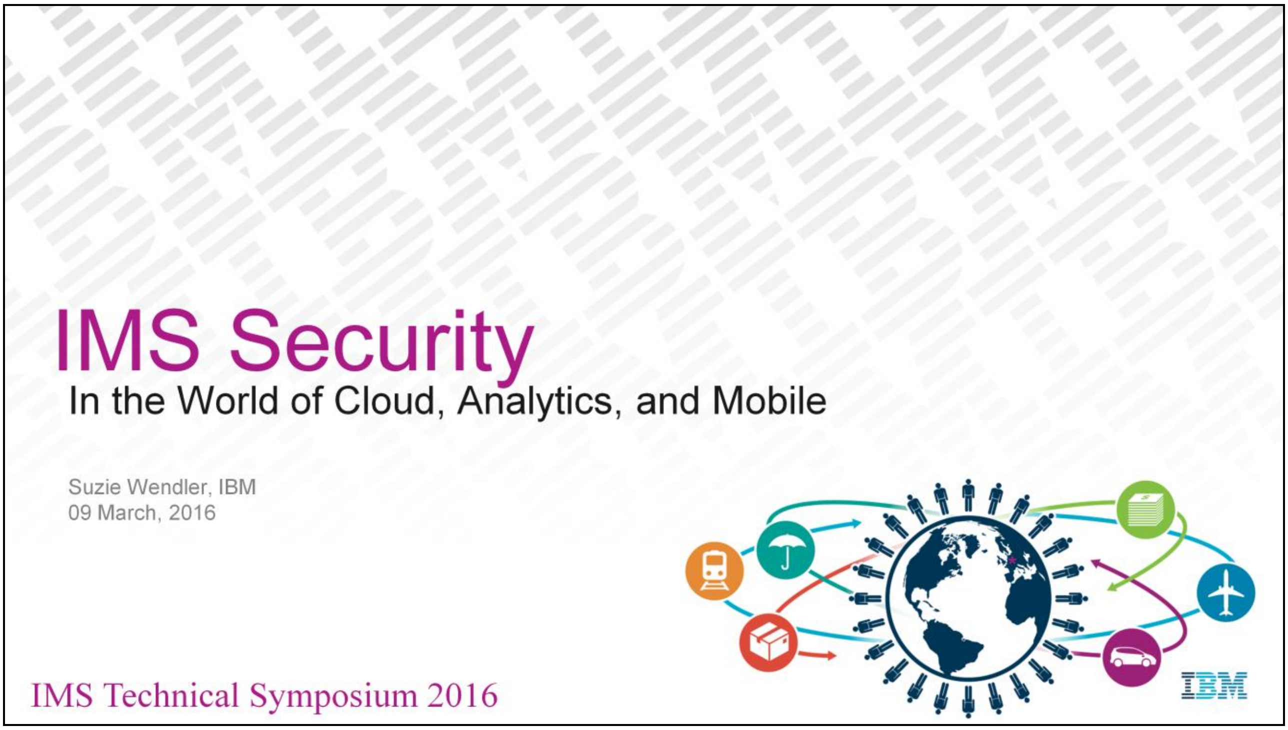
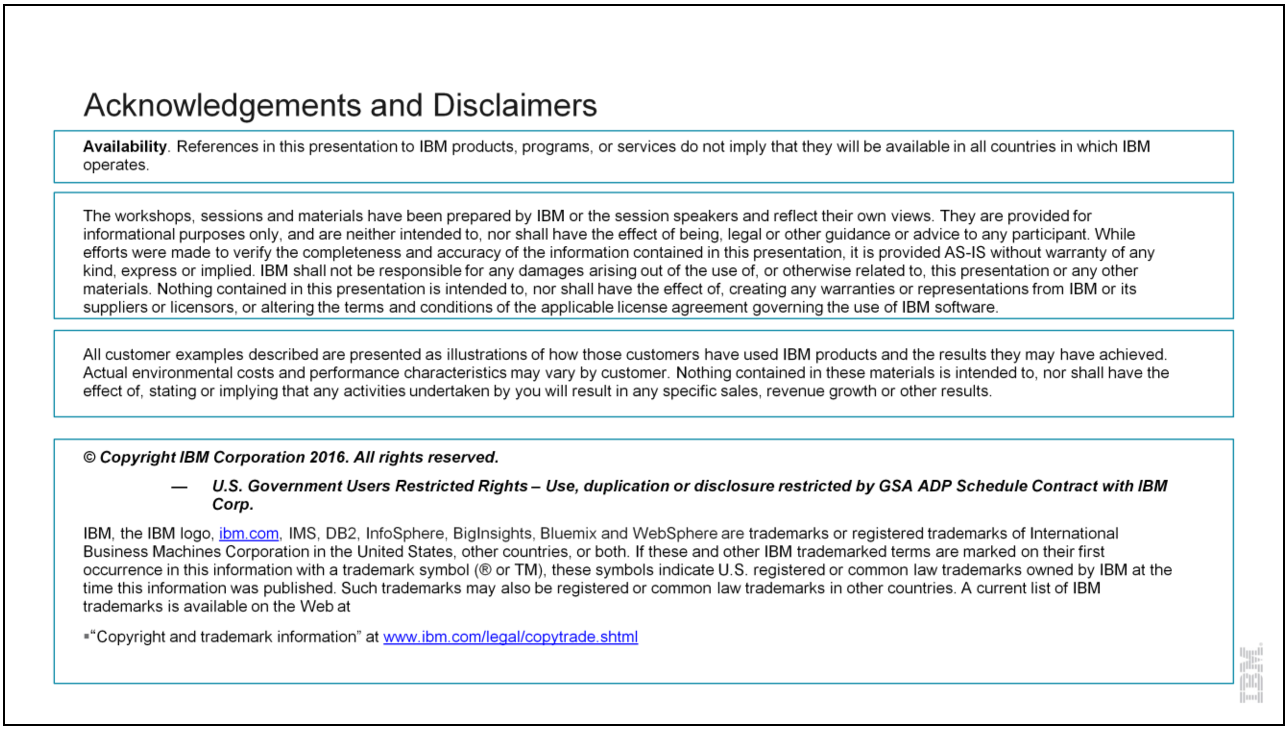


IMS Security

In the World of Cloud, Analytics, and Mobile

Suzie Wendler, IBM
09 March, 2016

IMS Technical Symposium 2016



Acknowledgements and Disclaimers

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

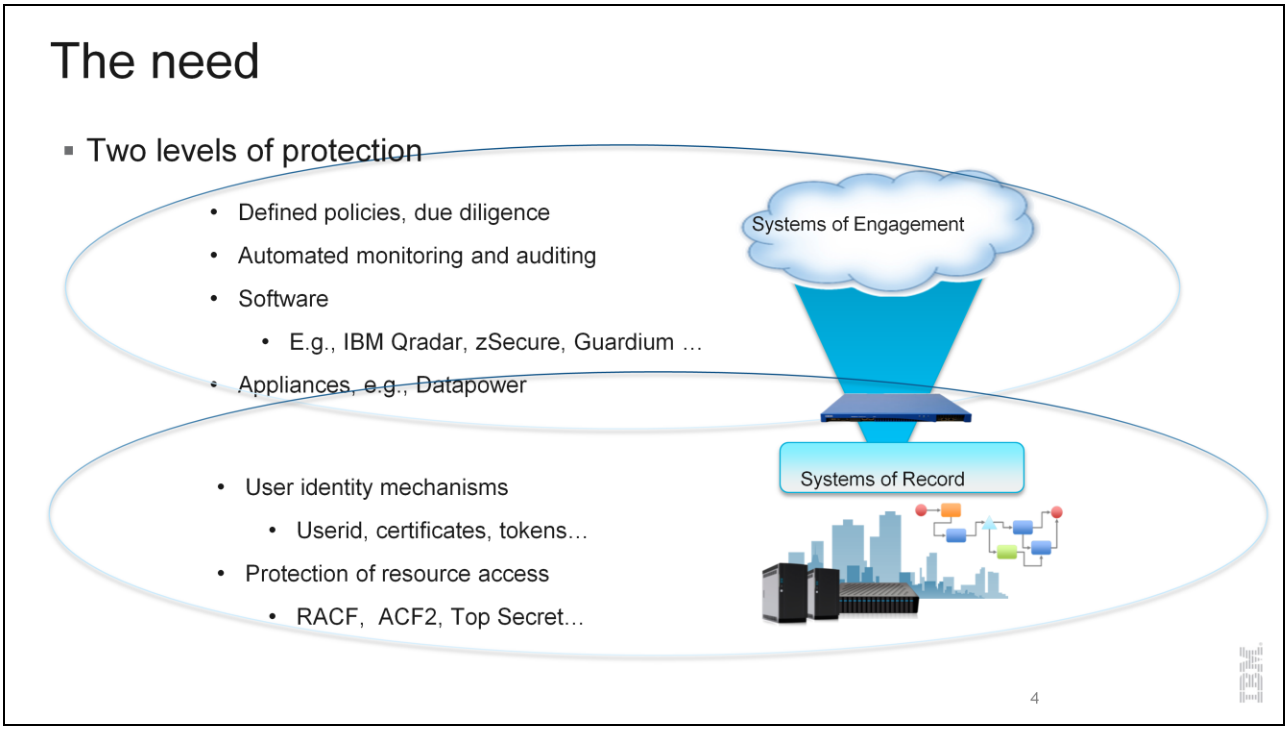
All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

The Evolution

- The 4 pillars of the new computing environment

- Benefits
  - Cost savings
  - Rapid procurement and deployment
  - Support for new technologies
  - Scalability, performance, flexibility

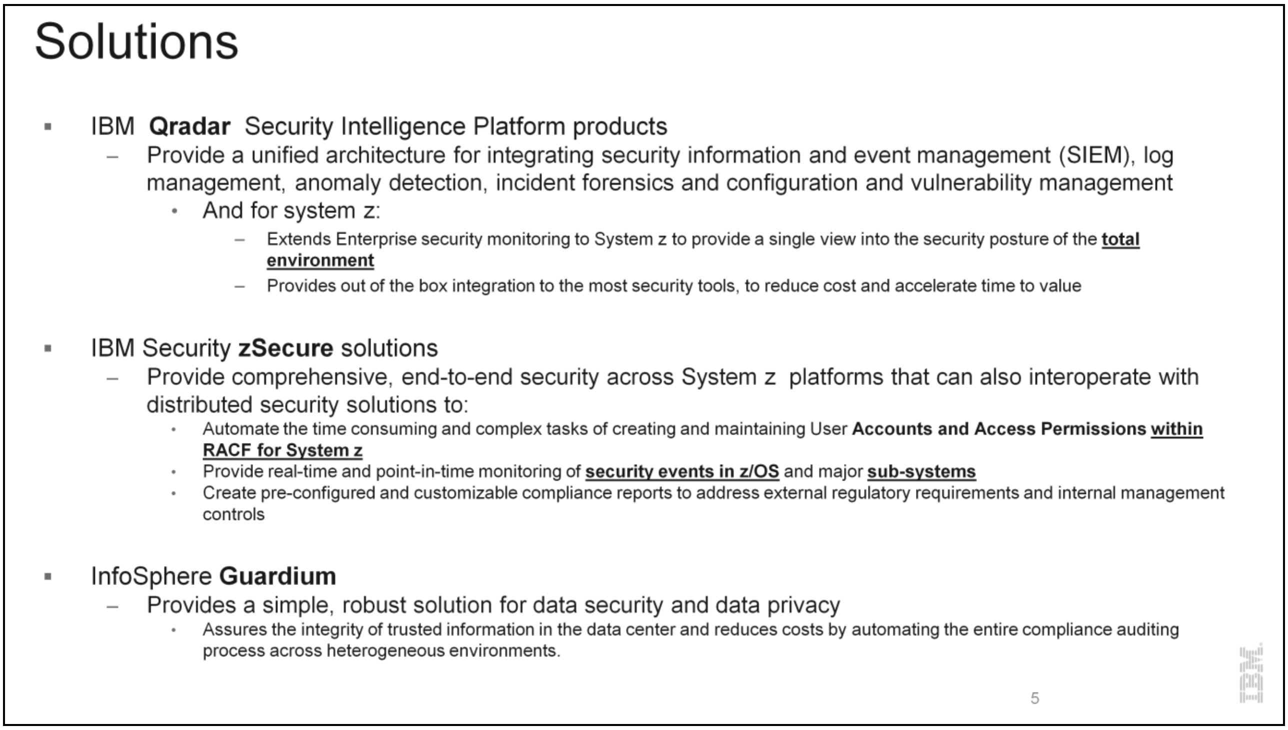
- Challenge
  - Greater need for due diligence, compliance, and security
    - The “notorious nine” threats: (2013 Cloud Computing Alliance)
      - Data breaches
      - Data loss
      - Account Hijacking
      - Insecure APIs
      - Denial of Services
      - Malicious insiders
      - Abuse of cloud services
      - Insufficient due diligence
      - Shared technology issue



The need

- Two levels of protection
  - Defined policies, due diligence
  - Automated monitoring and auditing
  - Software
    - E.g., IBM Qradar, zSecure, Guardium …
  - Appliances, e.g., Datapower
  - User identity mechanisms
    - Userid, certificates, tokens…
  - Protection of resource access
    - RACF, ACF2, Top Secret…

Datapower is both a security appliance & can provide a firewall mechanism to get into Systems of Record



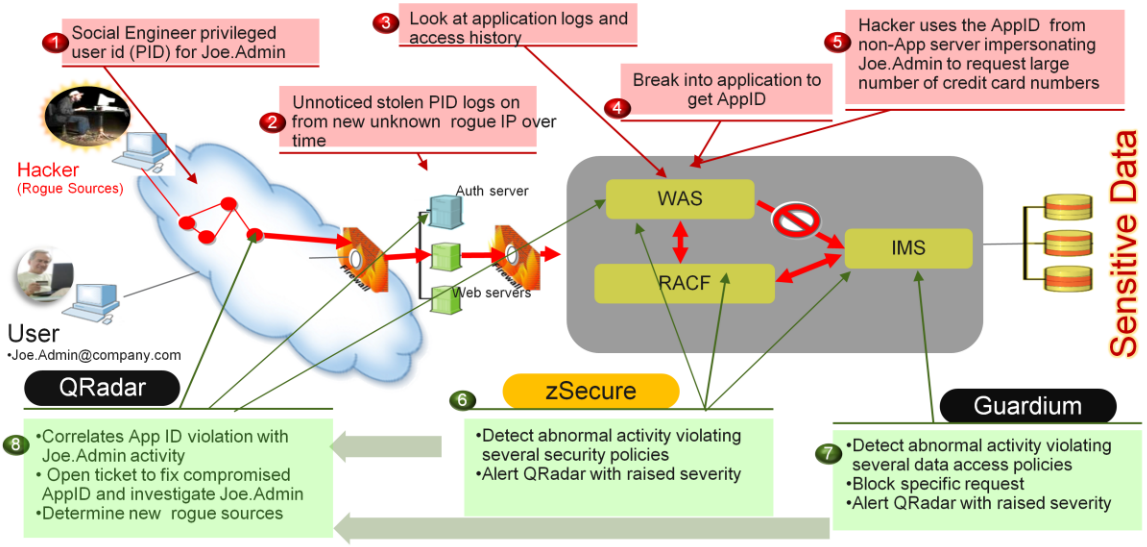
Solutions

- IBM **Qradar** Security Intelligence Platform products
  - Provide a unified architecture for integrating security information and event management (SIEM), log management, anomaly detection, incident forensics and configuration and vulnerability management
    - And for system z:
      - Extends Enterprise security monitoring to System z to provide a single view into the security posture of the **total environment**
      - Provides out of the box integration to the most security tools, to reduce cost and accelerate time to value

- IBM Security **zSecure** solutions
  - Provide comprehensive, end-to-end security across System z platforms that can also interoperate with distributed security solutions to:
    - Automate the time consuming and complex tasks of creating and maintaining User **Accounts and Access Permissions within RACF for System z**
    - Provide real-time and point-in-time monitoring of **security events in z/OS** and major **sub-systems**
    - Create pre-configured and customizable compliance reports to address external regulatory requirements and internal management controls

- InfoSphere **Guardium**
  - Provides a simple, robust solution for data security and data privacy
    - Assures the integrity of trusted information in the data center and reduces costs by automating the entire compliance auditing process across heterogeneous environments.

