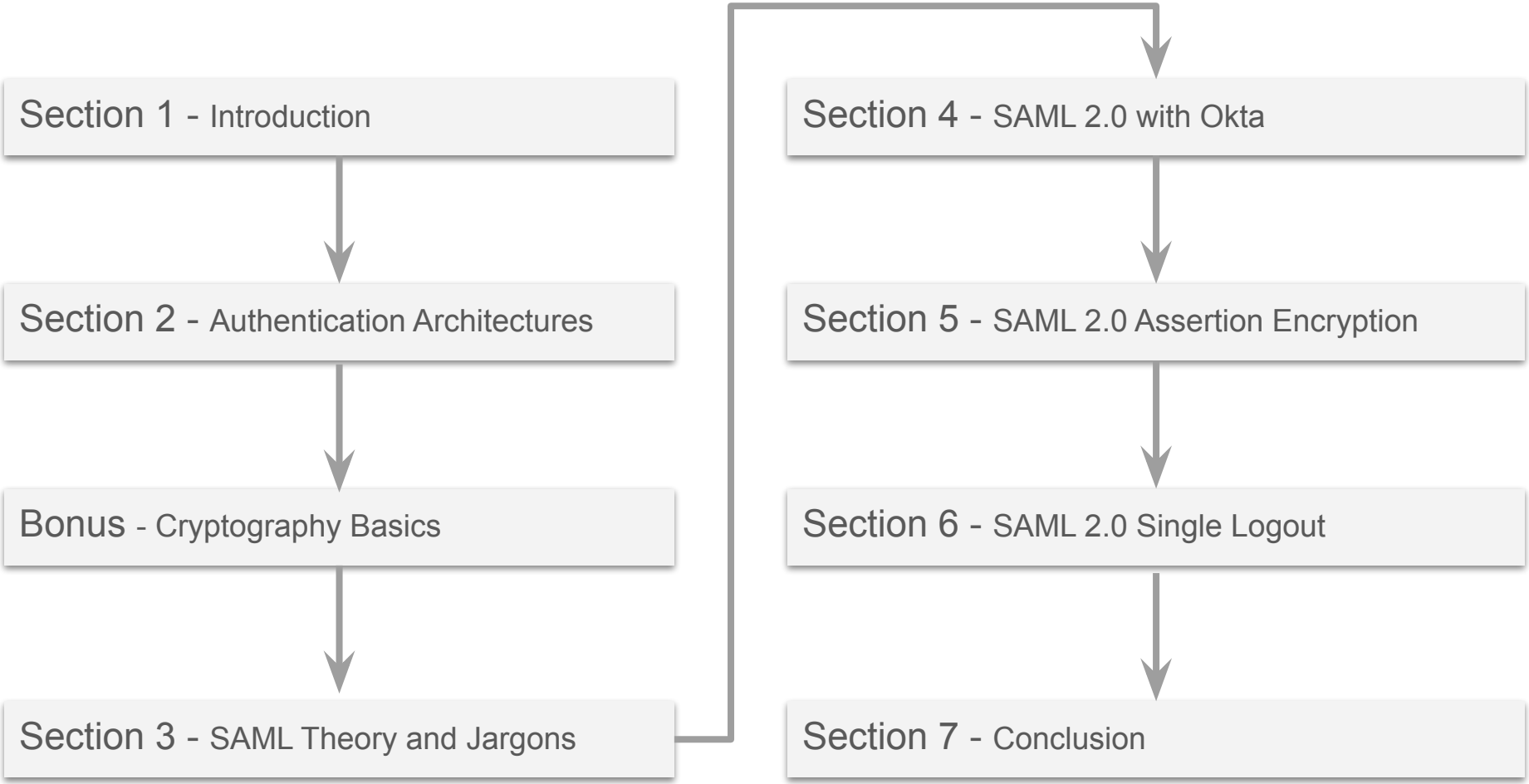
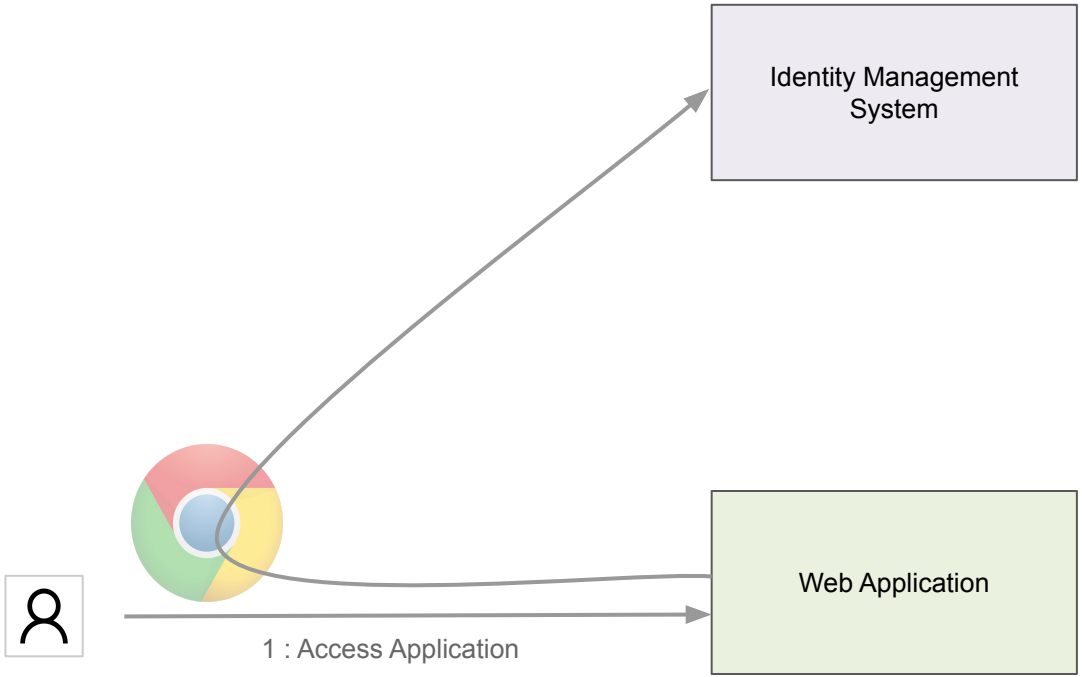
Course Content



Diagrams

The Private key and the Certificate are Base64 encoded so that they can be displayed as text.



Authentication Architectures

- Custom Security Architecture
  - Identity Provisioning, Authentication, Authorization
- Directory Services and LDAP
- LDAP Security Architecture
- Delegated Authentication

Custom Security Architecture

- Identity
- Authentication (AuthN)
- Authorization (AuthZ)
- Identity Provisioning

Business Logic

User Authentication

User Authorization

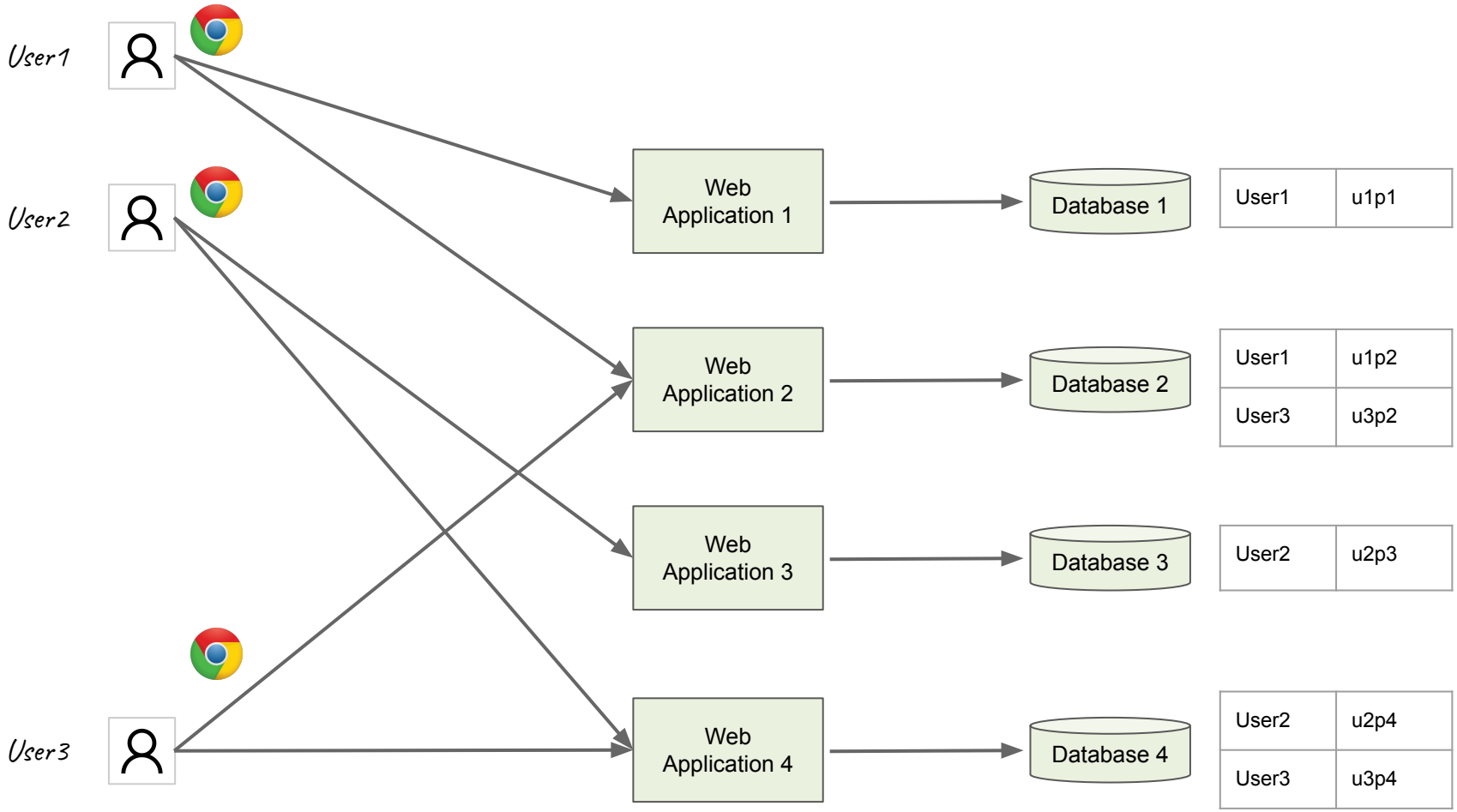
Identity Provisioning

Application Data

User Identity

User Roles

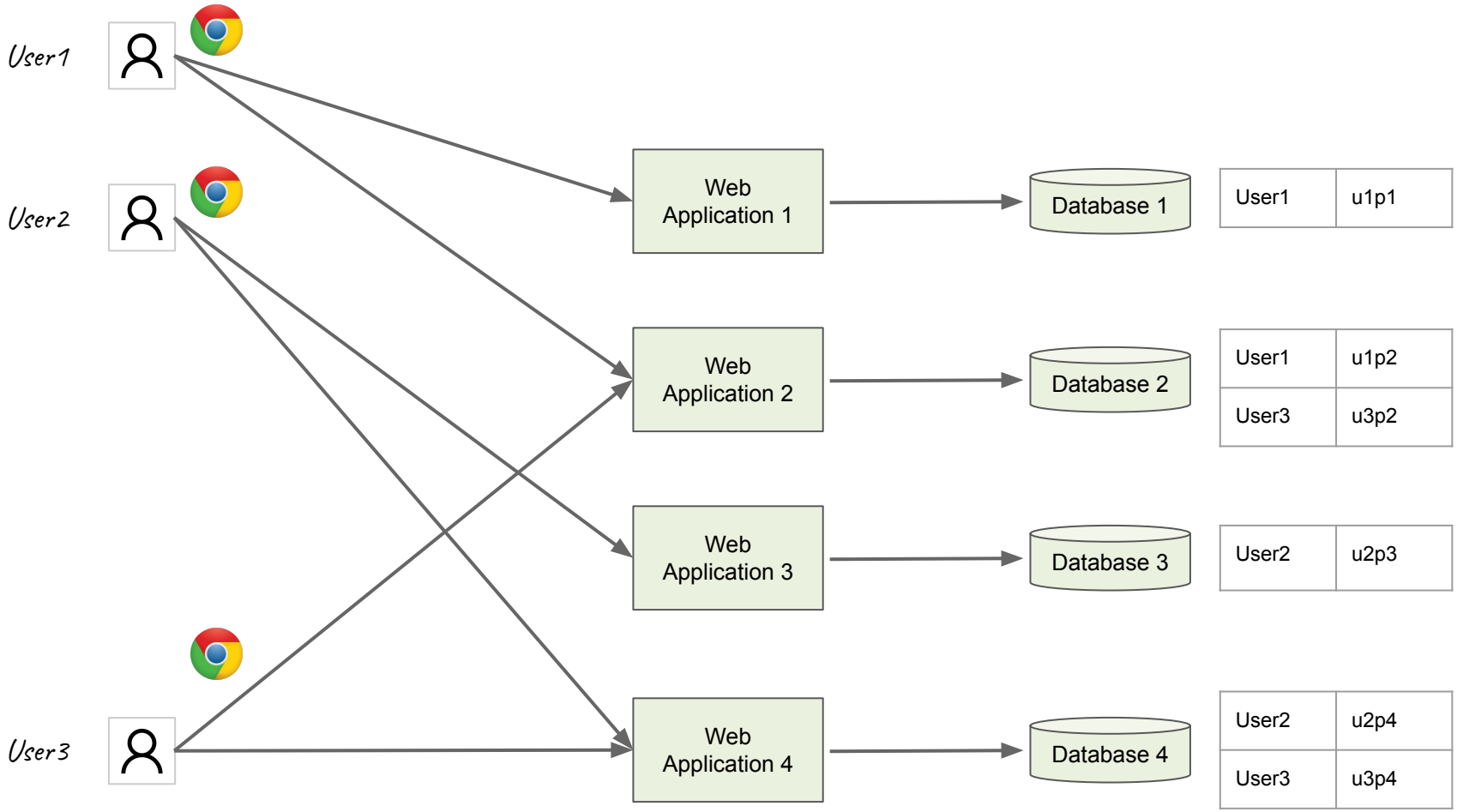
Custom Security Architecture Problems

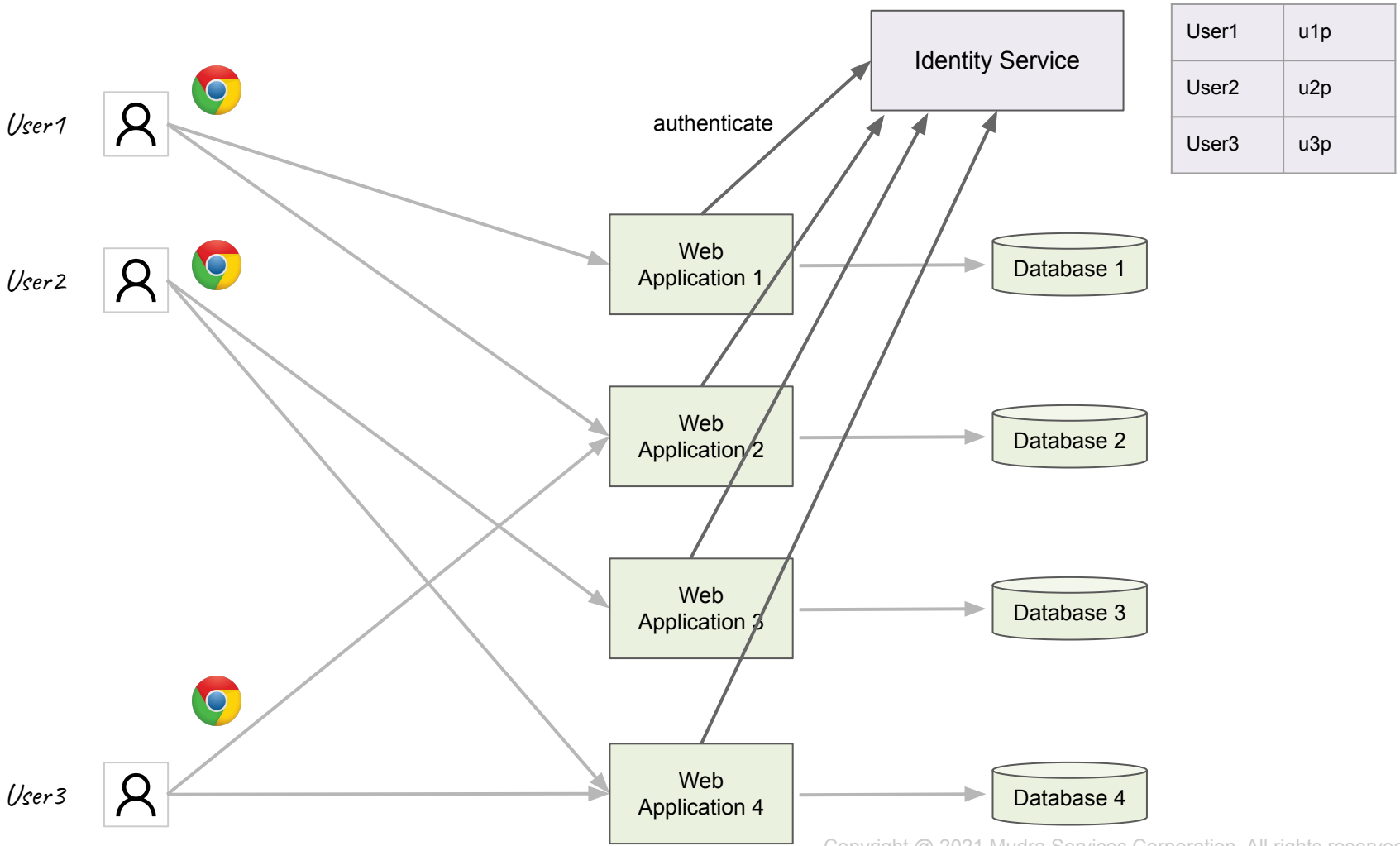


Custom Architecture - Problems

- User Identity is duplicated
- User has to remember too many passwords
- Application is responsible for User provisioning
- Provisioning and Deprovisioning of users is unmanageable
- No Single Sign-On
- Application responsible for Multi Factor Authentication (MFA)
- Credentials are sent to the Application (Security)

