LOCKHEED MARTIN

We never forget who we're working for™

GCSS-AF SOA and Web Services

Infrastructure & Support

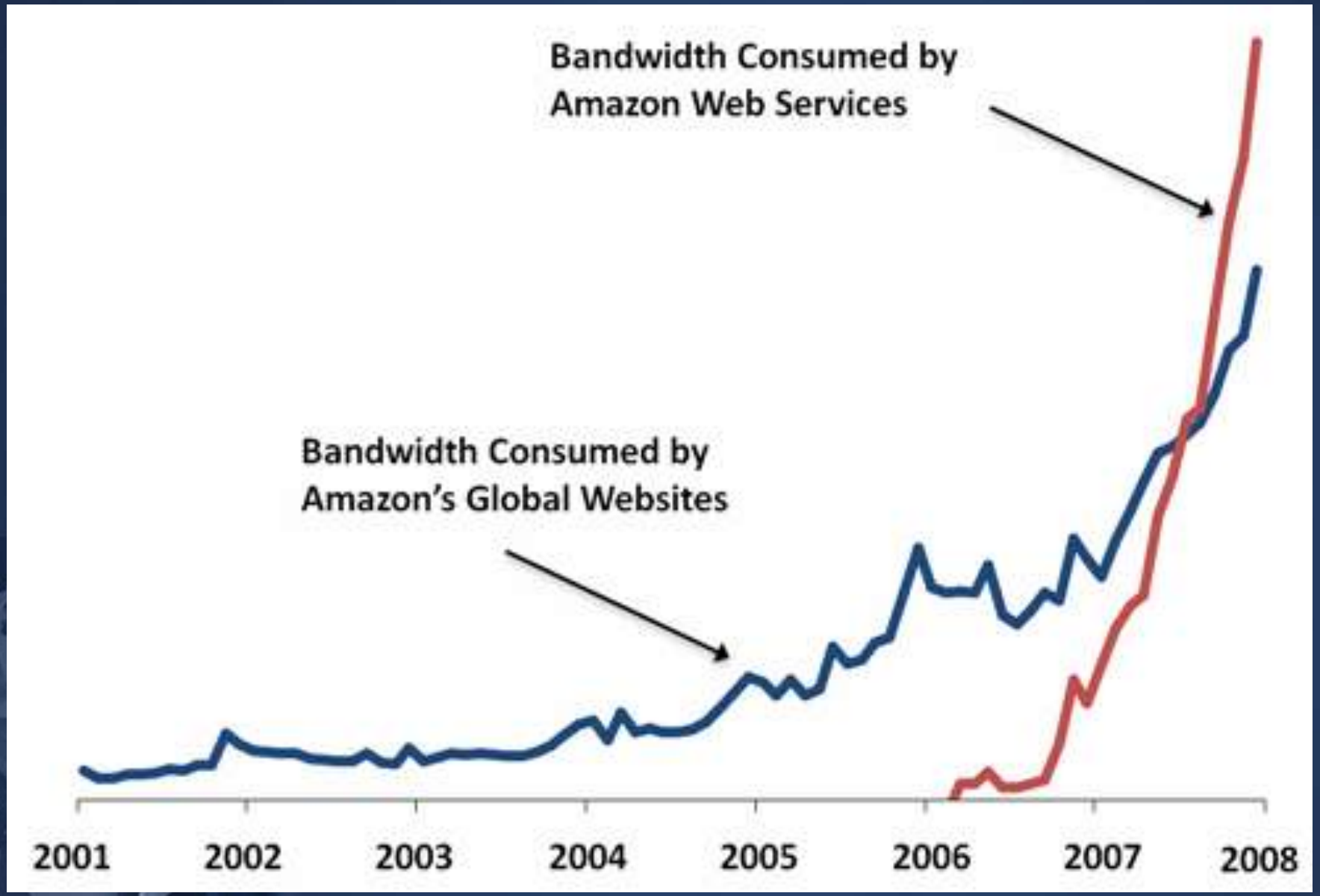
May 2009

Mike Acton
Lockheed Martin

Agenda

- Building the GCSS-AF SOA Infrastructure
- The SOA Journey
- GCSS-AF SOA Infrastructure
- Data Power SOA Appliance
- ESB: Web Services to/from Flat Files
- Web Services Security
- Consuming Web Services

The Web Services Effect



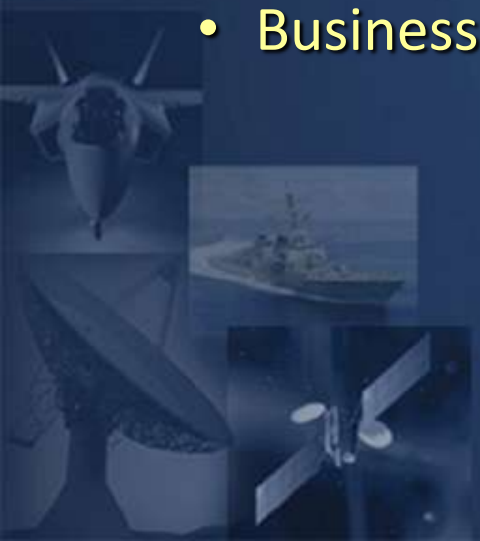
Building the GCSS-AF SOA Infrastructure

- GCSS-AF provides a **Modern Enterprise SOA and Information Integration Platform** that facilitates data exchange among modern and legacy systems