First Level



1 Social Engineer privileged user id (PID) for Joe.Admin

2 Unnoticed stolen PID logs on from new unknown rogue IP over time

3 Look at application logs and access history

4 Break into application to get AppID

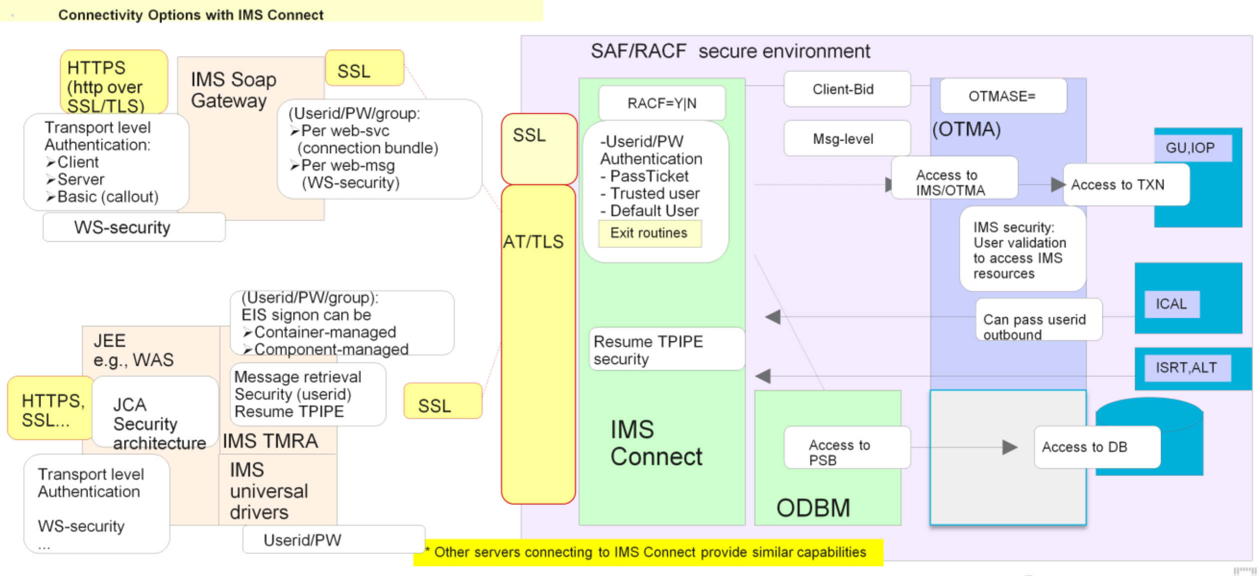
5 Hacker uses the AppID from non-App server impersonating Joe.Admin to request large number of credit card numbers

6
- Detect abnormal activity violating several security policies
- Alert QRadar with raised severity

7
- Detect abnormal activity violating several data access policies
- Block specific request
- Alert QRadar with raised severity

8
- Correlates App ID violation with Joe.Admin activity
- Open ticket to fix compromised AppID and investigate Joe.Admin
- Determine new rogue sources

Second level: Security Points for an Enterprise Server



Connectivity Options with IMS Connect

SAF/RACF secure environment

HTTPS (http over SSL/TLS)

IMS Soap Gateway

SSL

Transport level Authentication:
- Client
- Server
- Basic (callout)

(Userid/PW/group:
- Per web-svc (connection bundle)
- Per web-msg (WS-security)

WS-security

SSL

AT/TLS

RACF=Y|N

-Userid/PW Authentication
- PassTicket
- Trusted user
- Default User

Exit routines

Client-Bid

Msg-level

OTMASE=

(OTMA)

Access to IMS/OTMA

Access to TXN

GU,IOP

IMS security: User validation to access IMS resources

ICAL

Can pass userid outbound

ISRT,ALT

(Userid/PW/group): EIS signon can be
- Container-managed
- Component-managed

JEE e.g., WAS

HTTPS, SSL...

JCA Security architecture

Message retrieval Security (userid) Resume TPIPE

SSL

IMS TMRA

Resume TPIPE security

IMS Connect

Access to PSB

Access to DB

ODBM

Transport level Authentication

WS-security

...

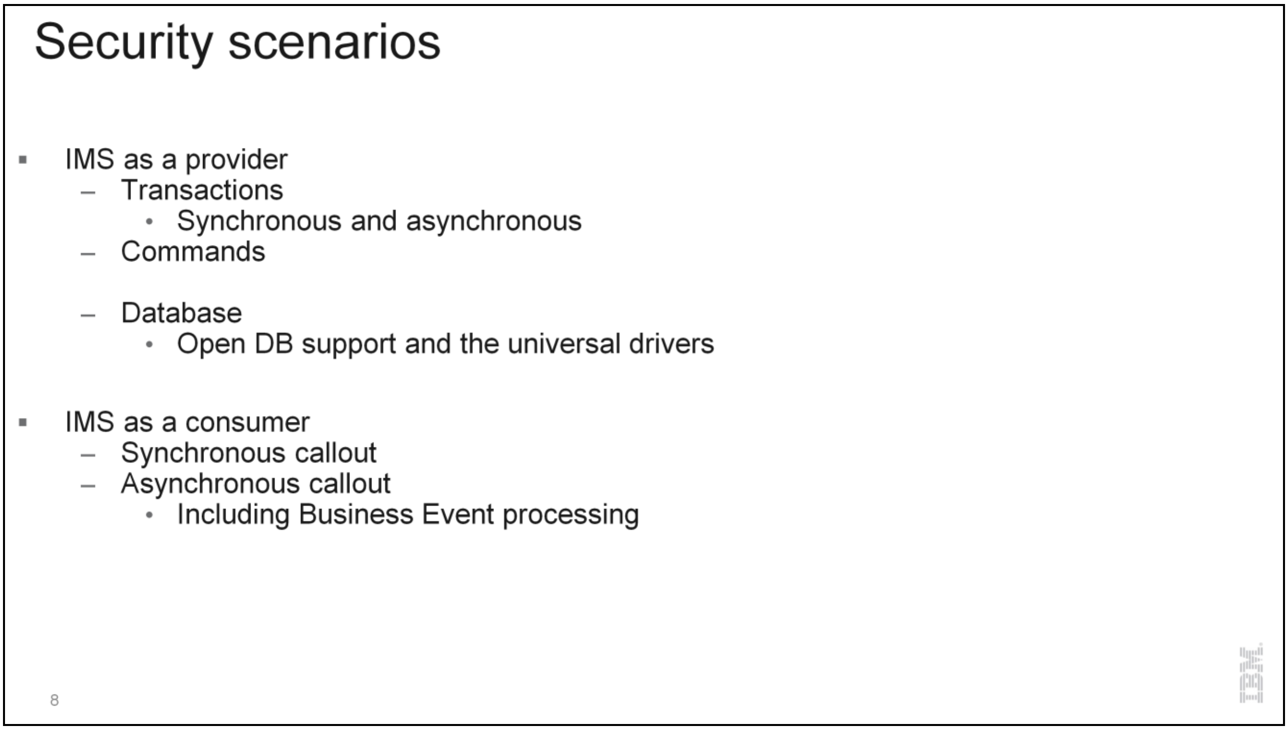
IMS universal drivers

Userid/PW

* Other servers connecting to IMS Connect provide similar capabilities

White boxes show the access points for different kinds of security.

That’s what we will talk about: Security points for an Enterprise Server.



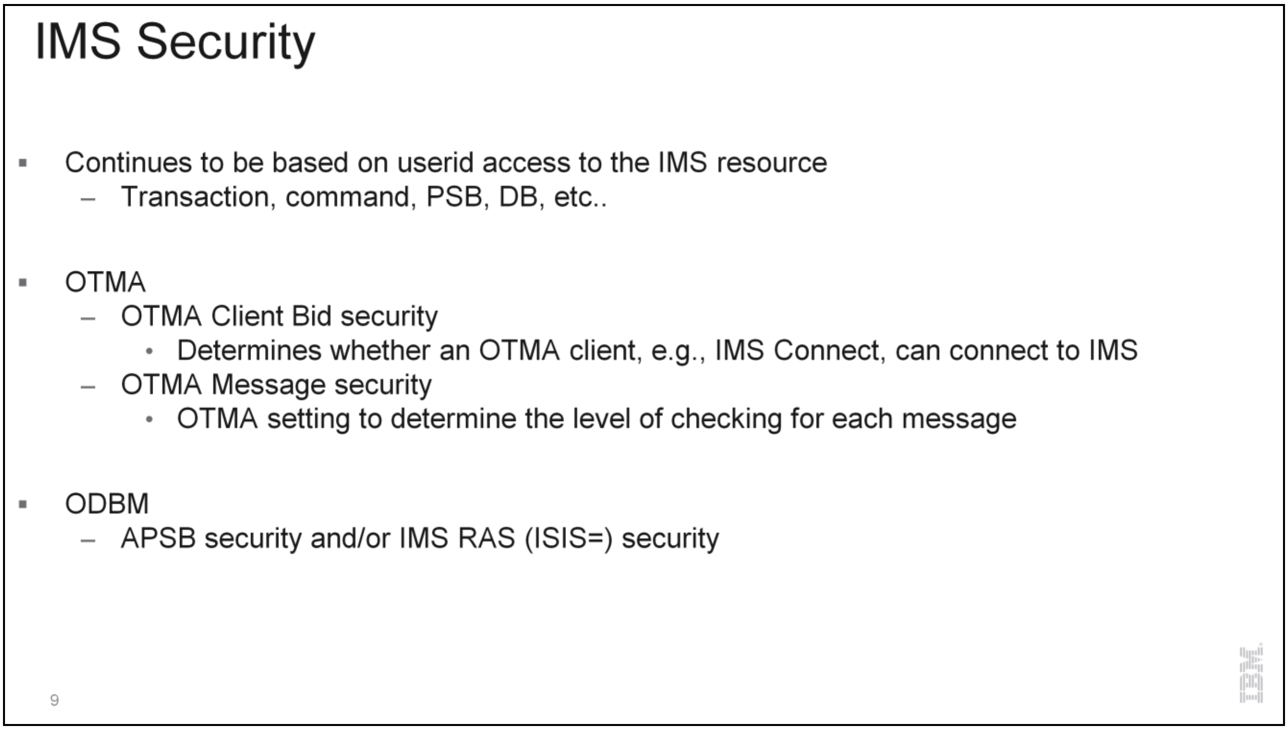
Security scenarios

- IMS as a provider
  - Transactions
    - Synchronous and asynchronous
  - Commands
  - Database
    - Open DB support and the universal drivers
- IMS as a consumer
  - Synchronous callout
  - Asynchronous callout
    - Including Business Event processing

What are the Security scenarios:

Inbound: protecting transactions, commands & Data

Outbound: messages will need to carry some form of an authentication mechanism



IMS Security

- Continues to be based on userid access to the IMS resource
  - Transaction, command, PSB, DB, etc..

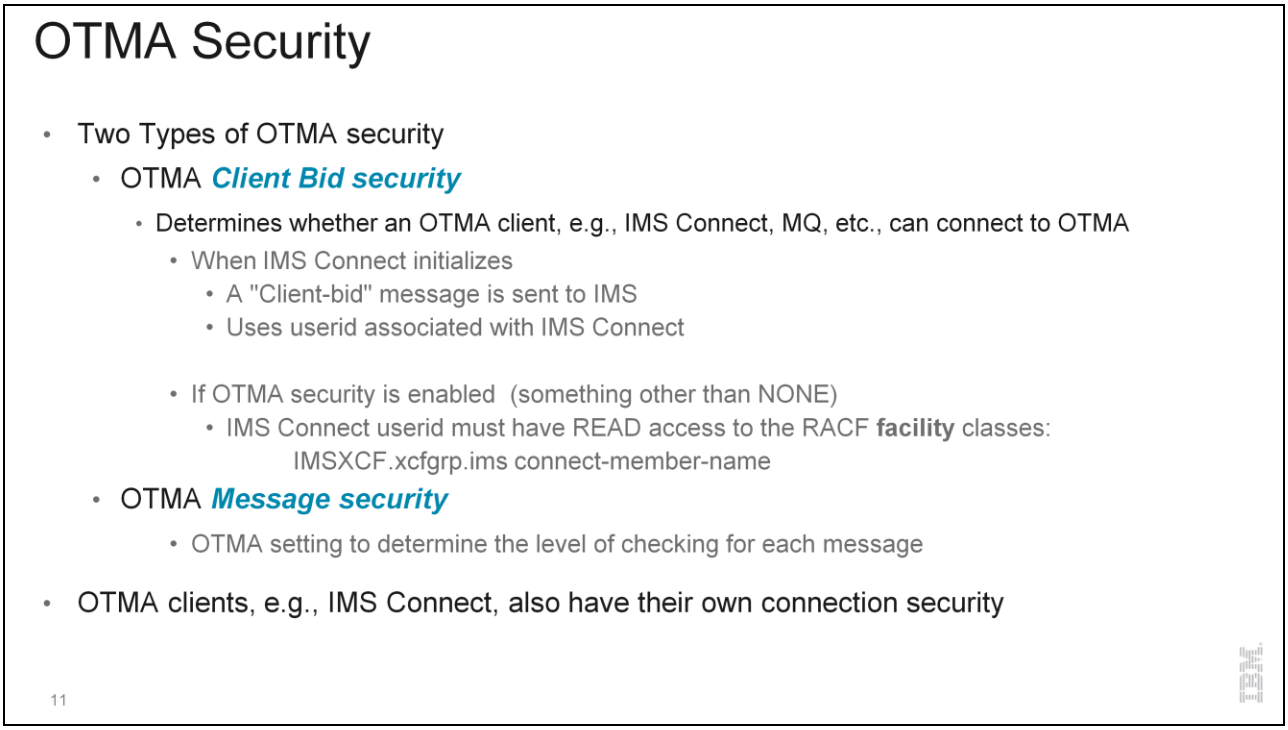
- OTMA
  - OTMA Client Bid security
    - Determines whether an OTMA client, e.g., IMS Connect, can connect to IMS
  - OTMA Message security
    - OTMA setting to determine the level of checking for each message

- ODBM
  - APSB security and/or IMS RAS (ISIS=) security

Userid

- Control blocks that represent the secured user
  - ***ACEE*** – Accessor Environment Element
    - z/OS control block which represents a user’s identity
    - Created by the SAF security manager, e.g., RACF
    - IMS security validation uses ACEEs
  - ***User Token***
    - z/OS control block that can be passed and used by IMS to build an ACEE
      - an 80-byte value which can be used to represent a user
        - » Contains a user ID, default group ID, and some credential information
  - ***RACO*** – RACF Environment Object
    - A “flattened” ACEE which can be transported from one address space to another address space and reconstituted into an ACEE
    - Can be used across MVS images