Common Identity





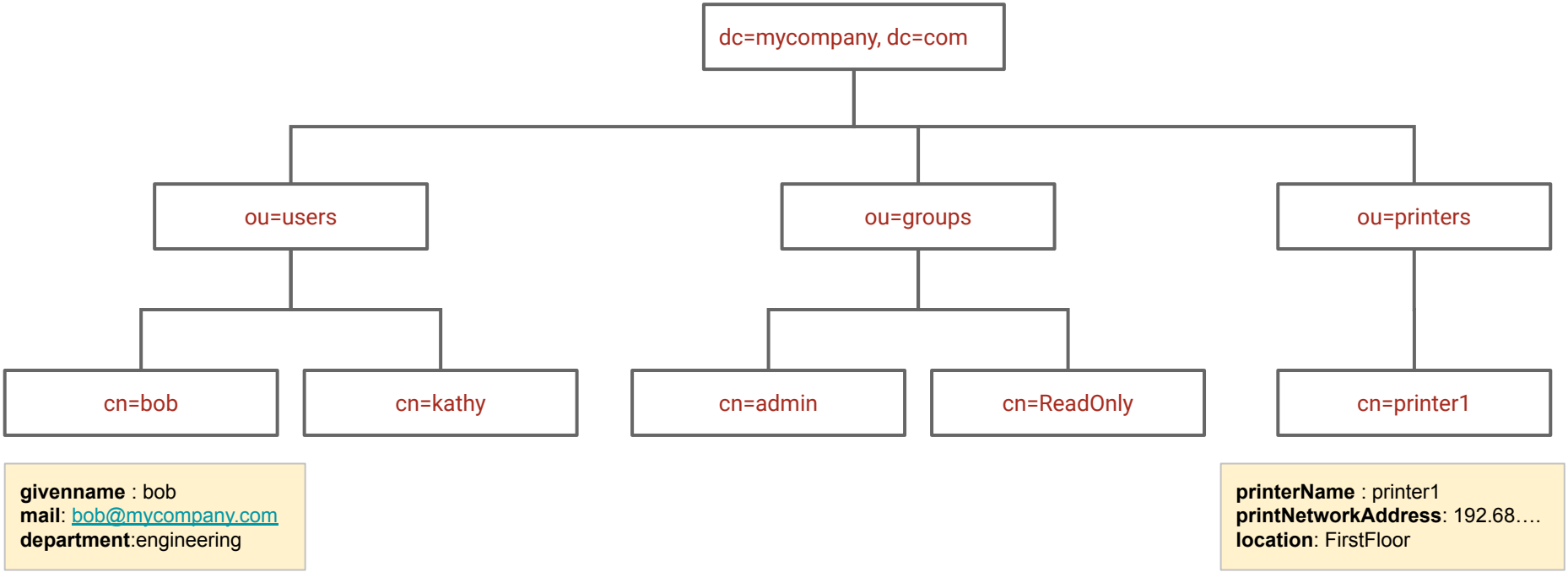
User and Group Data

- User Credentials
- First Name
- Last Name
- Email
- Phone Number
- Picture
- Groups (Application Roles can be mapped to this)
- Manager

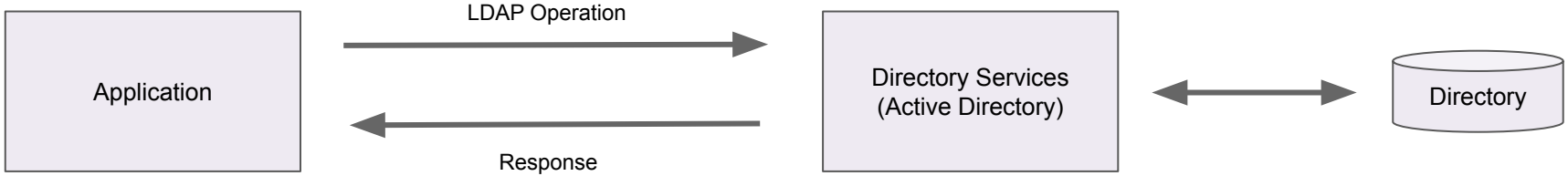
Lightweight Directory Access Protocol (LDAP)

- Directory Services
  - e.g Active Directory, Novell Directory Services
  - Entities (User, Group) stored in a tree structure
  - Fast Retrieval and Searches

- LDAP v3
  - Protocol to access Directory Services
  - Vendor neutral
  - All major languages have an implementation



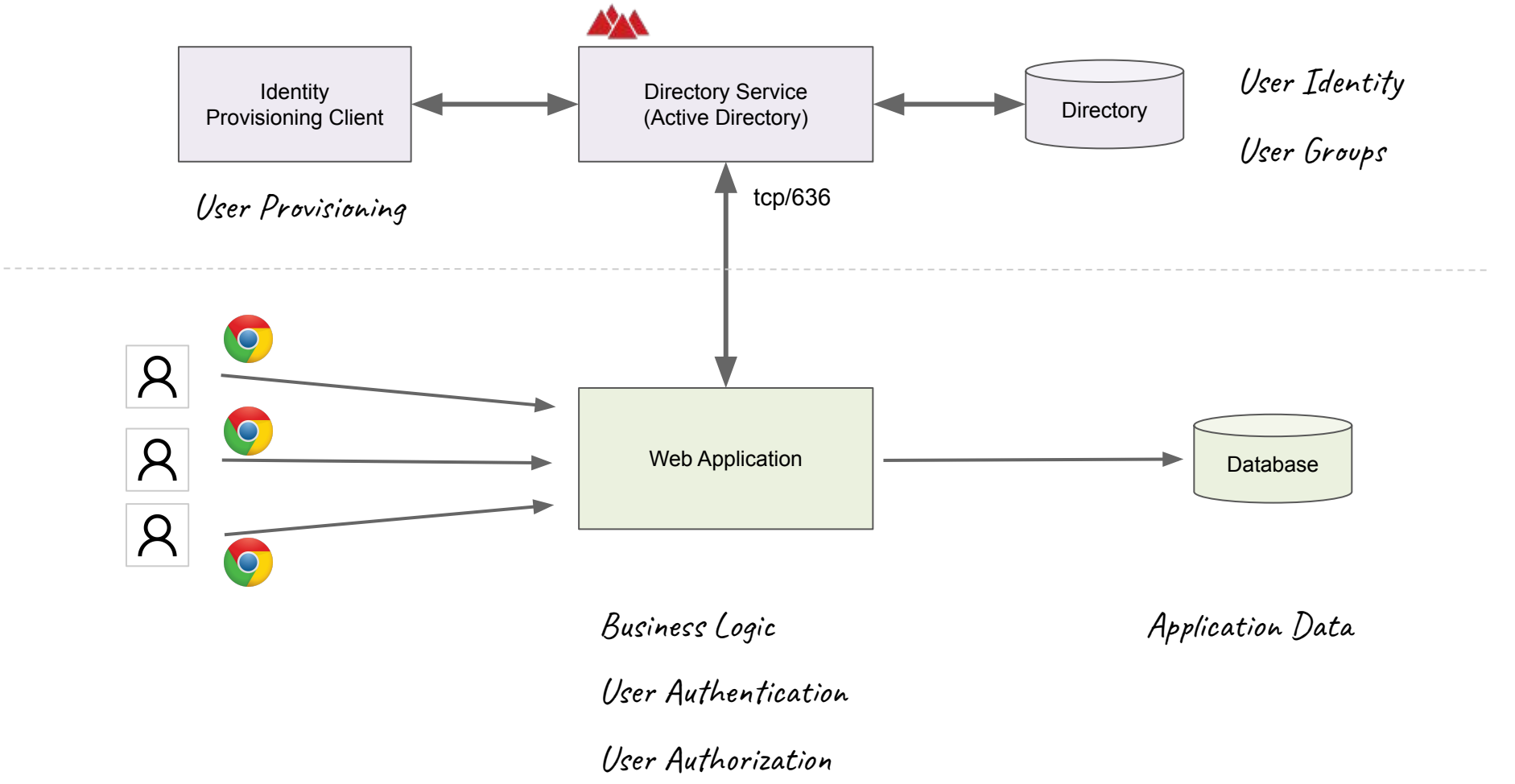
Distinguished Name (**DN**) for Bob	**cn**=bob,**ou**=users,**dc**=mycompany,**dc**=com
Distinguished Name (**DN**) for admin	**cn**=admin,**ou**=groups,**dc**=mycompany,**dc**=com
Distinguished Name (**DN**) for printer1	**cn**=printer1,**ou**=printers,**dc**=mycompany,**dc**=com

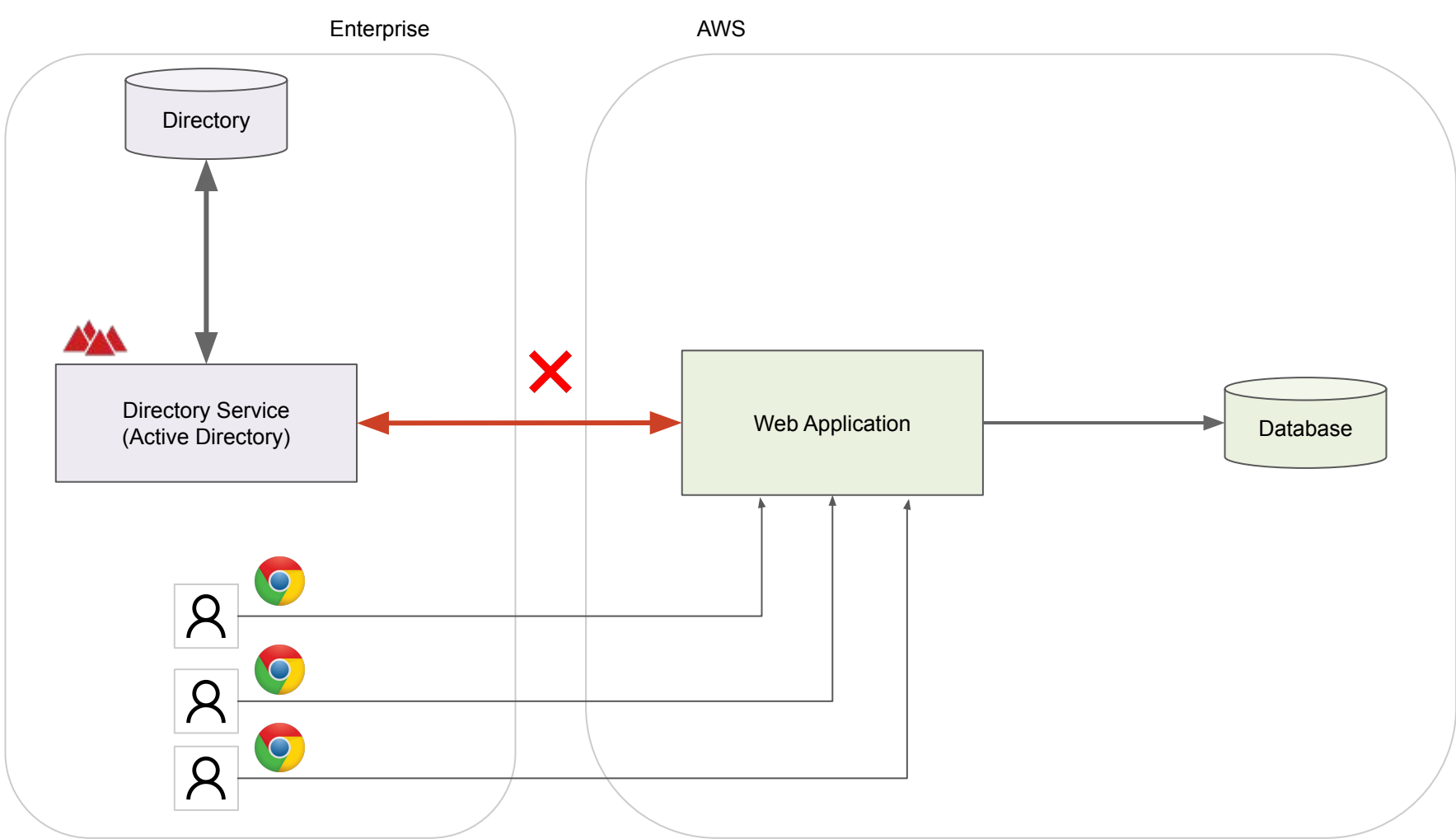


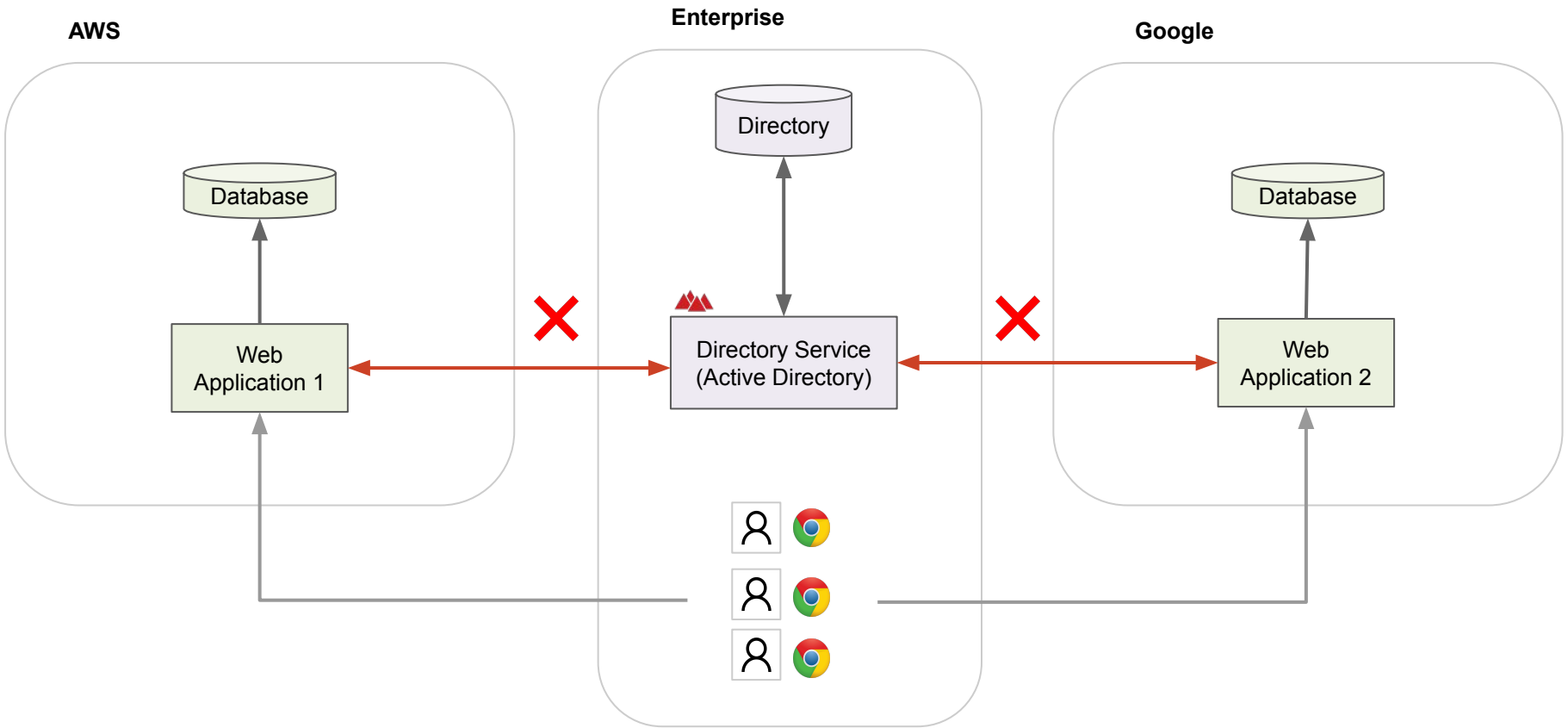
Operation	Explanation
Bind	Authenticate user
Search	Search for and/or retrieve directory entries
Add	Add a new Entry
Delete	Delete an Entry
Modify	Modify an Entry
Unbind	Close connection

