

# The Secured Enterprise: Leverage OpenID with Web Services

Prabath Siriwardena  
Technical Lead & Product Manager  
WSO2



The open source SOA company

WSO2 is an innovative Open Source technology company devoted to building Web services middleware for your SOA. Offering leading products, support and other services, WSO2 was founded in August 2005. It is a global corporation with offices located in USA, UK and Sri Lanka.



## Notebook: Credit Card Theft

August 6, 2008 4:46 PM

Eleven people were recently arrested for allegedly stealing over 40 million credit card numbers from an online source. Katie Couric notes that this is a telling sign of online shopper vulnerability.

# 40,000,000

credit card numbers stolen

**internetnews.com**

## Report: 85 Percent of U.S. Businesses Breached

**UPDATED:** The report has a recommendation for organizations that fear a breach and the reporting requirements that go with it.

July 13, 2009

By Alex Goldman: [More stories by this author.](#)



The fourth annual U.S. Encryption Trends Study was released today by The Ponemon Institute. The study says that 85 percent of surveyed businesses have experienced a data breach in the past year, up from 60 percent in the 2008 study. The report was sponsored by encryption supplier [PGP Corp.](#)

"A data breach is defined as the loss or theft of confidential or sensitive data including information about people and households," said Dr. Larry Ponemon, chairman and founder of [The Ponemon Institute](#), in an e-mail to [InternetNews.com](#).

The numbers are comparable to a similar study released last week concerning UK businesses. There, the Ponemon Institute found that [70 percent](#) had been breached in the last year.

The report was based on surveys with nearly a thousand (997) U.S.-based executives.

Security needs to be by design

NOT an after thought

What do we need  
to secure...



*We have a bunch  
of services  
already developed  
and some under  
development....*



*Yes... we need to  
make sure all the  
data transferred  
are secured....*



How about  
securing data  
transfer between  
service and the  
client through  
HTTPS....



*HTTPS is not bad.. But still it has certain limitations...*



Transport level encryption

Point to point

Entire message needs to be encrypted

Adds less weight on message payload

Applies only to HTTP



*How about  
message level  
security?*

End to End

Parts of the message can be encrypted

Adds more weight on message payload

Transport Independent

*Yes – let's  
finalize on  
Message level  
security....*



*How can we use  
Message Level  
Security to protect  
our services...*



Confidentiality

Integrity

Authentication

The assurance that a message has not been read by anyone other than the intended reader



The assurance that data is complete and accurate



The verification of a claimed identity



Can we make  
sure we  
interoperate with  
the rest...



Yes... we need not to re-implement the wheel... what is the standard to achieve C-I-A with message level security...?



Defines how to achieve confidentiality, integrity and authentication with SOAP messages

Does not define a new security technology only focuses on applying existing security technologies to SOAP messages

*With UsernameToken  
defined in WS-  
Security enables us to  
authenticate users  
with  
username/password...*



```
<wsse:UsernameToken wsu:Id="Example-1">
  <wsse:Username> ... </wsse:Username>
  <wsse:Password
    Type="..."> ... </wsse:Password>
  <wsse:Nonce
    EncodingType="..."> ... </wsse:Nonce>
  <wsu:Created> ... </wsu:Created>
</wsse:UsernameToken>
```

*WS-Security brings  
XML Encryption to  
enable confidentiality  
in SOAP Messages...*



Shared Key

Key Wrapping

A shared key for both encryption and decryption

Can operate on large plain text messages

Uses public key encryption to manage shared key distribution securely

Fast

Both the client & the service  
need not to have a certificate

A shared key is derived through  
the service's certificate

Further communication being encrypted  
with the derived shared key



*Integrity comes  
through the XML  
Signature....*

Integrity

Non repudiation

## WS - Security

XML Signature

XML  
Encryption

Username  
Token Profile

X.509 Token  
Profile

Okay... now all  
our services are  
secured with ws-  
security... What is  
next?



*We need to see  
who should be  
given access to  
our services....*



*Definitely all the  
internal users...*



*...also some of our  
partner  
companies...*



Okay... we can easily authenticate internal users with `UserNameToken` - since we have their credentials internally....



But we don't  
maintain  
credentials of  
external users...  
coming from our  
partner  
companies...

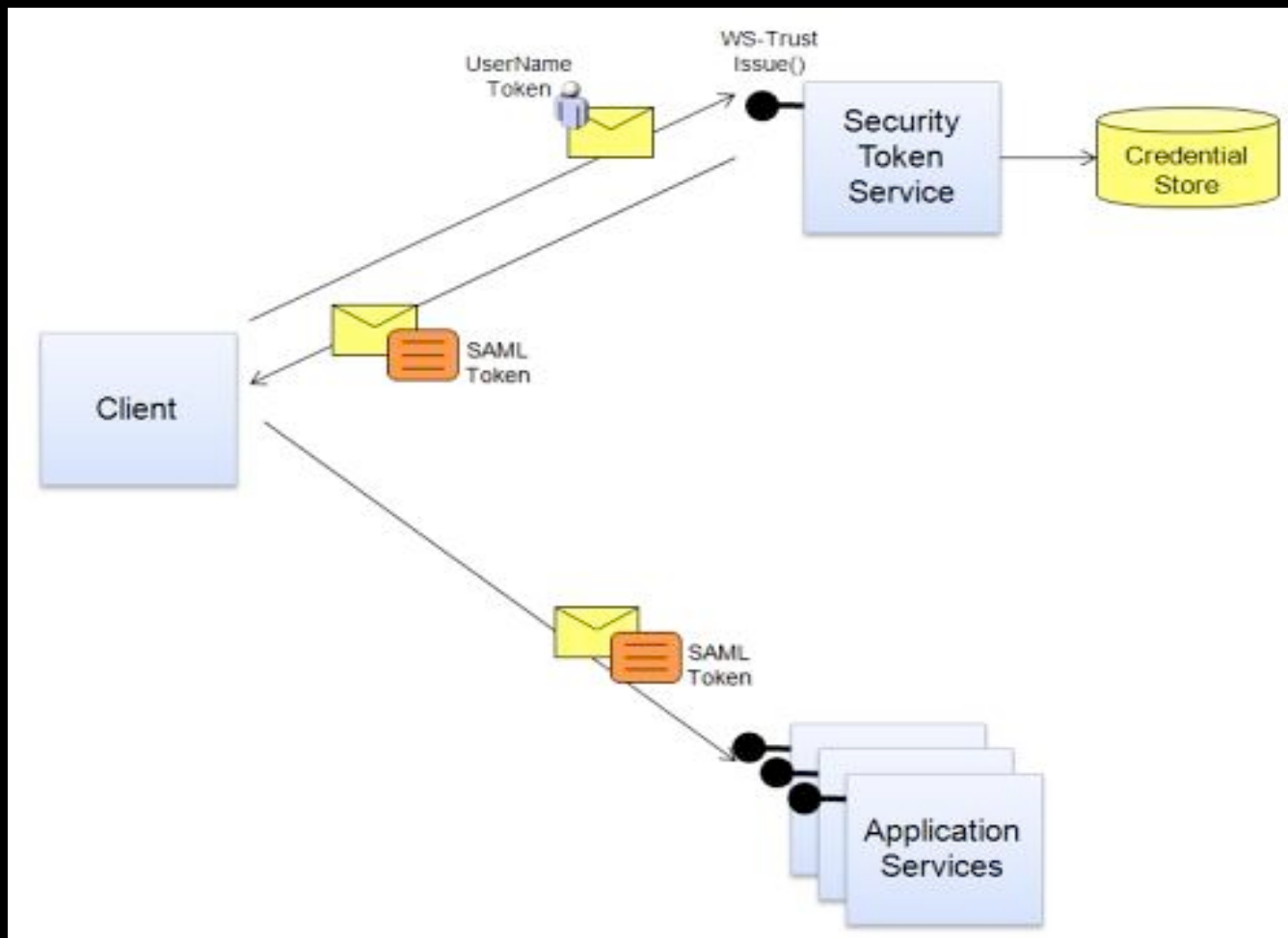




*We need not to  
maintain  
external user  
credentials... we  
only need to trust  
our partners....*



*...and that is  
what WS-Trust  
does....*



We need not to authenticate individual external users

We only TRUST external partners

All the requests coming through external users need to be signed by the corresponding partner companies

Only the requests signed by TRUSTED partners will let in

*...also our users  
need access to  
external systems.  
Out of our  
domain...*



*That is exactly  
the other side of  
what we just  
discussed.. We  
need to maintain  
an internal STS*



All the requests going out side from internal users need to have a security token issued by the internal STS

Internal users should authenticate themselves with the internal STS - prior to obtaining a security token

External services need to trust our STS

WS - Trust

WS - Security

XML  
Signature

XML  
Encryption

Username  
Token  
Profile

X.509  
Token  
Profile

Now... the question is how are we going to communicate our security requirements to the rest...



*Let's first list the  
security  
requirements.....*



Internal users should authenticate with user name / password when accessing services directly

External users should present a security token from a trusted STS

Email address should be present in the security token comes with the external users.

Only some parts of the message needs to be encrypted.

Encryption algorithm should be AES.

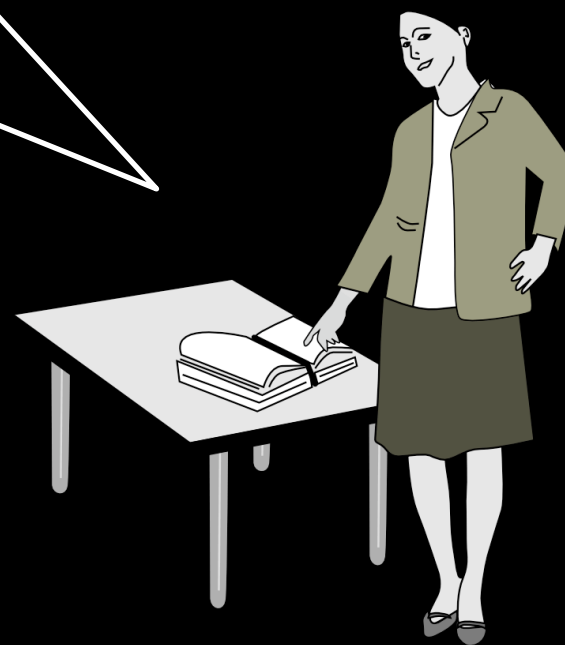
Encryption key size needs to be 256.