► Created by components, e.g., IMS Connect, where authentication takes place



OTMA Security

- Two Types of OTMA security
  - OTMA ***Client Bid security***
    - Determines whether an OTMA client, e.g., IMS Connect, MQ, etc., can connect to OTMA
      - When IMS Connect initializes
        - A "Client-bid" message is sent to IMS
        - Uses userid associated with IMS Connect

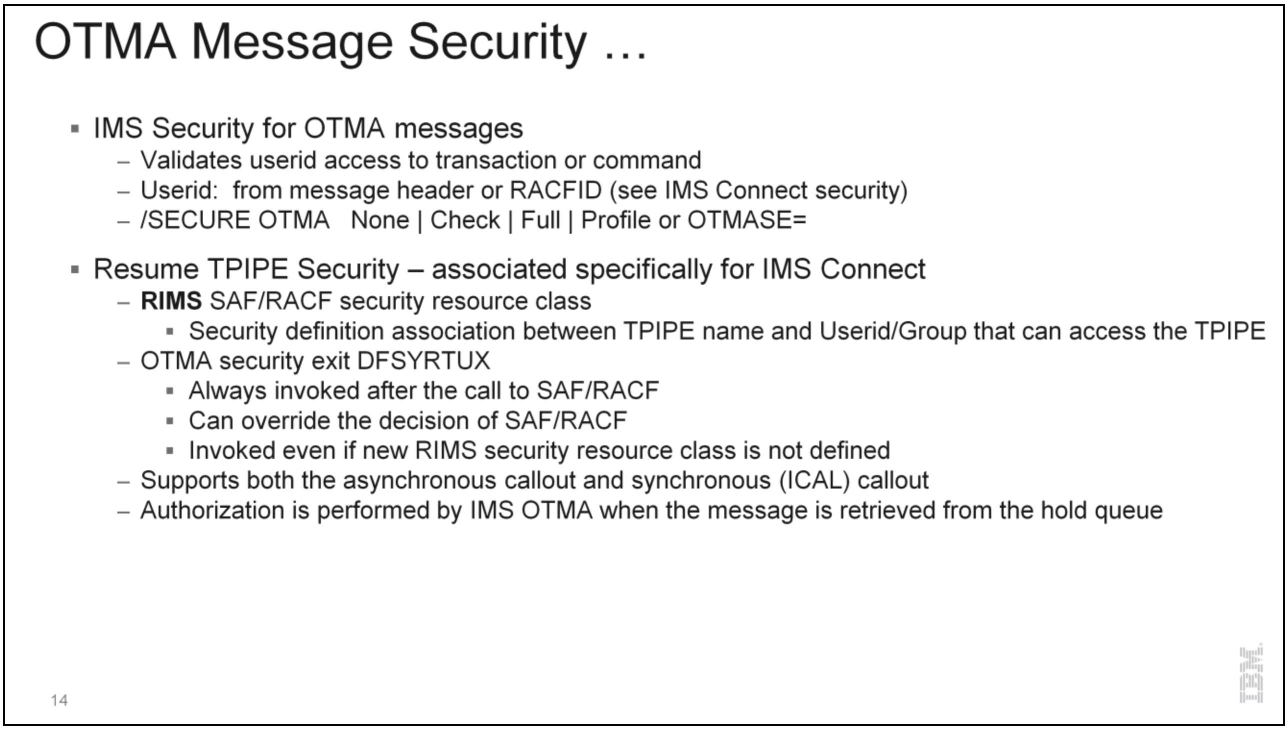
      - If OTMA security is enabled (something other than NONE)
        - IMS Connect userid must have READ access to the RACF **facility** classes:
          IMSXCF.xcfgrp.ims connect-member-name
  - OTMA ***Message security***
      - OTMA setting to determine the level of checking for each message
- OTMA clients, e.g., IMS Connect, also have their own connection security

OTMA Message Security

- OTMASE = *option (in DFSPBxx member of IMS.PROCLIB)*
  - Where *option* can be set to NONE, CHECK, FULL, PROFILE, JOIN (IMS V14) determines the level of checking to validate that a userid can access a transaction
    - **NONE** - If OTMA security is set to NONE there is no Message security
      - No checking of the RACF TIMS Class for user authority to execute tran
        - DFSCTRN0 is invoked if it exists and can enforce security
      - No checking of the RACF CIMS Class for user authority to issue cmd
        - DFSCCMD0 is invoked if it exists and can enforce security
    - **CHECK, FULL, or PROFILE** - OTMA builds a Userid Hash Table for OTMA Clients (TMEMBERs) and a table to hold RACF ACEEs for verified users
      - The table is dynamically increased if it fills up
      - Userids are checked against the transaction/command resource
        - DFSCTRN0 is invoked if it exists and can enforce security
        - DFSCCMD0 is invoked if it exists and can enforce security

OTMA Message Security

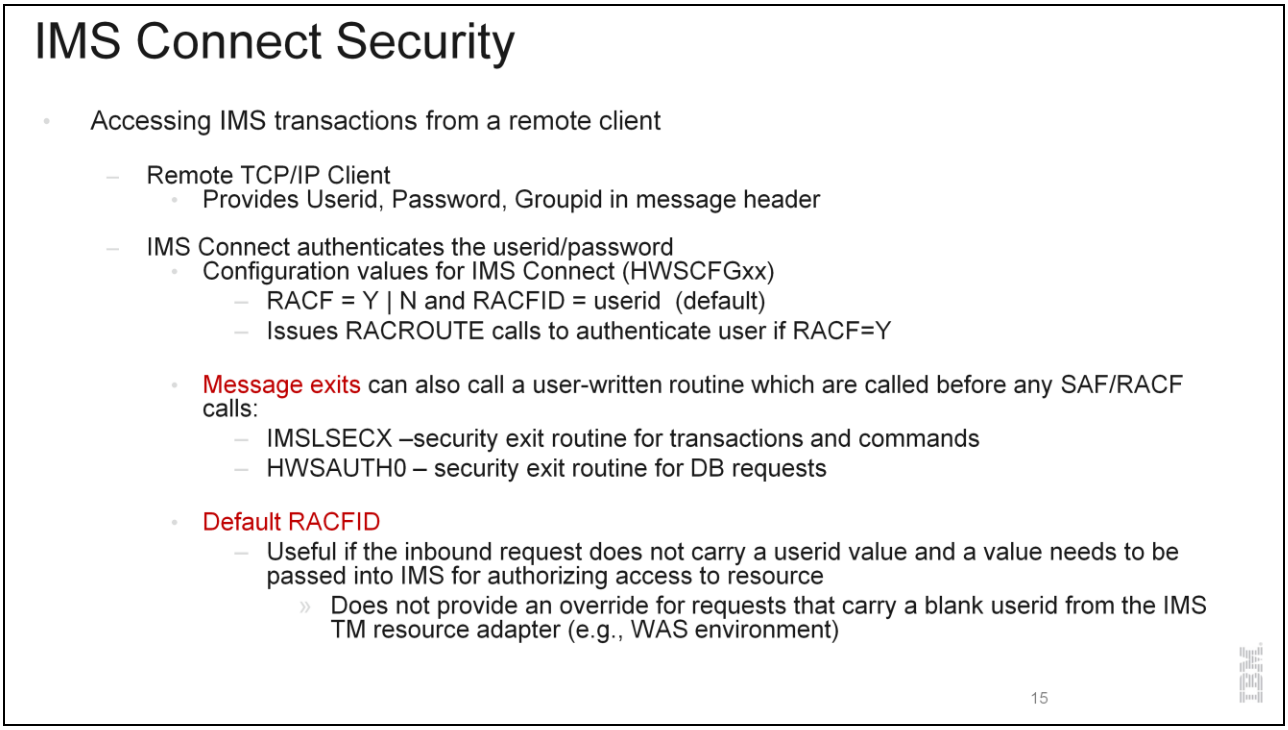
- OTMASE = *option (in DFSPBxx member of IMS.PROCLIB)*
    - **Join (IMS V14)** – Provides Connection security
        - Allows/Rejects an OTMA member client-bid request
            - RACF Resource class – class **FACILITY** Resource: IMSXCF.xcfgrp.client member-name
        - Acts like OTMASE=NONE for messages security



OTMA Message Security …

- IMS Security for OTMA messages
  - Validates userid access to transaction or command
  - Userid: from message header or RACFID (see IMS Connect security)
  - /SECURE OTMA None | Check | Full | Profile or OTMASE=
- Resume TPIPE Security – associated specifically for IMS Connect
  - **RIMS** SAF/RACF security resource class
    - Security definition association between TPIPE name and Userid/Group that can access the TPIPE
  - OTMA security exit DFSYRTUX
    - Always invoked after the call to SAF/RACF
    - Can override the decision of SAF/RACF
    - Invoked even if new RIMS security resource class is not defined
  - Supports both the asynchronous callout and synchronous (ICAL) callout
  - Authorization is performed by IMS OTMA when the message is retrieved from the hold queue

The OTMA Resume TPIPE Security exit routine (DFSYRTUX) is one of two possible methods that you can use to secure messages queued on the OTMA asynchronous hold queue. The other possible method of securing messages on the asynchronous hold queue is to use an external security product, such as RACF. The DFSYRTUX exit routine and an external security product can each by used by itself or in combination with each other. The DFSYRTUX exit routine runs in the IMS control region.



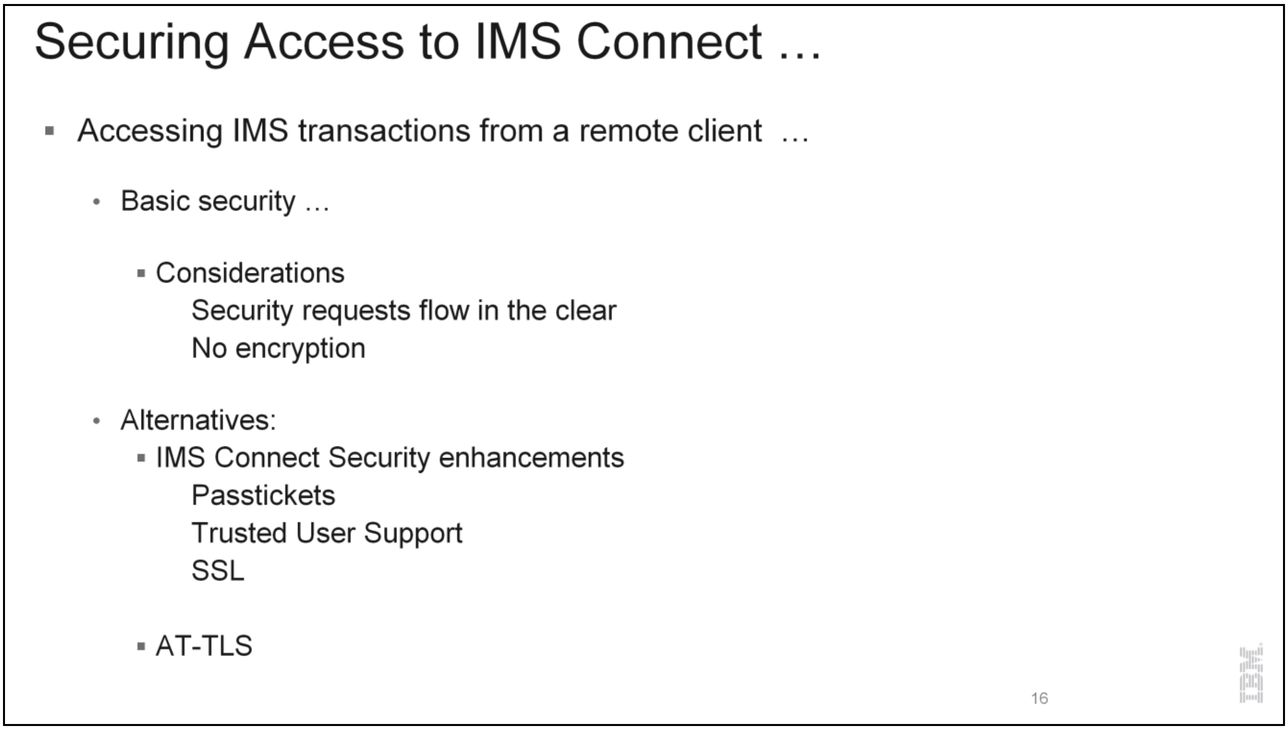
IMS Connect Security

- Accessing IMS transactions from a remote client
  - Remote TCP/IP Client
    - Provides Userid, Password, Groupid in message header
  - IMS Connect authenticates the userid/password
    - Configuration values for IMS Connect (HWSCFGxx)
      - RACF = Y | N and RACFID = userid (default)
      - Issues RACROUTE calls to authenticate user if RACF=Y
    - Message exits can also call a user-written routine which are called before any SAF/RACF calls:
      - IMSLSECX –security exit routine for transactions and commands
      - HWSAUTH0 – security exit routine for DB requests
    - Default RACFID
      - Useful if the inbound request does not carry a userid value and a value needs to be passed into IMS for authorizing access to resource
        - » Does not provide an override for requests that carry a blank userid from the IMS TM resource adapter (e.g., WAS environment)

You can set a default RACF user ID for IMS Connect to use when the input message either does not contain a userid in the header or the field is blank. When the default RACF userid is used, IMS Connect passes it in the OMSECUID field of the input message to OTMA. When OTMA security checking is enabled, OTMA uses the RACF userid for authorizing commands, transactions, and RESUME TPIPE calls with RACF. When both a default RACF userid is defined and the incoming message header userid field is not blank, IMS Connect uses the userid value in the message header.

A lot of people also use IMS Connect Extensions.

Connect extensions also implements these security exit routines.



Securing Access to IMS Connect …

- Accessing IMS transactions from a remote client …
  - Basic security …
    - Considerations
      - Security requests flow in the clear
      - No encryption
  - Alternatives:
    - IMS Connect Security enhancements
      - Passtickets
      - Trusted User Support
      - SSL
    - AT-TLS

Passtickets – alternative to a password.

Trusted User Support – your own thing with exit routines

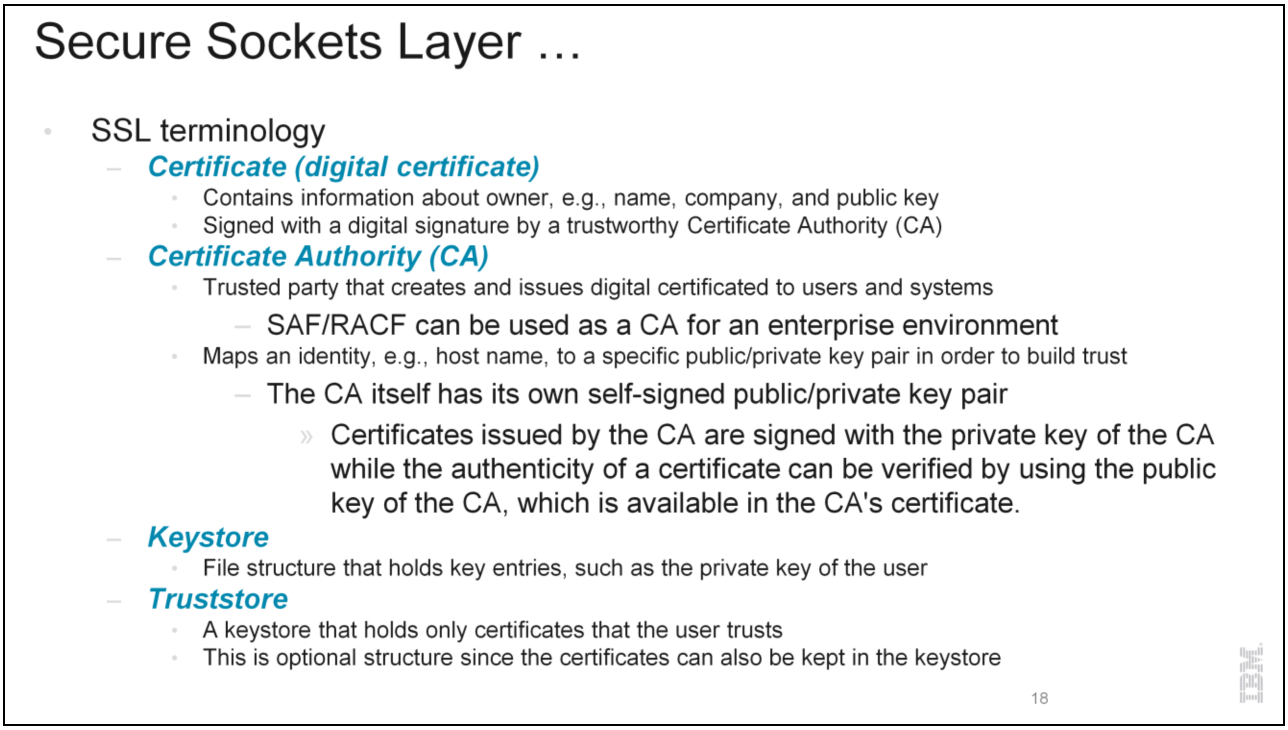
Not too many people use Trusted User Support.