Port	389
Port (TLS/SSL)	636
Protocol	TCP
Encoding	BER

LDAP Security Architecture



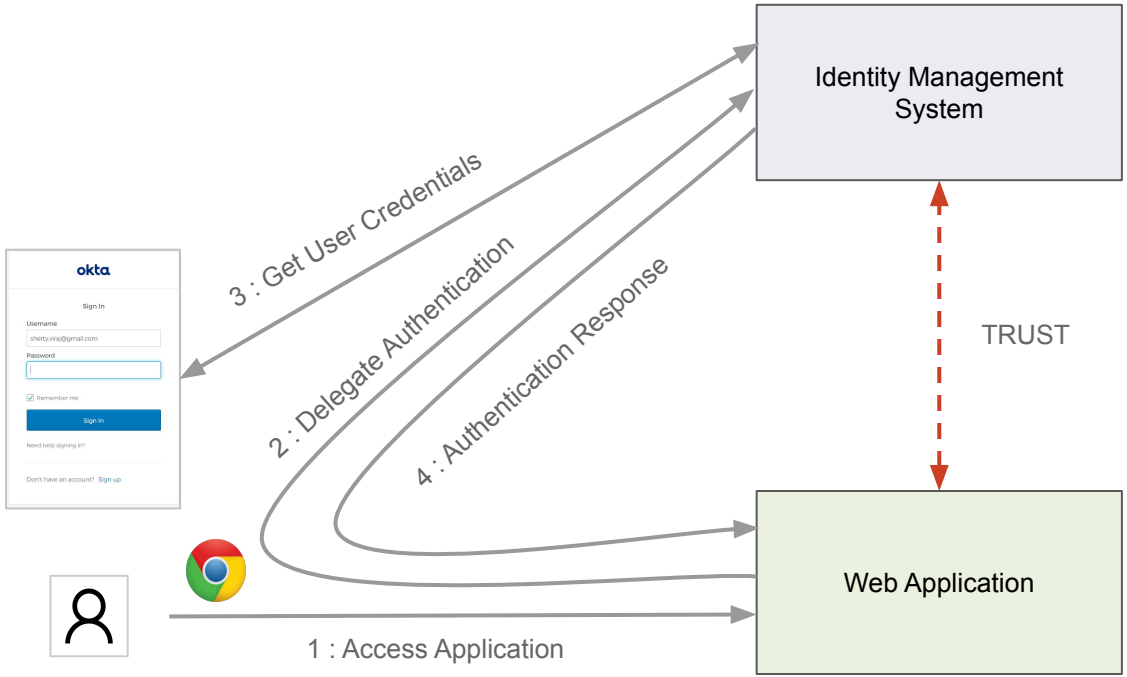




LDAP Architecture - Problems

- Credentials sent to the Application (Security)
- Applications and LDAP Server must be in the same security domain
- No Single Sign-On
- Application responsible for Multi Factor Authentication (MFA)

Delegated Authentication

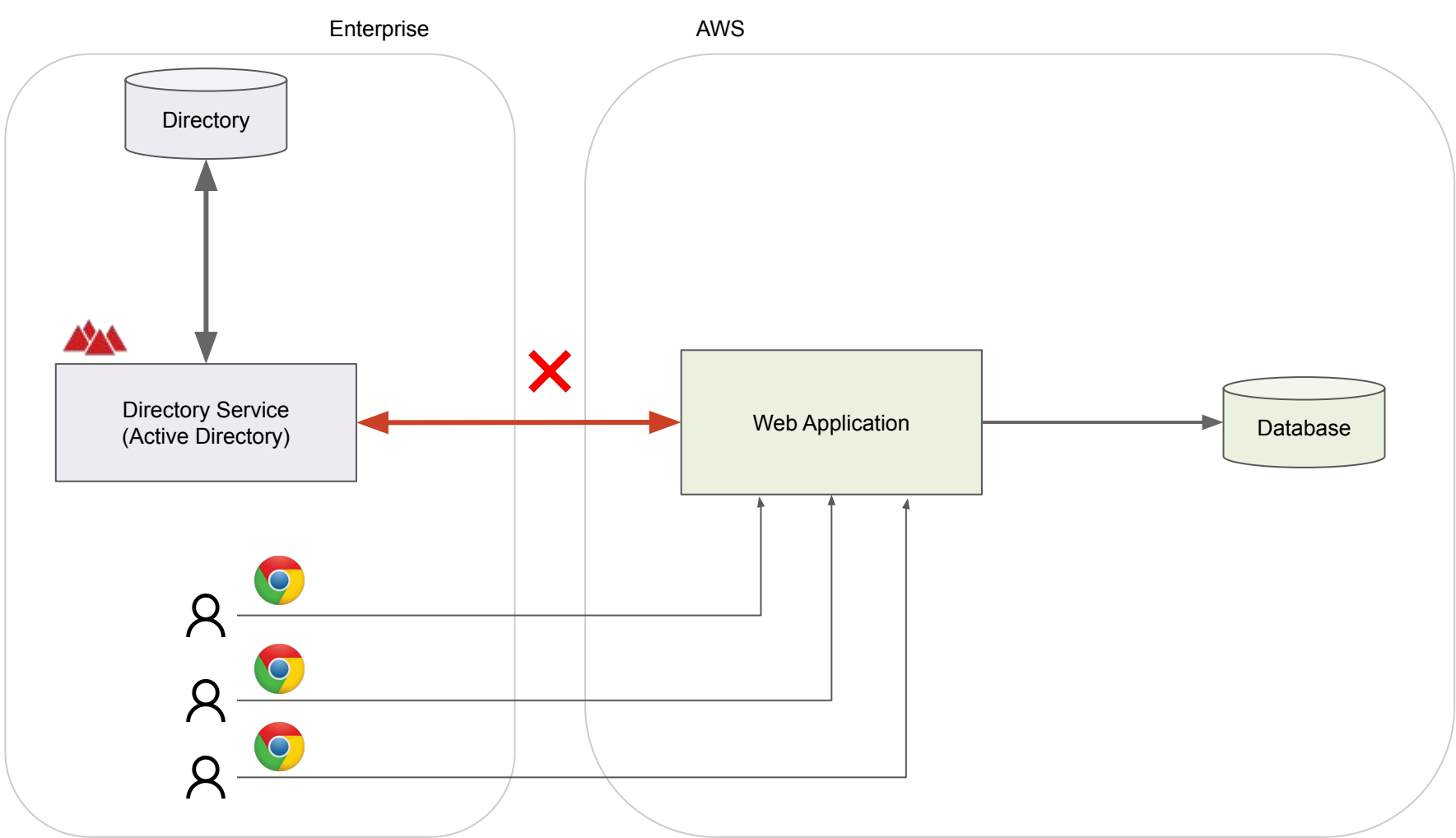


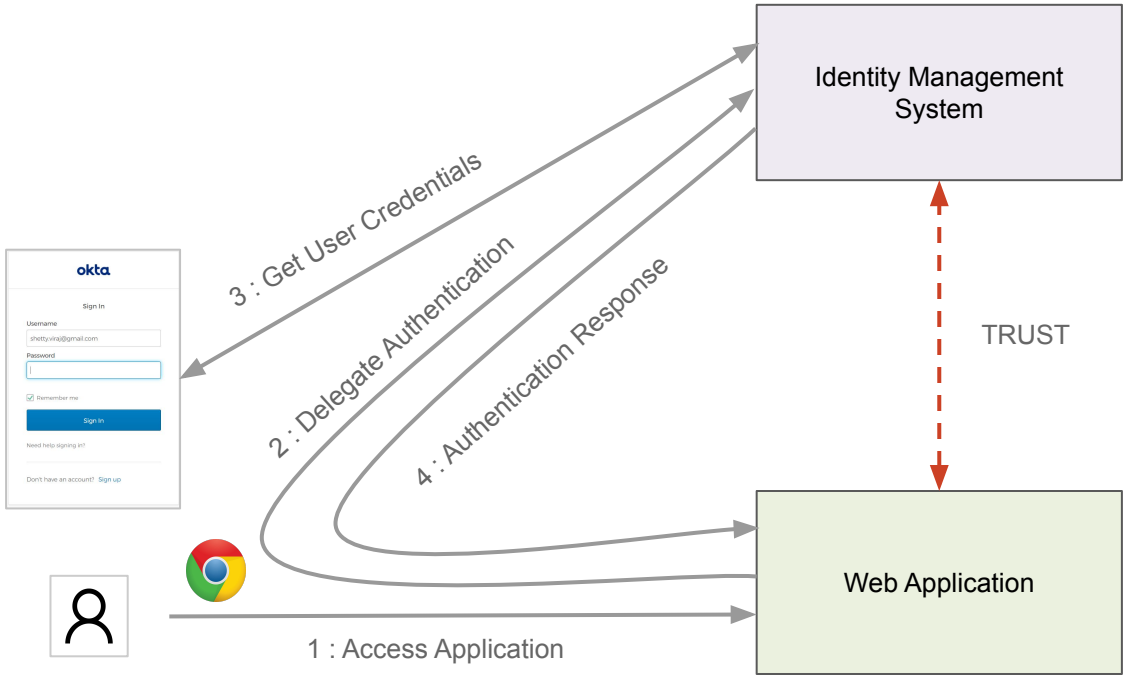
SAML 2.0 Single Sign On

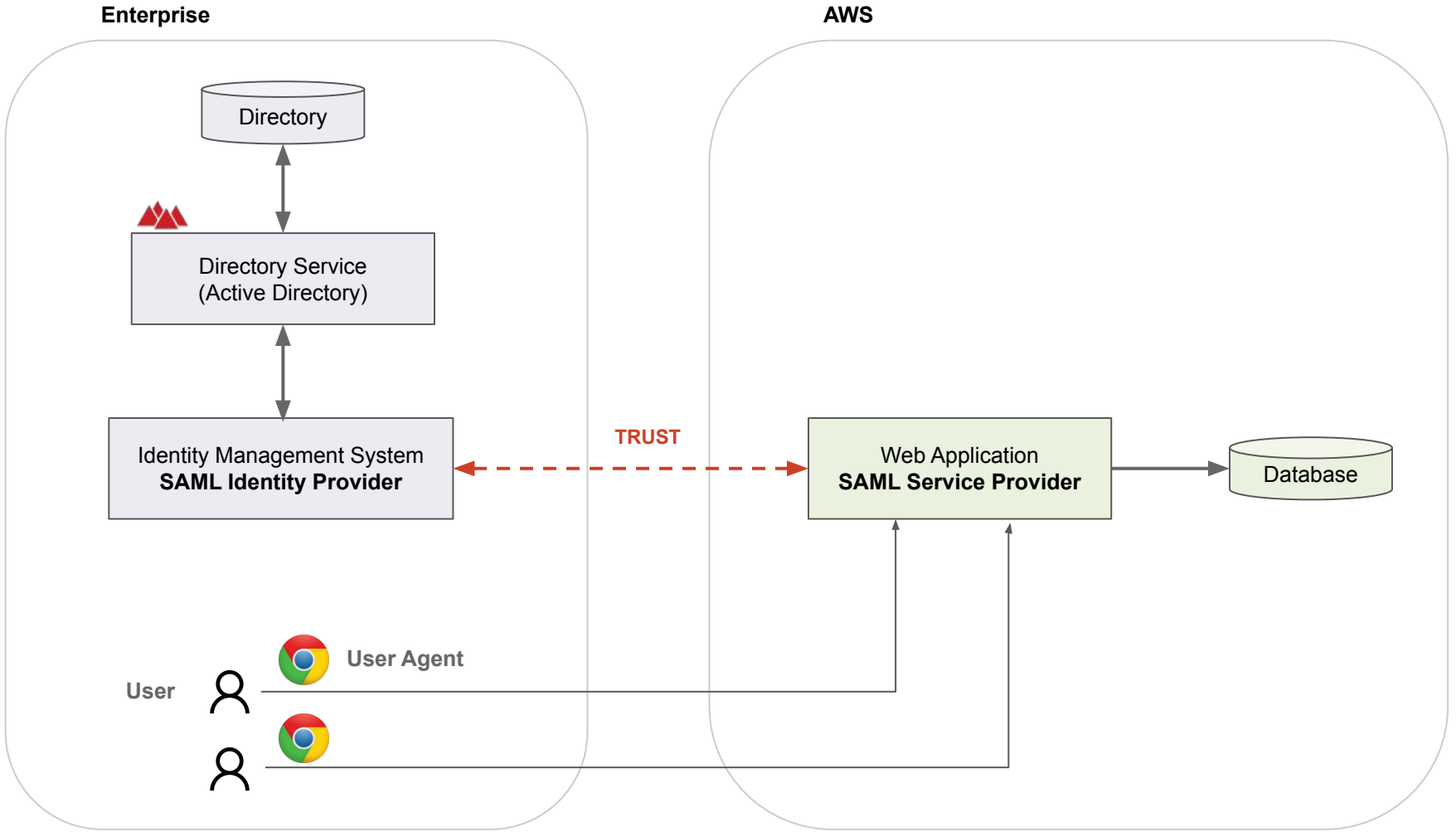
- SAML Theory
  - Identity Provider, Service Provider, Trust
  - SAML Metadata, SAML Assertion
  - Protocol Binding