All the parts in the <Body> must be signed

*We need a way  
to express all  
these in a  
standard way....*

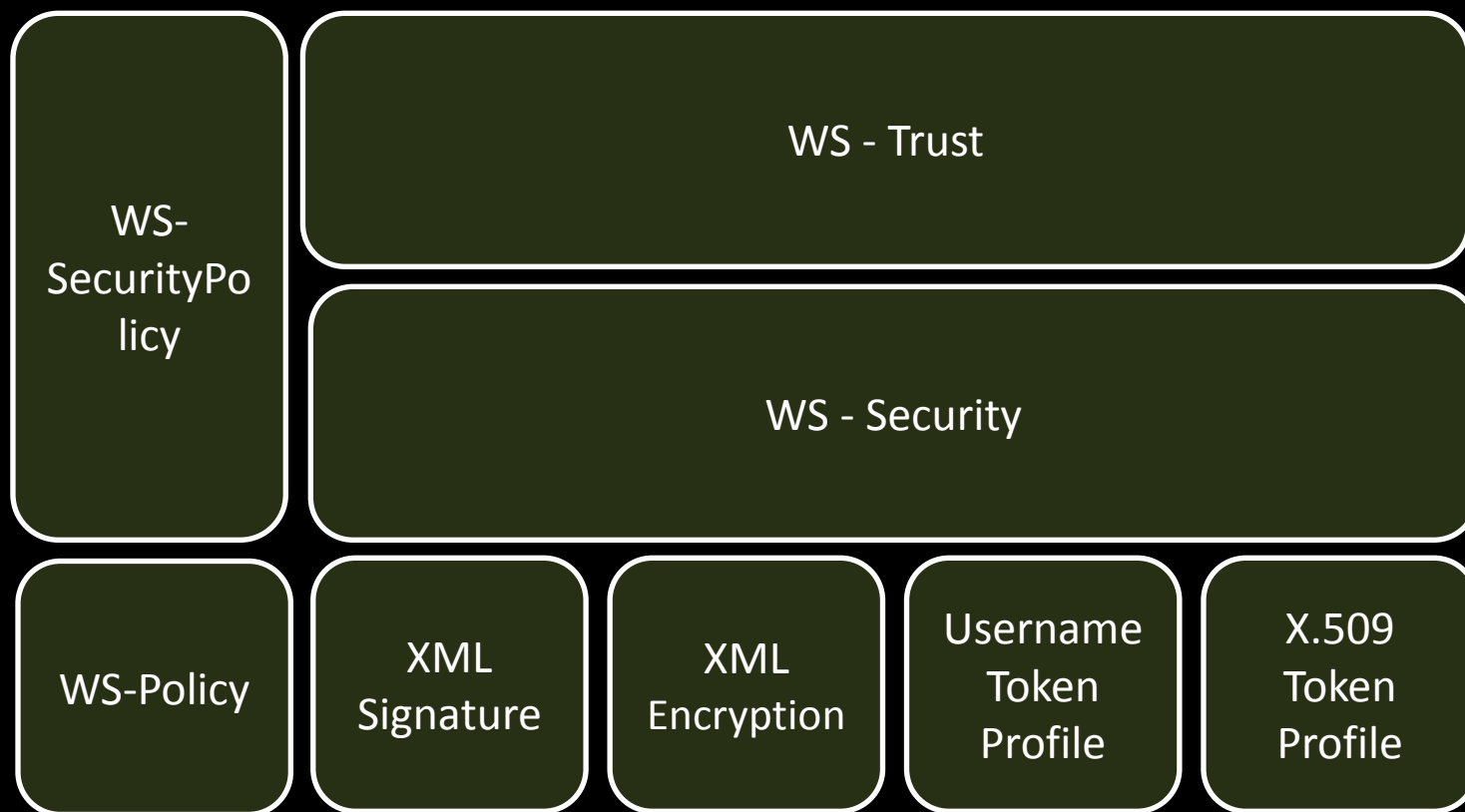


*Ws-security  
policy exactly  
addresses that...*



Used to express security requirements of a Web service according to, What needs to be protected.. What tokens to use.. Algorithms, reference types, etc...

Security policies can be defined at the binding level / operation level



Everything looks good.... Is there a way we could make sure we strictly follow the security polices defined...





*Okay - that means we need to validate each and every service developed...*

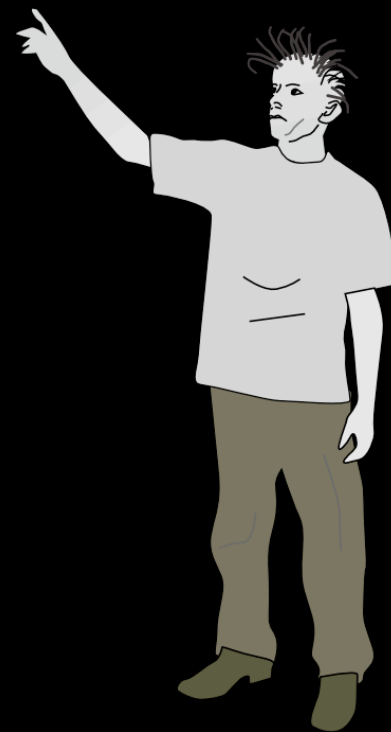
*Yes – validation  
needs to happen  
at two stages...*



*Design time  
validations will  
make sure we  
adhere to proper  
standards and  
policies at the  
time we develop ...*



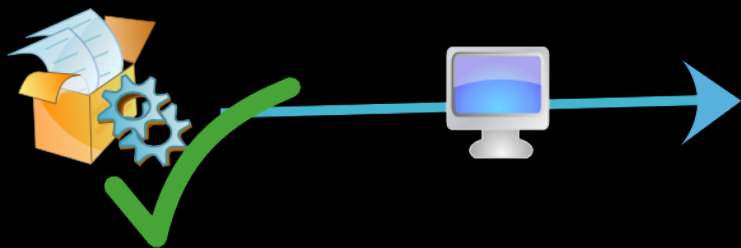
Runtime  
validations will  
make sure we  
evaluate all the  
requests coming in  
against the  
defined security  
policies....

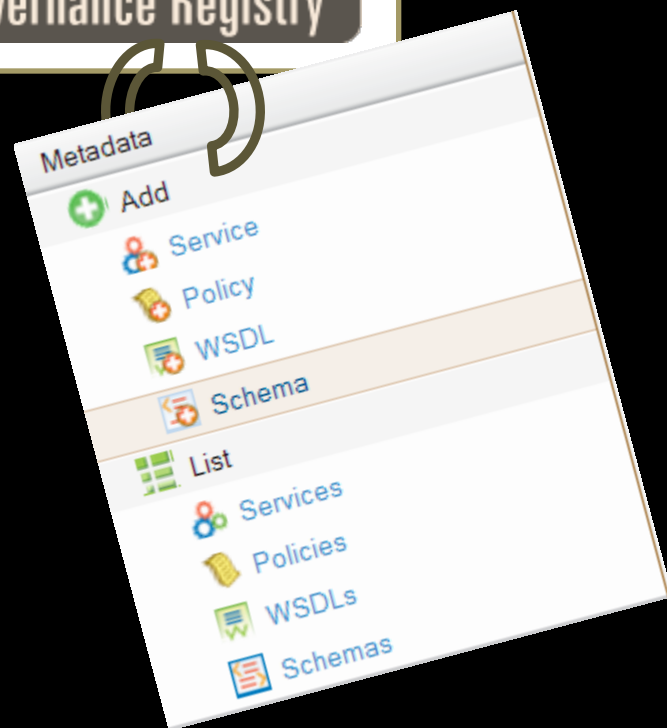


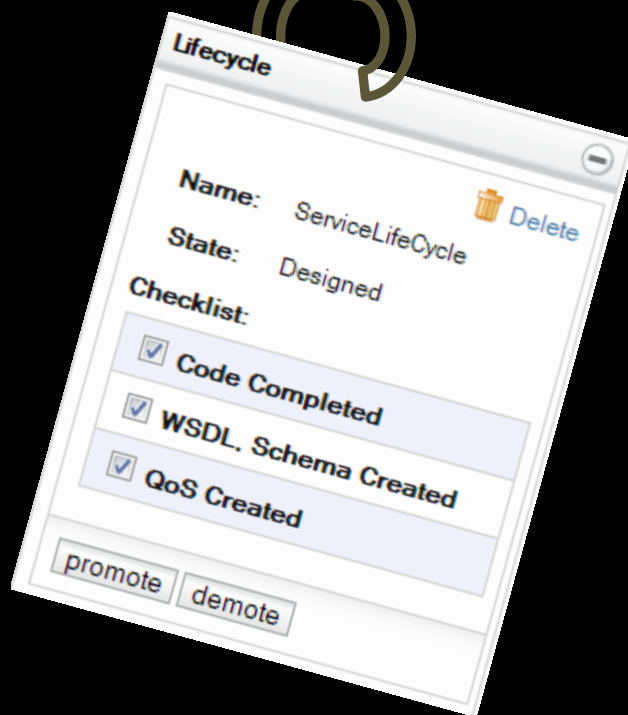
Design time governance



Runtime time governance







A screenshot of a "Subscriptions" dialog box. It has a title bar with a close button. The main content includes a green plus icon and the text "Add Subscription". Below this are two dropdown menus: "Event \*" and "Notification \*", both currently showing "-SELECT-". At the bottom are "Subscribe" and "Cancel" buttons. A help icon and a message "Select Event Type and Notification Method to Subscribe" are also present. At the very bottom, it says "No subscriptions for events of this entry." There are two overlapping circular arrows pointing clockwise, one above the other, positioned over the dialog box.

**Subscriptions**

+ Add Subscription

Event \* -SELECT-

Notification \* -SELECT-

Subscribe Cancel

? Select Event Type and Notification Method to Subscribe

No subscriptions for events of this entry.