SSL – generic name for TLS

SSL provides for encryption & authentication

Secure Sockets Layer

- SSL - TCP/IP encryption and authentication protocol
  - Secure transfer of sensitive information
    - Provides a private channel between client and server that ensures:
      - Privacy of data
      - Authentication of partners
      - Message integrity
  - SSL Standard
    - Handshake protocol for initial authentication/transfer of encryption keys
      - Agreement on how to encrypt/decrypt data and the format to transmit the encrypted data
      - Authentication of each side using assymetric public/private key mechanism with digital certificates
    - SSL Record protocol
      - protocol for transferring data using agreed upon encryption / decryption
      - Symmetric key encryption uses the negotiated session keys



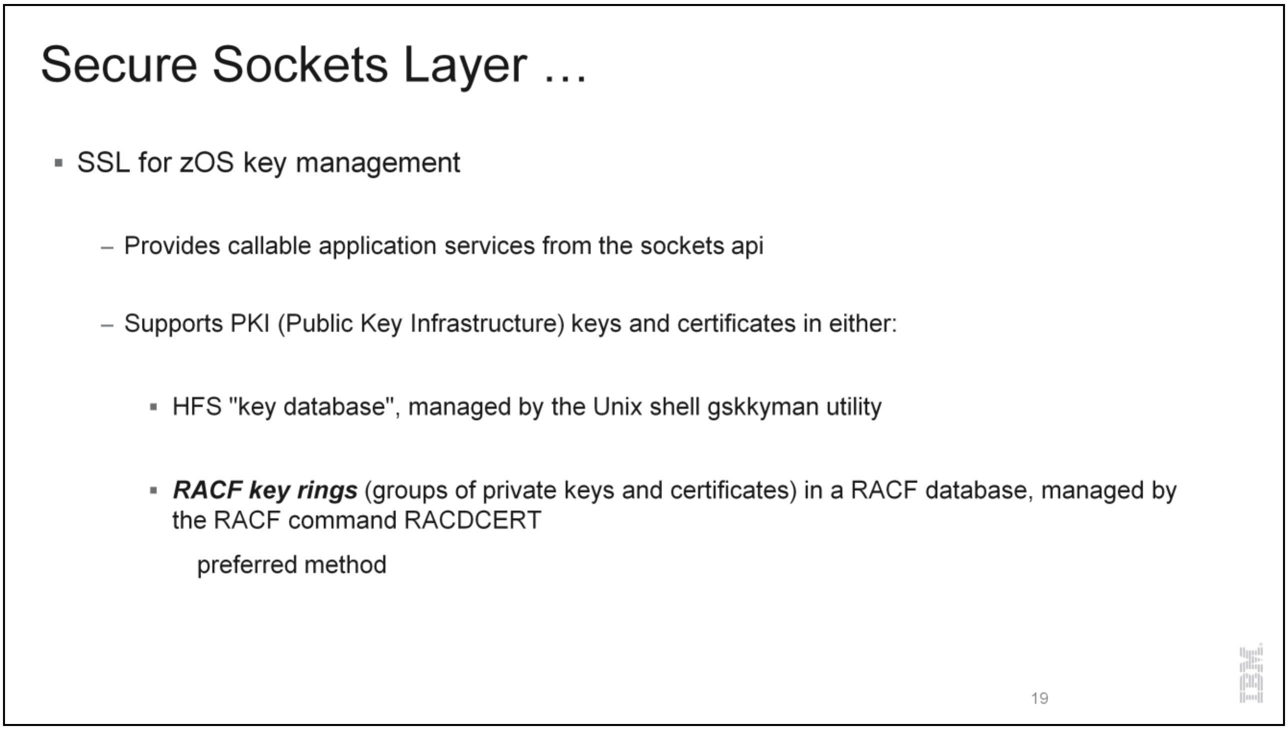
Secure Sockets Layer …

- SSL terminology
  - ***Certificate (digital certificate)***
    - Contains information about owner, e.g., name, company, and public key
    - Signed with a digital signature by a trustworthy Certificate Authority (CA)
  - ***Certificate Authority (CA)***
    - Trusted party that creates and issues digital certificated to users and systems
      - SAF/RACF can be used as a CA for an enterprise environment
    - Maps an identity, e.g., host name, to a specific public/private key pair in order to build trust
      - The CA itself has its own self-signed public/private key pair
        - » Certificates issued by the CA are signed with the private key of the CA while the authenticity of a certificate can be verified by using the public key of the CA, which is available in the CA's certificate.
  - ***Keystore***
    - File structure that holds key entries, such as the private key of the user
  - ***Truststore***
    - A keystore that holds only certificates that the user trusts
    - This is optional structure since the certificates can also be kept in the keystore

A digital certificate is a digital document that validates the identity of the owner of the certificate. A digital certificate contains information about its owner, such as its name, company, and public key. The certificate is signed with a digital signature by a Certificate Authority (CA).

A Certificate Authority (CA) is a trusted party that creates and issues digital certificates to users and systems. The CA establishes the foundation of trust in the certificates. The major task of a trusted CA is to map an identity, such as a host name, to a specific public/private key pair in order to build trust. The CA itself has its own self-signed public/private key pair. As with any public/private key pair the private key is kept secret. Certificates issued by the CA are signed with the private key of the CA, and the authenticity of a certificate can be verified by using the public key of the CA, which is available in the CA's certificate. SAF/RACF in z/OS can be used as the certificate authority to generate and sign certificates for internal systems or applications.

Certificates and private keys are stored in files called keystores. A keystore is a database of key material. Keystore information can be grouped into two categories: key entries and trusted certificate entries. The two entries can be stored in the same keystore or separately in a keystore and truststore for security purposes. Keystores and truststores are used by both the SSL client, e.g. IMS SOAP Gateway, and the SSL server, e.g., IMS Connect.



Secure Sockets Layer …

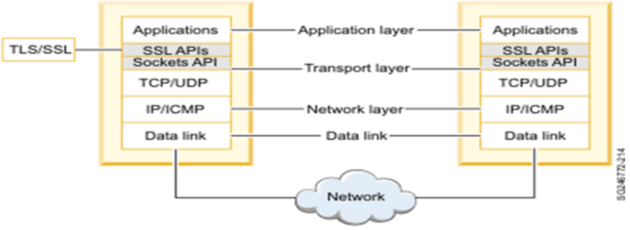
- SSL for zOS key management
  - Provides callable application services from the sockets api
  - Supports PKI (Public Key Infrastructure) keys and certificates in either:
    - HFS "key database", managed by the Unix shell gskkyman utility
    - ***RACF key rings*** (groups of private keys and certificates) in a RACF database, managed by the RACF command RACDCERT

      preferred method

A key ring is a named collection of certificates and Certificate Authorities that is associated with a specific user. A certificate is identified by its label and the key ring to which it is connected.

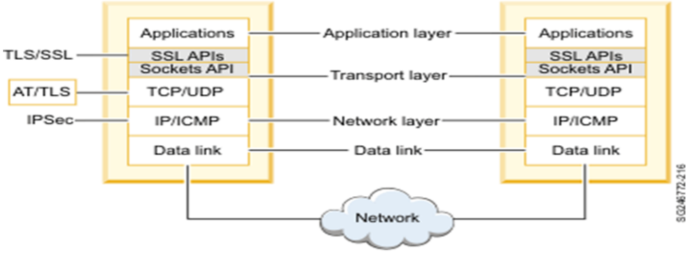
SSL → TLS

- Transport Layer Security – an evolution from SSL
  - SSL is a protocol standard developed by the Netscape Communications Corporation that uses encryption to provide confidentiality and authentication between two TCP/IP applications
  - As SSL gained in popularity, the IETF formally standardized SSL, made a few improvements and changed the name to Transport Layer Security (TLS)
  - TLS is defined in Request for Comments (RFC) 2246
    - Authentication of the server
    - A decision about how the data is to be encrypted
    - Optionally, the authentication of the client



TLS -> Application Transparent TLS (AT-TLS)

- Application Transparent TLS (AT-TLS) is a unique usage of TLS on z/OS
  - Instead of having the application itself (IMS Connect) be aware of TLS
    - Establishing the TLS connection is <u>pushed down the stack into the TCP layer</u>
  - Remote clients cannot distinguish between "normal" TLS (where the z/OS server application does the socket calls necessary for TLS) and AT-TLS (where the TCP layer handles the connection)
    - Application on z/OS can run without even being aware that the underlying connection to the remote client is using TLS



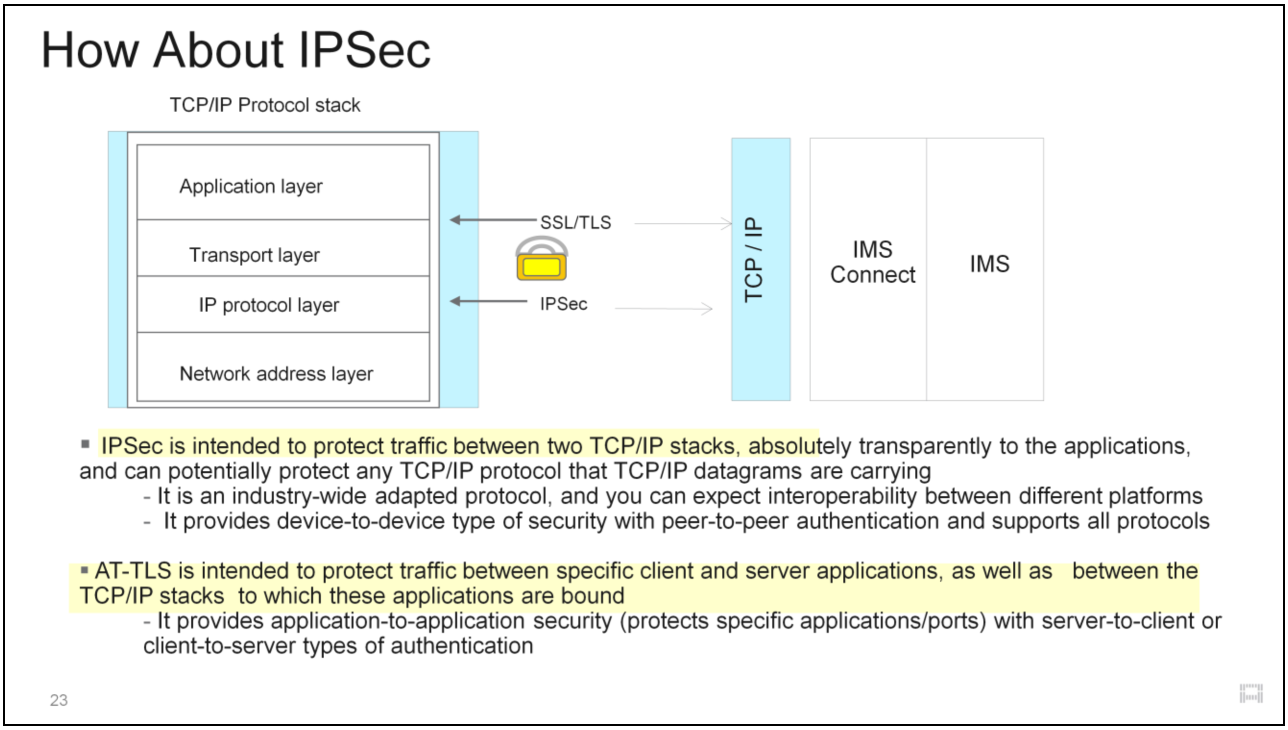
- AT-TLS is activated by specifying the TTLS option in the TCPCONFIG statement block in the TCP/IP profile data

And … Why choose to use AT-TLS?

- Participation in AT-TLS is transparent to IMS Connect
  - IMS Connect can therefore be invoked by a remote client using TLS
  - and
  - Rely on the z/OS TCPIP stack to perform the handshaking protocol to negotiate as well as perform all the require authentications and encryption

- Supports multiple ports
  - SSL support in IMS Connect is limited to a single port for the IMS Connect instance

- No additional configuration specifications in IMS Connect

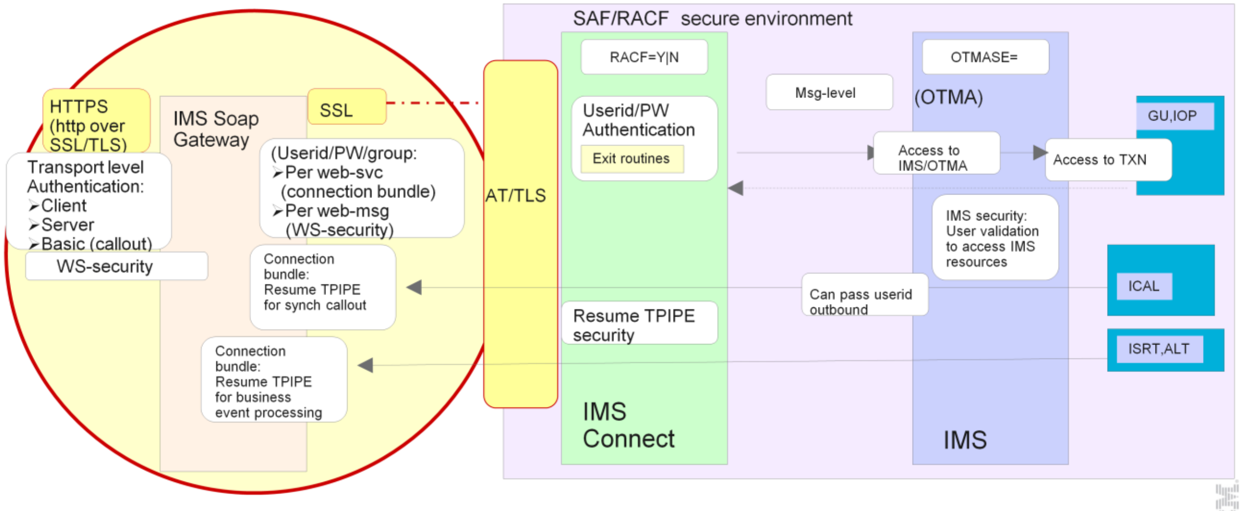


How About IPSec

- IPSec is intended to protect traffic between two TCP/IP stacks, absolutely transparently to the applications, and can potentially protect any TCP/IP protocol that TCP/IP datagrams are carrying
  - It is an industry-wide adapted protocol, and you can expect interoperability between different platforms
  - It provides device-to-device type of security with peer-to-peer authentication and supports all protocols