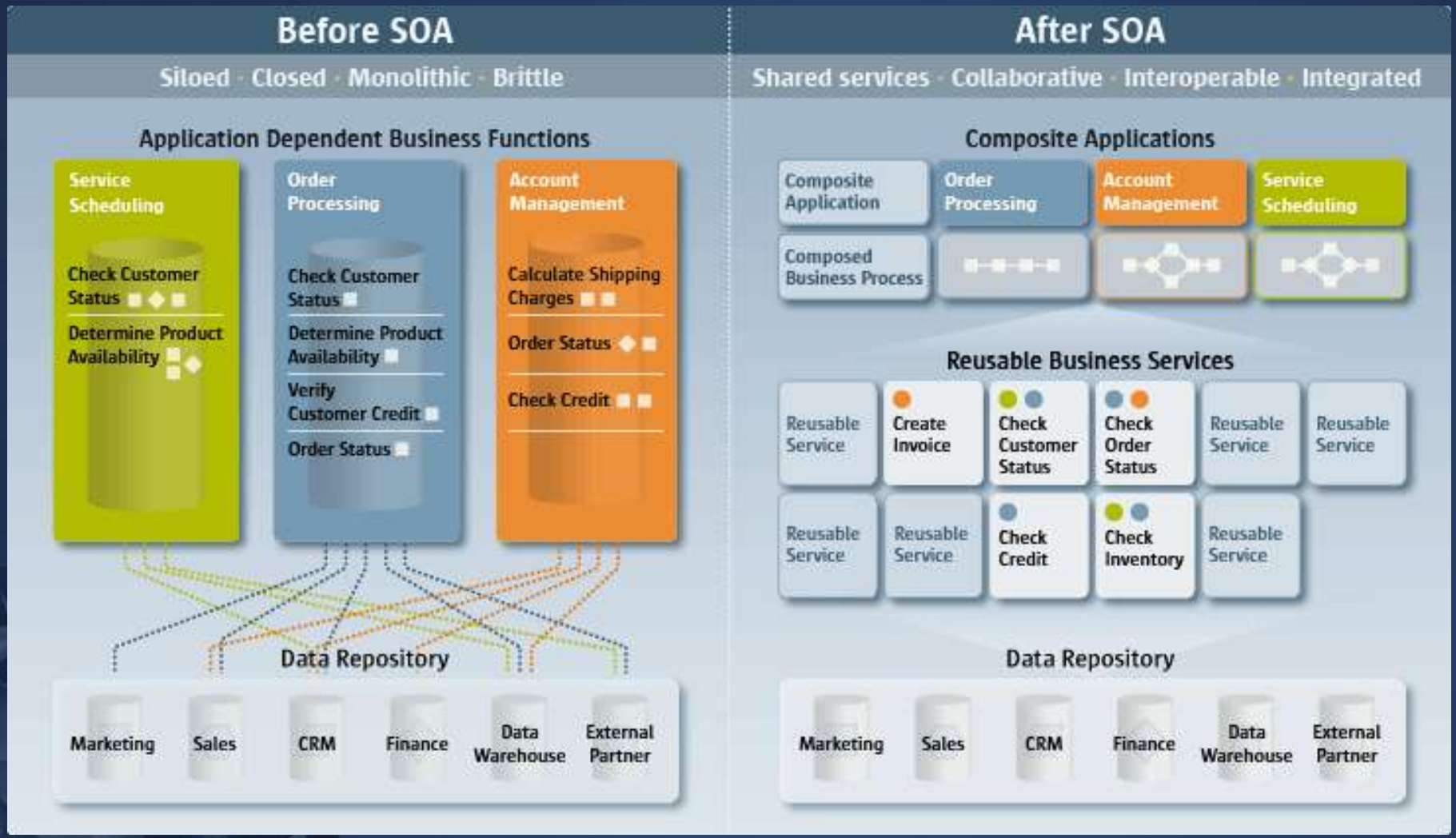
Building the GCSS-AF SOA Infrastructure

Drivers:

- GCSS-AF only hosts infrastructure services that applications need and actively use
- Business Case-Driven
- Implement GCSS-AF RMB Approved Requirements (including approved community Change Requests)
- Strategic Planning Process
  - SOA Maturity Models
  - Business Maturity Models