**MONITORING**

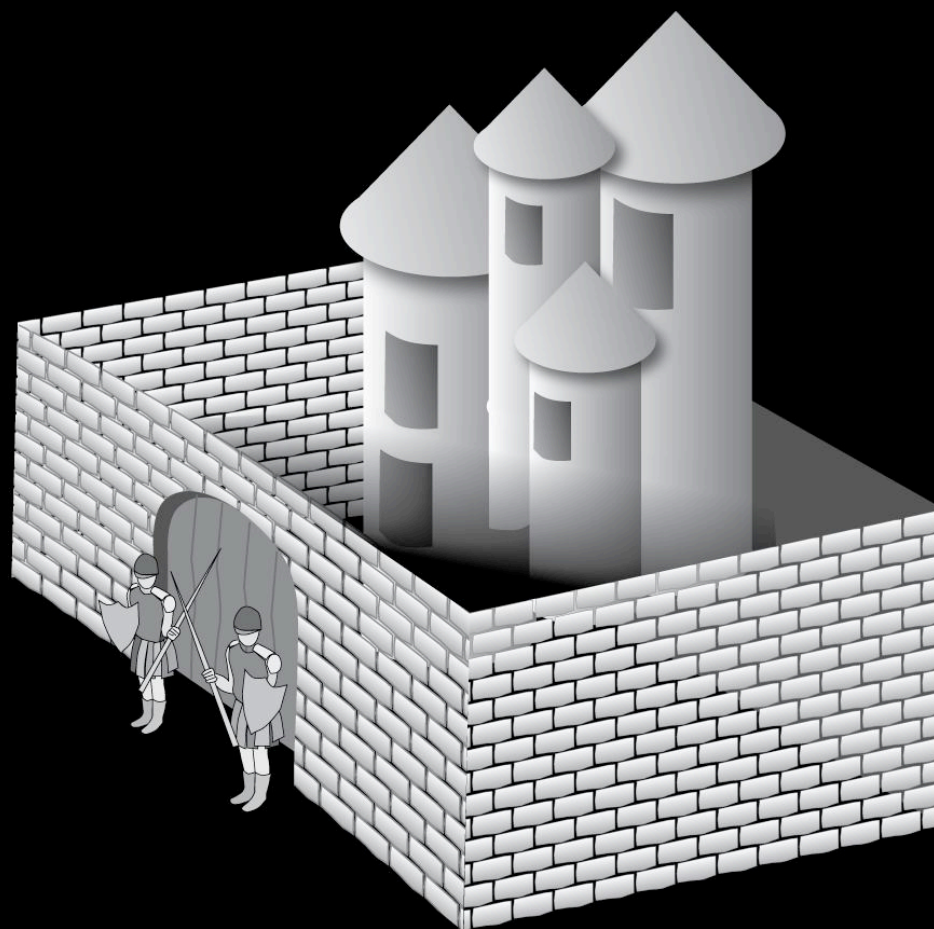
Yet... we haven't figure out how to enforce policies on users – or the requests coming through to our services...



*Yes... we need to make sure all the requests comply with the defined security policies...*



NOTES..... MESSAGE INTERCEPTOR  
GATEWAY PATTERN



Provides a single entry point and allows centralization of security enforcement for incoming and outgoing messages.

Helps to apply transport-level and message-level security mechanisms required for securely communicating with a Web services endpoint.

The logo for WSO2 Web Services Application Server, featuring the WSO2 logo on the left and the text 'Web Services Application Server' on the right, all within a rounded rectangular frame.

All the services can be deployed inside  
WSO2 Web Services Application Server  
[WSAS] – not publicly accessible

An open source web services engine powered  
by Apache Axis2



**Configure**

- User Management
- Key Stores
- Logging

**Manage**

Service

- List
- Add
  - POJO Service
  - JAX-WS Service
  - Axis1 Service
  - Data Service
    - Create
    - Upload
  - Spring Service
  - EJB Service
  - Axis2 Service
  - Jar Service