- AT-TLS is intended to protect traffic between specific client and server applications, as well as between the TCP/IP stacks to which these applications are bound
  - It provides application-to-application security (protects specific applications/ports) with server-to-client or client-to-server types of authentication

IMS Soap Gateway Security



HTTPS (http over SSL/TLS)

Transport level Authentication:
- Client
- Server
- Basic (callout)

WS-security

IMS Soap Gateway

SSL

(Userid/PW/group:
- Per web-svc (connection bundle)
- Per web-msg (WS-security)

Connection bundle: Resume TPIPE for synch callout

Connection bundle: Resume TPIPE for business event processing

AT/TLS

SAF/RACF secure environment

RACF=Y|N

Userid/PW Authentication

Exit routines

Resume TPIPE security

IMS Connect

Msg-level

OTMASE=

(OTMA)

Access to IMS/OTMA

Access to TXN

IMS security: User validation to access IMS resources

Can pass userid outbound

IMS

GU,IOP

ICAL

ISRT,ALT

2 types of security that Soap Gateway provides:

1) Support for network connection security
2) WS Security (web services security) on a message by message basis

IMS Soap Gateway

- Supports HTTPS connections with clients and SSL/AT-TLS with IMS Connect

- Transport-level (connection) security between remote client and IMS SG
  - Server authentication
    - During the SSL handshake, the server sends a certificate to the client to authenticate itself

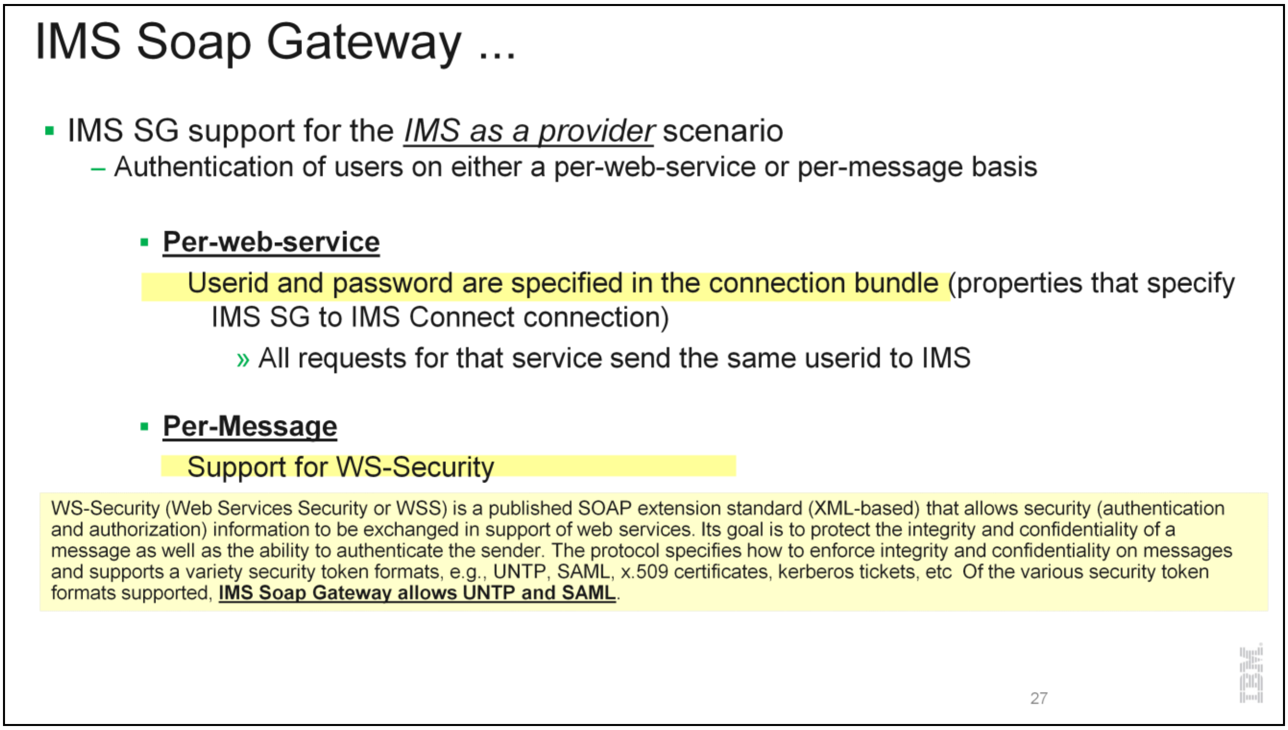
      Client authenticates the identity the certificate represents

        Certificate contains a public key and the cryptographic algorithms used in SSL
  - Client (Mutual) authentication
    - Server authentication (making it mutual) plus the requirement for the client to send its certificate to the server for authentication

  - * Certificates come from a trusted CA (Certification Authority)
    - They are exchanged at the transport level to establish trust before the connection can be established or a web service invoked

IMS Soap Gateway...

- Basic authentication (for IMS as a web service consumer, e.g. IMS callout)
  - When IMS Soap Gateway is the client, the server that hosts the web service may require basic authentication credentials in order to call the remote service
  - Connection bundle optional properties
    - Basic authentication user ID: Specifies the user ID to send to the server that hosts the web service for basic authentication
      - basic authentication password
    - Callout truststore name: Specifies the fully qualified path name of the truststore on SOAP Gateway that stores the certificates of trusted external web service servers(required for client or server authentication)
      - Trustore password
    - Callout keystore name: Specifies the fully qualified path name of the keystore on SOAP Gateway that stores the trusted client certificates for a callout application to authenticate with a target web service (Required for client authentication)
      - Keystore password
  - Security certificates can be sent at the transport level for server authentication or client authentication



IMS Soap Gateway ...

- IMS SG support for the *IMS as a provider* scenario
  - – Authentication of users on either a per-web-service or per-message basis
    - **Per-web-service**
      Userid and password are specified in the connection bundle (properties that specify IMS SG to IMS Connect connection)
      - » All requests for that service send the same userid to IMS
    - **Per-Message**
      Support for WS-Security

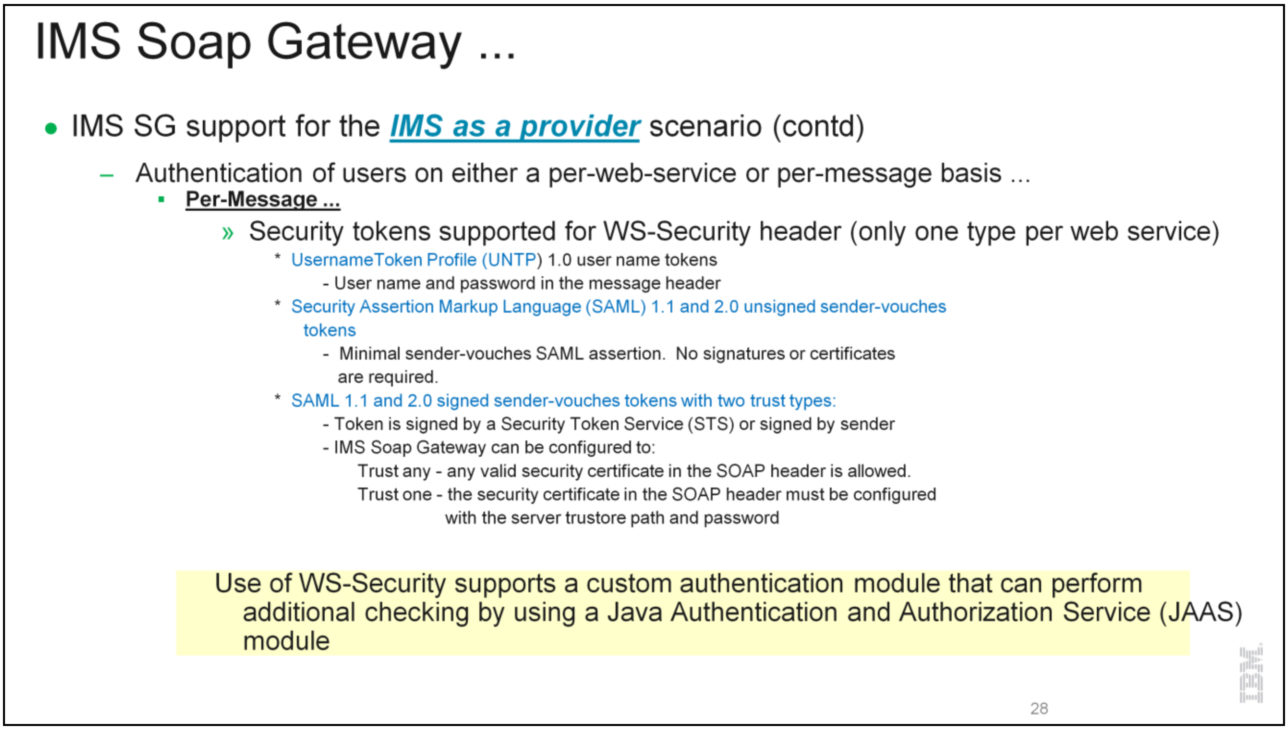
WS-Security (Web Services Security or WSS) is a published SOAP extension standard (XML-based) that allows security (authentication and authorization) information to be exchanged in support of web services. Its goal is to protect the integrity and confidentiality of a message as well as the ability to authenticate the sender. The protocol specifies how to enforce integrity and confidentiality on messages and supports a variety security token formats, e.g., UNTP, SAML, x.509 certificates, kerberos tickets, etc Of the various security token formats supported, **IMS Soap Gateway allows UNTP and SAML**.

WS-Security (Web Services Security or WSS) is a published SOAP extension standard (XML-based) that allows security (authentication and authorization) information to be exchanged in support of web services. Its goal is to protect the integrity and confidentiality of a message as well as the ability to authenticate the sender. The protocol specifies how to enforce integrity and confidentiality on messages and supports a variety security token formats, e.g., UNTP, SAML, x.509 certificates, kerberos tickets, etc Of the various security token formats supported, IMS Soap Gateway allows UNTP and SAML.

•A WS-Security *Username Token* (UNTP) enables an end-user identity to be passed over multiple hops before reaching the destination Web Service. The user identity (username and password) are inserted into the message header. When the token is received, the EIS server can ensure that: the timestamp on the token is still valid as well as authentication of user identity.

•The SAML standard provides the means by which authentication and authorization assertions can be exchanged across web service transactions, and how a security identity can be obtained and transferred from one business entity to another.

The sender-vouches confirmation method is used when a server (SOAP Gateway) needs to further propagate the client identity and attributes on behalf of the client (to IMS Connect and OTMA). An attesting entity uses the sender-vouches confirmation method to assert that it is acting on behalf of the subject of the SAML statements attributed with the sender-vouches SubjectConfirmation element. SAML support requires an SSL connection with client authentication to enable sender-vouches security tokens. You must configure client authentication to use the SAML sender-vouches confirmation method. The SOAP response message does not carry any security token information.



IMS Soap Gateway ...

- IMS SG support for the ***IMS as a provider*** scenario (contd)
  - Authentication of users on either a per-web-service or per-message basis ...
    - **Per-Message ...**
      - » Security tokens supported for WS-Security header (only one type per web service)
        - * UsernameToken Profile (UNTP) 1.0 user name tokens
          - User name and password in the message header
        - * Security Assertion Markup Language (SAML) 1.1 and 2.0 unsigned sender-vouches tokens
          - Minimal sender-vouches SAML assertion. No signatures or certificates are required.
        - * SAML 1.1 and 2.0 signed sender-vouches tokens with two trust types:
          - Token is signed by a Security Token Service (STS) or signed by sender
          - IMS Soap Gateway can be configured to:
            Trust any - any valid security certificate in the SOAP header is allowed.
            Trust one - the security certificate in the SOAP header must be configured with the server trustore path and password

Use of WS-Security supports a custom authentication module that can perform additional checking by using a Java Authentication and Authorization Service (JAAS) module

A SAML token can be signed or unsigned:

•When the token is unsigned, the request contains a minimal sender-vouches SAML assertion with no optional elements included. There are no signatures or certificates required. The response does not contain a security header.

•When the token is signed, the request contains a sender-vouches SAML assertion. The assertion element is signed. A reference to the certificate used to verify the signature is provided in the header. The response does not contain a security header.

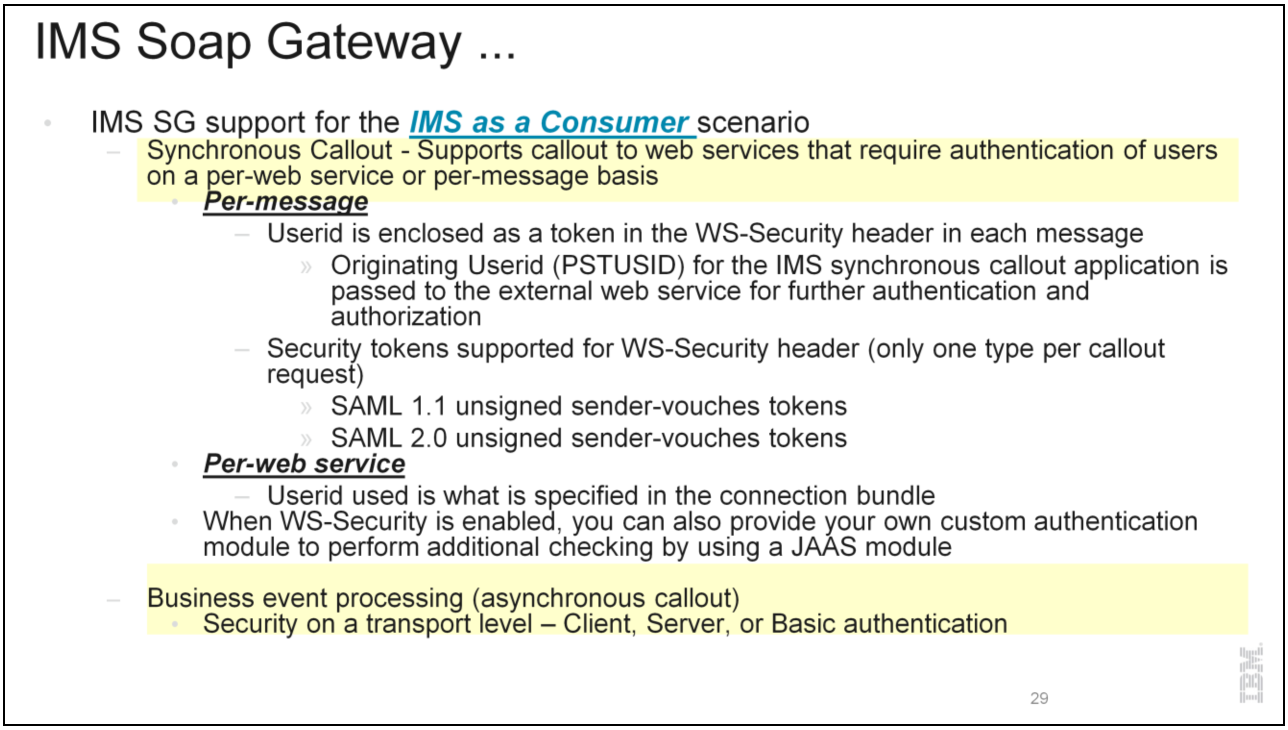
A SAML token can be signed by a Security Token Service (STS) or self-issued. SOAP Gateway can be configured to trust the SAML token and the signature (certificate) embedded, or to use the certificates in a specified truststore to verify the signature before trusting the SAML token.

