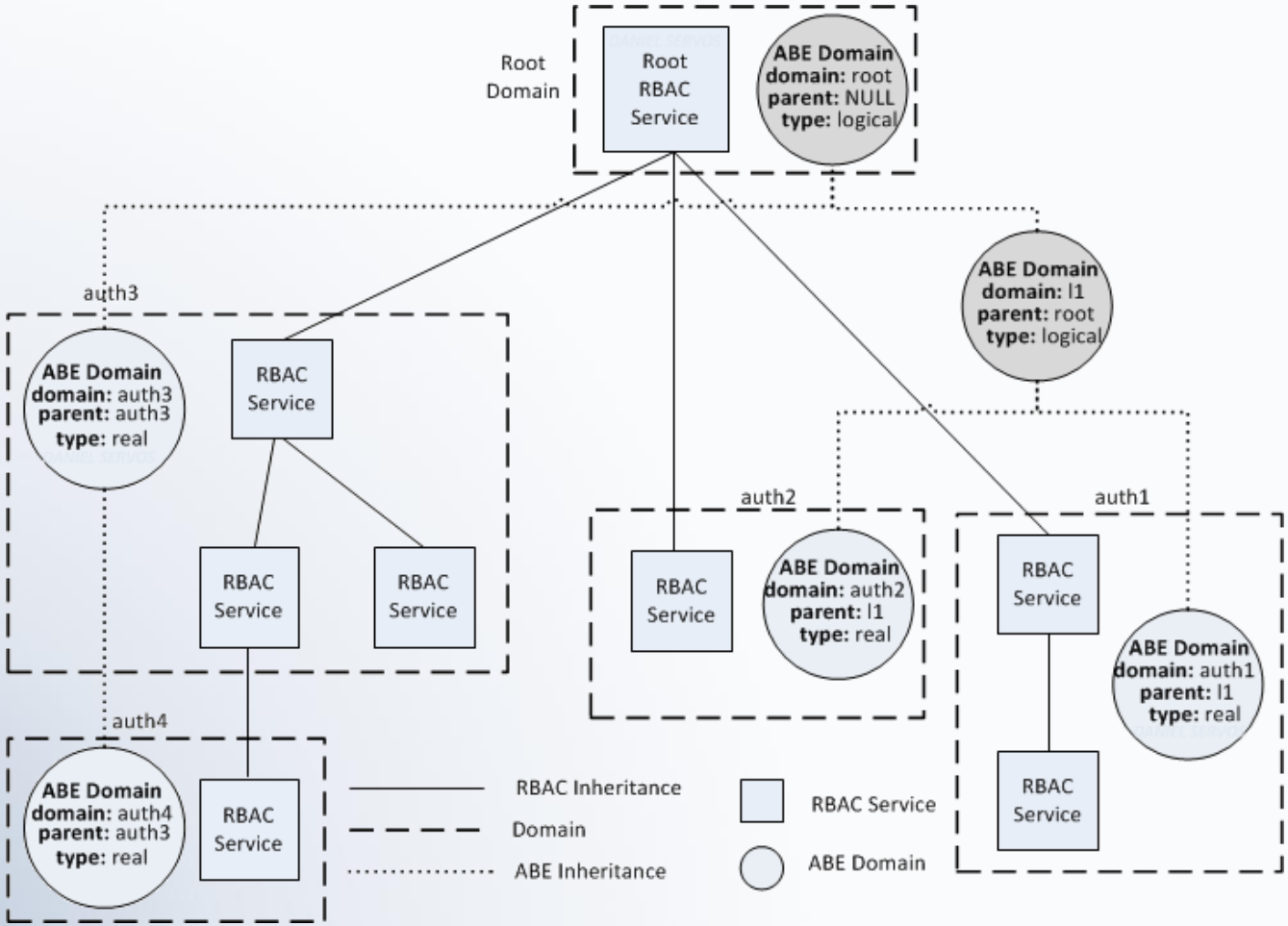
# Putting it all Together



# RBACaaS Integration with DMACPSABE



# RBACaaS Integration with DMACPSABE



# RBACaa

| Attribute Name | Type | Description |
|---|---|---|
| SYSTEM:TIME_STAMP | Integer | The date and time on the auth server when the session was started as a Unix time stamp. |
| SYSTEM:TIME_DAY | Integer | A number [1, 31] representing the day when the session was started in the current month. Based on gregorian calendar and UTC. |
| SYSTEM:TIME_HOUR | Integer | A number [0, 23] representing the hour when the session was started in UTC. |
| SYSTEM:TIME_MINUTE | Integer | A number [0, 59] representing the minute the session was started in UTC. |
| SYSTEM:TIME_SECOND | Integer | A number [0, 59] representing the second the session was started in UTC. |
| SYSTEM:TIME_WEEK_DAY | Integer | The week day when the session was started represented by a number starting at 0 for Sunday and ending at 6 for Saturday. Based on UTC. |
| SYSTEM:TIME_MONTH | Integer | A number [1, 12] representing the UTC gregorian calendar month when the session was started. |
| SYSTEM:TIME_YEAR | Integer | A number representing the gregorian calendar year when the session was started in UTC. |
| SYSTEM:USER_IP | Integer | An integer representation of the user's version 4 IP at the time they authenticated with the server. |
| SYSTEM:USER_IP_1 | Integer | An integer representation of the first byte of a user's version 4 IP at the time they authenticated with the server. |