Modules

- List
- Add
- Transports
- Shutdown/Restart

Registry

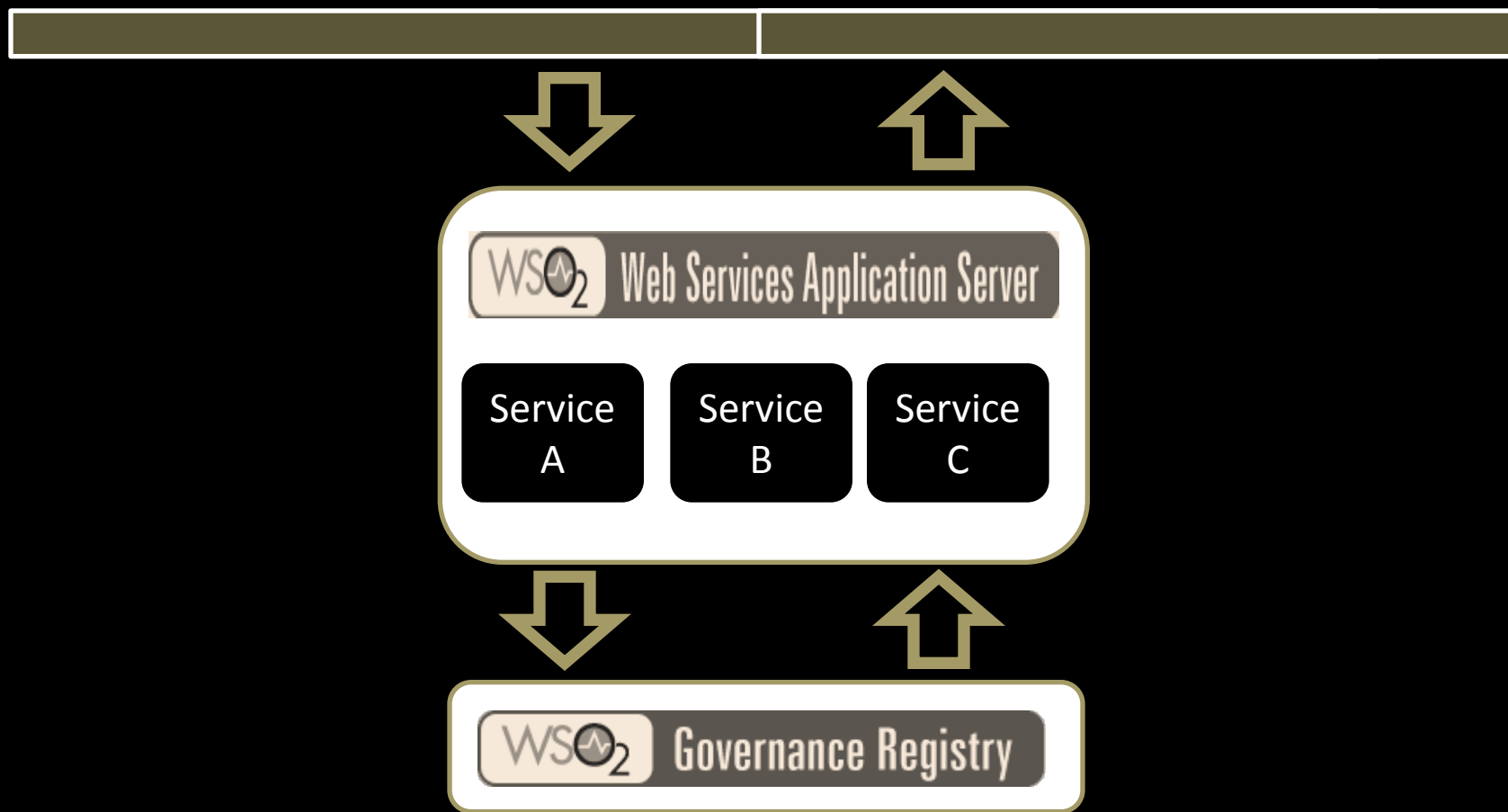
- Browse
- Search

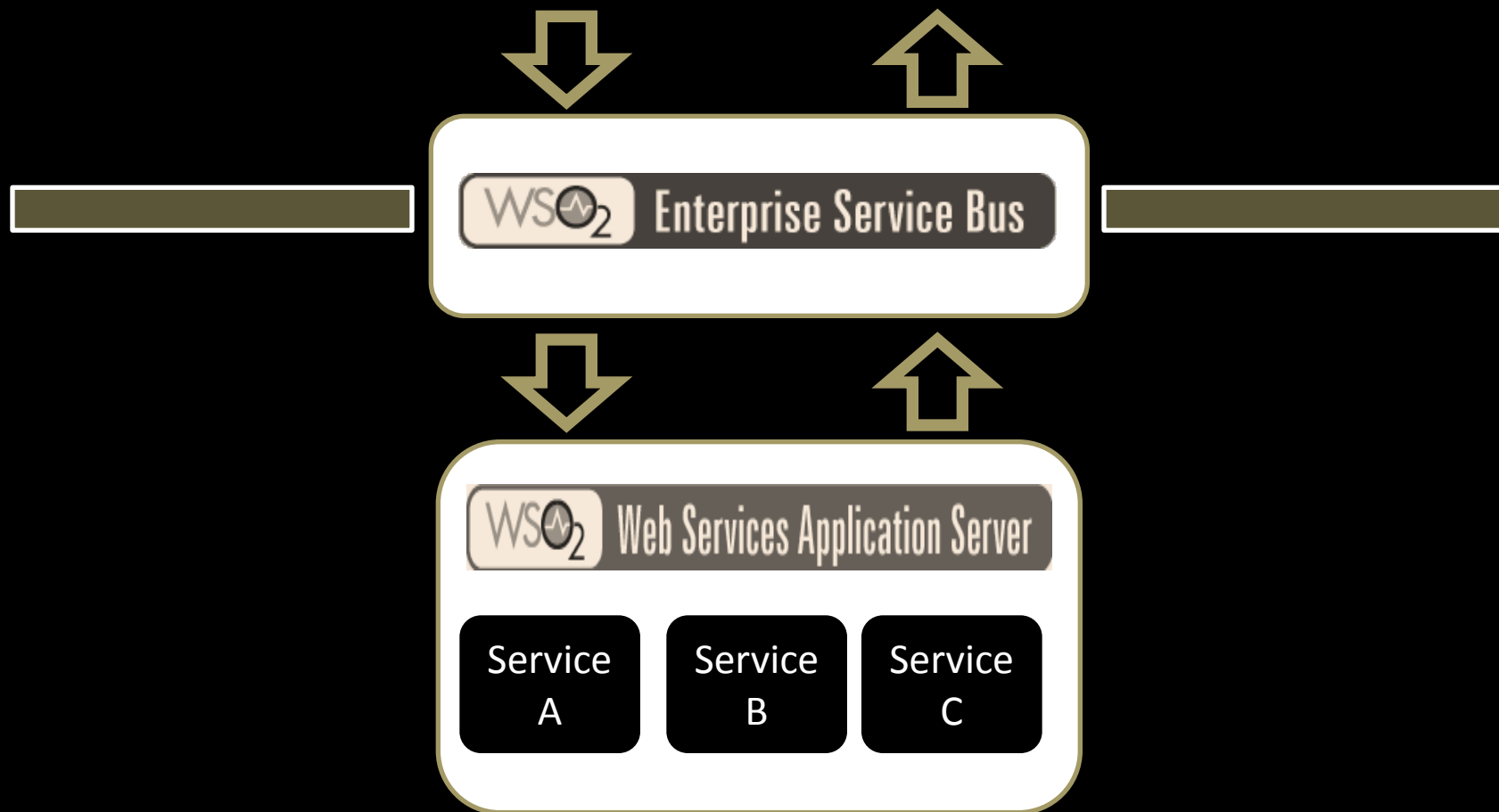
Monitor

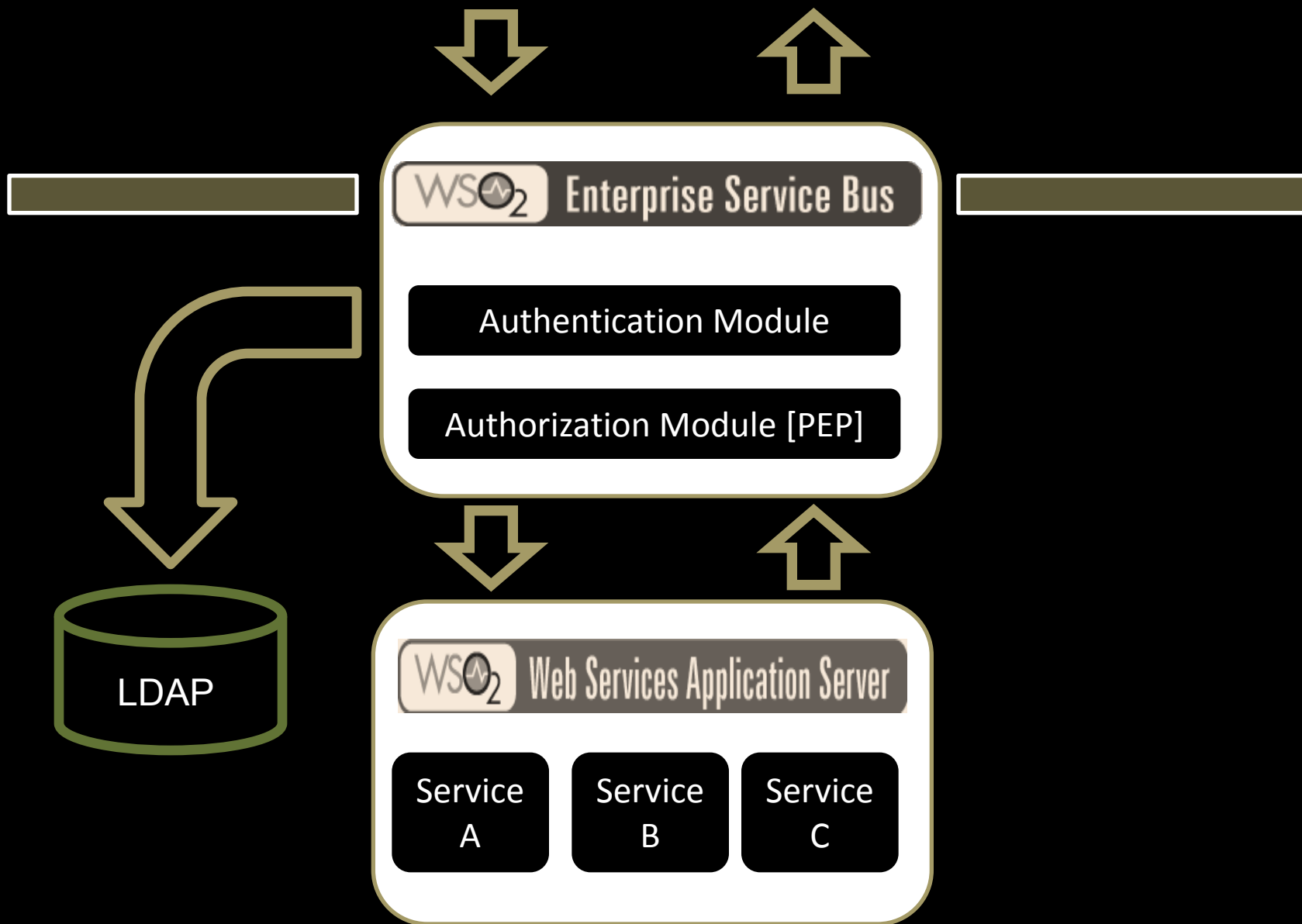
- System Statistics
- System Logs
- SOAP Tracer
- Message Flows

Tools

- WSDL2Java
- Java2WSDL
- WSDL Converter
- Try It
- Service Validator
- Module Validator









## Enterprise Service Bus

Quality of Service Configuration	
✔ Active [ Deactivate ]	
🛡 Security	📄 Policies
📧 Reliable Messaging	📡 Transports
⚙ Response Caching	📦 Modules
🚦 Access Throttling	📅 Operations
⚙ MTOM <input type="text" value="False"/>	📄 Parameters



Quality of Service Configuration

Active [ Deactivate ]

Security	Policies
Reliable Messaging	Transports
Response Caching	Modules
Access Throttling	Operations
MTOM <input type="checkbox"/> False	Parameters

### Security for the service

The service "MyProxyService" is not secured.

Enable Security?  Yes

Basic Scenarios		
1.	<input type="radio"/>	UsernameToken
2.	<input type="radio"/>	Non-repudiation
3.	<input type="radio"/>	Integrity
4.	<input type="radio"/>	Confidentiality
Advanced Scenarios		
5.	<input type="radio"/>	Sign and encrypt - X509 Authentication
6.	<input type="radio"/>	Sign and Encrypt - Anonymous clients
7.	<input type="radio"/>	Encrypt only - Username Token Authentication
8.	<input type="radio"/>	Sign and Encrypt - Username Token Authentication
9.	<input type="radio"/>	SecureConversation - Sign only - Service as STS - Bootstrap policy - Sign and Encrypt , X509 Authentication
10.	<input type="radio"/>	SecureConversation - Encrypt only - Service as STS - Bootstrap policy - Sign and Encrypt , X509 Authentication
11.	<input type="radio"/>	SecureConversation - Sign and Encrypt - Service as STS - Bootstrap policy - Sign and Encrypt , X509 Authentication
12.	<input type="radio"/>	SecureConversation - Sign Only - Service as STS - Bootstrap policy - Sign and Encrypt , Anonymous clients
13.	<input type="radio"/>	SecureConversation - Encrypt Only - Service as STS - Bootstrap policy - Sign and Encrypt , Anonymous clients
14.	<input type="radio"/>	SecureConversation - Encrypt Only - Service as STS - Bootstrap policy - Sign and Encrypt , Username Token Authentication



### Security for the service

The service "MyProxyService" is not secured.

Enable Security?  Yes  No

Basic Scenarios		
1.	<input type="radio"/>	UsernameToken
2.	<input type="radio"/>	Non-repudiation
3.	<input type="radio"/>	Integrity

Quality of Service Configuration

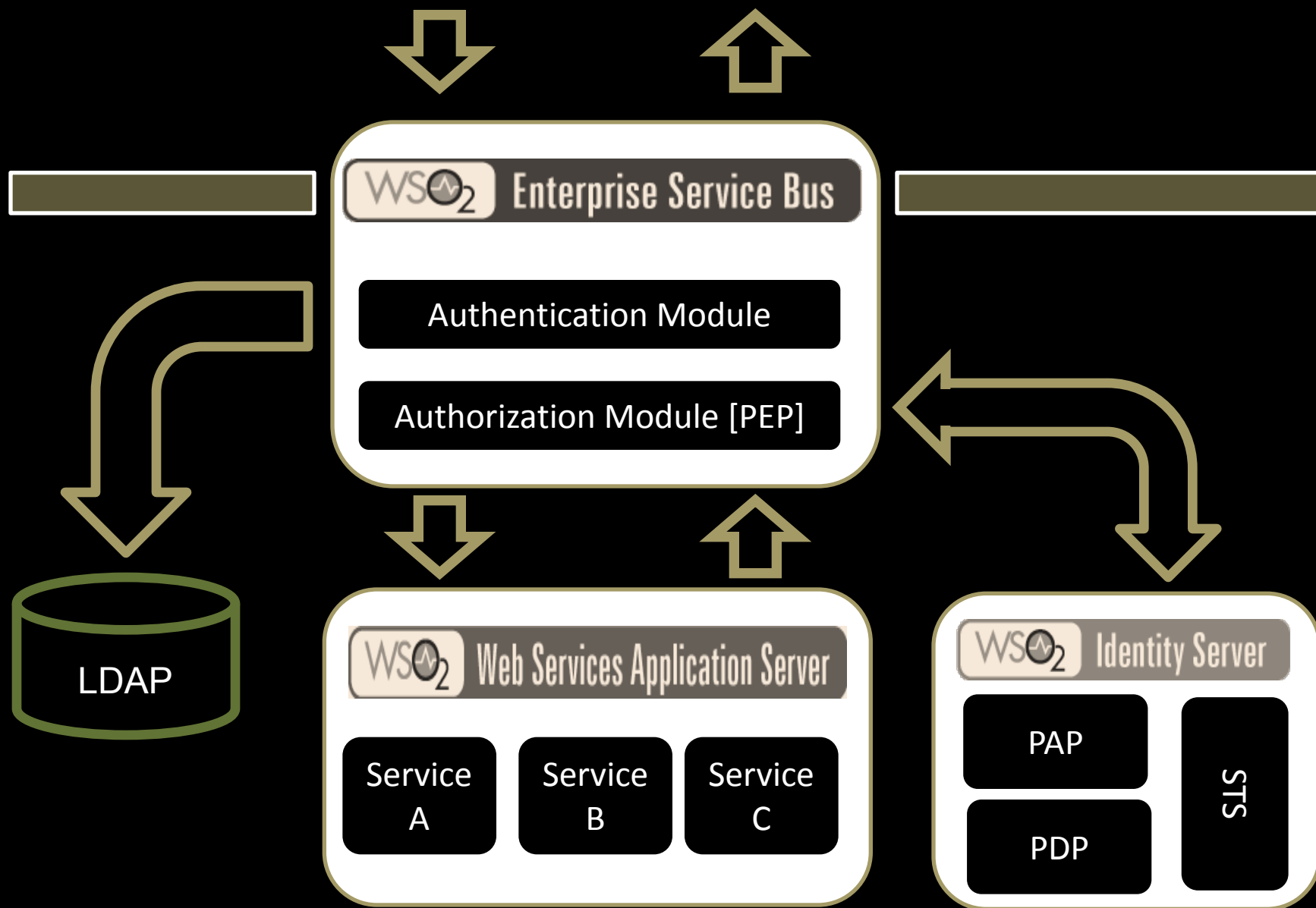
Active [ Deactivate ]