From Silos to Reusable Services

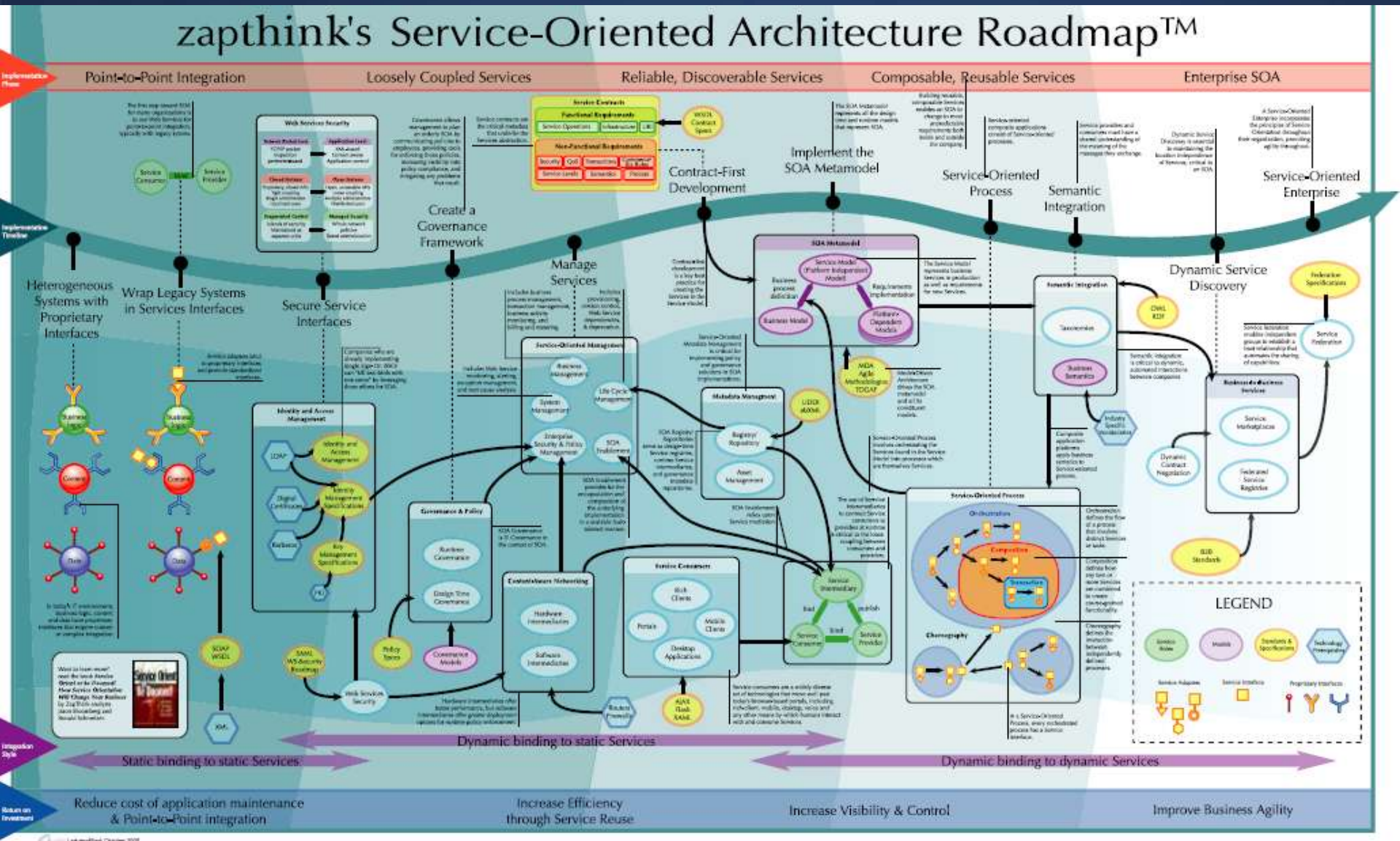


The SOA Journey

Stage Name	Business Silos	Standardized Technology	Optimized Core	Business Modularity
IT Capability	Local IT applications	Shared technical platforms	Enterprise-wide hardwired processes or databases	Plug & play business process modules
Business Objectives	ROI of local business initiatives	Reduced IT costs	Cost and quality of business operations	Speed to market; Strategic agility
Funding Priorities	Individual applications	Shared infrastructure services	Enterprise applications and data stores	Reusable business process components
Key Management Capability	Technology-enabled change management	Design and update of standards; funding shared services	Core enterprise process definition and measurement	Management of reusable business processes
Who Defines Applications	Local business leaders	IT & business unit leaders	Senior management and process leaders	IT, business and industry leaders
Key IT Governance Issues	Measure and communicate value	Establish local/ regional/global responsibilities	Align project priorities with architecture objectives	Define, source & fund business modules

Source: *Architecture as Strategy: Creating a Foundation for Business Execution*, J. Ross, P. Weill, D. Robertson, HBS Press, June 2006.

SOA Roadmap


zapthink's Service-Oriented Architecture Roadmap™
Point-to-Point Integration
Loosely Coupled Services
Reliable, Discoverable Services
Composable, Reusable Services
Enterprise SOA
Heterogeneous Systems with Proprietary Interfaces
Wrap Legacy Systems in Services Interfaces
Secure Service Interfaces
Create a Governance Framework
Manage Services
Contract-First Development
Implement the SOA Metamodel
Service-Oriented Process
Semantic Integration
Service-Oriented Enterprise
Dynamic Service Discovery
LEGEND
Static binding to static Services
Dynamic binding to static Services
Dynamic binding to dynamic Services
Reduce cost of application maintenance & Point-to-Point integration
Increase Efficiency through Service Reuse
Increase Visibility & Control
Improve Business Agility

GCSS-AF Conceptual SOA

Service Discovery includes the Repository and Registry

Automated workflow-based Governance is the KEY to success!