| SYSTEM:USER_IP_2 | Integer | An integer representation of the second byte of a user's version 4 IP at the time they authenticated with the server. |
| SYSTEM:USER_IP_3 | Integer | An integer representation of the third byte of a user's version 4 IP at the time they authenticated with the server. |
| SYSTEM:USER_IP_4 | Integer | An integer representation of the fourth byte of a user's version 4 IP at the time they authenticated with the server. |
| SYSTEM:USER_DOMAIN_ID<br>AKA: auth_key | Integer | The ID assigned to the auth server's domain. |
| SYSTEM:USER_GID | Integer | The user's GID. |
| SYSTEM:USER_START_DATE | Integer | A unix time stamp containing the date the user's account was activated. |
| SYSTEM:USER_END_DATE | Integer | A unix time stamp containing the date the user's account will be or was deactivated or "0" if no such date is set. |
| SYSTEM:SESSION_START | Integer | A unix time stamp containing the date and time the user's session was started. |
| SYSTEM:SESSION_EXPIRE | Integer | A unix time stamp containing the date and time the user's session will expire. |
| SYSTEM:CLIENT_VERSION | Integer | An integer representation of the version number of the client software the user used to authenticate with the server. |
| SYSTEM:SERVER_VERSION | Integer | An integer representation of the version number of the server software being used. |
| SYSTEM:AUTH_METHOD | Integer | An integer representing the authentication method used to authorize the user. |