- Security
- Reliable Messaging
- Response Caching
- Access Throttling
- MTOM  False

#### Policy Selection

Service Hierarchy		
Service MyProxyService		<a href="#">Edit Policy</a>
Operation mediate		<a href="#">Edit Policy</a>
Operation mediate	<input checked="" type="radio"/> In Message <input type="radio"/> Out Message	<a href="#">Edit Policy</a>
Binding Hierarchy		
Binding MyProxyServiceSoap11Binding		<a href="#">Edit Policy</a>
Operation mediate		<a href="#">Edit Policy</a>
Operation mediate	<input checked="" type="radio"/> In Message <input type="radio"/> Out Message	<a href="#">Edit Policy</a>
Binding MyProxyServiceSoap12Binding		<a href="#">Edit Policy</a>
Operation mediate		<a href="#">Edit Policy</a>
Operation mediate	<input checked="" type="radio"/> In Message <input type="radio"/> Out Message	<a href="#">Edit Policy</a>

14.	<input type="radio"/>	SecureConversation - Encrypt Only - Service as STS - Bootstrap policy - Sign and Encrypt , Username Token Authentication
-----	-----------------------	--



The logo for WSO2 Identity Server, featuring the WSO2 logo and the text "Identity Server" inside a rounded rectangular box.

WSO<sub>2</sub> Identity Server

Claim-based security token service - mapping user attributes to defined claims, which can be used to enable identity federation with claim aware web services.


XACML Policy Administration Point & Policy Decision Point



## Identity Server

## STS Configuration

 Apply Security Policy  <https://localhost:9443/services/wso2carbon-sts>

Service Endpoint Address	Certificate Alias	
<a href="http://localhost:8280/services/echo">http://localhost:8280/services/echo</a>	wso2carbon.cert.cer	 Delete

## Add new trusted service

Endpoint Address\*

Certificate Alias

Apply



### STS Configuration

Apply Security Policy https://localhost:9443/services/wso2carbon-sts

Service Endpoint Address	Certificate
http://localhost:8280/services/echo	wso2c...

Add new trusted service

Endpoint Address\*

Certificate Alias

### Security for the service

The service "wso2carbon-sts" is secured using "UsernameToken"

Enable Security?

Basic Scenarios		
1.	<input checked="" type="radio"/>	UsernameToken
2.	<input type="radio"/>	Non-repudiation
3.	<input type="radio"/>	Integrity
4.	<input type="radio"/>	Confidentiality

Advanced Scenarios		
5.	<input type="radio"/>	Sign and encrypt - X509 Authentication
6.	<input type="radio"/>	Sign and Encrypt - Anonymous clients
7.	<input type="radio"/>	Encrypt only - Username Token Authentication
8.	<input type="radio"/>	Sign and Encrypt - Username Token Authentication
9.	<input type="radio"/>	SecureConversation - Sign only - Service as STS - Bootstrap policy - Sign and Encrypt , X509 Authentication
10.	<input type="radio"/>	SecureConversation - Encrypt only - Service as STS - Bootstrap policy - Sign and Encrypt , X509 Authentication
11.	<input type="radio"/>	SecureConversation - Sign and Encrypt - Service as STS - Bootstrap policy - Sign and Encrypt , X509 Authentication
12.	<input type="radio"/>	SecureConversation - Sign Only - Service as STS - Bootstrap policy - Sign and Encrypt , Anonymous clients
13.	<input type="radio"/>	SecureConversation - Encrypt Only - Service as STS - Bootstrap policy - Sign and Encrypt , Anonymous clients
14.	<input type="radio"/>	SecureConversation - Encrypt Only - Service as STS - Bootstrap policy - Sign and Encrypt , Username Token Authentication

WSO<sub>2</sub> Identity Server

## User Entitlement

 Add New Entitlement Policy  Import New Entitlement Policy  Evaluate Entitlement Policies

Available Entitlement Policies

*No policies defined*



## Identity Server

## User Entitlement

+ Add New Entitlement Policy    Import New Entitlement Policy

Available Entitlement Policies

*No policies defined*

## Policy Editor

Source View

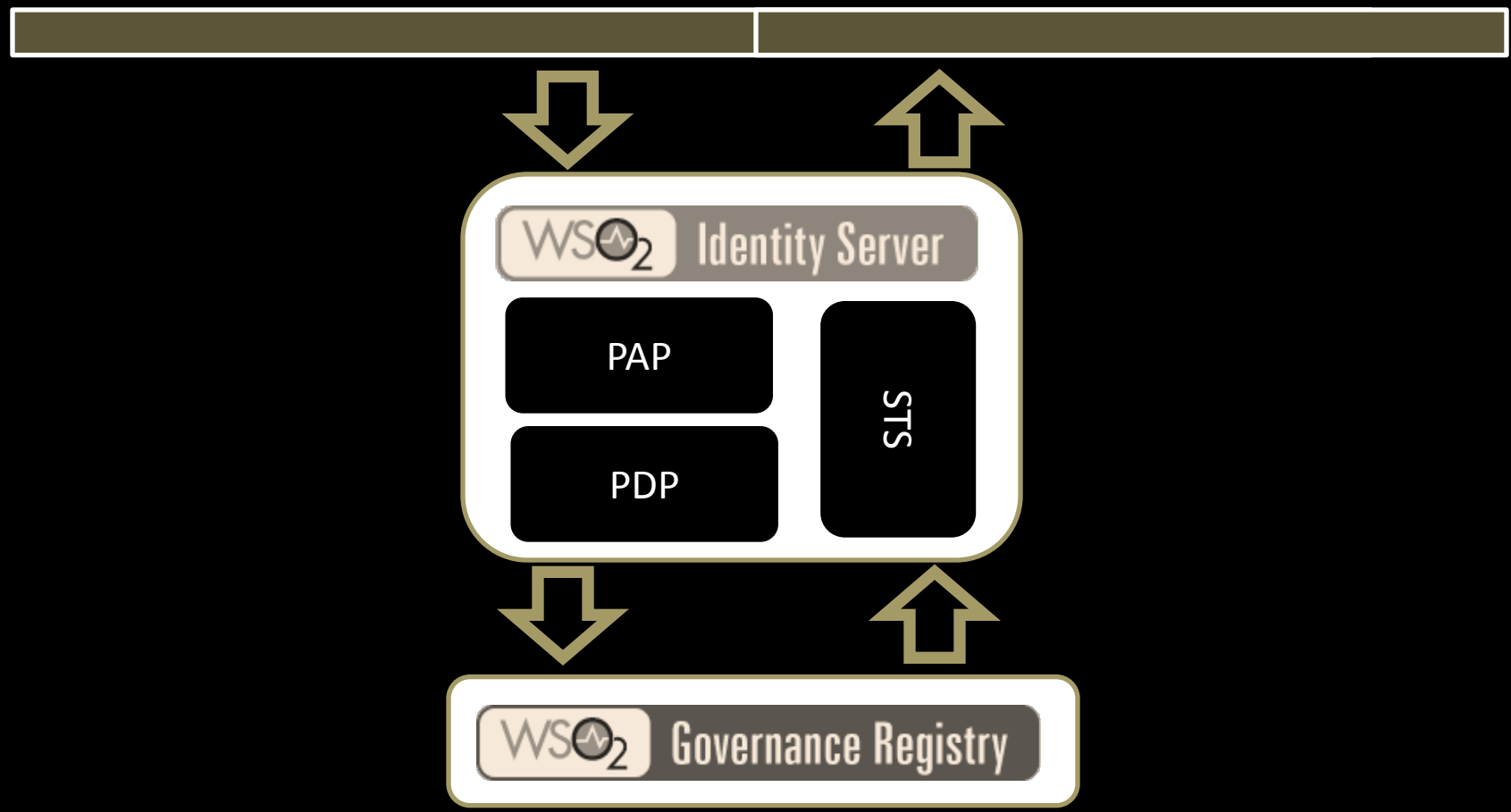
Design View

```

RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:first-applicable"
xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os">
  <Description>Sample XACML Authorization Policy</Description>
  <Target>
    <Subjects>
      <AnySubject/>
    </Subjects>
    <Actions>
      <AnyAction/>
    </Actions>
    <Resources>
      <Resource>
        <ResourceMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-match">
          <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">
http://localhost:8280/services/echo/</AttributeValue>
          <ResourceAttributeDesignator

AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
    <Rule Effect="Permit" RuleId="primary-user-rule">
      <Target>
        <Subjects>
          <AnySubject/>
        </Subjects>

```



WS-Security / WS-Trust / WS-Security Policy

Message Interceptor Gateway Pattern

WSO2 Governance Registry / WSO2 WSAS /  
WSO2 ESB / WSO2 Identity Server

*We have secured  
access to all our  
backend services...*



*Let's think of  
securing the front  
end....*





Yes... our backend services can be accessed through either with a direct client or with our web portal....