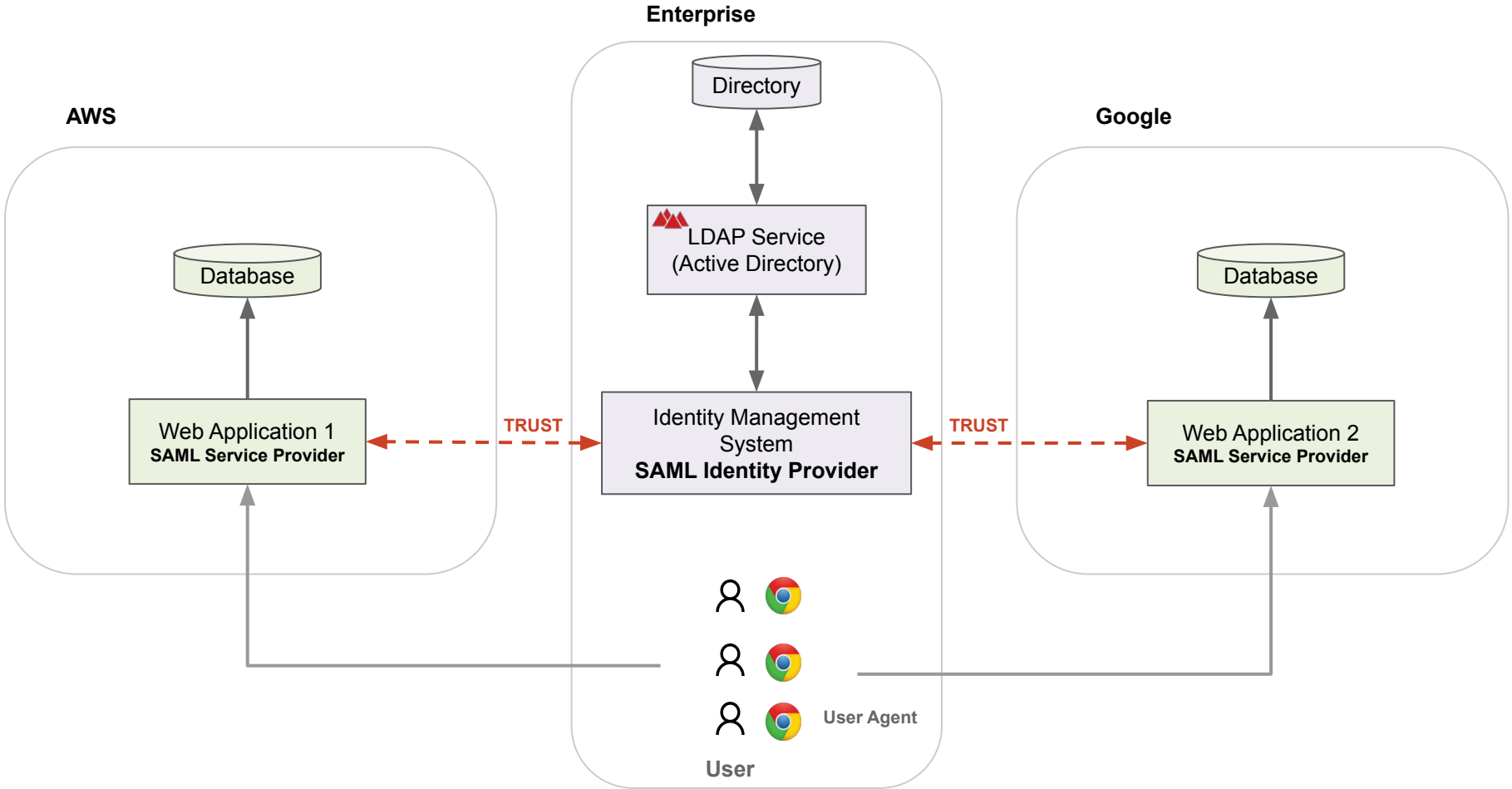
- SAML Authentication Flow
  - SP Initiated SAML Single Sign On
  - IDP Initiated SAML Single Sign On

Introducing SAML 2.0
(Security Assertion Markup Language)









SAML Authentication Flow

Enterprise/Other Cloud

AWS

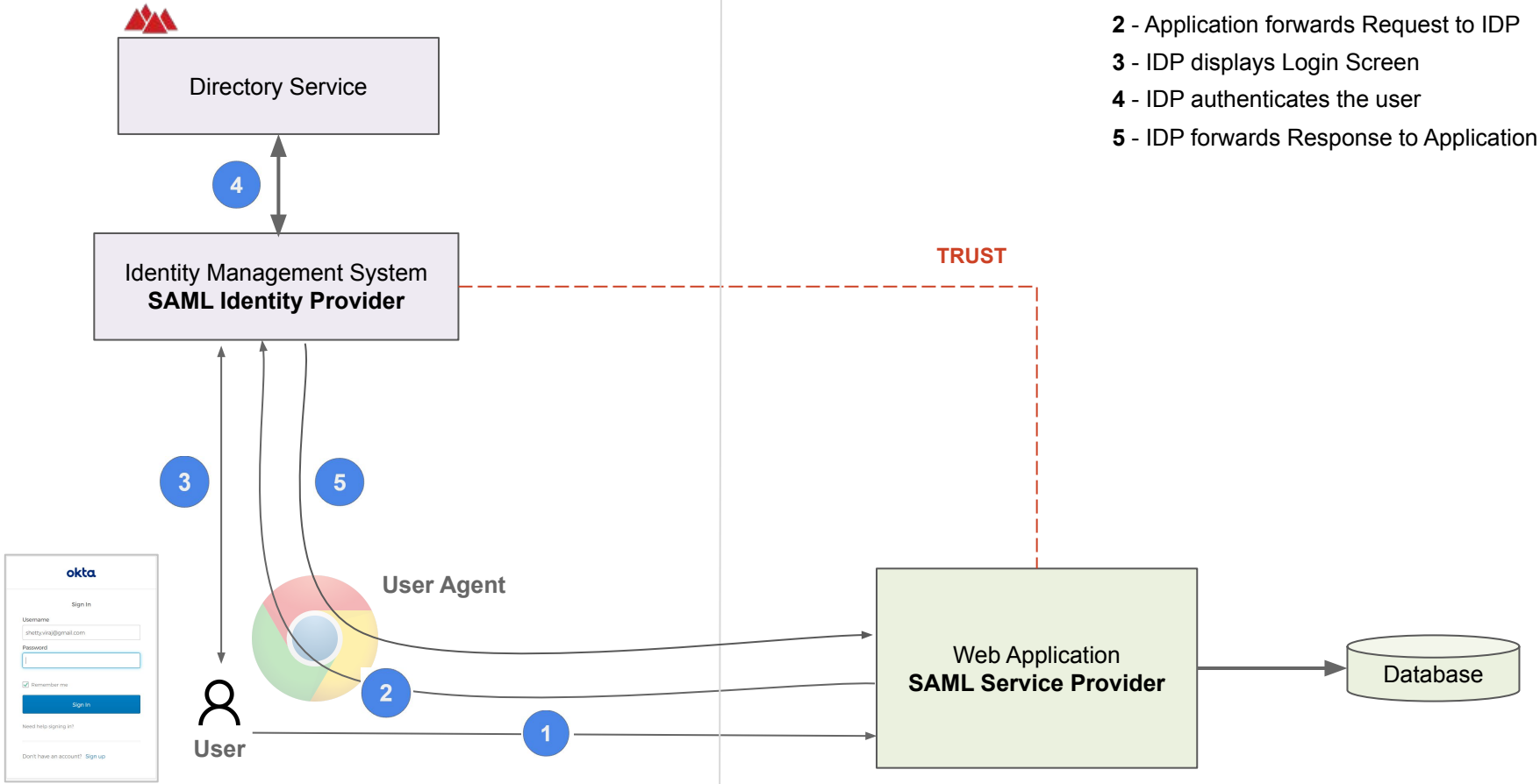
1 - User Accesses Application

2 - Application forwards Request to IDP

3 - IDP displays Login Screen

4 - IDP authenticates the user

5 - IDP forwards Response to Application



SP Initiated SSO Flow

Enterprise/Other Cloud

SP INITIATED SAML SSO

AWS

Directory Service

4

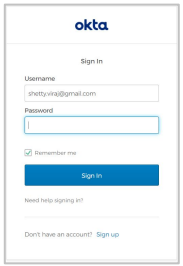
Entity ID

Identity Management System
SAML Identity Provider

TRUST

3

5



User Agent

User

2

1

Entity ID

Web Application
SAML Service Provider

Database

Relying Party

1 - User Accesses Application
2 - Application forwards Request to IDP
3 - IDP displays Login Screen
4 - IDP authenticates the user
5 - IDP forwards Response to Application

IDP INITIATED SAML SSO

Enterprise/Other Cloud

AWS

Directory Service

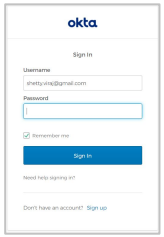
2

Identity Management System
SAML Identity Provider

TRUST

1

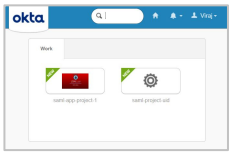
3



User Agent

4

User



5

Web Application
SAML Service Provider

Database

1 - User Accesses and logs into IDP

2 - IDP authenticates the user

3 - IDP displays User Dashboard

4 - IDP forwards Response to Application

5 - Application displays Page

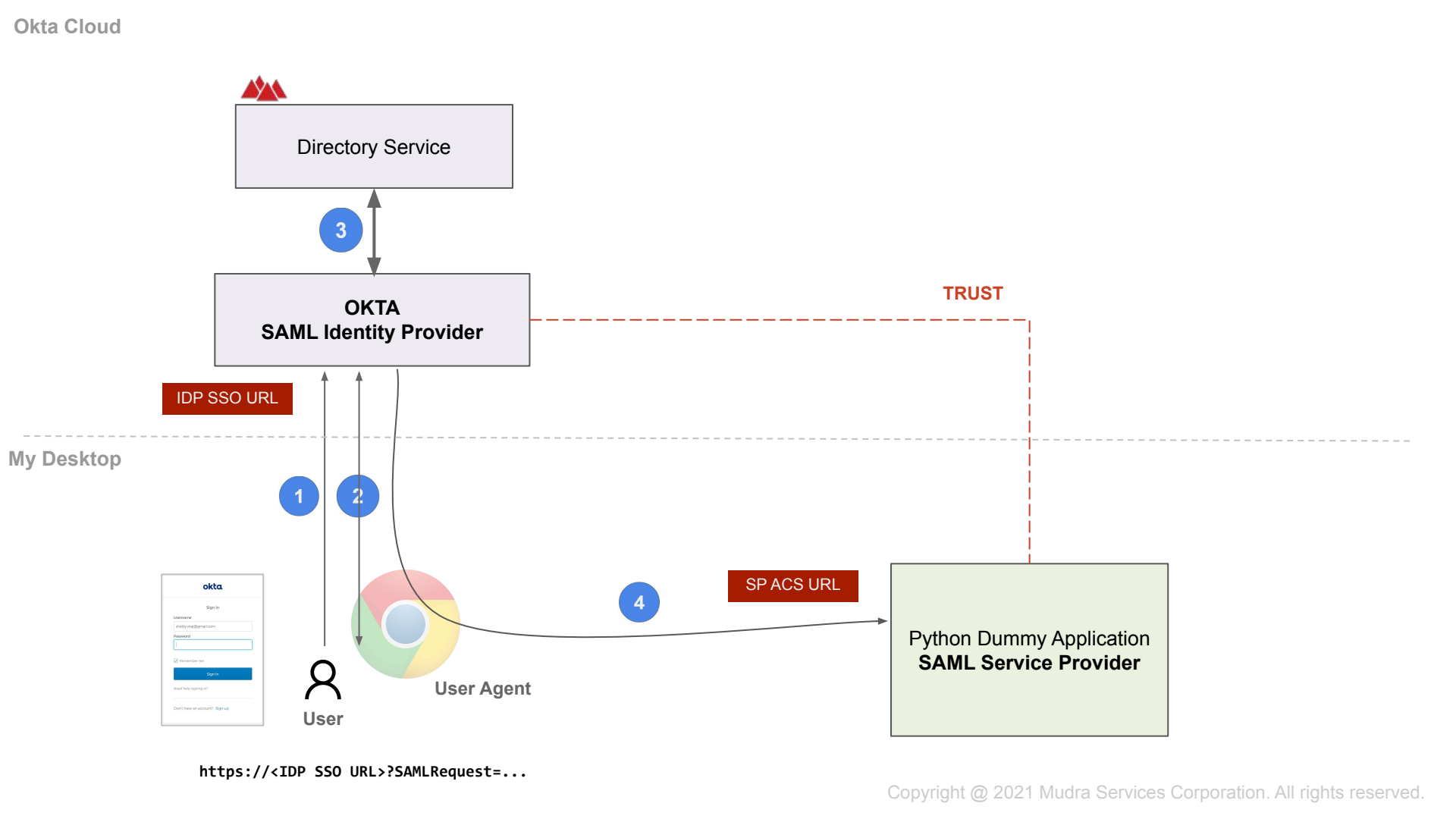
Hands On SAML 2.0 with Okta

- Okta SAML Setup for Project 1
- Hands-On SAML Experimentation with Okta
  - Construct XML SSO Request
  - Analyze XML SSO Response
- Signing Assertion and Response
- SAML Debugging
  - Google Chrome
  - SAML Devtools Extension
  - SAML Developer Tools Website

Okta Setup

Project 1 Explanation

- Use Okta Identity provider
- Use Dummy Python Application
- Construct SAML Request
- Inspect the Signed SAML Response
- No Encryption of SAML Response
- No SAML Groups



Enterprise/Other Cloud

AWS

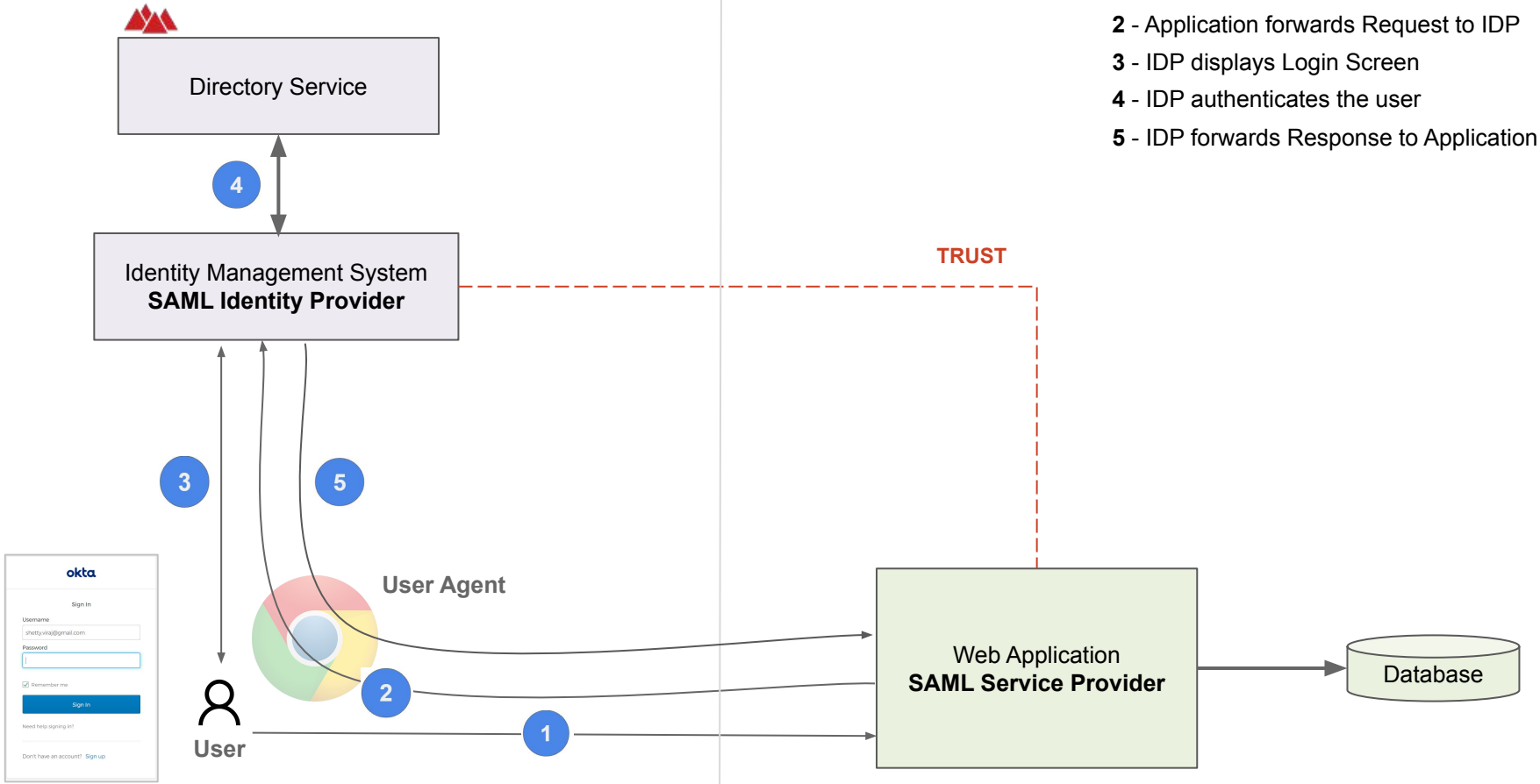
1 - User Accesses Application

2 - Application forwards Request to IDP

3 - IDP displays Login Screen

4 - IDP authenticates the user

5 - IDP forwards Response to Application



Project 1

SAML 2.0 Assertion Encryption

- Okta SAML Setup for Project 2
- Hands-On SAML Experimentation with Okta
  - Construct XML SSO Request
  - Analyze XML SSO Response
- Signing Assertion and Response
- Assertion Encryption
  - Generate RSA Keys and Certificate
- Handling custom user and group claims