# RBACa

| Rule | Explanation |
|---|---|
| ( {CURRENT_DATE} < SYSTEM:SESSION_EXPIRE ) | "{CURRENT_DATE}" is replaced with the time of encryption. This prevents expired keys from decrypting documents created after the key's expiration date. |
| ( SYSTEM:USER_END_DATE = 0 OR {CURRENT_DATE} < SYSTEM:USER_END_DATE ) | "{CURRENT_DATE}" is replaced with the time of encryption. This prevents expired user accounts from decrypting documents created after the accounts expiration date. |
| ( {MIN_CLIENT} ≤ SYSTEM:CLIENT_VERSION ) | Optional rule for limiting client versions.<br><br>"{MIN_CLIENT}" is replaced with the minimum allowed client version to access a file. Allows banning of out of date clients for newly encrypted files. |
| ( {MIN_SERVER} ≤ SYSTEM:SERVER_VERSION ) | Optional rule for limiting authentication server versions.<br><br>"{MIN_SERVER}" is replaced with the minimum allowed server version to access a file. Allows banning of out of date servers for newly encrypted files. |
| ( SYSTEM:USER_DOMAIN_ID ≠ 0 ) | Ensures that an attribute for a domain is set to anything but 0. (Note that not equals ensures that the user has some value for the attribute so long as it is not the given constant). |
| ( SYSTEM:USER_GID ≠ 0 ) | Ensures that an attribute for a user's global ID is set to anything but 0. (Note that not equals ensures that the user has some value for the attribute so long as it is not the given constant). |
| ( SYSTEM:USER_GID NOT IN ( {USER_BLACK_LIST_SET} )) | Optional rule for blocking a set of users from accessing newly encrypted files.<br><br>"{USER_BLACK_LIST_SET}" is replaced with the set of black listed user's GIDs, blocking them from accessing newly encrypted files. |
| ( SYSTEM:USER_DOMAIN_ID NOT IN ( {DOMAIN_BLACK_LIST_SET} )) | Optional rule for blocking a set of domains from access newly encrypted files.<br><br>"{DOMAIN_BLACK_LIST_SET}" is replaced with the set of black listed domains. Domains may be blocked from reading newly encrypted files in the case they are compromised. |
| ( SYSTEM:AUTH_METHOD NOT IN ( {AUTH_METHOD_BLACK_LIST_SET} )) | Optional rule for blocking weak or compromised authentication methods.<br><br>"{AUTH_METHOD_BLACK_LIST_SET}" is replaced with the set of black listed authentication methods. |
| ( SYSTEM:USER_IP NOT IN ( {IP_BLACK_LIST} )) | Optional rule for blocking users by IP rather than GID.<br><br>"{IP_BLACK_LIST}" is replaced by the set of IPs to be blocked from decrypting newly encrypted files. Blocked IPs are based on the IP used to authenticate with the authentication service. |

# HCX Integration with RBACaaS and RBSSO

- Create permissions for each user task
  - For example:

```
EHR.*          EHR.view.*            EHR.edit.*
               EHR.view.ident.*      EHR.edit.ident.*
               EHR.view.medical.*    EHR.edit.medical.*
               EHR.view.lab.*        EHR.edit.lab.*
               EHR.view.insurace.*   EHR.edit.insurace.*
```

- Add conditional permissions in RBACaaS
  - For example:

```
EHR.edit.medical.intranet
          SYSTEM:USER_IP_1 == 192 AND SYSTEM:USER_IP_2 == 168
          AND (SYSTEM:USER_IP_3 == 100 OR SYSTEM:USER_IP_3 == 110)

EHR.eidt.lab.intranet
          SYSTEM:USER_IP_1 == 192 AND SYSTEM:USER_IP_2 == 168
          AND (SYSTEM:USER_IP_3 == 100 OR SYSTEM:USER_IP_3 == 110)
EHR.view.insurace.bizhours
          SYSTEM:TIME_HOUR >= 9 AND SYSTEM:TIME_HOUR <= 17
```

- Accept and check user RequestToken and AuthToken from RBSSO

# HCX Integration with RBACaaS and RBSSO

- Check that user has a given permission using the RBACaaS API
  - For Example:

```
hasPermission("EHR.* OR EHR.edit.* OR EHR.edit.medical.* OR EHR.edit.medical.intranet");
```

- Periodically request revocation lists of users, domains, authentication methods, client versions, or server versions