*Also we already  
have different  
web applications  
managed  
internally...*



*And it's hard to  
have different  
credentials to  
each web  
application....*



*Let's redesign  
authentication  
for all our web  
applications....*





*One more thing...  
we also need to  
give access to  
external users to  
the web portal as  
well...*

Too many passwords

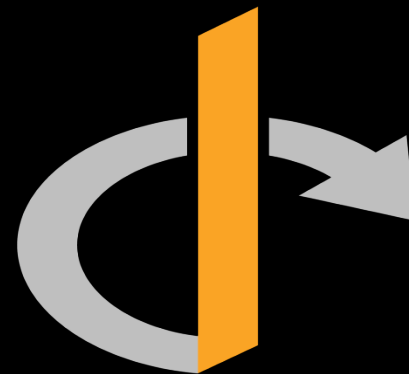
Single Sign On

Giving access to external domain users

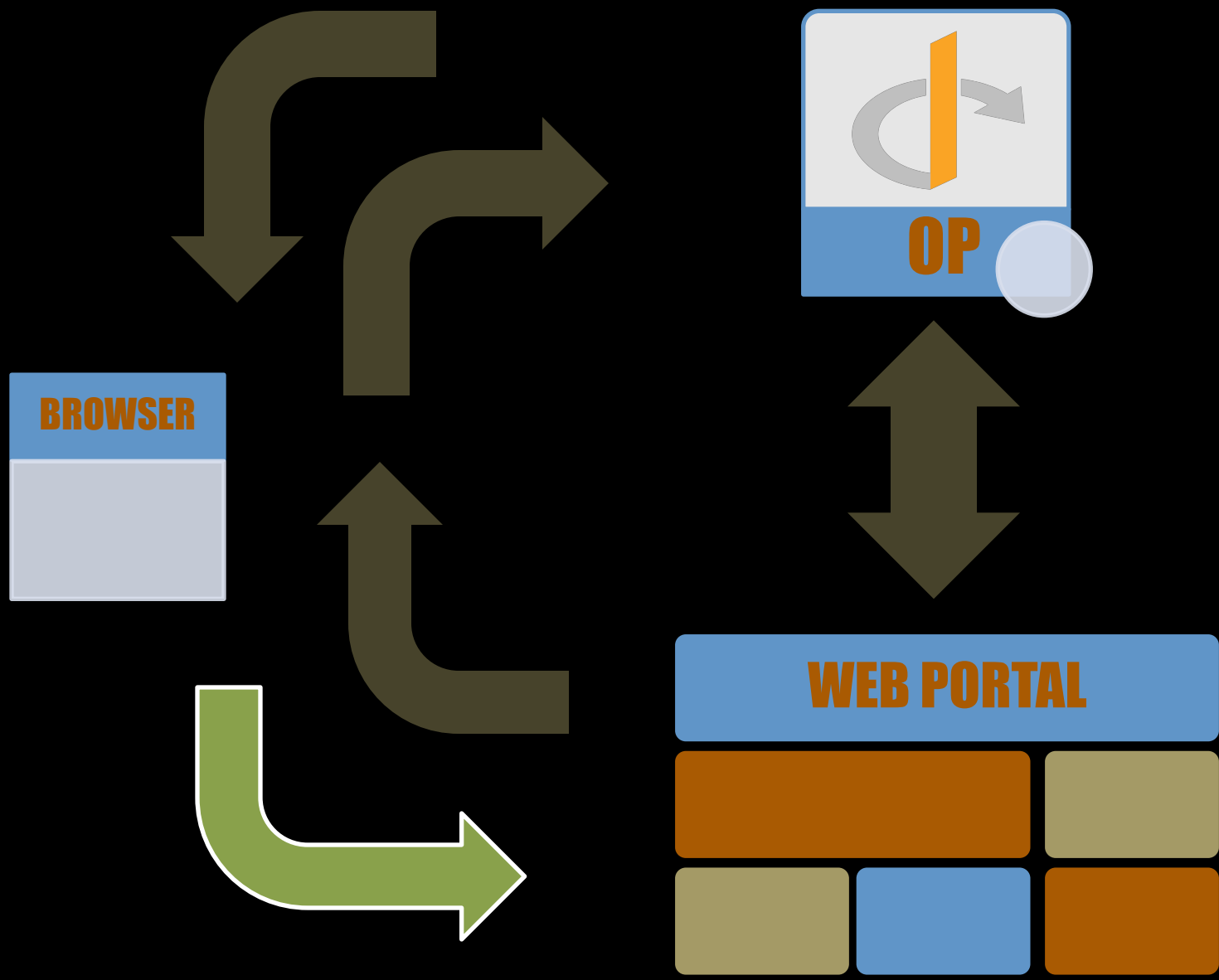