GCSS-AF SOA and Web Services Infrastructure

Governance			
	Design-Time Governance	EIT-GP (GCSS-AF EIT Governance Portal, aka: GCSS-AF.com)	
	Run-Time Governance	**N/A**	Ex: AmberPoint
Service Discovery			
	Web Service Repository	EIT-GP (GCSS-AF EIT Governance Portal, aka: GCSS-AF.com)	
	Web Services Registry	HP Systinet	
Service Security			
	Web Service Proxy	DataPower XI50 SOA Appliance	
	Access Management	Tivoli Access Manager	
	Identity Management	Tivoli Identity Manager	
	Federation	Tivoli Federated Identity Manager	
	Reduced Sign On (RSO) Integration	Tivoli Access Manager	
Service Mediation			
	Hardware Mediation	DataPower XI50 SOA Appliance	
	Software Mediation	ESB Mediation, Informatica	
Machine to Machine (M2M) Messaging			
	Enterprise Service Bus (ESB)	MQSeries, Message Broker	
	ESB Flat File Processing	ESB and DataPower XI50 SOA Appliance	
Business Process (BPEL) Orchestration			
	Business Process (BPEL) Engine	IBM Process Server, Oracle BPM Server	
Enterprise Service Management (ESM)			
	URI Monitoring	HP Site Scope	DISA Provided
	Web Services Monitoring	**N/A**	Ex: AmberPoint
	Web Services Runtime Policy Enforcement	**N/A**	Ex: AmberPoint
Web Application Servers			
	Java	IBM	
	Java	Oracle	
	.NET	Microsoft	
Extract, Transform, Load (ETL)			
	Informatica Suite		

NOTE: Complete GCSS-AF Product and Version Listing: https://www.gcss-af.com/cfs/outreach/tools/vdd/

Data Power XI50 SOA Integration and Security Appliance

OVERVIEW

XML-to-'Any' Conversion at Wirespeed
Integrated message-level security

X150 is a network device capable of transforming between disparate message formats, including binary, legacy, and XML, and providing message routing and security. XI50 can be used for cost-effective XML enablement of mainframes, wirespeed enterprise message buses, and enterprise application integration

Problems Solved:
WS Application Integration
Integrates SOA and Web services deployments
Transforms between disparate message formats (binary, legacy, XML, etc.)
Bridges wireline transport-level protocols (HTTP, MQ, FTP, JMS, TIBCO EMS, etc.)

Security Features:

- XML/SOAP Firewall - Filter on any content, metadata or network variables
- Web Application Firewall - Cookie rewriting, HTTP header injection /suppression
- Data Validation - Approve incoming/outgoing XML and SOAP at wirespeed
- XML Threat Protection - Mitigate risk from new and emerging application-layer threats
- XML Security - Encrypt and sign message/fields, provide message confidentiality and integrity
- Web Services Access Control - SAML, WS-Security, LDAP, Kerberos, LTPA, RADIUS, digital certificates, usernames /passwords, etc.
- MultiStep and XML/SOAP Routing - Sophisticated multi-stage pipeline
- Web Services Management - Secure Web services proxy, Service-Level Management (SLM)
- Transport Layer Flexibility - SSL acceleration; secure ↔ insecure zone bridging
- Service Virtualization - Reverse proxy masks end-points and back-end resources
- Configuration and Administration - Ease of use, integration for management
- Extensibility - Extending the base functionality using standards-based approaches

Web Services Security - Enterprise Proxy Approach

Service Messaging and Security Extensions

Supports internal, external and federated service access.

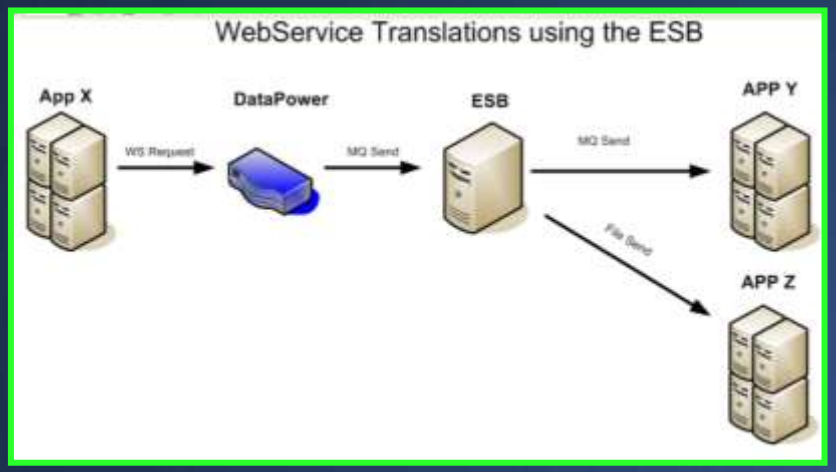
The entire process takes place in between the requesting client and hosted service. Clients remain unaware of the entire process, while the intended Web services do not have to be concerned with performing security functions.