# HCX Ir

| Permission | Function |
|---|---|
| RBACAdmin.* | Grants access rights to all administrative functions. |
| RBACAdmin.user.* | Grants access rights to all user related functions (add user, mapping user to roles/groups, removing user, etc.). |
| RBACAdmin.user.add | Grants right to add a user. |
| RBACAdmin.user.remove | Grants right to remove a user |
| RBACAdmin.user.maprole.* | Grants right to map or unmap any role to a user. |
| RBACAdmin.user.mapgroup.* | Grants right to map or unmap any group from a user. |
| RBACAdmin.user.maprole.{role_name} | Grants the right to map or unmap the role {role_name} to a user. |
| RBACAdmin.user.mapgroup.{group_name} | Grants the right to map or unmap the group {group_name} to a user. |
| RBACAdmin.user.addparam.* | Grants the right to add any parameter/value pair to a user. |
| RBACAdmin.user.removeparam.* | Grants the right to remove any parameter/value pair from a user. |
| RBACAdmin.user.addparam.{param_name} | Grants the right to add the parameter/value pair for {param_name} to a user. |
| RBACAdmin.user.removeparam.{param_name} | Grants the right to remove the parameter/value pair for {param_name} from a user. |
| RBACAdmin.group.* | Grants all access rights on group functions. |
| RBACAdmin.group.add | Grants the right to add a user group. |
| RBACAdmin.group.remove | Grants the right to remove a user group. |
| RBACAdmin.group.parent | Grants the right to set a groups parent. |
| RBACAdmin.group.mapcon.* | Grants the right to add or remove a condition to a group. |
| RBACAdmin.group.mapcon.{group_name} | Grants the right to add or remove a condition to the group {group_name}. |
| RBACAdmin.group.maprole.* | Grants the right to add or remove a role to a group. |

# HCX In

| | |
|---|---|
| RBACAdmin.group.maprole.* | Grants the right to add or remove a role to a group. |
| RBACAdmin.group.maprole.{role_name}.* | Grants the right to add or remove the role {role_name} to any group. |
| RBACAdmin.role.* | Grants all access rights on role functions. |
| RBACAdmin.role.add | Grants the right to add a role. |
| RBACAdmin.role.remove | Grants the right to remove a role. |
| RBACAdmin.role.parent | Grants the right to set a roles parent. |
| RBACAdmin.role.mapperm.* | Grants the right to map or unmap permission to a role. |
| RBACAdmin.role.mapperm.{perm_name} | Grants the right to map or unmap the permission {perm_name} to a role. |
| RBACAdmin.perm.* | Grants all rights to permission functions. |
| RBACAdmin.perm.add | Right to register a permission. |
| RBACAdmin.perm.remove | Right to unregister a permission. |
| RBACAdmin.perm.mapcon.* | Grants right to add or remove a condition to a permission. |
| RBACAdmin.perm.mapcon.{perm_name} | Grants right to add a condition to the permission {perm_name}. |
| RBACAdmin.ssd.* | Grants all right to SSD constraint functions. |
| RBACAdmin.ssd.add | Grants right to add an SSD constraint |
| RBACAdmin.ssd.remove | Grants right to remove an SSD constraint |
| RBACAdmin.revoke | Grants right to revoke a session. |
| RBACAdmin.view.* | Grants right to view all RBAC elements. |
| RBACAdmin.view.user.* | Grants right to view all user records. |
| RBACAdmin.view.user.{user_name} | Grants right to view record for given {user_name}. |
| RBACAdmin.view.role.* | Grants right to view all role records. |
| | Grants right to view record for given {role_name} |

# HCX I

| | |
|---|---|
| RBACAdmin.view.user.{user_name} | Grants right to view record for given {user_name}. |
| RBACAdmin.view.role.* | Grants right to view all role records. |
| RBACAdmin.view.role.{role_name} | Grants right to view record for given {role_name}. |
| RBACAdmin.view.group.* | Grants right to view all group records. |
| RBACAdmin.view.group.{group_name} | Grants right to view record for given {group_name}. |
| RBACAdmin.view.con.* | Grants right to view all condition records. |
| RBACAdmin.view.con.{con_name} | Grants right to view record for given {con_name}. |
| RBACAdmin.view.perm.* | Grants right to view all permission records. |
| RBACAdmin.view.perm.{perm_name} | Grants right to view record for given {perm_name}. |
| RBACAdmin.view.ssd | Grants right to view all ssd records. |
| RBACAdmin.view.rl | Grants right to view all revocation list records. |
| RBACAdmin.view.param.* | Grants right to view all parameter records. |
| RBACAdmin.view.param.{param_name} | Grants right to view record for given {param_name}. |
| RBACAdmin.view.sessions | Grants right to view all active sessions. |
| RBACAdmin.view.log | Grants right to view the auditlog. |
| RBACAdmin.system.* | Grants all system commands. |
| RBACAdmin.system.shutdown | Grants right to shut down the RBAC service. |
| RBACAdmin.system.restart | Grants right to reboot the RBAC service. |
| RBACAdmin.system.setdomain | Grants right to set RBAC domain name. |
| RBACAdmin.system.addchilddomain | Grants right to add a child domain. |
| RBACAdmin.system.setparentdomain | Grants right to set the domain's parent. |