When a SAML token is signed by the sender, the SOAP Gateway server can be configured to:

•Trust the embedded signature (certificate) within the SOAP header along with the signing SAML token, or

•Trust the certificates in a specified truststore. All certificates in the referenced keystore or truststore are the trusted source for verifying the SAML signature.

After the SAML signature is verified and the token is trusted, SOAP Gateway extracts the SAML ID together with the security attributes from the SOAP header and propagates the SAML ID to IMS Connect for further authorization. SOAP Gateway includes a WS-Security API for creating self-issued SAML tokens. You can also use any RSA-SHA1 signature method.



IMS Soap Gateway ...

- IMS SG support for the ***IMS as a Consumer*** scenario
  - Synchronous Callout - Supports callout to web services that require authentication of users on a per-web service or per-message basis
    - ***Per-message***
      - Userid is enclosed as a token in the WS-Security header in each message
        - Originating Userid (PSTUSID) for the IMS synchronous callout application is passed to the external web service for further authentication and authorization
      - Security tokens supported for WS-Security header (only one type per callout request)
        - SAML 1.1 unsigned sender-vouches tokens
        - SAML 2.0 unsigned sender-vouches tokens
    - ***Per-web service***
      - Userid used is what is specified in the connection bundle
    - When WS-Security is enabled, you can also provide your own custom authentication module to perform additional checking by using a JAAS module
  - Business event processing (asynchronous callout)
    - Security on a transport level – Client, Server, or Basic authentication

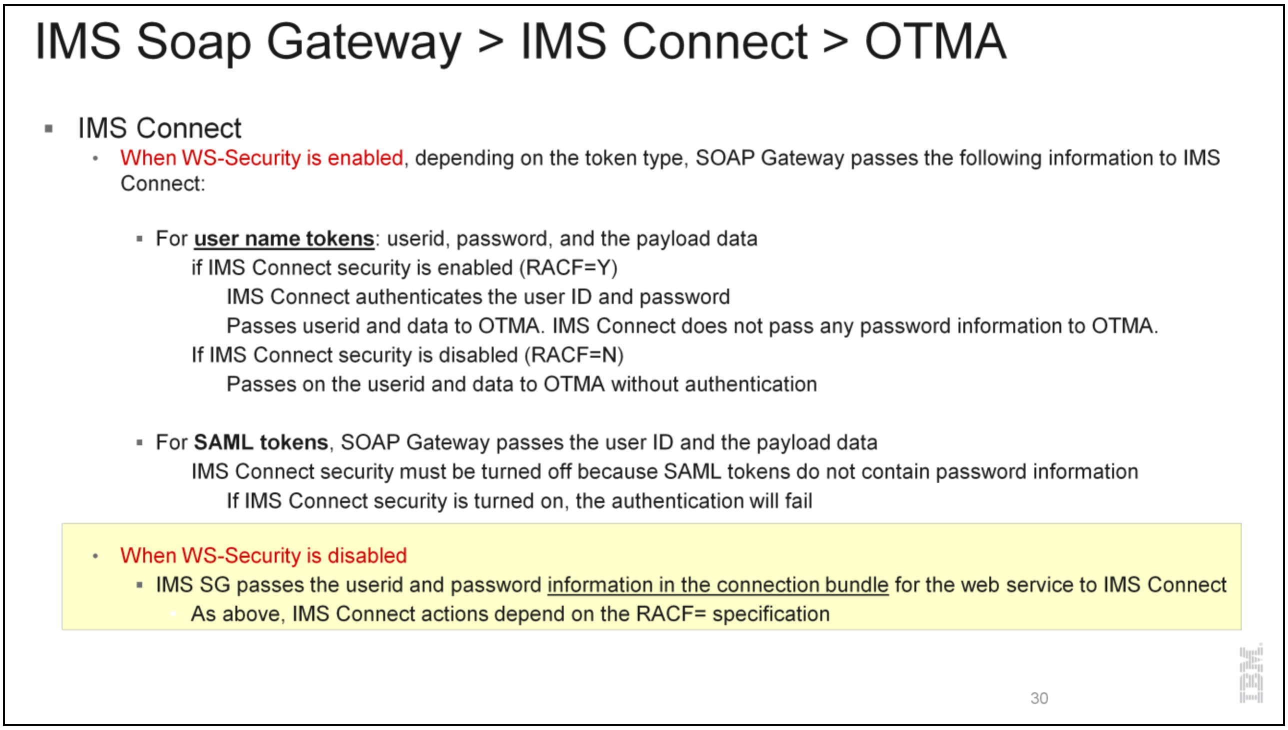
For the synchronous callout scenarios, in addition to transport-level security through basic authentication, server authentication, or mutual authentication, SOAP Gateway now supports message-level security with SAML 1.1 and SAML 2.0 sender-vouches unsigned tokens. SAML is an XML-based standard developed by Security Services Technical Committee (SSTC) of Organization for the Advancement of Structured Information Standards (OASIS). This standard facilitates:

•The exchange of user identity and security attributes information between communicating parties at the SOAP message level.

•The exchange of authentication and authorization assertions across web service transactions.

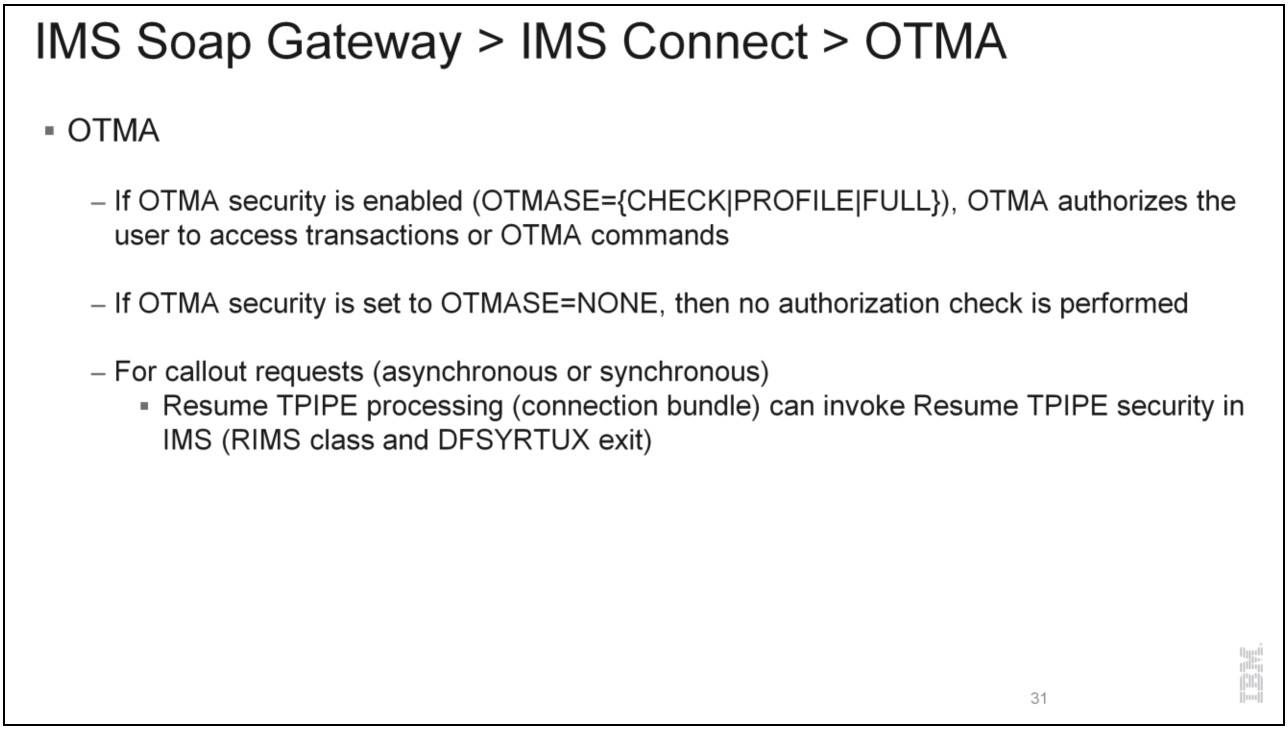
WS-Security SAML confirmation method is supported for synchronous callout applications by extracting the user ID (the user that initiates the synchronous callout application) from the correlation token and passing it to the external web service.

The IMS SOAP Gateway also supports custom authentication modules for accessing the security header for validation before the SOAP request messages are sent out to the external web service server.



IMS Soap Gateway > IMS Connect > OTMA

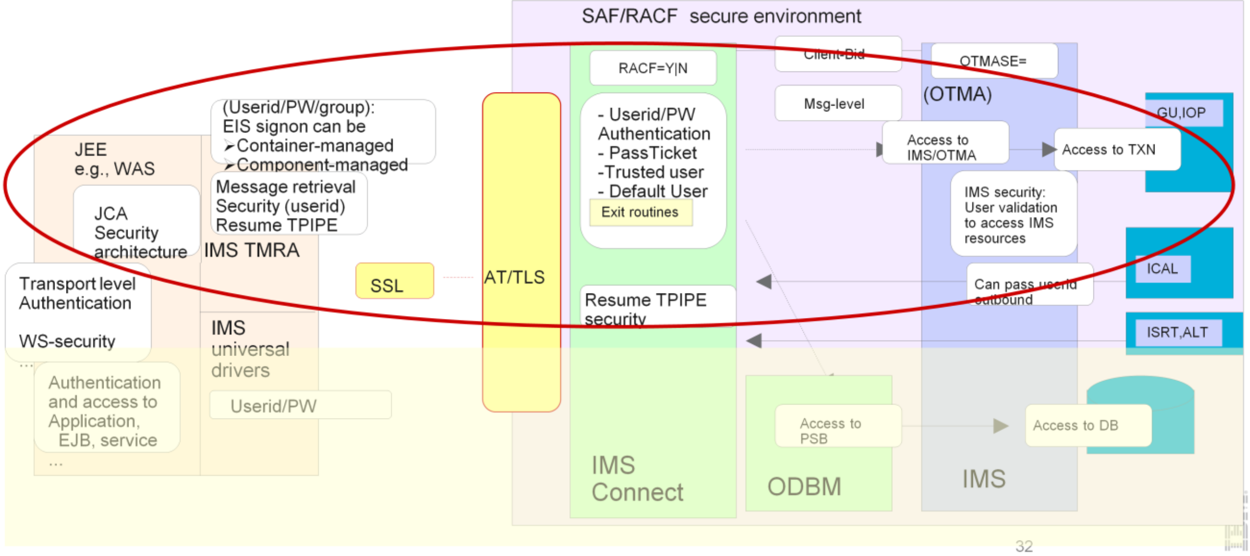
- IMS Connect
  - When WS-Security is enabled, depending on the token type, SOAP Gateway passes the following information to IMS Connect:
    - For **user name tokens**: userid, password, and the payload data
      - if IMS Connect security is enabled (RACF=Y)
        - IMS Connect authenticates the user ID and password
        - Passes userid and data to OTMA. IMS Connect does not pass any password information to OTMA.
      - If IMS Connect security is disabled (RACF=N)
        - Passes on the userid and data to OTMA without authentication
    - For **SAML tokens**, SOAP Gateway passes the user ID and the payload data
      - IMS Connect security must be turned off because SAML tokens do not contain password information
        - If IMS Connect security is turned on, the authentication will fail
  - When WS-Security is disabled
    - IMS SG passes the userid and password information in the connection bundle for the web service to IMS Connect
      - As above, IMS Connect actions depend on the RACF= specification



IMS Soap Gateway > IMS Connect > OTMA

- OTMA
  - If OTMA security is enabled (OTMASE={CHECK|PROFILE|FULL}), OTMA authorizes the user to access transactions or OTMA commands
  - If OTMA security is set to OTMASE=NONE, then no authorization check is performed
  - For callout requests (asynchronous or synchronous)
    - Resume TPIPE processing (connection bundle) can invoke Resume TPIPE security in IMS (RIMS class and DFSYRTUX exit)

IMS TM Resource Adapter (IMS TMRA)



SAF/RACF secure environment

JEE e.g., WAS

JCA Security architecture

(Userid/PW/group): EIS signon can be
- Container-managed
- Component-managed

Message retrieval Security (userid) Resume TPIPE

IMS TMRA

Transport level Authentication

WS-security

…

Authentication and access to Application, EJB, service

…

IMS universal drivers

Userid/PW

SSL

AT/TLS

RACF=Y|N

- Userid/PW Authentication
- PassTicket
-Trusted user
- Default User

Exit routines

Resume TPIPE security

IMS Connect

Client-Bid

Msg-level

Access to IMS/OTMA

OTMASE=

(OTMA)