Project 2 Explanation

- Run Dummy Python Application at port 8081
- New user field UID
- New Groups and associate with users
- Add encryption to Response
- Inspect the Signed/Encrypted SAML Response

Okta Cloud

Project 2

Directory Service

3

OKTA
SAML Identity Provider

TRUST

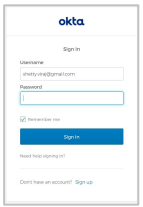
IDP SSO URL

My Desktop

1

2

➔ Encrypted Assertion
➔ Group Claims



User

User Agent

4

SP ACS URL

Python Dummy Application
SAML Service Provider

8081

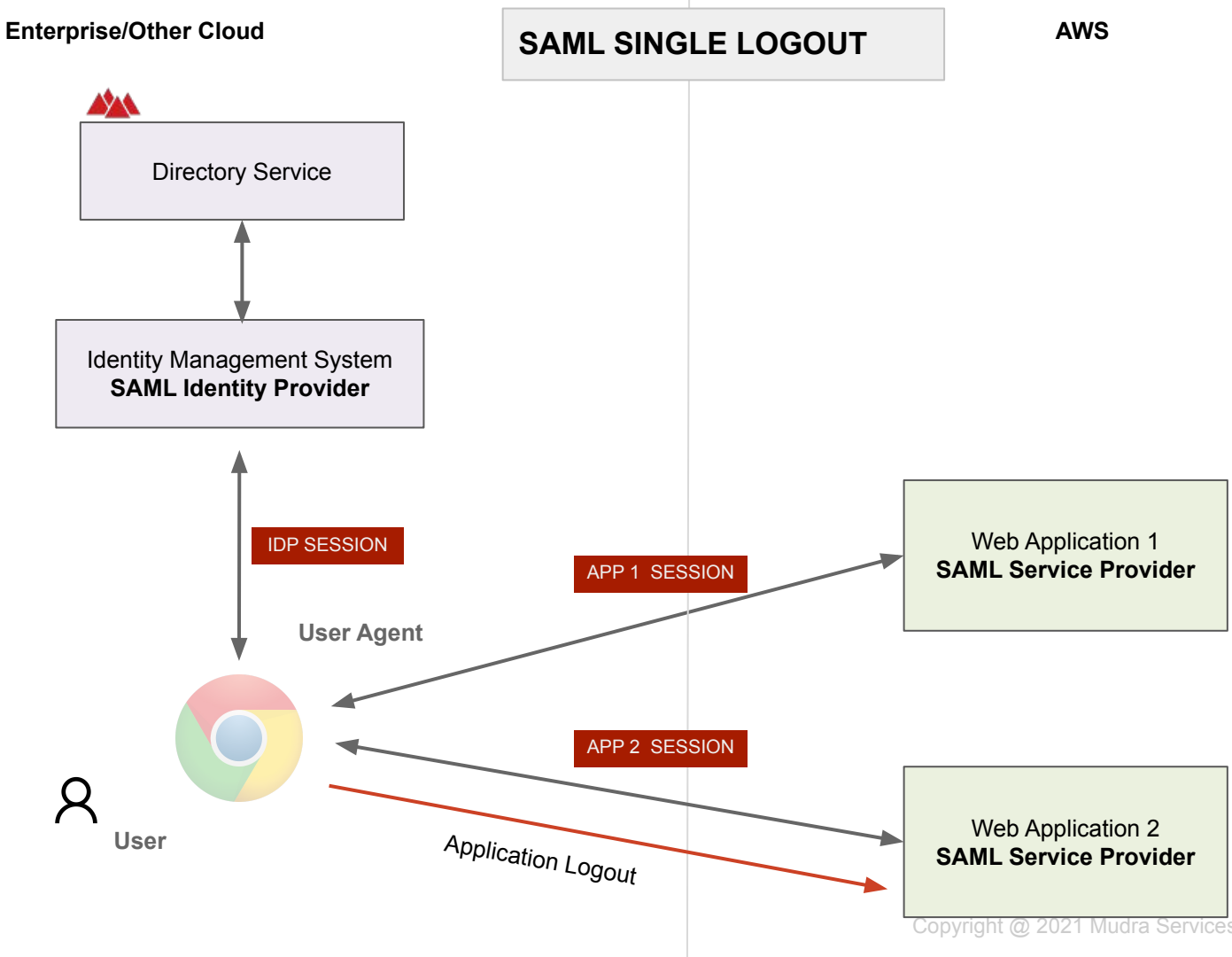
`https://<IDP SSO URL>?SAMLRequest=...`

Project 2

SAML 2.0 Single Logout

- Okta SAML Setup
- Hands-On SAML Experimentation with Okta
  - Construct XML Logout Request
  - Analyze XML Logout Response
- HTTP Sessions
- Okta User Provisioning
- Okta MFA and SAML

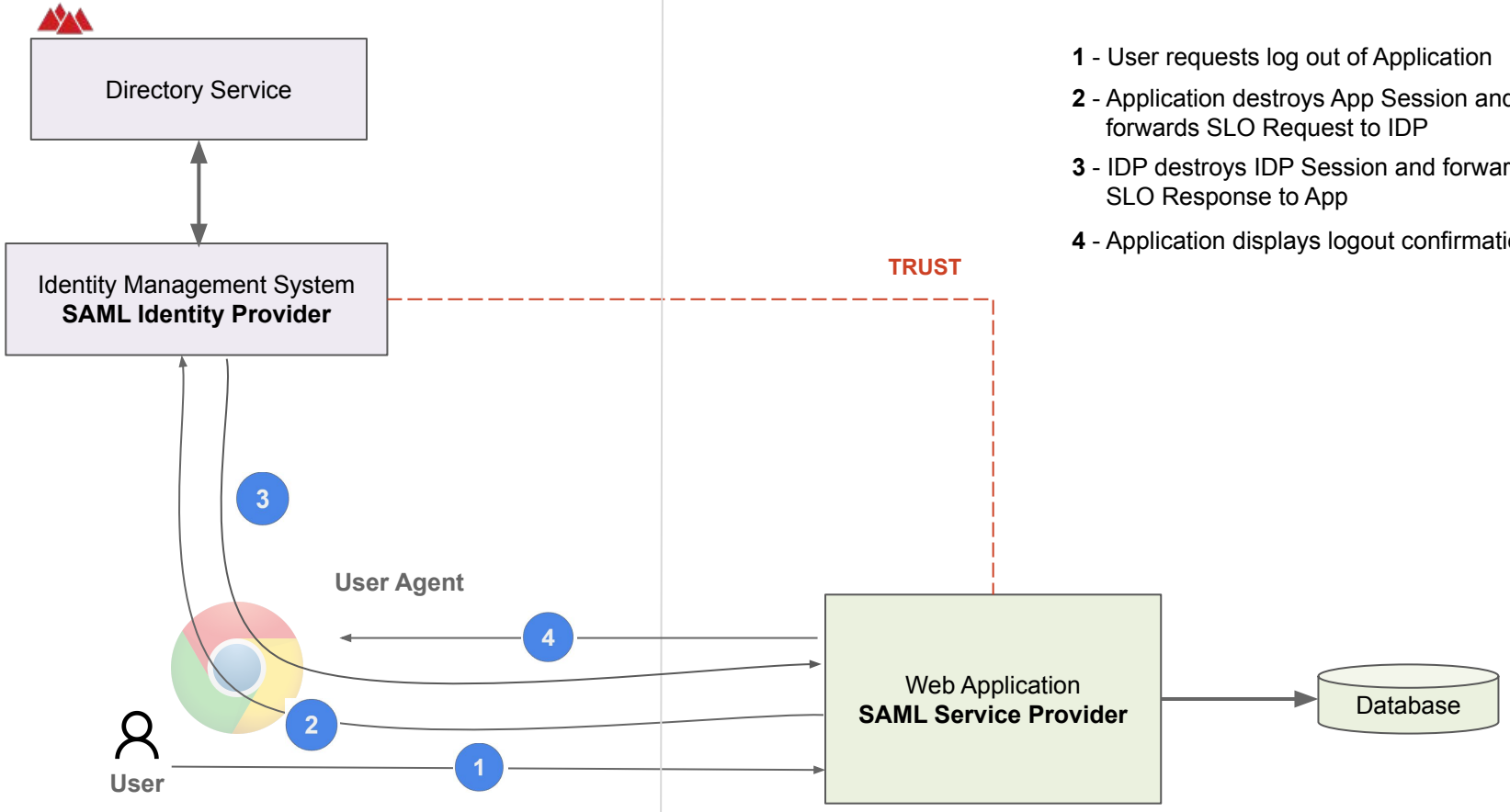
SAML Single Logout



Enterprise/Other Cloud

SP INITIATED SAML SLO

AWS



1 - User requests log out of Application

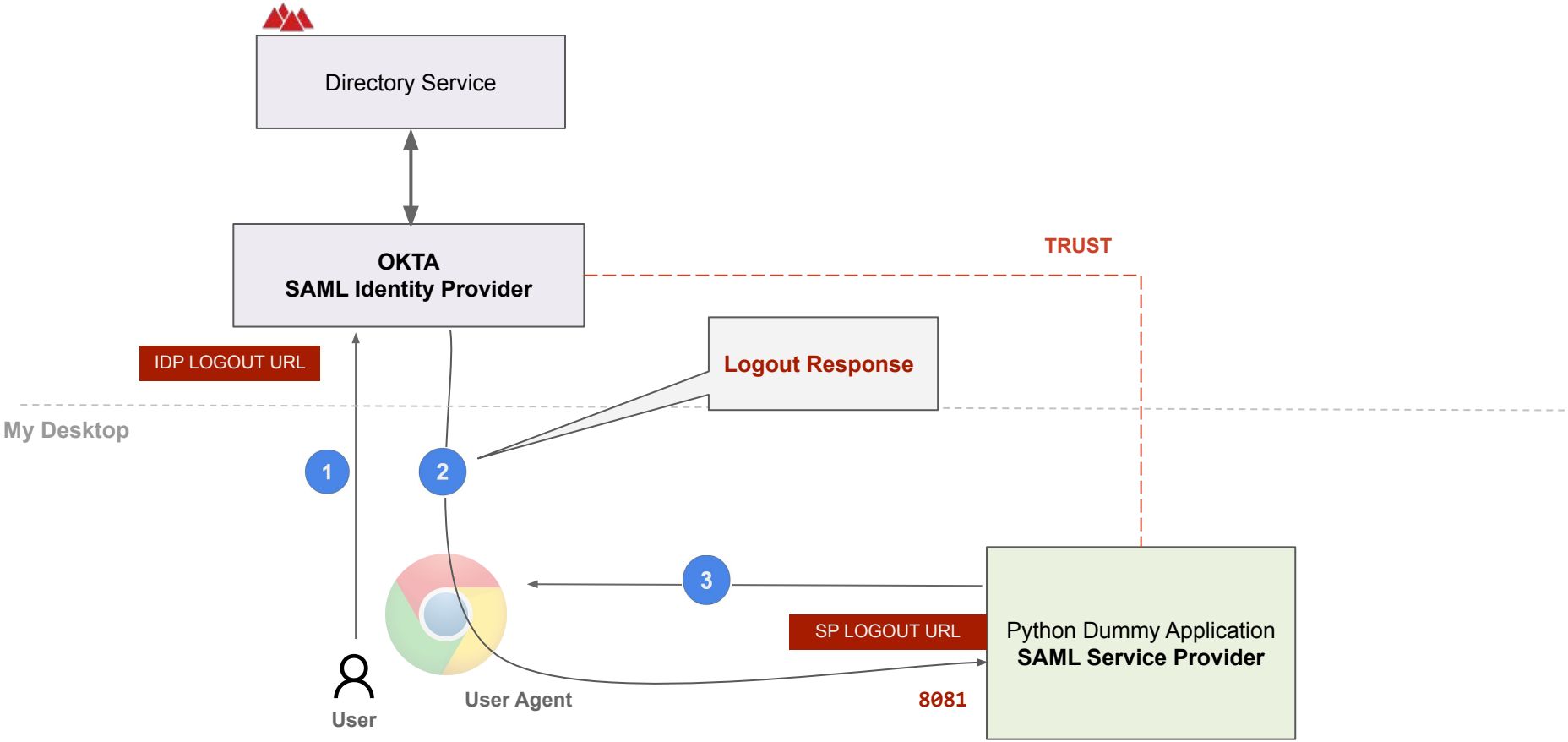
2 - Application destroys App Session and forwards SLO Request to IDP