# Extensions to CCR and Other XML Formats

```
<?xml version="1.0"?>
<DMACPSABE>
    <DMACPSABE:header>
            <DMACPSABE:meta>
                    <DMACPSABE:versions>
                            <DMACPSABE:encryption>{ENCRYT_VER}</DMACPSABE:encryption>
                            <DMACPSABE:rbac>{RBAC_VER}</DMACPSABE:rbac>
                            <DMACPSABE:format>{FORMAT_VER></DMACPSABE:format>
                    </DMACPSABE:versions>
                    <DMACPSABE:id>{RECORD_ID}</DMACPSABE:id>
                    … Other meta data needed by an implementation. …
            </DMACPSABE:meta>
            <DMACPSABE:permissions>
                    <DMACPSABE:view>{PERM_VIEW}</DMACPSABE:view>
                    <DMACPSABE:edit>{PERM_EDIT}</DMACPSABE:edit>
                    <DMACPSABE:perm>{PERM_PERM}</DMACPSABE:perm>
            </DMACPSABE:permissions>
            <DMACPSABE:keys>
                    <DMACPSABE:public>
                            {PUB_KEY}
                    </DMACPSABE:public>
                    <DMACPSABE:private>
                            ENCRYPTED WITH {PERM_EDIT} POLICY:
                            ____________________________________
                            <DMACPSABE:id>{RECORD_ID}</DMACPSABE:id>
                            <DMACPSABE:sigkey>{PRVI_KEY}</DMACPSABE:sigkey>
                            <DMACPSABE:nonce>{NONCE}</DMACPSABE:nonce>
                            ____________________________________
                    </DMACPSABE:private>
            </DMACPSABE:keys>
    </DMACPSABE:header>
    <DMACPSABE:body>
            … Any unecrypted XML data …
                    <DMACPSABE:element>
                            <DMACPSABE:permissions>
                                    <DMACPSABE:view>{PERM_VIEW}</DMACPSABE:view>
                                    <DMACPSABE:edit>{PERM_EDIT}</DMACPSABE:edit>
                                    <DMACPSABE:perm>{PERM_PERM}</DMACPSABE:perm>
                            </DMACPSABE:permissions>
                            <DMACPSABE:ciphertext>
                                    ENCRYPTED WITH {PERM_VIEW} POLICY:
                                    ____________________________________
```

```
            </DMACPSABE:meta>
            <DMACPSABE:permissions>
                        <DMACPSABE:view>{PERM_VIEW}</DMACPSABE:view>
                        <DMACPSABE:edit>{PERM_EDIT}</DMACPSABE:edit>
                        <DMACPSABE:perm>{PERM_PERM}</DMACPSABE:perm>
            </DMACPSABE:permissions>
            <DMACPSABE:keys>
                        <DMACPSABE:public>
                                    {PUB_KEY}
                        </DMACPSABE:public>
                        <DMACPSABE:private>
                                    ENCRYPTED WITH {PERM_EDIT} POLICY:
                                    ____________________________________
                                    <DMACPSABE:id>{RECORD_ID}</DMACPSABE:id>
                                    <DMACPSABE:sigkey>{PRVI_KEY}</DMACPSABE:sigkey>
                                    <DMACPSABE:nonce>{NONCE}</DMACPSABE:nonce>
                                    ____________________________________
                        </DMACPSABE:private>
            </DMACPSABE:keys>
    </DMACPSABE:header>
    <DMACPSABE:body>
            … Any unecrypted XML data …
                        <DMACPSABE:element>
                                    <DMACPSABE:permissions>
                                                <DMACPSABE:view>{PERM_VIEW}</DMACPSABE:view>
                                                <DMACPSABE:edit>{PERM_EDIT}</DMACPSABE:edit>
                                                <DMACPSABE:perm>{PERM_PERM}</DMACPSABE:perm>
                                    </DMACPSABE:permissions>
                                    <DMACPSABE:ciphertext>
                                                ENCRYPTED WITH {PERM_VIEW} POLICY:
                                                ____________________________________
                                                … Any XML data with elements sorted alphabetically …
                                                <DMACPSABE:cttail>
                                                            <DMACPSABE:id>{RECORD_ID}</DMACPSABE:id>
                                                            <DMACPSABE:nonce>{NONCE}</DMACPSABE:nonce>
                                                <DMACPSABE:cttail>
                                                ____________________________________
                                    </DMACPSABE:ciphertext>
                                    <DMACPSABE:signature>
                                                {CIPHERTEXT_SIG}
                                    </DMACPSABE:signature>
                                    <DMACPSABE:searchindex>
                                                … Optional implementation dependent search index …
                                    </DMACPSABE:searchindex>
                        </DMACPSABE:element>

            … Any unecrypted XML data …
    </DMACPSABE:body>
</DMACPSABE>
```