IMS security: User validation to access IMS resources

Access to TXN

GU,IOP

ICAL

ISRT,ALT

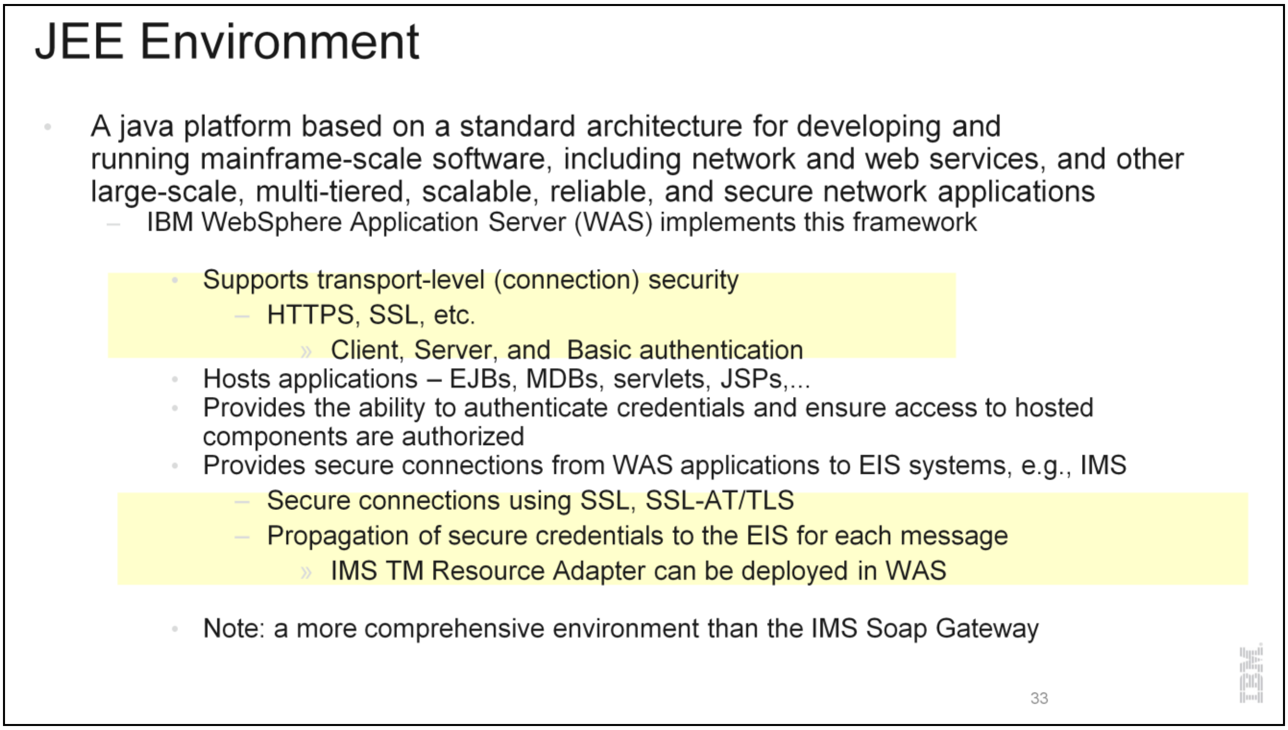
Can pass userid outbound

ODBM

Access to PSB

Access to DB

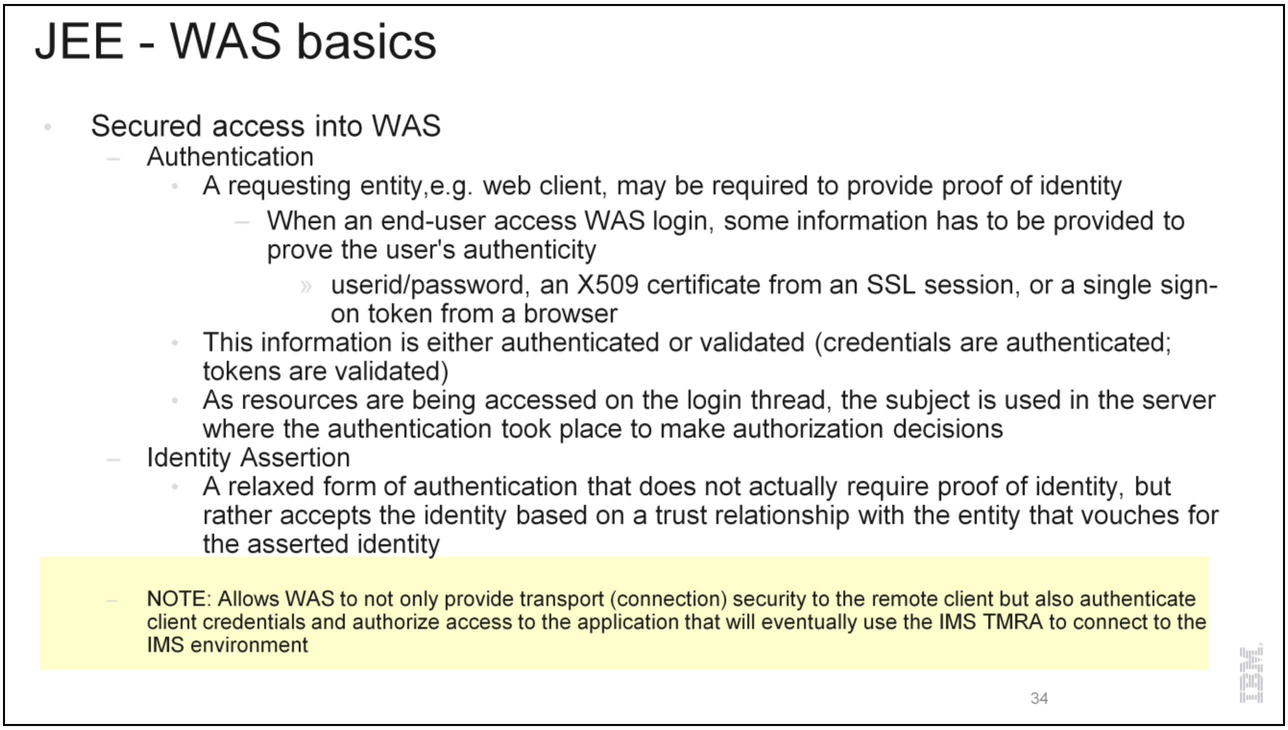
IMS



JEE Environment

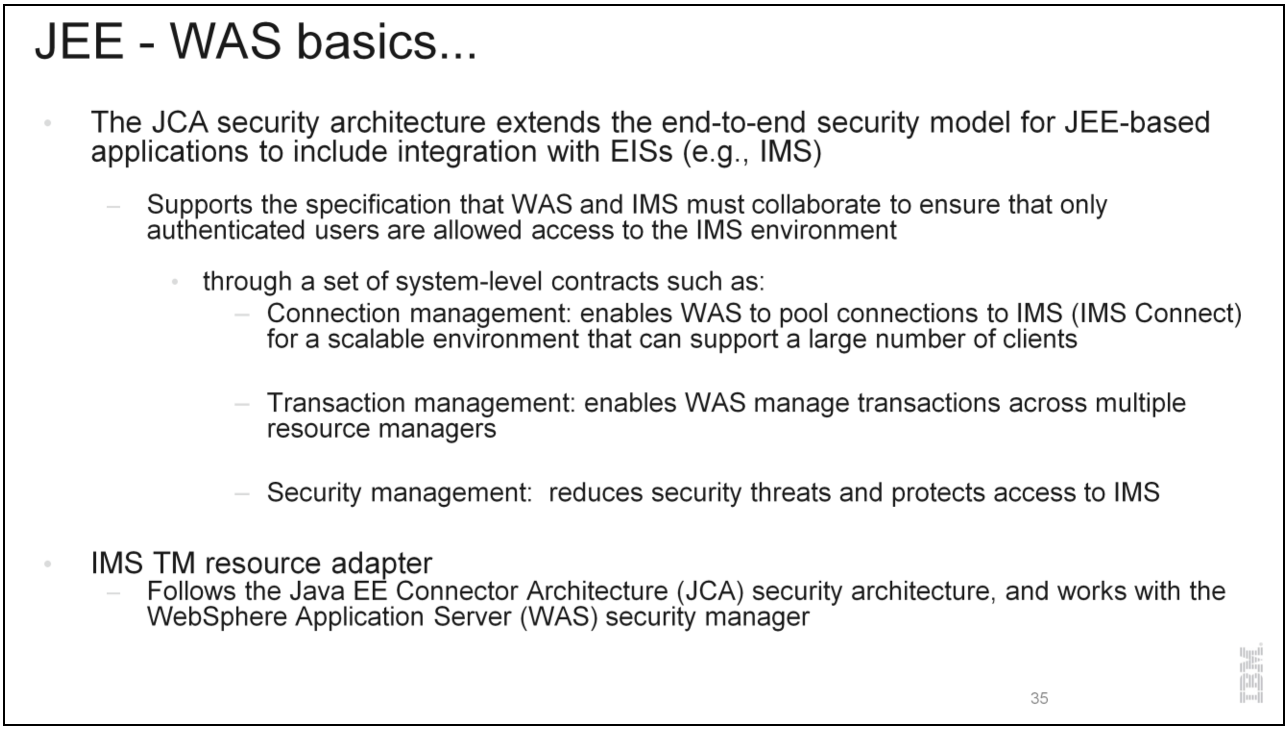
- A java platform based on a standard architecture for developing and running mainframe-scale software, including network and web services, and other large-scale, multi-tiered, scalable, reliable, and secure network applications
  - IBM WebSphere Application Server (WAS) implements this framework
    - Supports transport-level (connection) security
      - HTTPS, SSL, etc.
        - Client, Server, and Basic authentication
    - Hosts applications – EJBs, MDBs, servlets, JSPs,...
    - Provides the ability to authenticate credentials and ensure access to hosted components are authorized
    - Provides secure connections from WAS applications to EIS systems, e.g., IMS
      - Secure connections using SSL, SSL-AT/TLS
      - Propagation of secure credentials to the EIS for each message
        - IMS TM Resource Adapter can be deployed in WAS
    - Note: a more comprehensive environment than the IMS Soap Gateway

WebSphere Application server implements the JEE Security Framework specification and provides a unified, policy-based, and permission-based model for securing Web resources, Web service endpoints, and enterprise JavaBeans according to JEE standards. WAS also addresses enterprise end-to-end security requirements including: Authentication, Resource access control, Data integrity, Confidentiality, Privacy, and Secure interoperability. Authenticating a user involves retrieving information about users and groups to perform security-related functions, including authentication and authorization. WAS can be configured to leverage user registries, a trust association interpretor (TAI), single sign-on (SSO) across multiple WAS servers,...



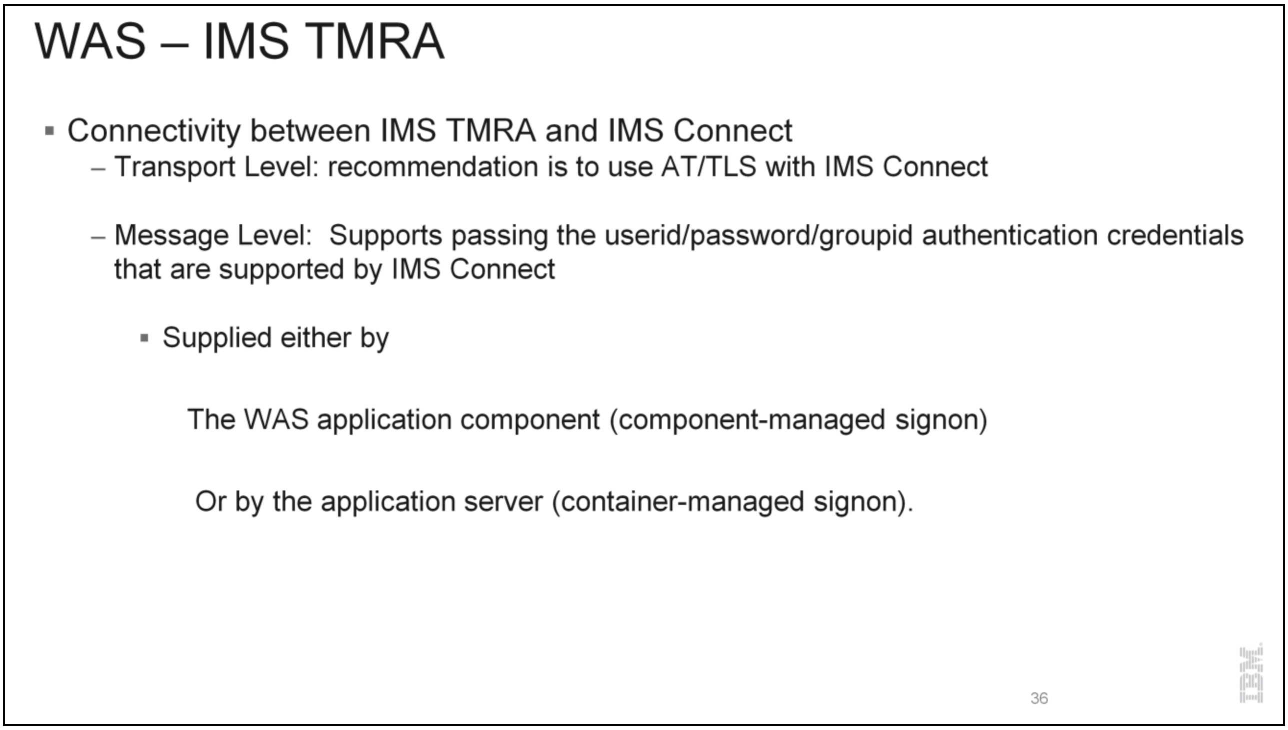
JEE - WAS basics

- Secured access into WAS
  - Authentication
    - A requesting entity,e.g. web client, may be required to provide proof of identity
      - When an end-user access WAS login, some information has to be provided to prove the user's authenticity
        - userid/password, an X509 certificate from an SSL session, or a single sign-on token from a browser
    - This information is either authenticated or validated (credentials are authenticated; tokens are validated)
    - As resources are being accessed on the login thread, the subject is used in the server where the authentication took place to make authorization decisions
  - Identity Assertion
    - A relaxed form of authentication that does not actually require proof of identity, but rather accepts the identity based on a trust relationship with the entity that vouches for the asserted identity
  - NOTE: Allows WAS to not only provide transport (connection) security to the remote client but also authenticate client credentials and authorize access to the application that will eventually use the IMS TMRA to connect to the IMS environment



JEE - WAS basics...

- The JCA security architecture extends the end-to-end security model for JEE-based applications to include integration with EISs (e.g., IMS)
  - Supports the specification that WAS and IMS must collaborate to ensure that only authenticated users are allowed access to the IMS environment
    - through a set of system-level contracts such as:
      - Connection management: enables WAS to pool connections to IMS (IMS Connect) for a scalable environment that can support a large number of clients
      - Transaction management: enables WAS manage transactions across multiple resource managers
      - Security management: reduces security threats and protects access to IMS
- IMS TM resource adapter
  - Follows the Java EE Connector Architecture (JCA) security architecture, and works with the WebSphere Application Server (WAS) security manager

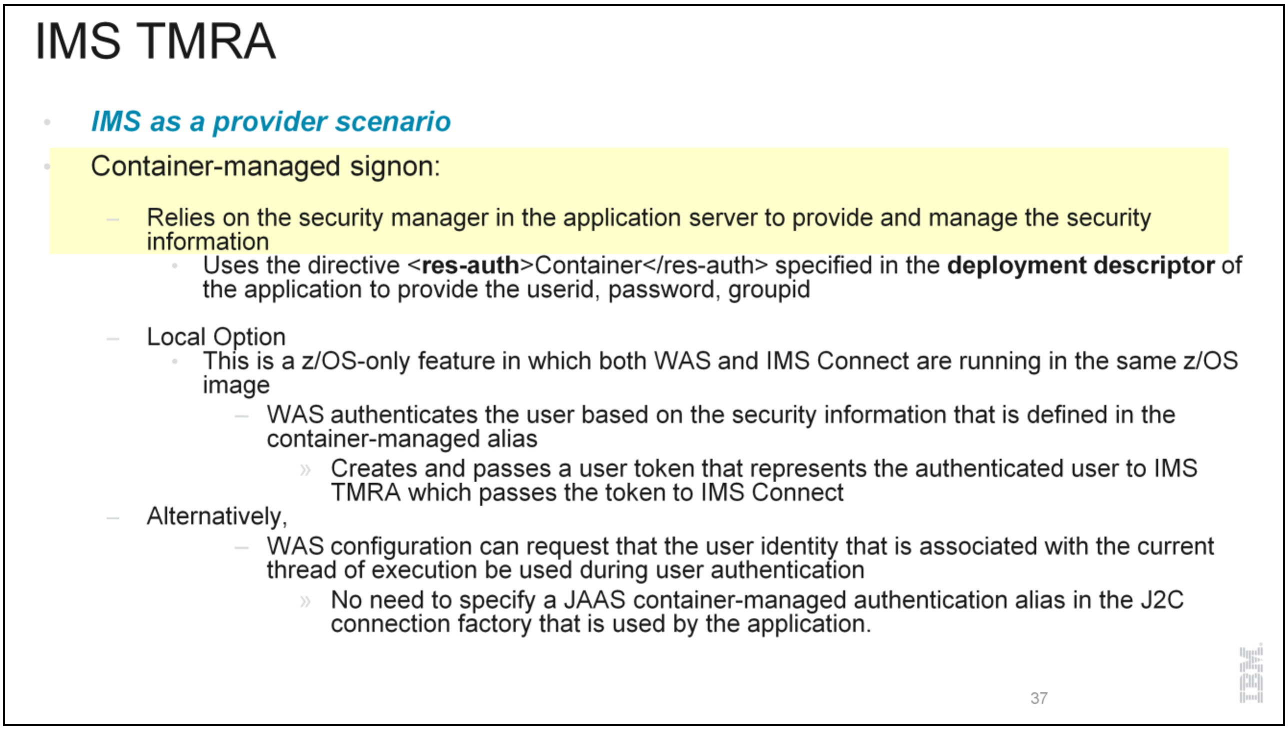


WAS – IMS TMRA

- Connectivity between IMS TMRA and IMS Connect
  - Transport Level: recommendation is to use AT/TLS with IMS Connect
  - Message Level: Supports passing the userid/password/groupid authentication credentials that are supported by IMS Connect
    - Supplied either by