Advantages:

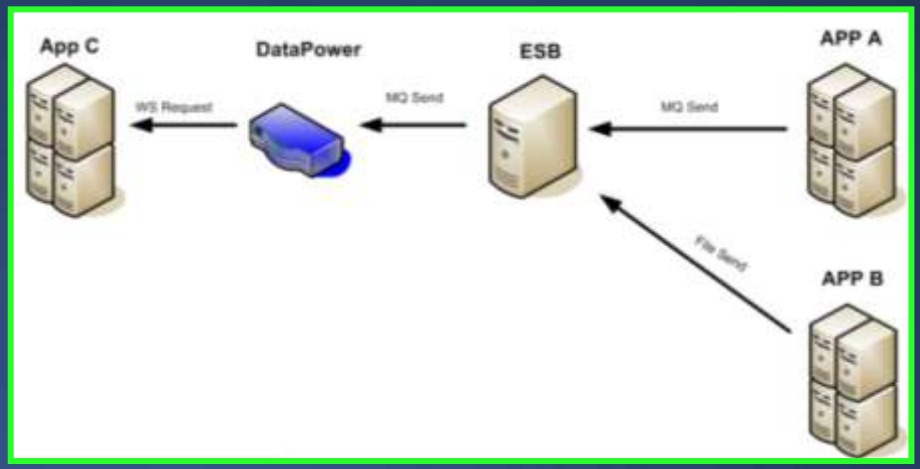
- Mitigates and protects against the major categories and many types of security threats to Web services and messaging.
- Security functions can be performed directly by the proxy rather than being implemented in the actual applications and services.
- Security functions are hidden from clients, who remain unaware of the process.
- Bi-directional security protects both the client and service provider.
- Web services are concealed by the proxy, which only establishes a connection for authorized clients.
- Web service providers are not affected by the function of the proxy, as it is an independent entity.
- Proxies can be easily added and removed to and from an architecture depending on system and service needs.
- The proxy promotes loose coupling and the separation of distinct services.

Modern/Legacy Mediation: Web Services to/from Flat Files

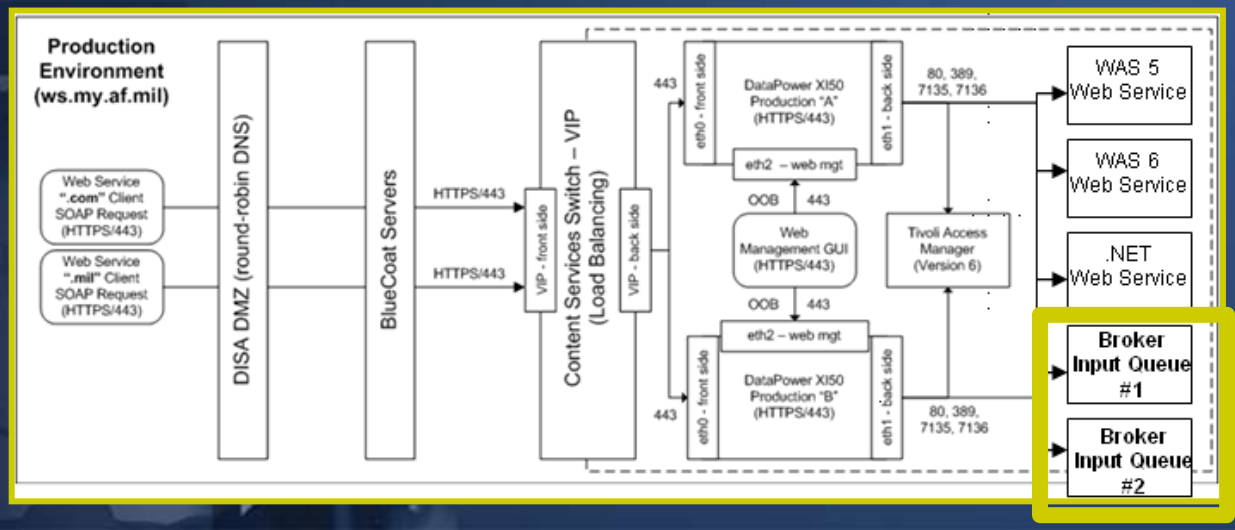
Change Request #1963



Web Service Sent to Flat File Recipient (w/data mediation)



Flat File Sent to Web Service Recipient (w/data mediation)

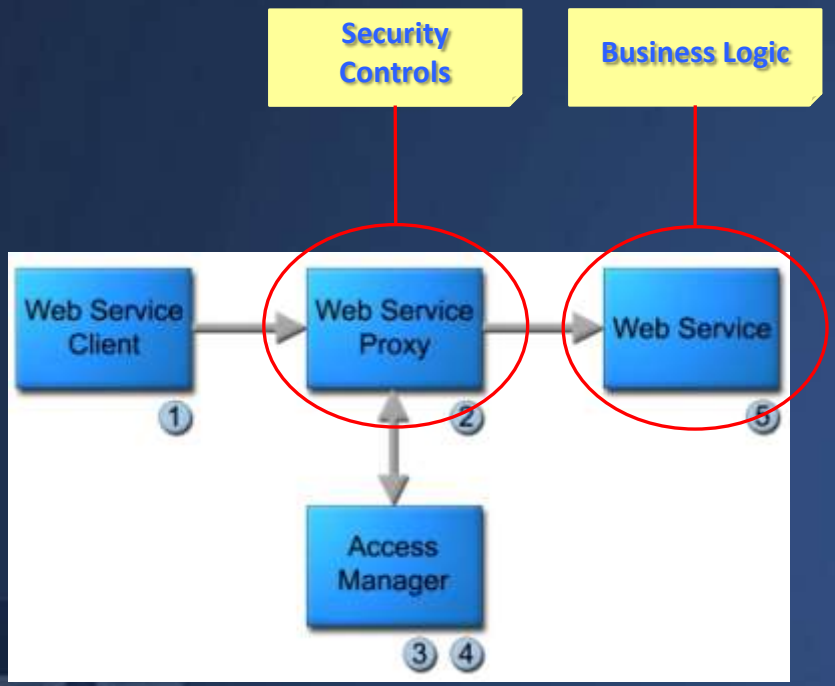


ESB

Production Configuration:

- ESB
- DataPower XI50s
- Authentication and Access
- Web Service Management GUI Tool
- Web App Servers (Java and .NET)

Web Services Security Proxy Sequence of Events



When a SOAP message is sent from a client to a Web service application protected by the GCSS-AF Web service proxy, the following process occurs each time:

1) The Web service client prepares the SOAP request, populating the SOAP header with a X.509 binary security token.