3 - IDP destroys IDP Session and forwards SLO Response to App

4 - Application displays logout confirmation

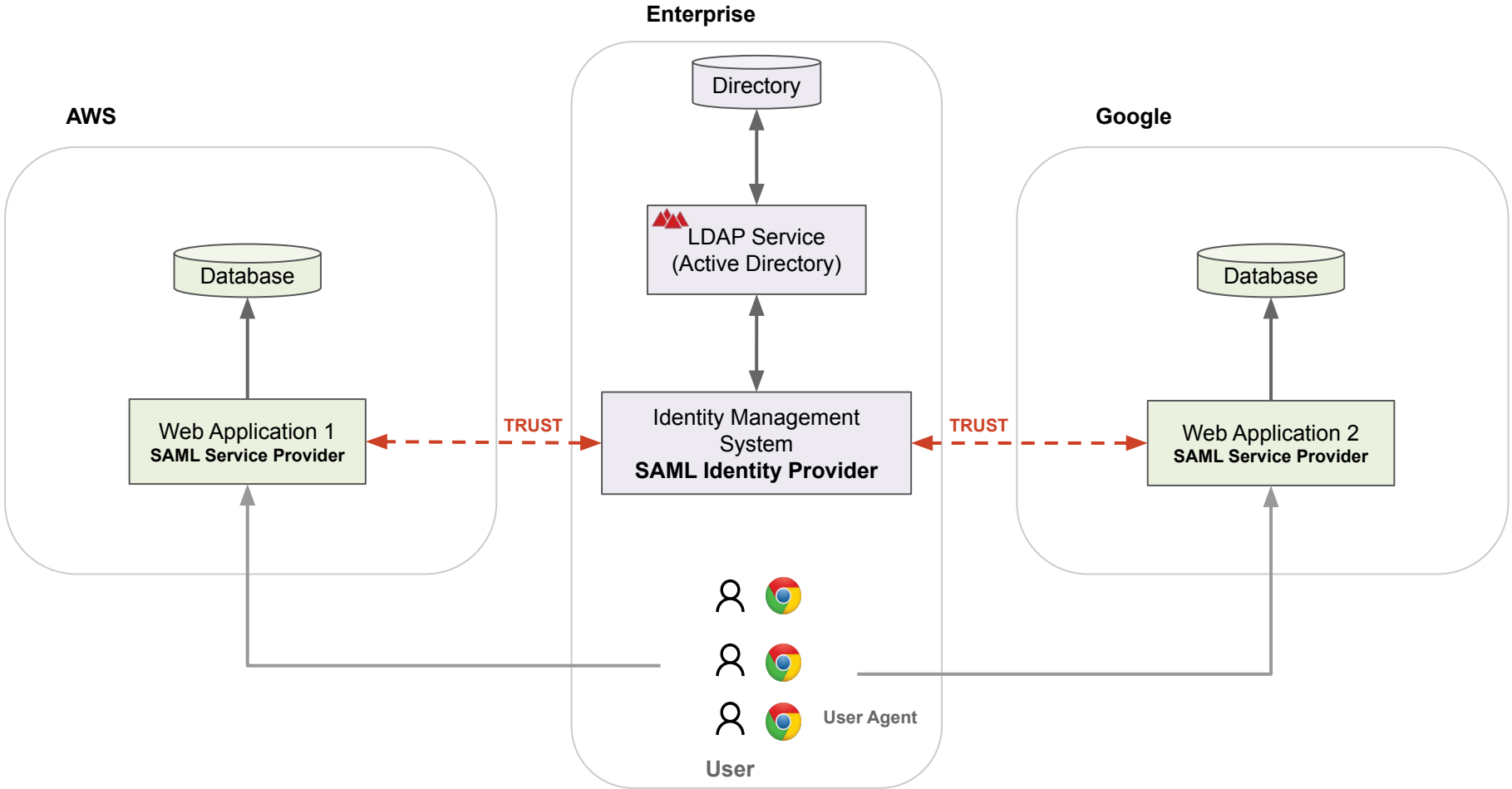
Okta Cloud

Project 2



`https://<IDP LOGOUT URL>?SAMLRequest=...`

User Provisioning



SAML 2.0 and Programming

Enterprise/Other Cloud

AWS

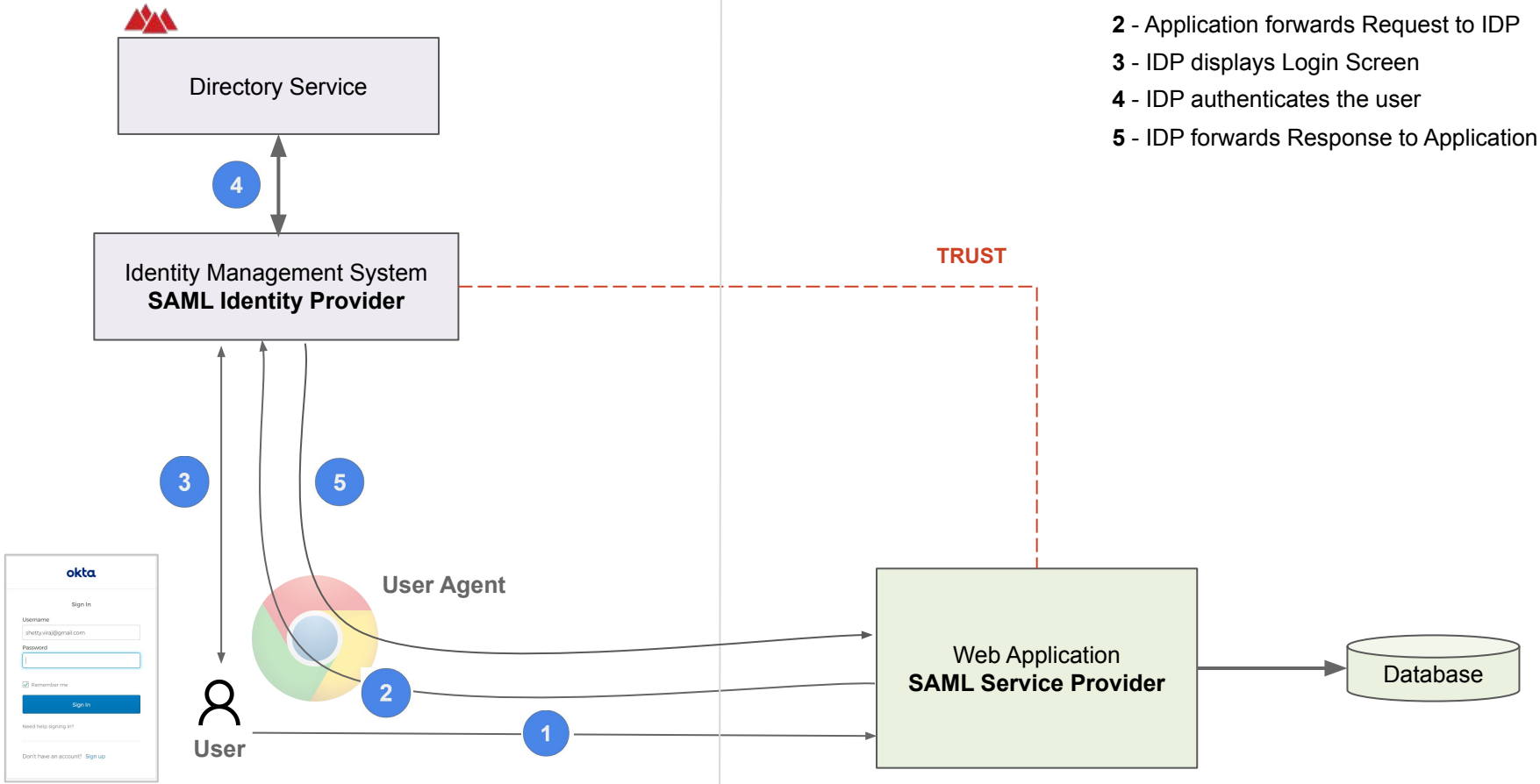
1 - User Accesses Application

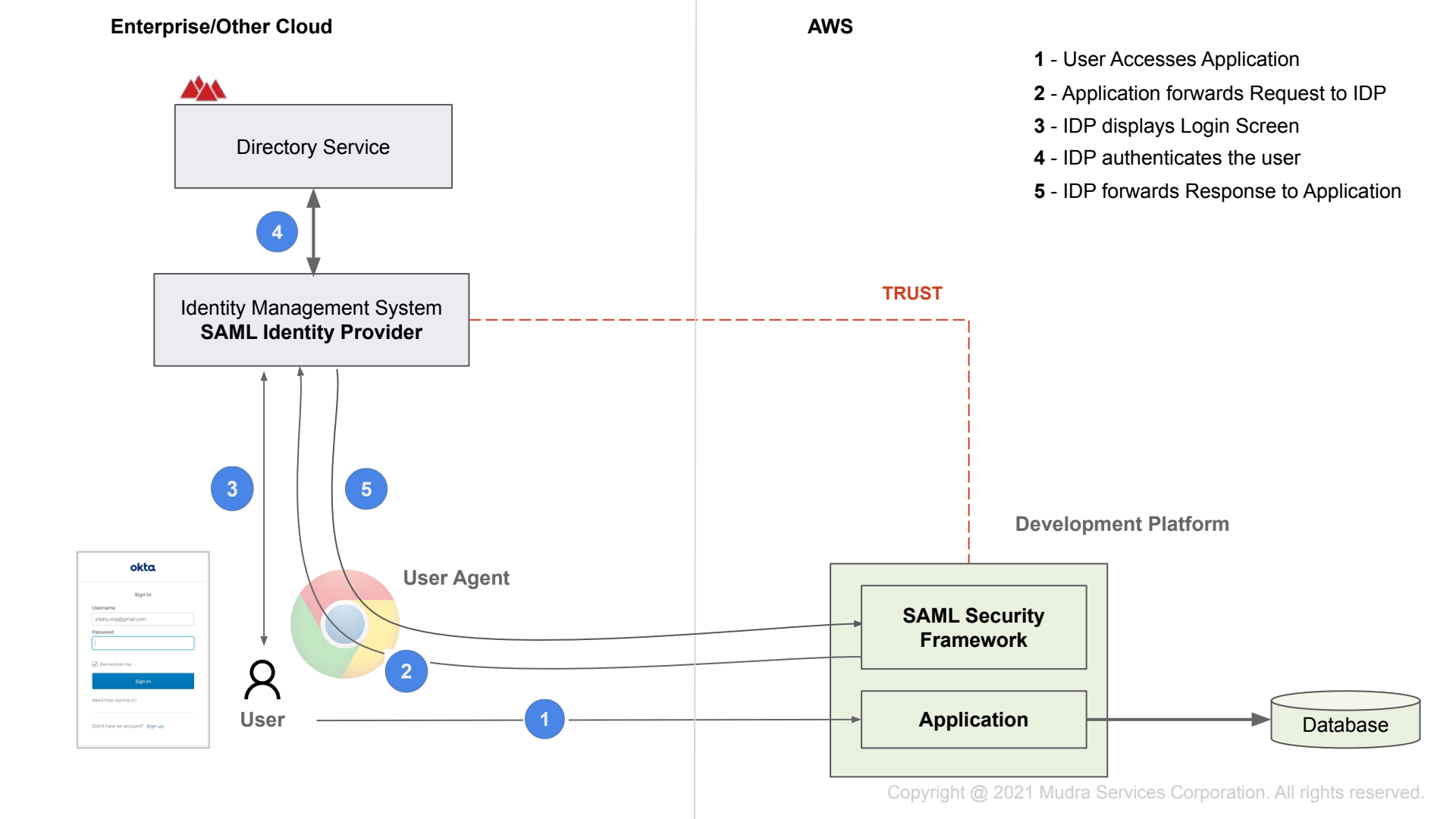
2 - Application forwards Request to IDP

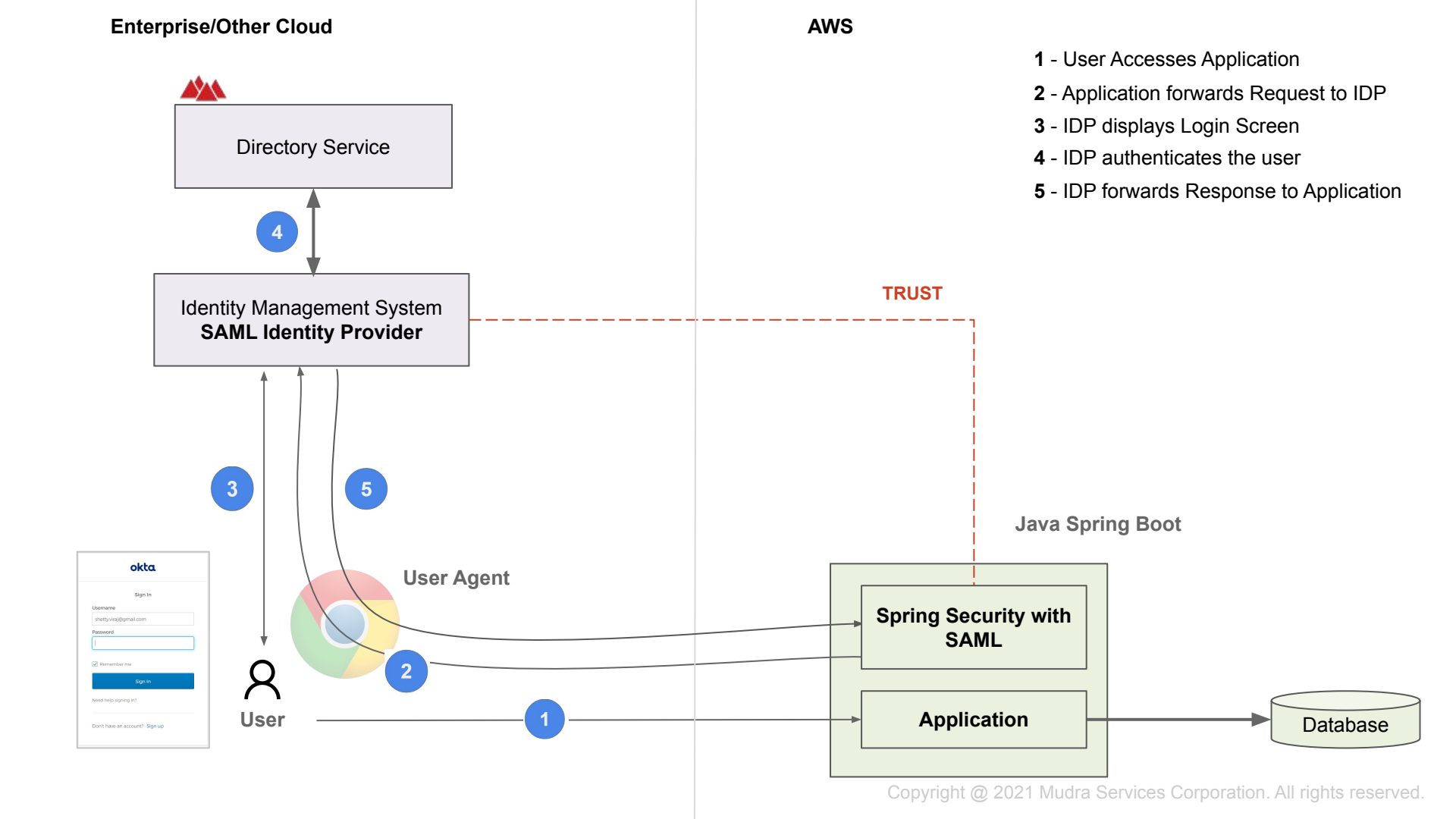
3 - IDP displays Login Screen

4 - IDP authenticates the user

5 - IDP forwards Response to Application




Enterprise/Other Cloud
AWS
1 - User Accesses Application
2 - Application forwards Request to IDP
3 - IDP displays Login Screen
4 - IDP authenticates the user
5 - IDP forwards Response to Application
Directory Service
4
Identity Management System
SAML Identity Provider
TRUST
3
5
Development Platform
okta
Sign In
Username
Password
Remember me
Sign In
Need help signing in?
Don't have an account? Sign up
User Agent
SAML Security Framework
2
User
1
Application
Database
Copyright @ 2021 Mudra Services Corporation. All rights reserved.

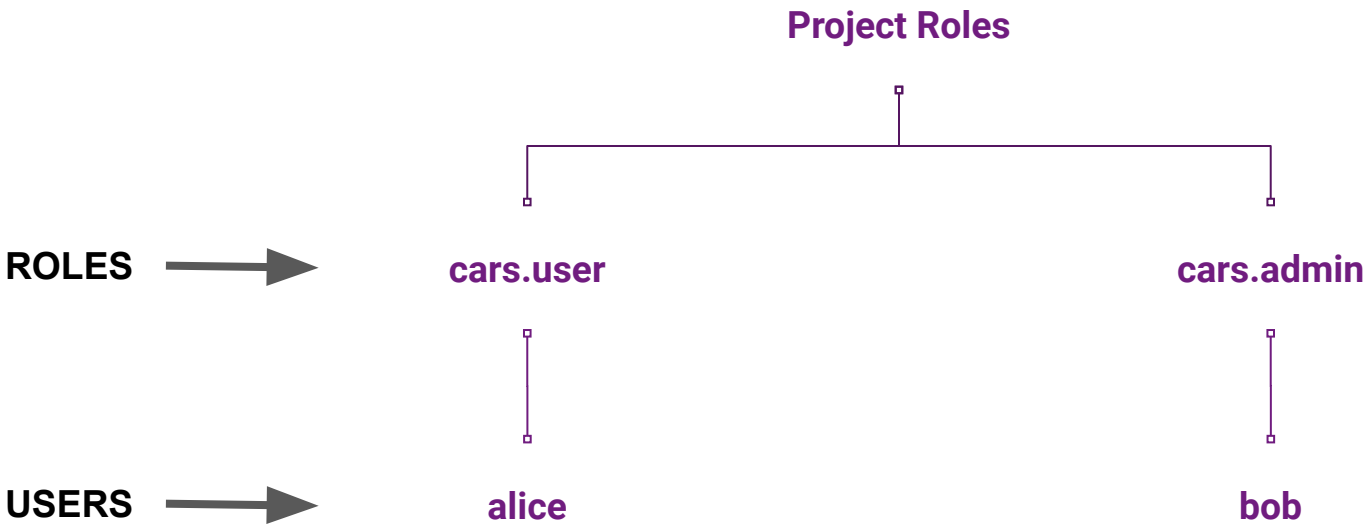


Cryptography Basics

- URL Encoding
- Base64 Encoding
- Hashing (SHA-512, SHA-256)
- Symmetric Encryption (AES, Blowfish)
- Asymmetric Encryption (RSA)
  - RSA Keys, Certificates
- Digital Signatures

PROJECT

Form Based Spring Boot Application



PROJECT

Spring Boot SAML Integration

Okta Cloud

1 - User Accesses Application
2 - Application forwards Request to IDP
3 - IDP displays Login Screen
4 - IDP authenticates the user
5 - IDP forwards Response to Application