Decentralized Single Sign On

Single User Profile

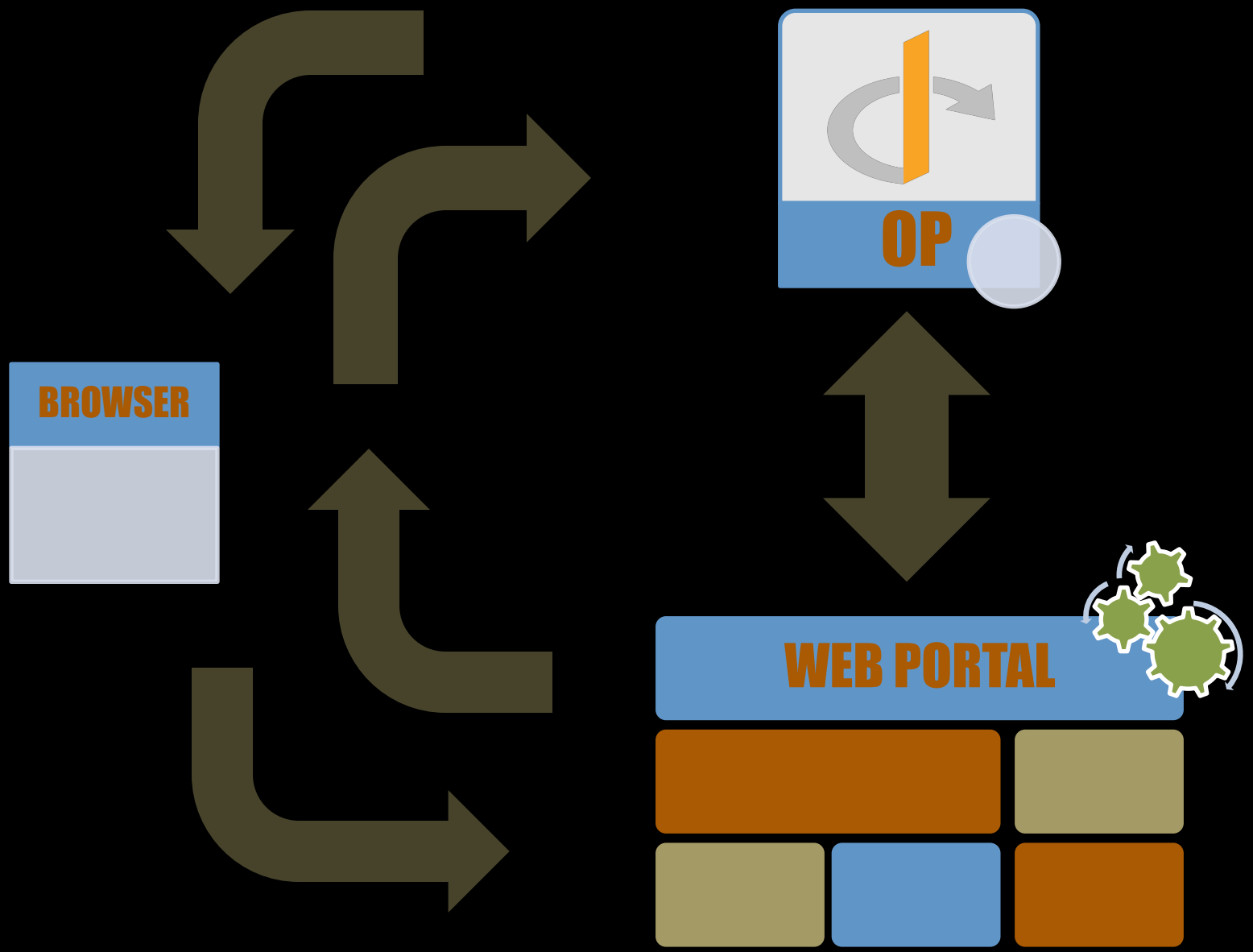
Identity Federation



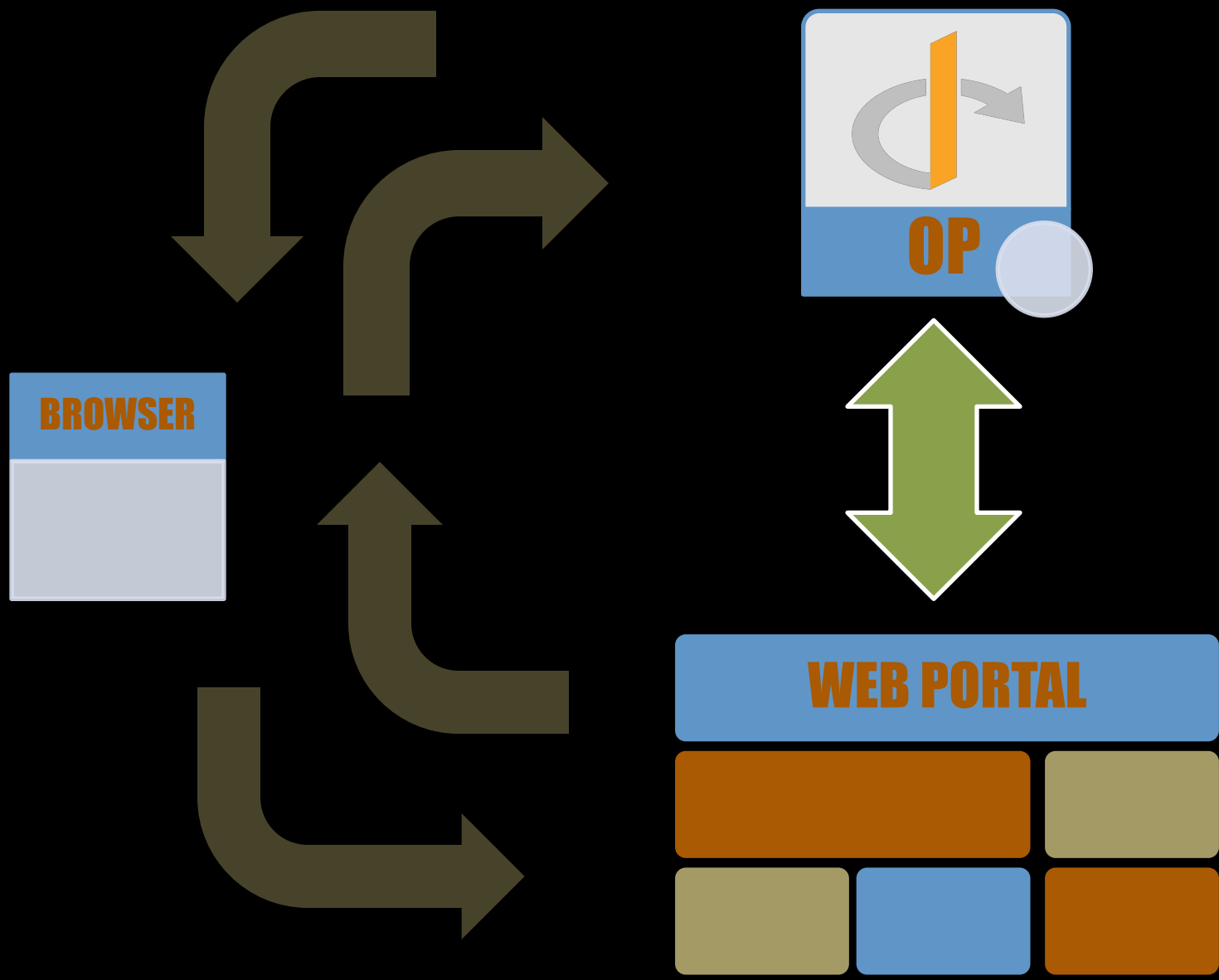
NOTES..... OPENID LOGIN FOR WEB PORTAL



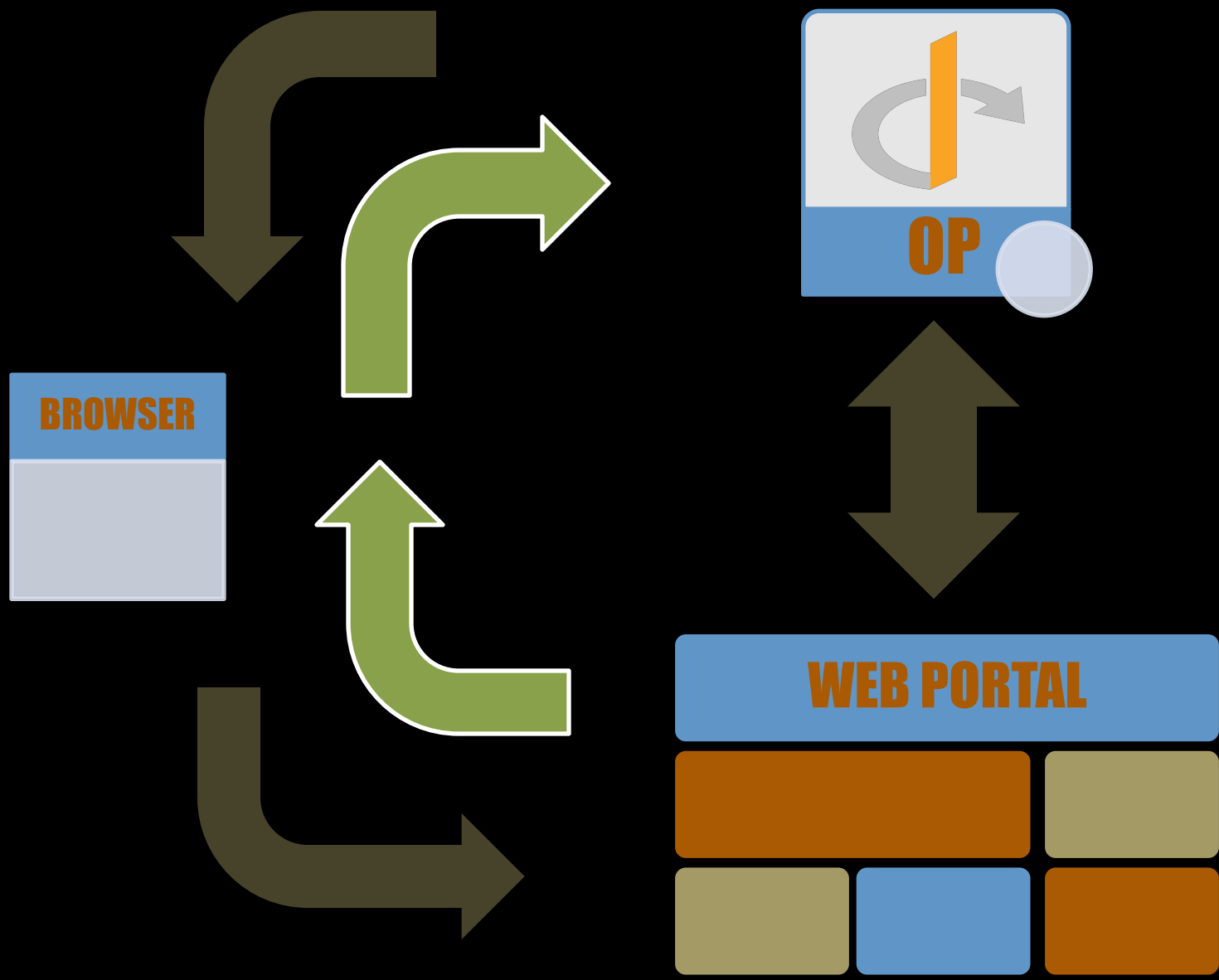
NOTES..... OPENID LOGIN FOR WEB PORTAL



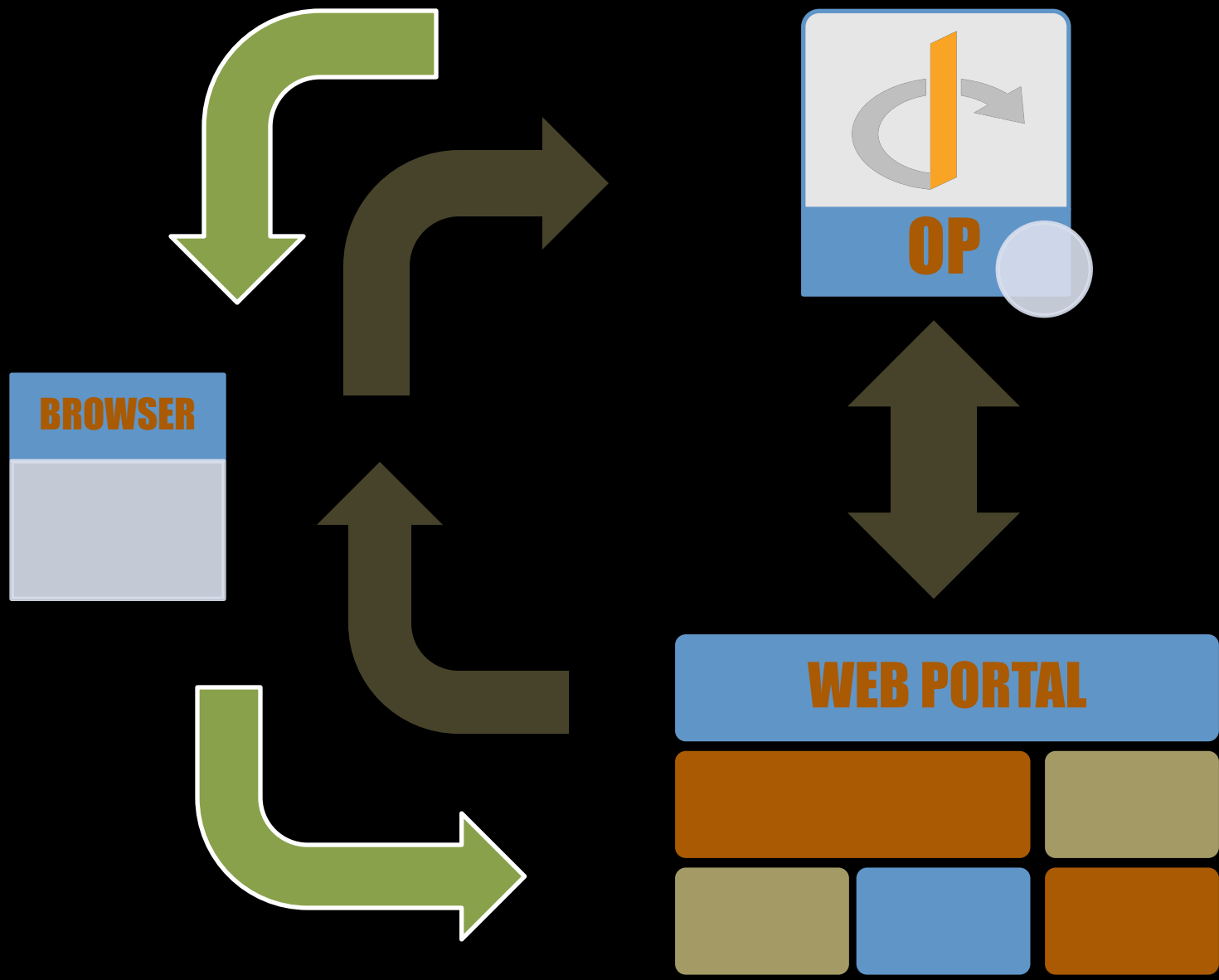
NOTES..... OPENID LOGIN FOR WEB PORTAL

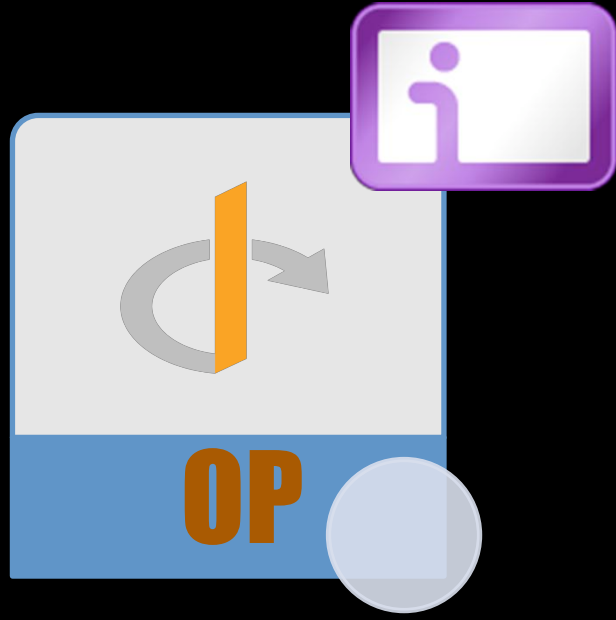


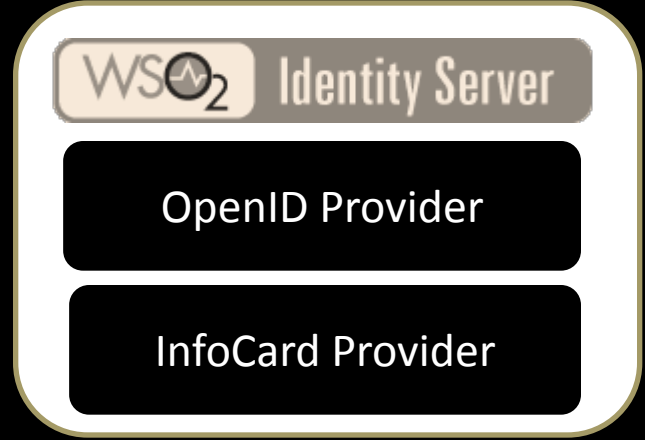
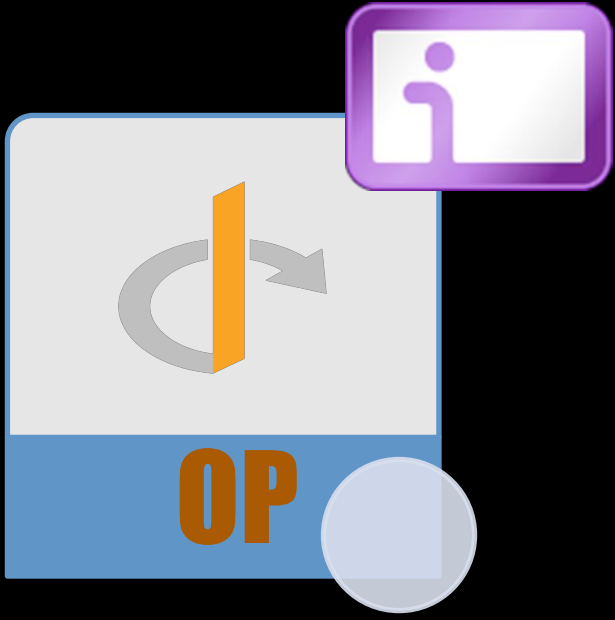
NOTES..... OPENID LOGIN FOR WEB PORTAL