2) Upon receipt of the message, the Web service proxy examines the X.509 binary security token signature for authenticity. If the signature is invalid, indicating some form of tampering occurred, a SOAP fault is returned to the Web service client. If the signature is valid, the next step in the process occurs.

3) A valid X.509 certificate associated with the X.509 binary security token is used to authenticate a service user through an access manager, which is an integrated COTS solution used for authorization and identity management. If the service user cannot be identified, a SOAP fault is returned to the Web service client.

4) If the service user is identified, the access manager begins authorizing the client based on the Web Service method. If the service user is not authorized to access the Web service method, a SOAP fault is returned to the Web service client.

5) If the service user IS AUTHENTICATED AND AUTHORIZED to call the Web service method, the Web service proxy calls the protected Web service method, and the data is returned to the Web service client.

REST Service Support

- **REST services are more free-form than Web Services and don't provide a WSDL which serves as the basis for the web service proxy configuration**
  - We lose the ws-security extensions and would therefore require mutual authn on all REST connections
  - XML validation and acceleration provided by DP would also be impacted

- **Bottom line is that we can support REST via DataPower**
  - Some design work is needed for those services since the wsdl and soap elements shown on the previous slides are not supported by REST services
    - It may make sense to secure REST services via WebSEAL security junctions

Consuming Web Services (5 Steps)

Governance Process

1. Find It!

- EIT-Governance Portal Services Repository (www.gcss-af.com)
- Workplans (Governance/Permissions)

2. View It!

- Descriptions, Guides
- WSDL
- XML Schemas

3. Secure It!

- X.509 Certificate (Authentication)
  - Use AF Process for obtaining certificate
- TAM Roles (Authorization)

4. Connect It!

- Java and .NET Web Services
- SOAP

5. Consume It!

- XML Payload
- XML Schemas
- Transformations

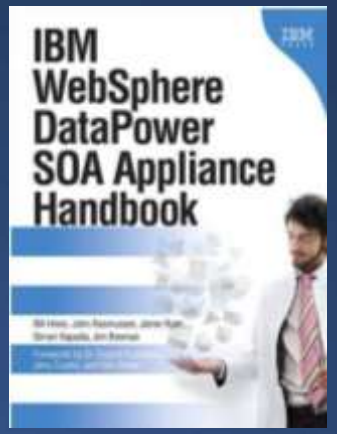
Contact Outreach Support

Developer's Tips

- Contact Outreach to discuss your data needs
- Outreach has Web Service and DataPower Expertise
- Complete a Work Plan on EIT-GP
- Open Change Requests and Follow Up w/GCSS-AF if you need additional capabilities
- Develop and Test your Web Service in the CIE
  - Full installation of GCSS-AF in CIE for development/testing
    - Including ESB and DataPower XI50

Resources

- An Introduction to IBM Rational Application Developer
  - ISBN-13: 978-1-931182-22-5
- IBM WebSphere DataPower SOA Appliance Handbook
  - ISBN-13: 978-0-13-714819-6
- IBM SOA and Web Services Developer's Website:
  - http://www.ibm.com/developerworks/webservices
- IBM SOA Sandbox
  - http://www.ibm.com/developerworks/downloads/soasandbox/?S_TACT=105AGX04&S_CMP=HP



BACKUP

DataPower Web Service Proxy Test Example

```
public class  GcssAfWsProxyTest {

        private static EngineConfiguation createClientconfig() {
                SimpleProvider clientConfig = new SimpleProvider();
                Handler securityHandler = (Handler)new WSDoAllSender();
                securityHandler.setOption(WSHandlerConstants.ACTION, WSHandlerConstants.SIGNATURE);
                securityHandler.setOption(WSHandlerConstants.USER,“gcss-af-ws-proxy-test-user");
                securityHandler.setOption(WSHandlerConstants.PASSWORD_TYPE, WSConstants.PW_TEXT);
                securityHandler.setOption(WSHandlerConstants.SIB_PROP_FILE, “client_crypto.properties");
                securityHandler.setOption(WSHandlerConstants.SIG_KEY_ID, "DirectReference");
                securityHandler.setOption(WSHandlerConstants.PW_CALLBACK_CLASS, "smartdemo.PWCallback");
                SimpleChain reqHandler = new SimpleChain();
                SimpleChain resHandler = new SimpleChain();
                reqHandler.addHandler(securityHandler);
                respHandler.addHandler(securityHandler);
                Handler pivot = (Handler)new HTTPSender();
                Handler transport = new SimpleTargetedChain(reqHandler, pivot, respHandler);
                clientConfig.deployTransport(HTTPTransport.DEFAULT_TRANSPORT_NAME,transport);
                return clientCofig;
        }
        public static void main(String[] args) {
                try {
                        System.setProperty( "javax.net.ssl.trustStore", "jssecacerts");
                        System.setProperty( "javax.net.ssl.trustStorePassword", "changeit");
                        java.net.URL url = new java.net.URL ( "url to web service");
                        SmartWebServicesLocator locator = new SmartWebServicesLocator();
                        EngineConfiguration clientConfig = createClientConfig();
                        locator.setEngineConfiguration(clientConfig);
                        locator.setEngine(new AxisClient(clientConfig));
                        SmartWebServiceSoap client = locator.getSmartWebServicesSoap(url);
                        String response = client.getAcquisitionGroupList(498);
                        System.out.println(response);
                } catch ( Exception e ) {
                        e.printStackTrace();
                }
        }
}
```