Directory Service

4

OKTA
SAML Identity Provider

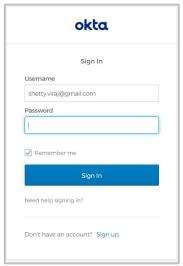
TRUST

My Desktop

3

5

Java Spring Boot



User Agent

2

User

1

Spring Security with SAML

Used Car Portal

Database

https://localhost:8080

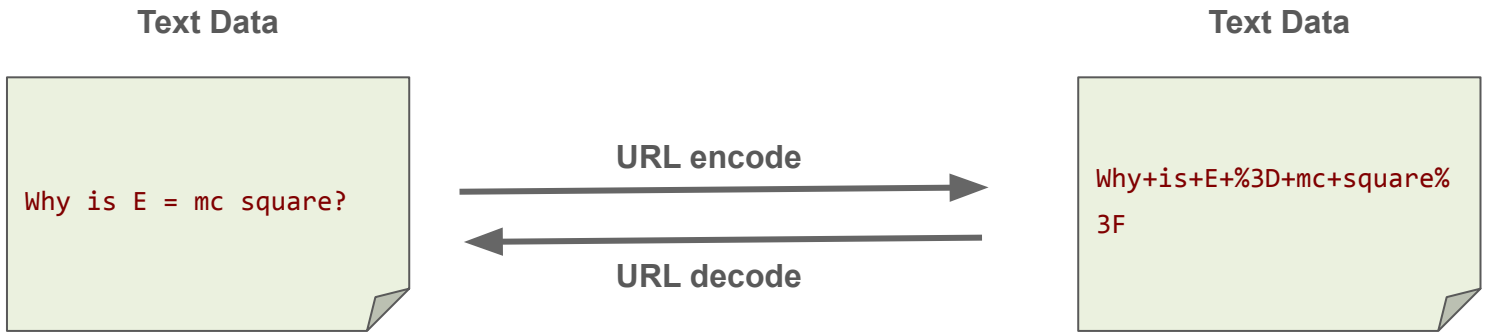
```yaml
spring:
  security:
    saml2:
      relyingparty:
        registration:
          carsonline:
            signing:
              credentials:
                - private-key-location: "classpath:credentials/private.key"
                  certificate-location: "classpath:credentials/certificate.crt"
            decryption:
              credentials:
                - private-key-location: "classpath:credentials/private.key"
                  certificate-location: "classpath:credentials/certificate.crt"
            identityprovider:
              metadata-uri: "classpath:okta-metadata.xml"
```

ACS URL	**<baseurl>**/login/saml2/sso/**<project-name>**
Single Logout URL	**<baseurl>**/logout/saml2/slo
Metadata URL(*)	**<baseurl>**/saml2/service-provider-metadata/**<project-name>**
EntityID	**<baseurl>**/saml2/service-provider-metadata/**<project-name>**

ACS URL	http://localhost:8080/login/saml2/sso/**carsonline**
Single Logout URL	http://localhost:8080/logout/saml2/slo
Metadata URL(*)	http://localhost:8080/saml2/service-provider-metadata/**carsonline**
EntityID	http://localhost:8080/saml2/service-provider-metadata/**carsonline**

URL Encoding

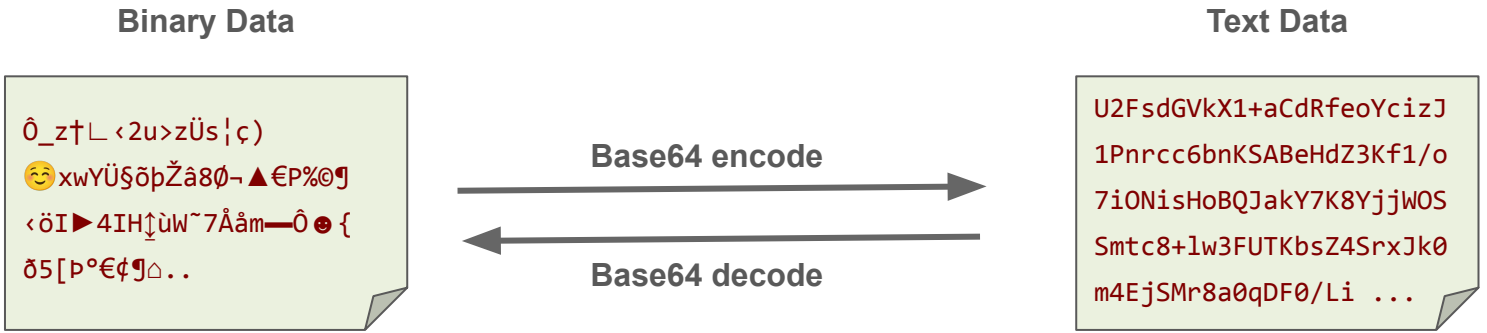
```
<url path>?param1=value1&param2=value2&param3=value3
```



- Conform data to URL rules
- No spaces, special characters

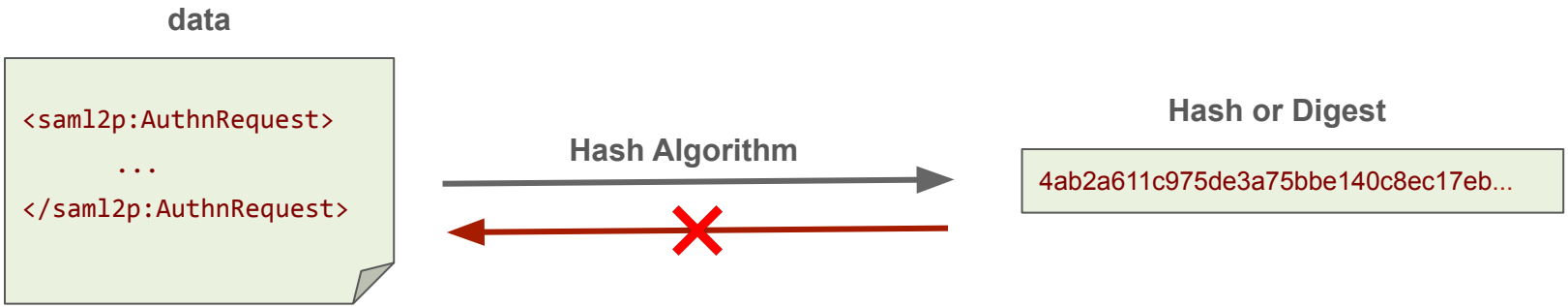
Space	%20 or +
?	%3F
=	%3D

Base64 Encoding



- Binary to text encoding
- Embed binary data (images) in Text (html. xml)
- Send data as HTTP Post

Hashing



- One way
- No collisions
- It's not Encryption

SHA-512

SHA-256

SHA-1

MD5

- Storing Passwords
- Digital Signatures
- Download files

Hashing

Java SE Development Kit 17.0.1 downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.

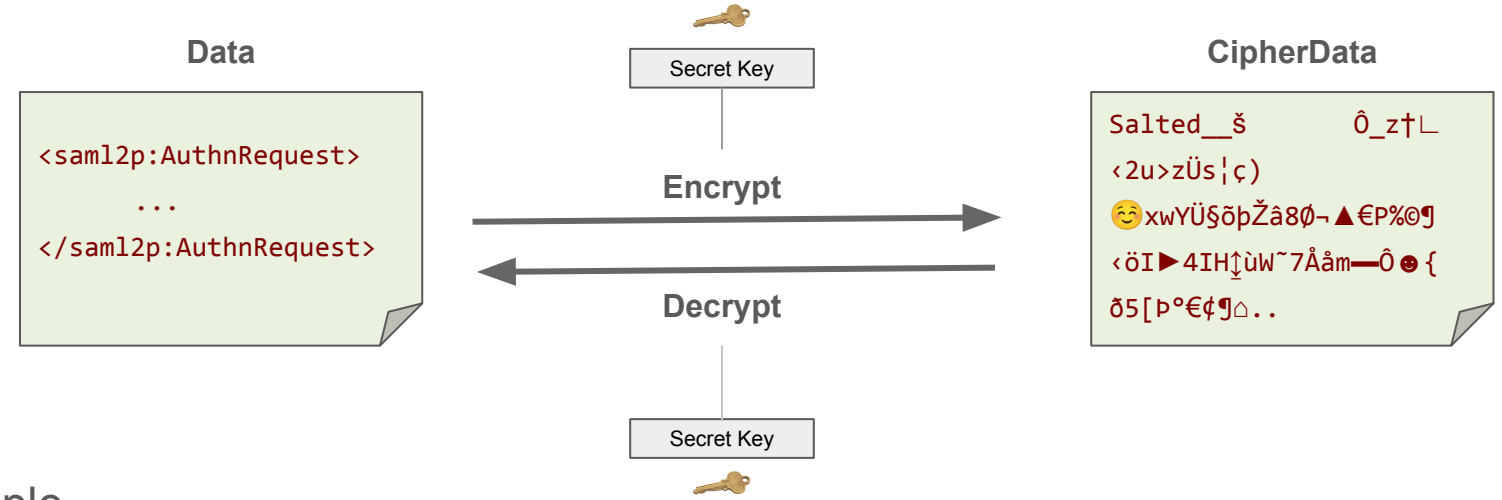
The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.

Linux **macOS** **Windows**

Product/file description	File size	Download
Arm 64 Compressed Archive	171.13 MB	https://download.oracle.com/java/17/latest/jdk-17_linux-aarch64_bin.tar.gz (sha256)
Arm 64 RPM Package	153.16 MB	https://download.oracle.com/java/17/latest/jdk-17_linux-aarch64_bin.rpm (sha256)
x64 Compressed Archive	172.35 MB	https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.tar.gz (sha256)
x64 Debian Package	148.02 MB	https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.deb (sha256)
x64 RPM Package	154.78 MB	https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.rpm (sha256)

```
a5e954a4e89b50277f20345034aea0ccf06f53705d4ec586b268b14ff42468f7
```

Symmetric Encryption



- Simple
- Secret Key
- Efficient for large data
- Hard to share secret key

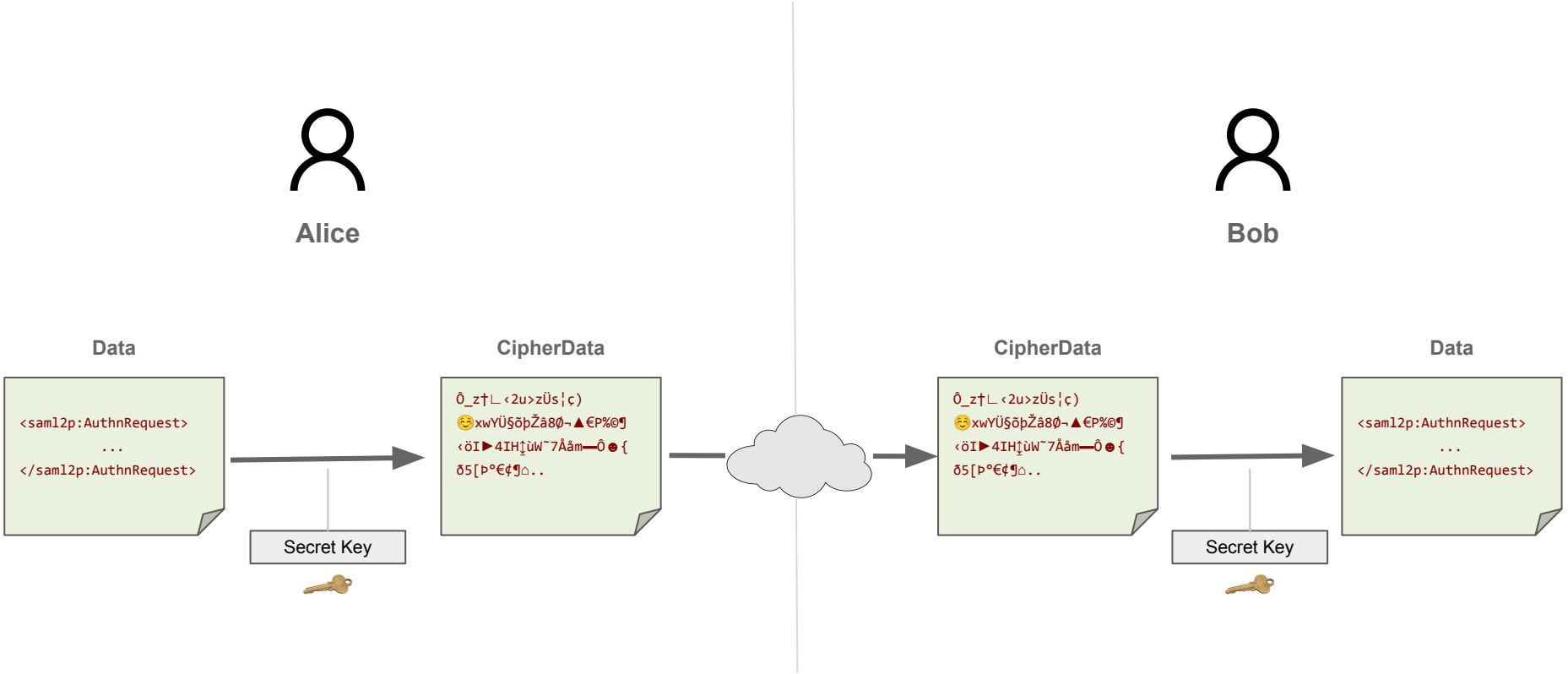
AES-256

AES-128

BlowFish

DES

Symmetric Encryption

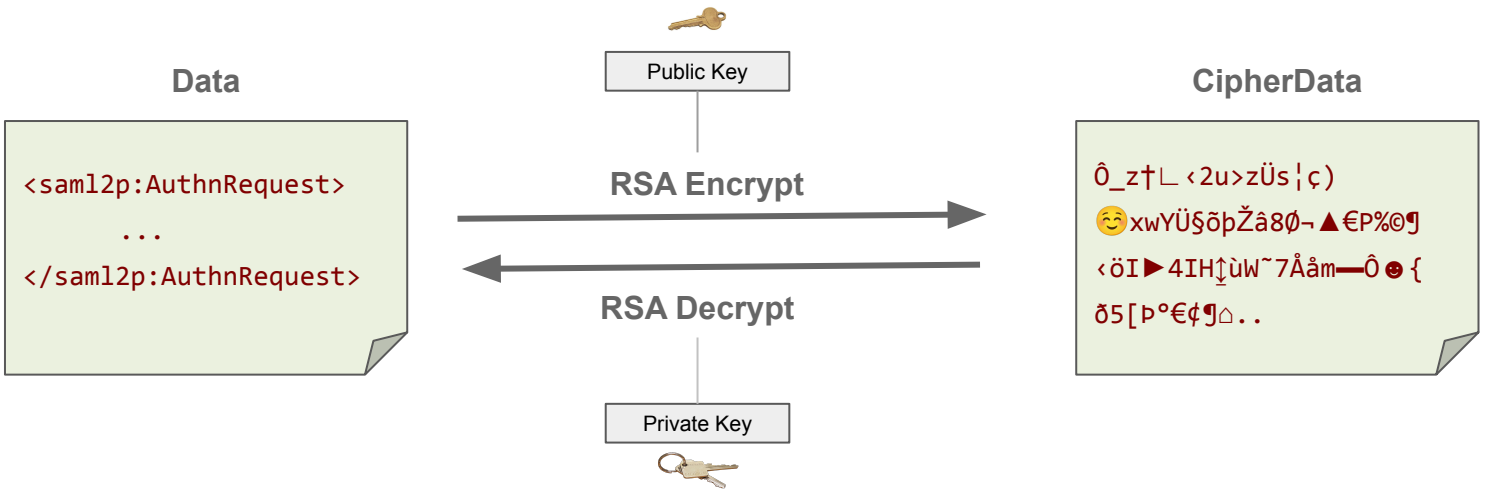


How will Alice share the Secret key with Bob ?