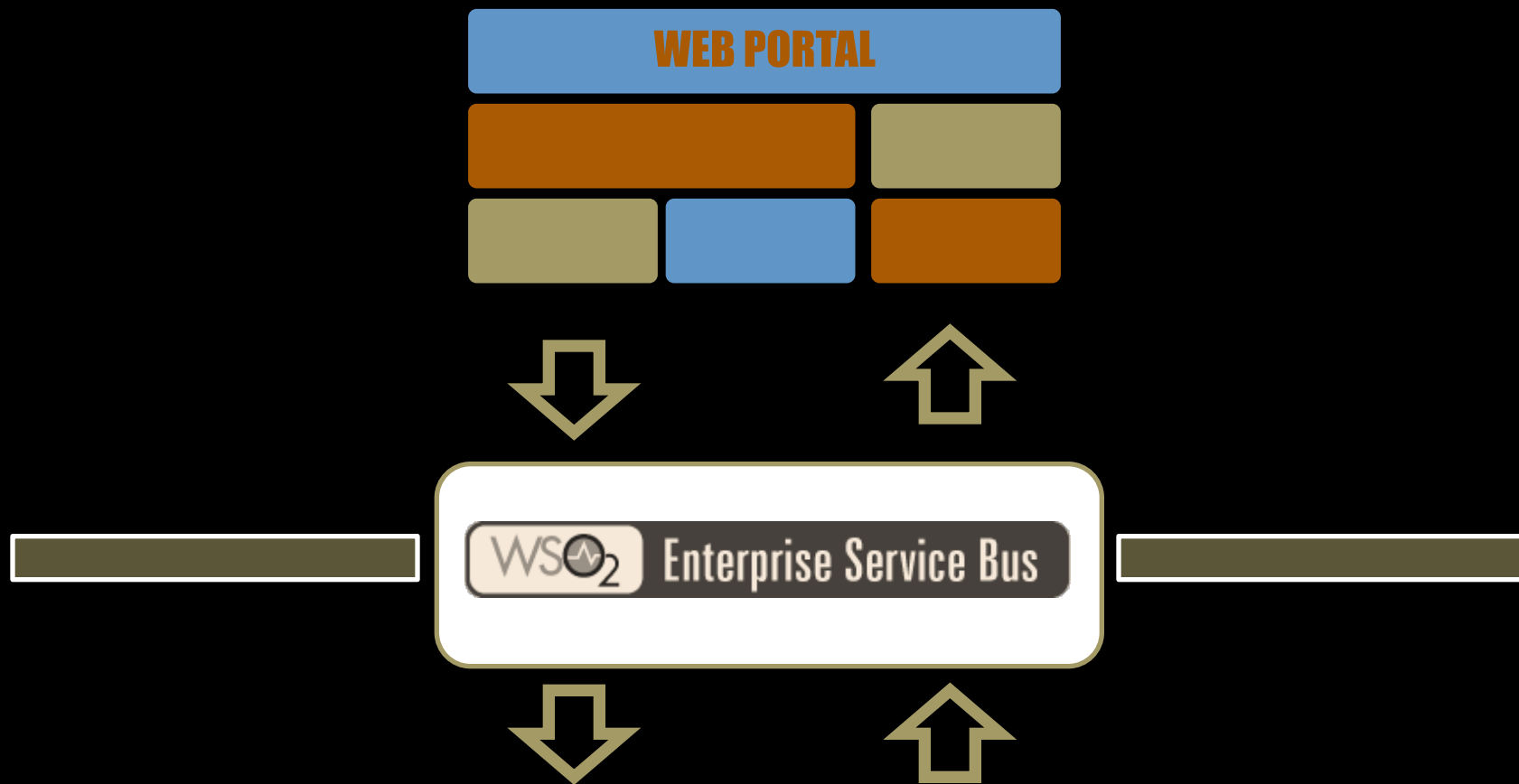


NOTES..... OPENID LOGIN FOR WEB PORTAL

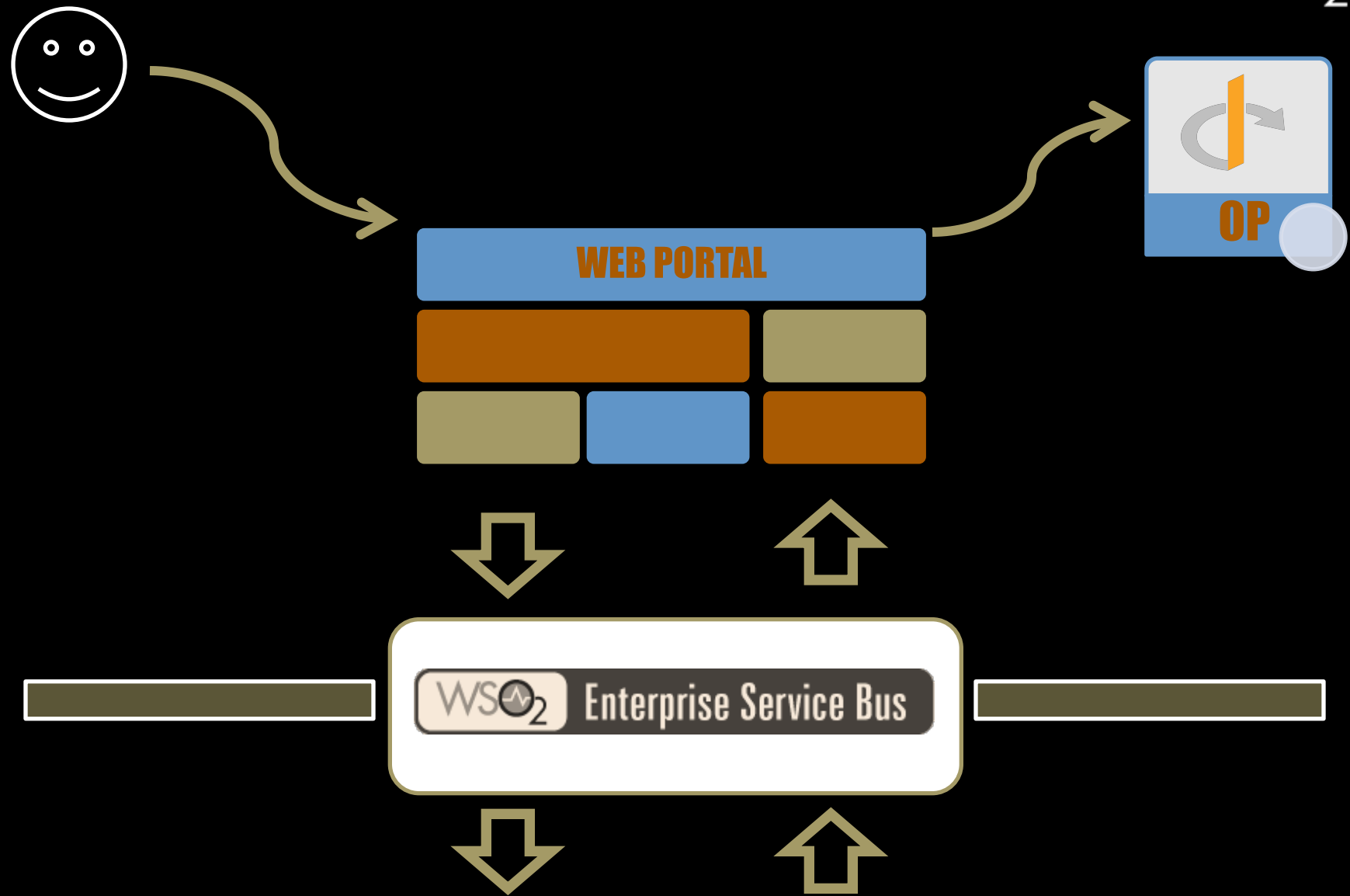








NOTES..... TRUSTED SUB SYSTEM



WS-Security / WS-Trust / WS-Security Policy

Message Interceptor Gateway Pattern

WSO2 Governance Registry / WSO2 WSAS /  
WSO2 ESB / WSO2 Identity Server

OpenID + InfoCard

Trusted Sub System Pattern

*<http://wso2.com>*

*<http://wso2.com/about/contact>*

*[bizdev@wso2.com](mailto:bizdev@wso2.com)*

*[prabath@wso2.com](mailto:prabath@wso2.com)*

Thank You...!!!