SOAP Request, WS-Security 1.0 Binary Security Token with Client Signature

```
<soapenv:Envelope xmlns:mil="http://Mil.AF.Smart.WebServices" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsse:BinarySecurityToken EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" wsu:Id="CertId-19551676"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">…</wsse:BinarySecurityToken>
      <ds:Signature Id="Signature-8092433" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
          <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
          <ds:Reference URI="#id-13592751">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>BvvBBtuhuPfbfOFNfQJEhjn2WfY=</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>

<ds:SignatureValue>pClxgJqFscP/Xjsc43Uj4nXrzp32g/EQzCpwehY0hd1bT4+P7OEcD49ccDs1tMIvQ8cqXBBM9TbxPw6ze7tPUiaqrFA3hUwLzaQi7BMrRtZDjN/lq
3Cp7m/3Ly1VmREXpryROw25ms12JQwBWcErpaqxEovudHe7ffy9jYBwWoo=</ds:SignatureValue>
        <ds:KeyInfo Id="KeyId-24401596">
          <wsse:SecurityTokenReference wsu:Id="STRId-15724536" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
            <wsse:Reference URI="#CertId-19551676" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"/>
          </wsse:SecurityTokenReference>
        </ds:KeyInfo>
      </ds:Signature>
      <wsu:Timestamp wsu:Id="Timestamp-25752838" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsu:Created>2008-06-24T19:10:27.741Z</wsu:Created>
        <wsu:Expires>2008-06-24T19:15:27.741Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body wsu:Id="id-13592751" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <mil:GetAcquisitionGroupList>
      <mil:AcquisitionGroupListRequestData>
        <mil:RootAcquisitionGroupID>498</mil:RootAcquisitionGroupID>
      </mil:AcquisitionGroupListRequestData>
    </mil:GetAcquisitionGroupList>
  </soapenv:Body>
</soapenv:Envelope>
```