Asymmetric Encryption

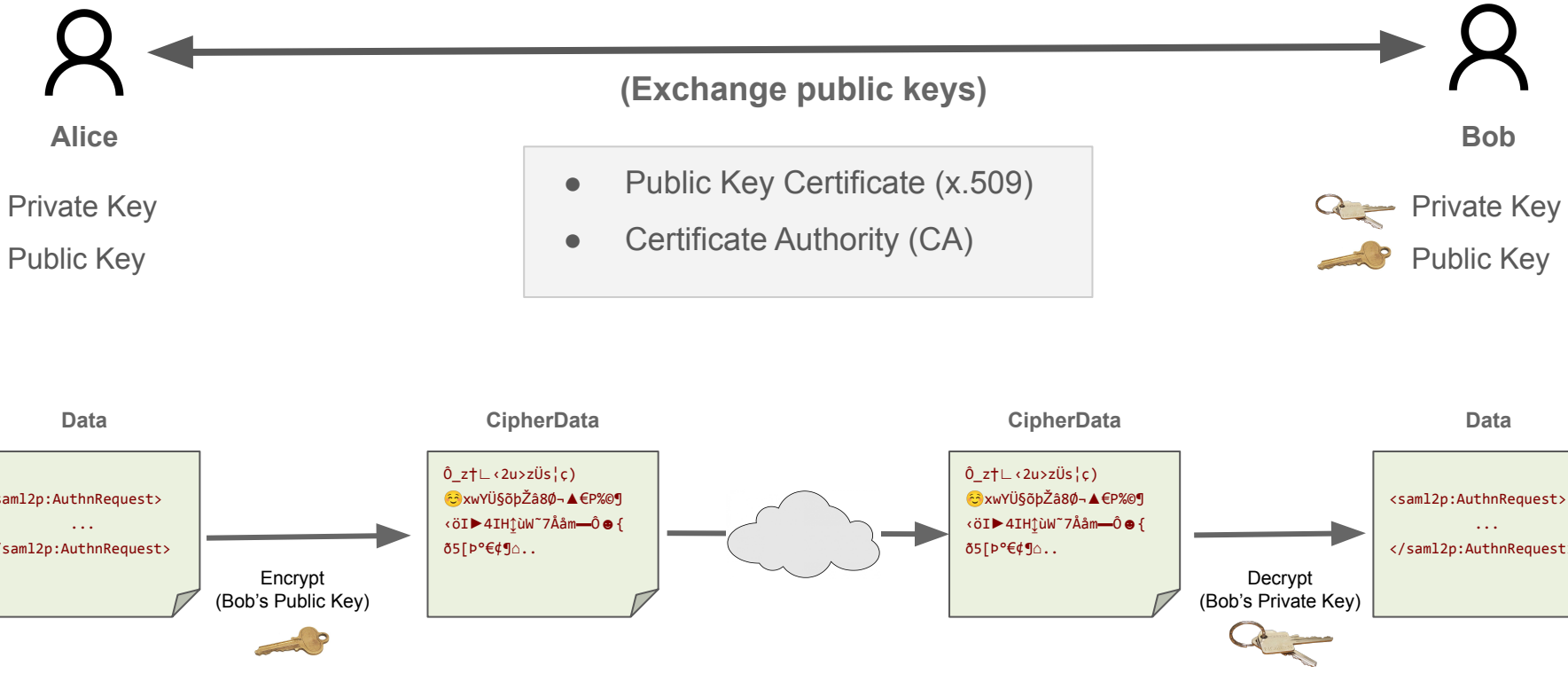

RSA Private Key


RSA Public Key



- Encrypt with Public Key, Decrypt with Private Key
- Sign with Private Key, verify with Public Key (more later)

Asymmetric Encryption

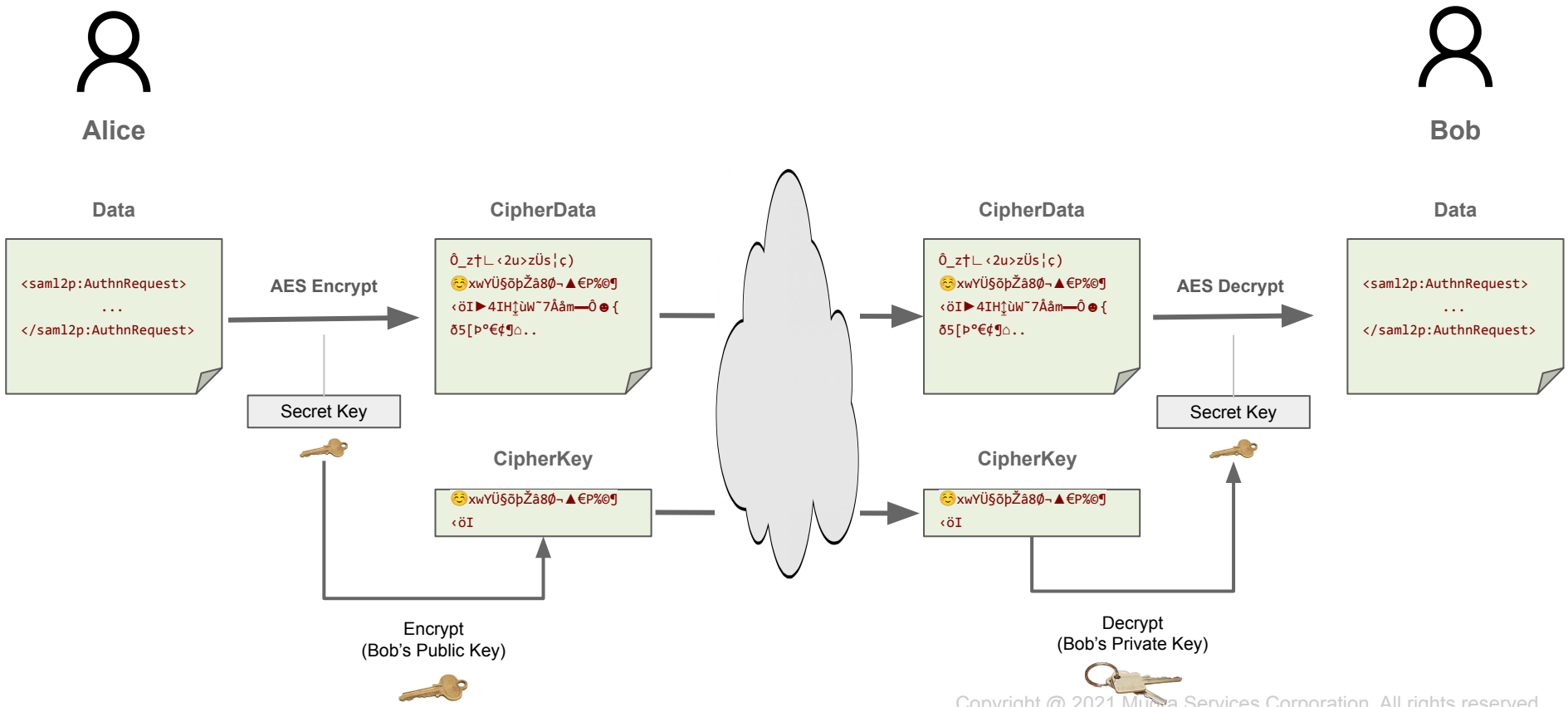


```
openssl req -x509 -nodes -sha256 -days 3650 -newkey rsa:2048 -keyout private.key -out certificate.crt
```

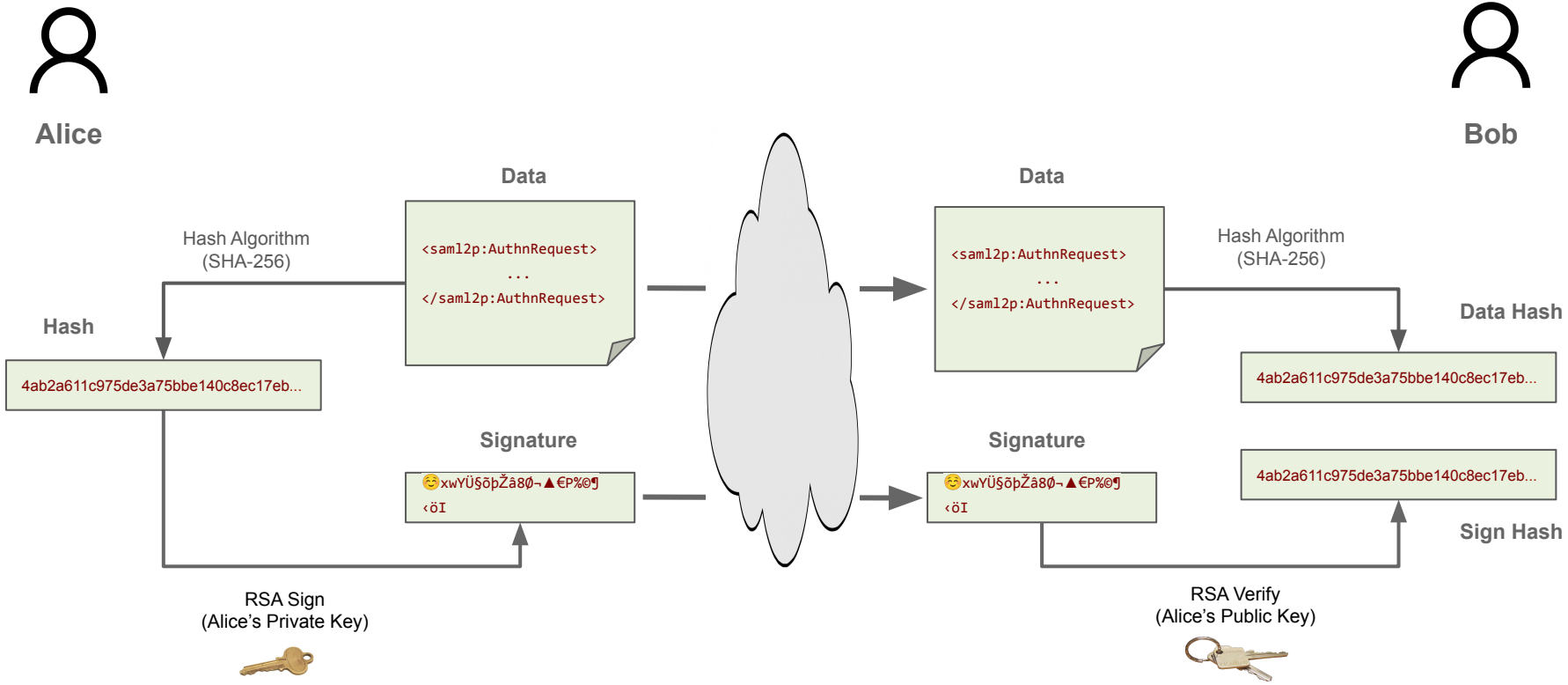
```
-----BEGIN PRIVATE KEY-----
MIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAgEAAoIBAQC34IZ3HWRioB+a
6kIsp3TEfKSbVIXQpnoprIhjJwi0fuMAhEgDK2Xf6oxXAwWc3wWNF9cqvCohcer3
191JGKqUfMJvA1VV8PBNyAMNC2SPaUcUcwIm3ZF7osYkVPqfk5FBQ4K9F4s7mIxQ
lPNvfgjgrw7ZOpD936U0yPUolpCNOa6XjmEa7fanhs9FLIE0PvFGMmycbdm71eOe
cx2TC8jOx38HsfdbZJloDTAso23vs7kB9aPXC6LFUsuE4RX9cqYUadomv+kl8MgN
xqfP98EbVMX7KGWZRZ92RfoccupegSCq2Yu9CvLB3SNJYZztASeZrhHvQvQV5p8p
pOFBRxupAgMBAAECggEBAK9DeIei4WNMYrOjZC3x80+xyqgvuVi2xaxhQqLXuulo
JHECpS04OWNjyh7Jx5jNzxm8Rp/6OGImliNkgwzAUR81h5KO59EB1dsdsSLG4DP9
0f2A3eUzvg7NiPVqdnSSEJrXgY9BGLpWXjAUubBkvouC8LFHrJn/iRW2Ek98rA2e
9seNgnbx+913Wcbo6A5GbFQhGq6hq/6/t/qAbvbPfsAhUvWQvdAA+oay8DA8r+nq
RgN9X+XFSg3D5zuJATLRQF7KYg2OfvjMZqZirx9iTHPV20BNxEVh7z41WkXmYemu
nTYfa5JEybXd6M5hLoy7e5QmrozuPBxIM4lzbVLkPbECgYEA3UPvEHT9SzzjaTLL
Vi4hbxkT6nKRm1KMAW6Q1nuglAp9XaTfenzd6nPfhLRIwcbNp2SmALq37Uht7pYO
AMQrsxOLEUcMUrJ2ep+CKkYLYi4Z1jtWRtd2s99yQIEL1ztP2J6c8uV/uB7hq6L1
pf8RJDW+BDModSNqSGzERKFgOw0CgYEA1L4N9yNqdai6ikDf4RvRMr5NlKr1kFPn
0uRMvfMUFtKwsd4DpdXlb1TGOXNG7le3IgEHtWy8fdONz1EnqGXReHKYZjs1CMOZ
c2Whn4higVFjWSD638BT9/xLjrXlAxjUhzOa7PybLuvhyiUaro9xJBltjIE/He3j
7+It1JE9jA0CgYB5G3jbeh01tDMI3nt7RS6Zn5FODr1x1YG6Ould7DbdMMyjl92W
LohMjnW2LfNw1L61s5pE5e5MGwvIvisV8km076p7n3a4QOQmg+3783DBoVgU8U9s
4PwgwdnIQpnXiahPeamQLVt1zLlad4ya+fxI5H2H0PSBAQxobdgTdHy2dQKBgQDA
ALeB8QjIH/wbgumLtRtdfLtwDcuK8u8rOri4RoY/AvBN113bceycShqNBylhOvEh
U2StrPGBbklyboAoT5x55JL+0TjQLBh2Oadg4CpnJBlR+53wCxi9m4plxmxmGybU
zxJoCKPV+TXSxyIJoZQvTHO+d3eLf4RUoAFU87BkkQKBgQCoQUveeHwLvQS3hnXj
u62wxGydSAF55OI5XuHeQuAQxYQVV8k8dOFjMXUy+qTQ9K78kjXFpGxahxBxc9aG
w9y7LNdnbTBqwwWUsdB6oJYOnz6yECtdBrnXe72vjTxLBMr00ZVtmlfoVho4RvPy
dQ1/L/w7P0Z9Z3OT2EfA1urWYw==
-----END PRIVATE KEY-----
```

```
-----BEGIN CERTIFICATE-----
MIIDtjCCAp6gAwIBAgIJALCM7qOEGLLVMA0GCSqGSIb3DQEBCwUAMHAxCzAJBgNV
BAYTAlVTMREwDwYDVQQIDAhWaXJnaW5pYTERMA8GA1UEBwwIU3Rlcmxpbmcx DjAM
BgNVBAoMBW11ZHJhMRQwEgYDVQQLDAtlbmdpbmVlcmluZzEVMBMGA1UEAwwMcHJv
amVjdDJkZW1vMB4XDTIxMTAxODE3MzI1MVoXDTMxMTAxNjE3MzI1MVowcDELMAkG
A1UEBhMCVVMxETAPBgNVBAgMCFZpcmdpbmlhMREwDwYDVQQHDAhTdGVybGluZzEO
MAwGA1UECgwFbXVkcmExFDASBgNVBAsMC2VuZ2luZWVyaW5nMRUwEwYDVQQDDAxw
cm9qZWN0MmRlbW8wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC34IZ3
HWRioB+a6kIsp3TEfKSbVIXQpnoprIhjJwi0fuMAhEgDK2Xf6oxXAwWc3wWNF9cq
vCohcer3191JGKqUfMJvA1VV8PBNyAMNC2SPaUcUcwIm3ZF7osYkVPqfk5FBQ4K9
F4s7mIxQlPNvfgjgrw7ZOpD936U0yPUolpCNOa6XjmEa7fanhs9FLIE0PvFGMmyc
bdm71eOecx2TC8jOx38HsfdbZJloDTAso23vs7kB9aPXC6LFUsuE4RX9cqYUadom
v+kl8MgNxqfP98EbVMX7KGWZRZ92RfoccupegSCq2Yu9CvLB3SNJYZztASeZrhHv
QvQV5p8ppOFBRxupAgMBAAGjUzBRMB0GA1UdDgQWBBR13m4wYjvebU0W/5khIXXN
JaIRYDAfBgNVHSMEGDAWgBR13m4wYjvebU0W/5khIXXNJaIRYDAPBgNVHRMBAf8E
BTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQAo+2zysvdZ3kwXiPIslhe2whbyAY+f
l8IymHburicoYES/fuSHFyQe59b/5o3HNNT+mm7qmnUkNbInttpSuPS8I+1nIZd9
qbJkj7pgSSA/vhbqtnupoDT1F7d5i/c05gwinN08r8Ohoc48GlH2uihVJdvcA+Ti
0yMQYtGiNoyyU4l2Bv0D2R/B1+0b4NaAkw7Dfx2iQw2TEEXEQe8M9AS/zQIRxQlS
FGxM1cVyIx7bIY9Z1xyetLjUtbt0gzGHt4OQdVAjbONbL3MqHzd7inV0f80gLF1E
0FqjoZHChavPgWl2Z77aavq5EglaC0iEJhY5FRe0A2g3Xbb2SQSr+qFQ
-----END CERTIFICATE-----
```

Hybrid Encryption



Digital Signature and Verification



Is **Data Hash = Sign Hash** ?