# Searching

- Several solutions exist:
  - (Chang & Mitzenmacher, 2005)
  - (Li, Wang, Wang, Cao, Ren, & Lou, 2010)
  - (Wang, Cao, Li, Ren, & Lou, 2010)
  - (Ballard, Green, Medeiros, & Monrose, 2005)

- Prototype Solution:
  - Hashed keyword plus salt
  - Hashed keywords paired with values

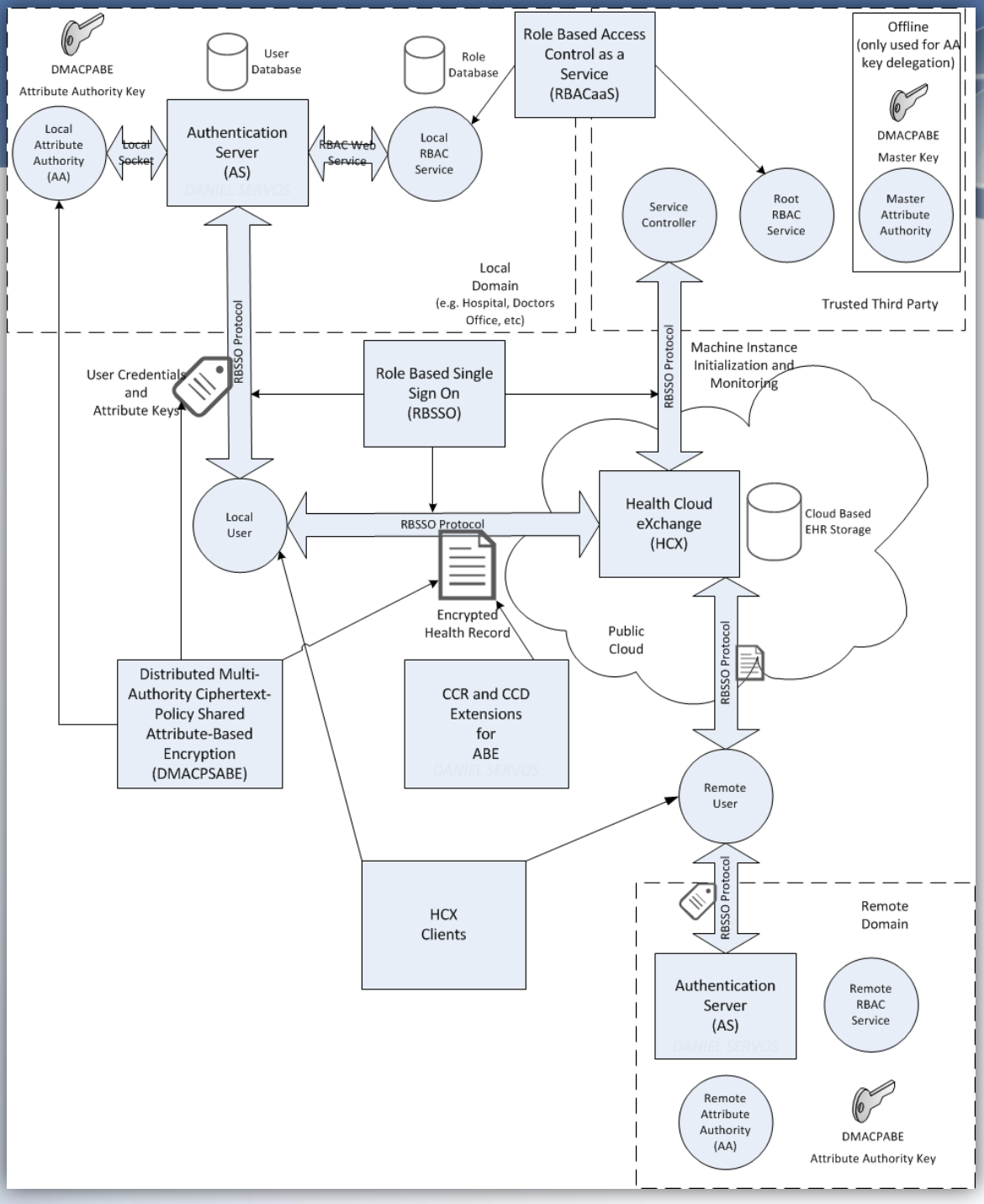
Compute (client side):

```
for each keyword, value in keyword_pair_set:
    keyword_table.add(hash(keyword + salt), value)
```

Search (service side):

```
for each file in file_set:
    value = file.keyword_table.get(keyword_hash)
    if value != null and value > threshold:
         reslut_table.add(file, value)
```