WS-Security Binary Security Token

Signature Reference

Signed Body

SOAP Response, WS-Security 1.0 Binary Security Token with ServerSignature

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soapenv:Header xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsu:Timestamp wsu:Id="Timestamp-a4fc714d-bb0f-4dbb-bda7-a0bce11c7762" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
        <wsu:Created>2008-06-24T19:12:12Z</wsu:Created>
        <wsu:Expires>2008-06-24T19:17:12Z</wsu:Expires>
      </wsu:Timestamp>
      <wsse:BinarySecurityToken wsu:Id="SecurityToken-52d5e357-c71f-4e26-9d58-661afca1a3ae" EncodingType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
x509-token-profile-1.0#X509v3" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">…</wsse:BinarySecurityToken>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
          <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
          <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
          <Reference URI="#Body-adec4fad-7f93-4189-be7e-5231b8c44c94">
            <Transforms>
              <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <DigestValue>yOkGiRm2xlx97APjTl8lC38luLE=</DigestValue>
          </Reference>
</SignedInfo>
<SignatureValue>hW00aGrz3p65s5Ouno0obtMc+mCNcbetwuIPGv2VWDCY09jpHgcDb6Mr4QxQbPx4ArMFSl669TlOl8z/jQDr2xqv/3TrDH16jVWq5qOzqoulqs8wzd
mnv5J3HRsaaX2ElNN9hgB88qr7nEVFzKCJ/xbpRMwQ3O2Sr/oHQGUTR5Q=</SignatureValue>
        <KeyInfo>
          <wsse:SecurityTokenReference xmlns="">
            <wsse:Reference URI="#SecurityToken-52d5e357-c71f-4e26-9d58-661afca1a3ae" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-x509-token-profile-1.0#X509v3"/>
          </wsse:SecurityTokenReference>
        </KeyInfo>
      </Signature>
    </wsse:Security>
  </soapenv:Header>
  <soap:Body wsu:Id="Body-adec4fad-7f93-4189-be7e-5231b8c44c94" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <GetAcquisitionGroupListResponse xmlns="http://Mil.AF.Smart.WebServices">
      <AcquisitionGroupListResponseData>…</AcquisitionGroupListResponseData>
    </GetAcquisitionGroupListResponse>
  </soap:Body>
</soap:Envelope>
```

WS-Security Binary Security Token

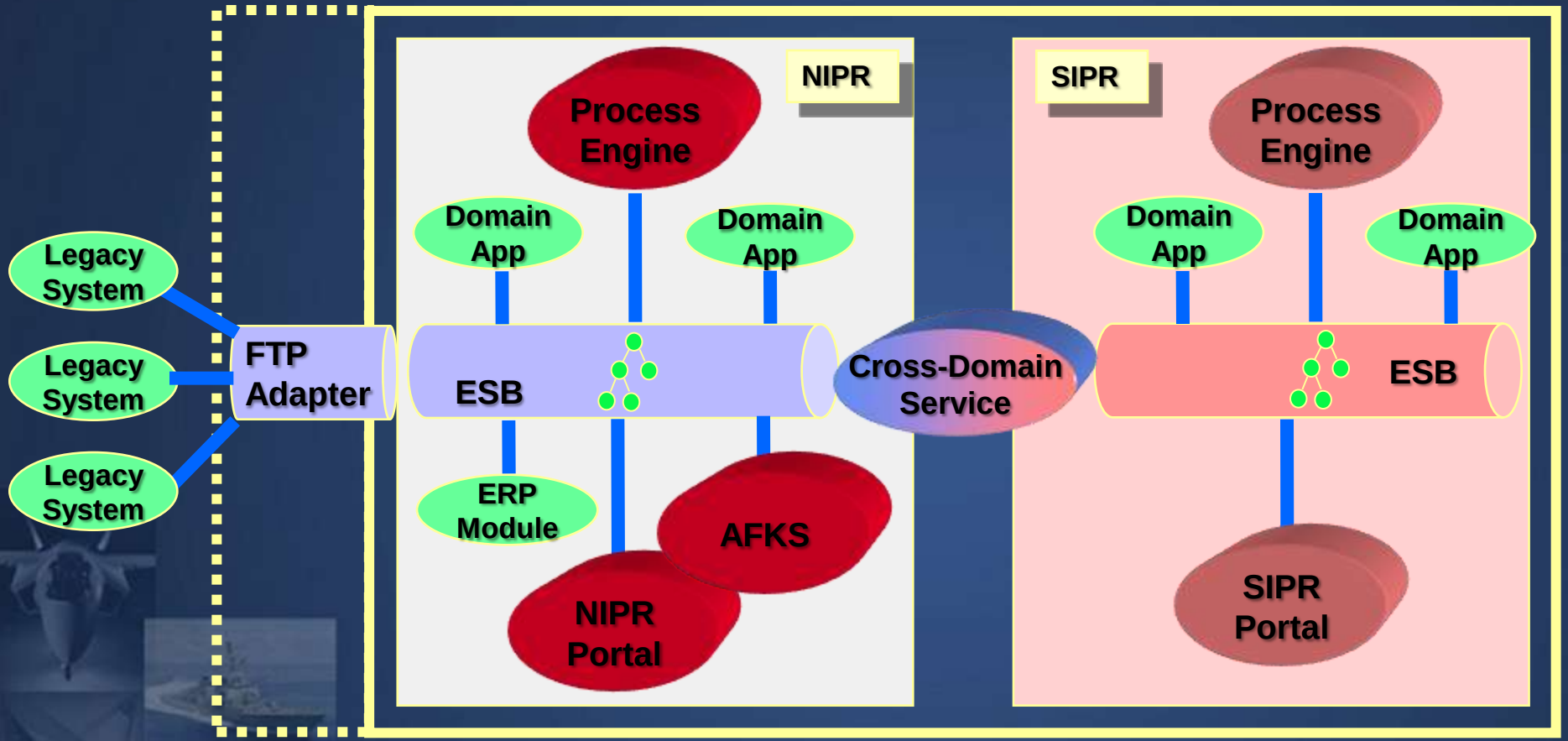
Signature Reference

Signed Body

Echo Web Service Example

```
package com.ibm.was.wssample.sei.echo;
import java.net.MalformedURLException;
import java.net.URL;
import javax.xml.namespace.QName;
import javax.xml.ws.Service;
import javax.xml.ws.WebEndpoint;
import javax.xml.ws.WebServiceClient;
/**
 * This class was generated by the JAXWS SI.
 * JAX-WS RI 2.0_01-b15-fcs
 * Generated source version: 2.0
*/
@WebServiceClient(name = "EchoService", targetNamespace = "http://com/ibm/was/wssample/sei/echo/", wsdlLocation =
"WEB-INF/wsdl/Echo.wsdl")
public class EchoService
    extends Service
{
    private final static URL ECHOSERVICE_WSDL_LOCATION;
    static {
        URL url = null;
        try {
            url = new URL("file:/WEB-INF/wsdl/Echo.wsdl");
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        ECHOSERVICE_WSDL_LOCATION = url;
    }
    public EchoService(URL wsdlLocation, QName serviceName) {
        super(wsdlLocation, serviceName);
    }
    public EchoService() {
        super(ECHOSERVICE_WSDL_LOCATION, new QName("http://com/ibm/was/wssample/sei/echo/", "EchoService"));
    }
    /**
     * @return
     * returns EchoServicePortType
     */
    @WebEndpoint(name = "EchoServicePort")
    public EchoServicePortTypeClient getEchoServicePort() {
        return (EchoServicePortTypeClient)super.getPort(new QName("http://com/ibm/was/wssample/sei/echo/",
                "EchoServicePort"), EchoServicePortTypeClient.class);
    }
}
```

ESB Architecture



DataPower XI50 Reference Architecture