# System

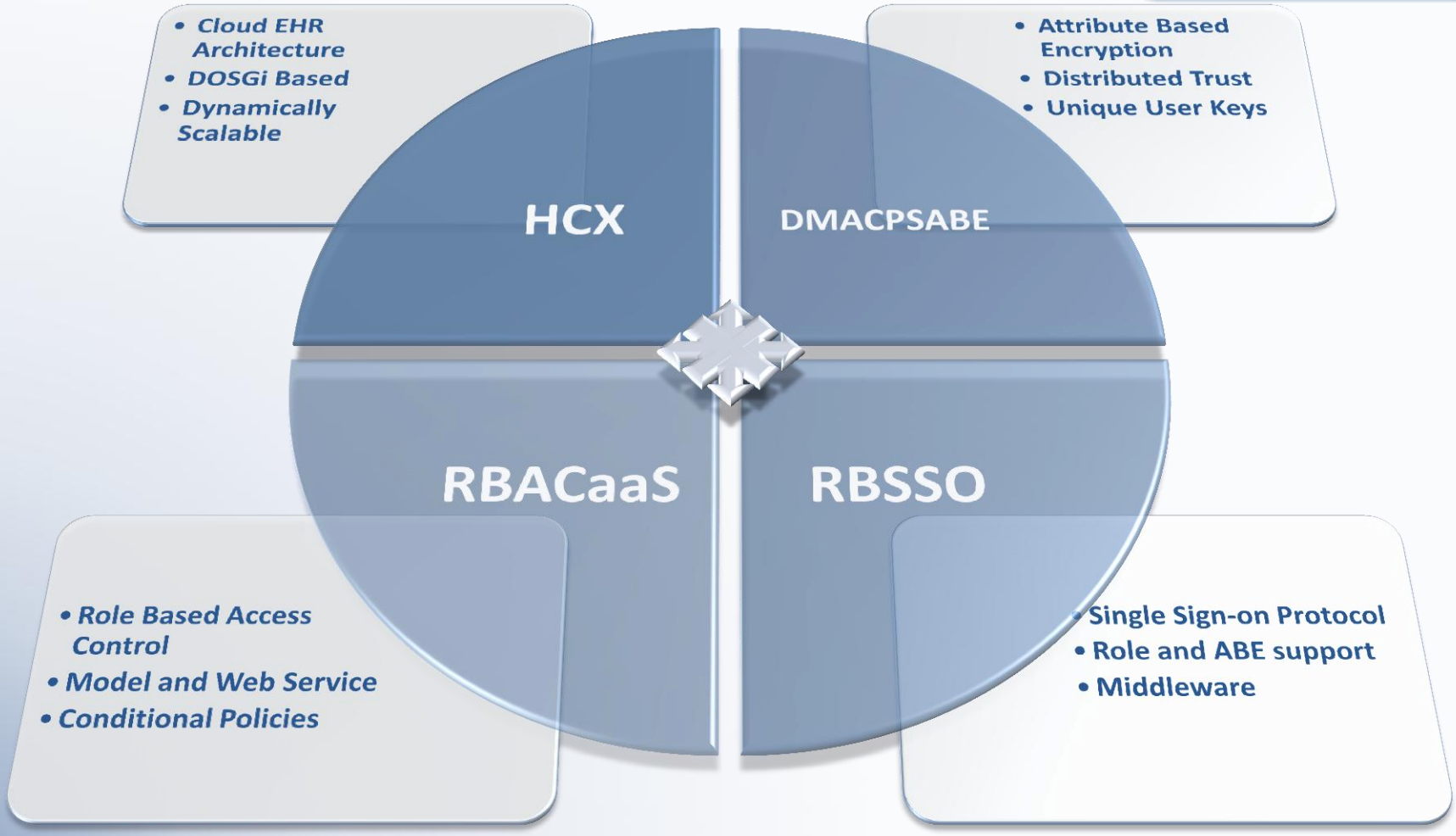

DMACPABE
Attribute Authority Key
User Database
Role Database
Role Based Access Control as a Service (RBACaaS)
Offline (only used for AA key delegation)
DMACPABE
Master Key
Master Attribute Authority
Local Attribute Authority (AA)
Local Socket
Authentication Server (AS)
RBAC Web Service
Local RBAC Service
Service Controller
Root RBAC Service
Local Domain (e.g. Hospital, Doctors Office, etc)
Trusted Third Party
RBSSO Protocol
User Credentials and Attribute Keys
Role Based Single Sign On (RBSSO)
Machine Instance Initialization and Monitoring
Local User
Health Cloud eXchange (HCX)
Cloud Based EHR Storage
Encrypted Health Record
Public Cloud
Distributed Multi-Authority Ciphertext-Policy Shared Attribute-Based Encryption (DMACPSABE)
CCR and CCD Extensions for ABE
Remote User
HCX Clients
Remote Domain
Authentication Server (AS)
Remote RBAC Service
Remote Attribute Authority (AA)
DMACPABE
Attribute Authority Key

# Prototype Demonstration

# Future Work

- Automated Policy Discovery/Creation
- Automated Role and Permission Discovery
- Automatic Role Activation
- Explore Alternative Hierarchy Structures
- Explore Alternative Access Control Models
- Removal of the Master Attribute Authority
- Human Readable Attribute Names
- Searchable DMACPSABE
- DMACPSABE Based Signing
- Mobile Support
- Real World Implementation and Use

# Conclusions



# More Information

http://flash.lakeheadu.ca/~dservos/thesis

# Questions?