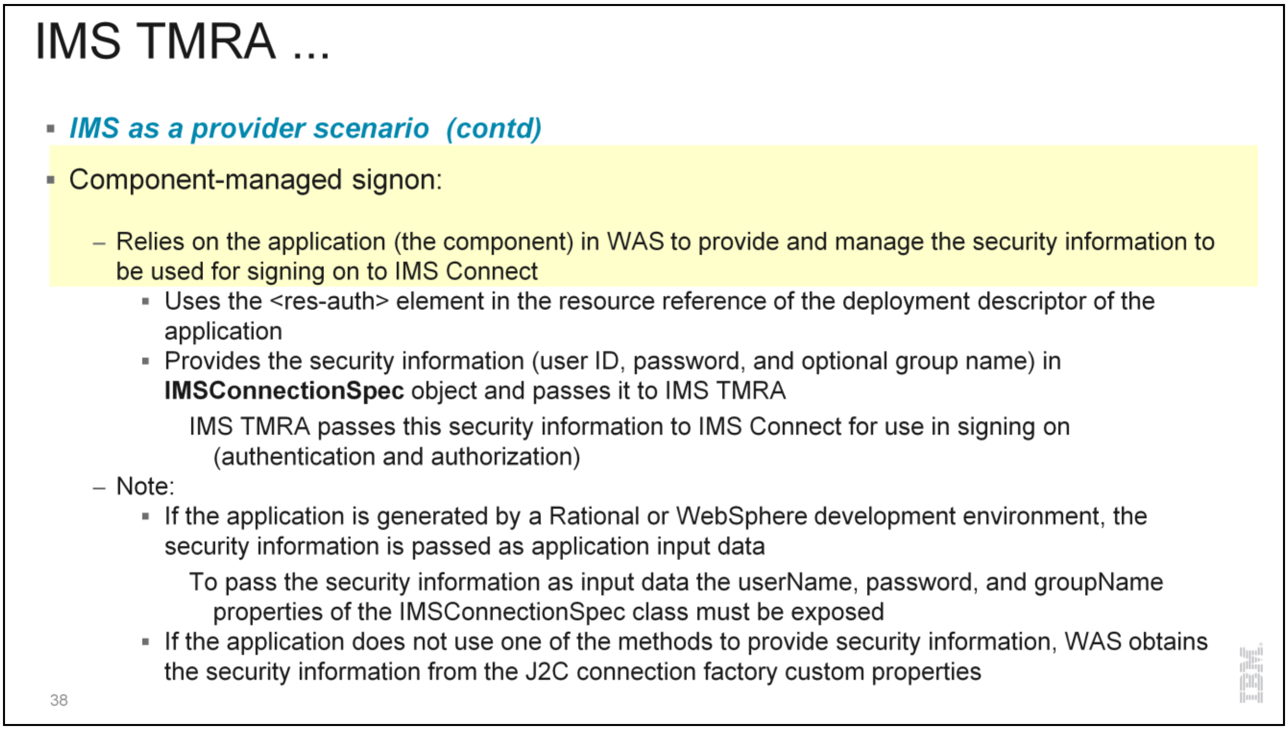
      The WAS application component (component-managed signon)

      Or by the application server (container-managed signon).



IMS TMRA

- ***IMS as a provider scenario***
- Container-managed signon:
  - Relies on the security manager in the application server to provide and manage the security information
    - Uses the directive <**res-auth**>Container</res-auth> specified in the **deployment descriptor** of the application to provide the userid, password, groupid
  - Local Option
    - This is a z/OS-only feature in which both WAS and IMS Connect are running in the same z/OS image
      - WAS authenticates the user based on the security information that is defined in the container-managed alias
        - Creates and passes a user token that represents the authenticated user to IMS TMRA which passes the token to IMS Connect
  - Alternatively,
      - WAS configuration can request that the user identity that is associated with the current thread of execution be used during user authentication
        - No need to specify a JAAS container-managed authentication alias in the J2C connection factory that is used by the application.

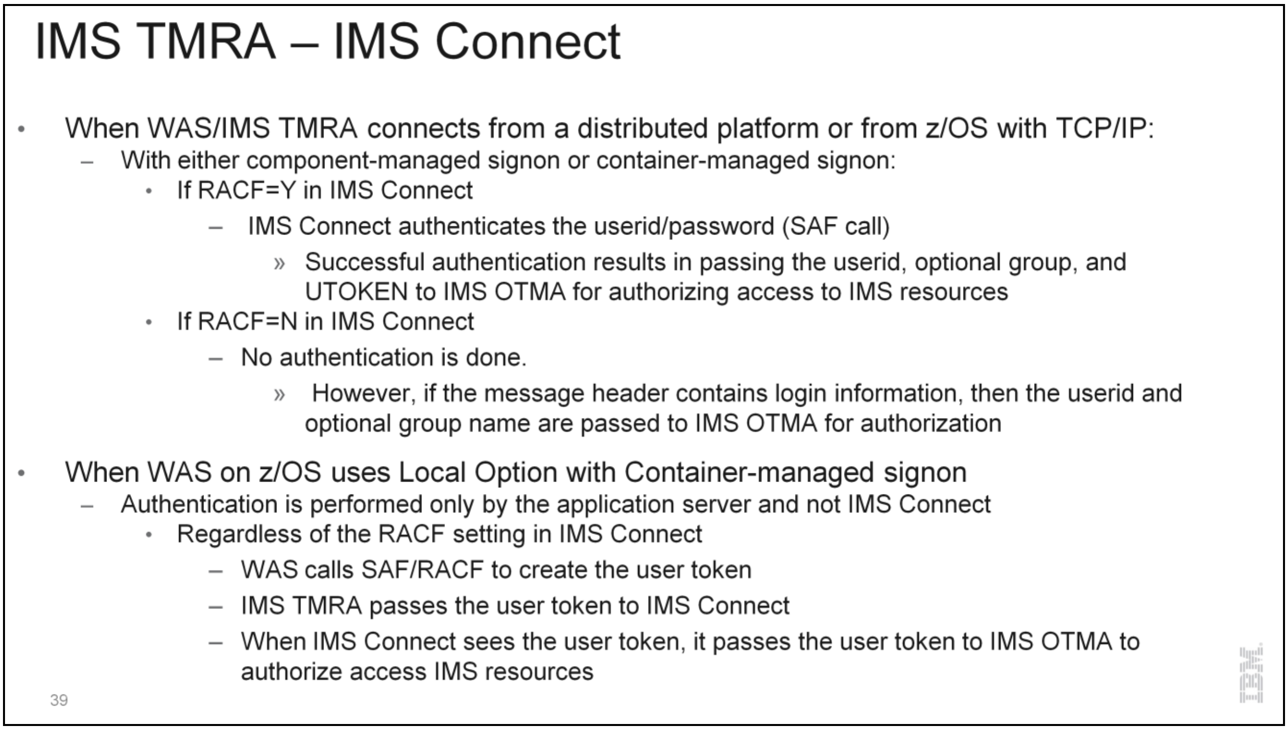


IMS TMRA ...

- ***IMS as a provider scenario (contd)***
- Component-managed signon:
  - Relies on the application (the component) in WAS to provide and manage the security information to be used for signing on to IMS Connect
    - Uses the <res-auth> element in the resource reference of the deployment descriptor of the application
    - Provides the security information (user ID, password, and optional group name) in **IMSConnectionSpec** object and passes it to IMS TMRA

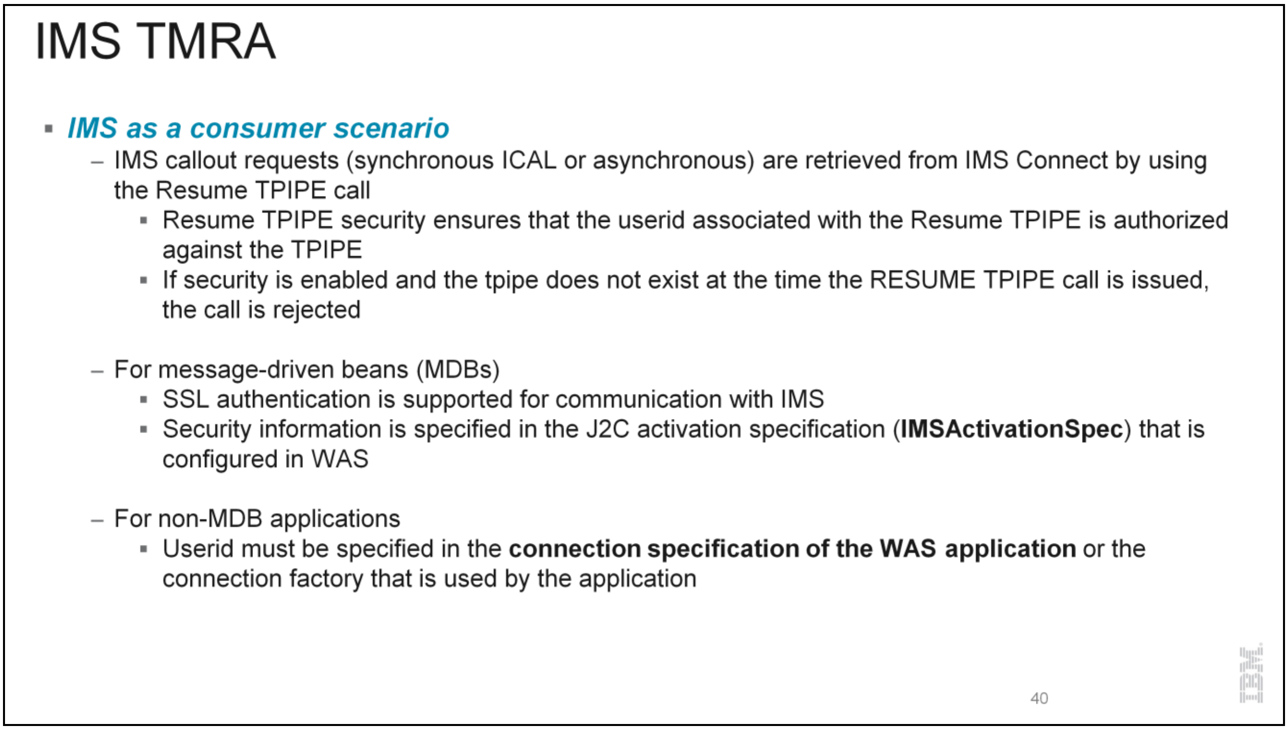
      IMS TMRA passes this security information to IMS Connect for use in signing on (authentication and authorization)
  - Note:
    - If the application is generated by a Rational or WebSphere development environment, the security information is passed as application input data

      To pass the security information as input data the userName, password, and groupName properties of the IMSConnectionSpec class must be exposed
    - If the application does not use one of the methods to provide security information, WAS obtains the security information from the J2C connection factory custom properties



IMS TMRA – IMS Connect

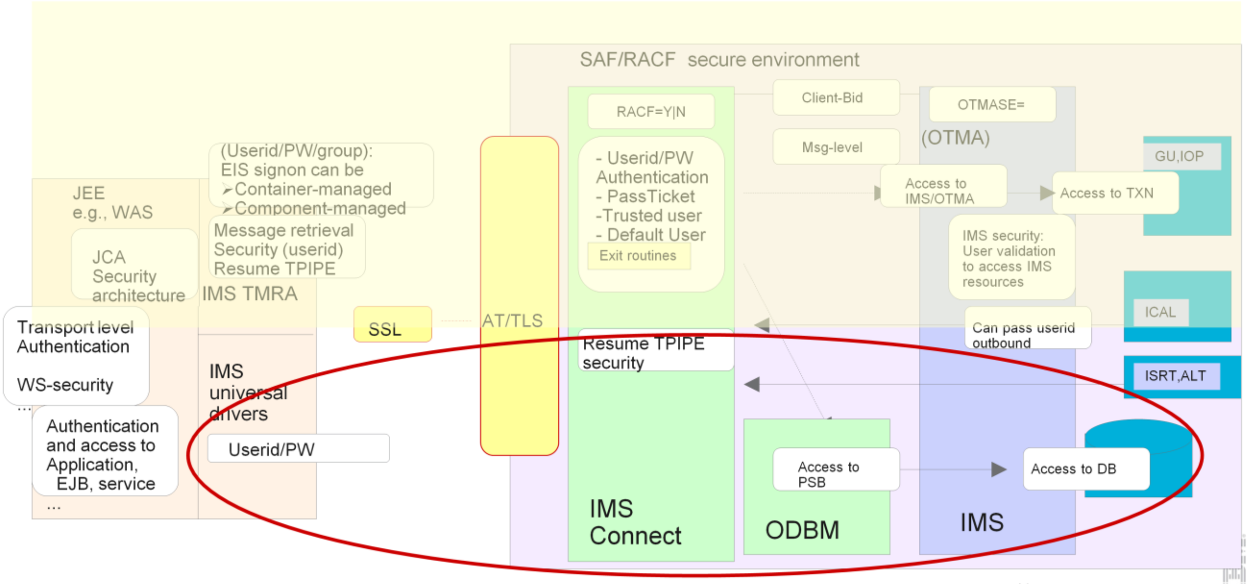
- When WAS/IMS TMRA connects from a distributed platform or from z/OS with TCP/IP:
  - With either component-managed signon or container-managed signon:
    - If RACF=Y in IMS Connect
      - IMS Connect authenticates the userid/password (SAF call)
        - Successful authentication results in passing the userid, optional group, and UTOKEN to IMS OTMA for authorizing access to IMS resources
    - If RACF=N in IMS Connect
      - No authentication is done.
        - However, if the message header contains login information, then the userid and optional group name are passed to IMS OTMA for authorization
- When WAS on z/OS uses Local Option with Container-managed signon
  - Authentication is performed only by the application server and not IMS Connect
    - Regardless of the RACF setting in IMS Connect
      - WAS calls SAF/RACF to create the user token
      - IMS TMRA passes the user token to IMS Connect
      - When IMS Connect sees the user token, it passes the user token to IMS OTMA to authorize access IMS resources



IMS TMRA

- ***IMS as a consumer scenario***
  - IMS callout requests (synchronous ICAL or asynchronous) are retrieved from IMS Connect by using the Resume TPIPE call
    - Resume TPIPE security ensures that the userid associated with the Resume TPIPE is authorized against the TPIPE
    - If security is enabled and the tpipe does not exist at the time the RESUME TPIPE call is issued, the call is rejected
  - For message-driven beans (MDBs)
    - SSL authentication is supported for communication with IMS
    - Security information is specified in the J2C activation specification (**IMSActivationSpec**) that is configured in WAS
  - For non-MDB applications
    - Userid must be specified in the **connection specification of the WAS application** or the connection factory that is used by the application

Open DB Security



SAF/RACF secure environment

JEE
e.g., WAS

JCA
Security
architecture

Transport level
Authentication

WS-security
…

Authentication
and access to
Application,
EJB, service
…

(Userid/PW/group):
EIS signon can be
➢Container-managed
➢Component-managed

Message retrieval
Security (userid)
Resume TPIPE

IMS TMRA

IMS
universal
drivers

Userid/PW

SSL

AT/TLS

RACF=Y|N

- Userid/PW
Authentication
- PassTicket
-Trusted user
- Default User
Exit routines

Resume TPIPE
security

IMS
Connect

Client-Bid

Msg-level

Access to
IMS/OTMA

OTMASE=

(OTMA)

Access to TXN

GU,IOP

IMS security:
User validation
to access IMS
resources

Can pass userid
outbound

ICAL

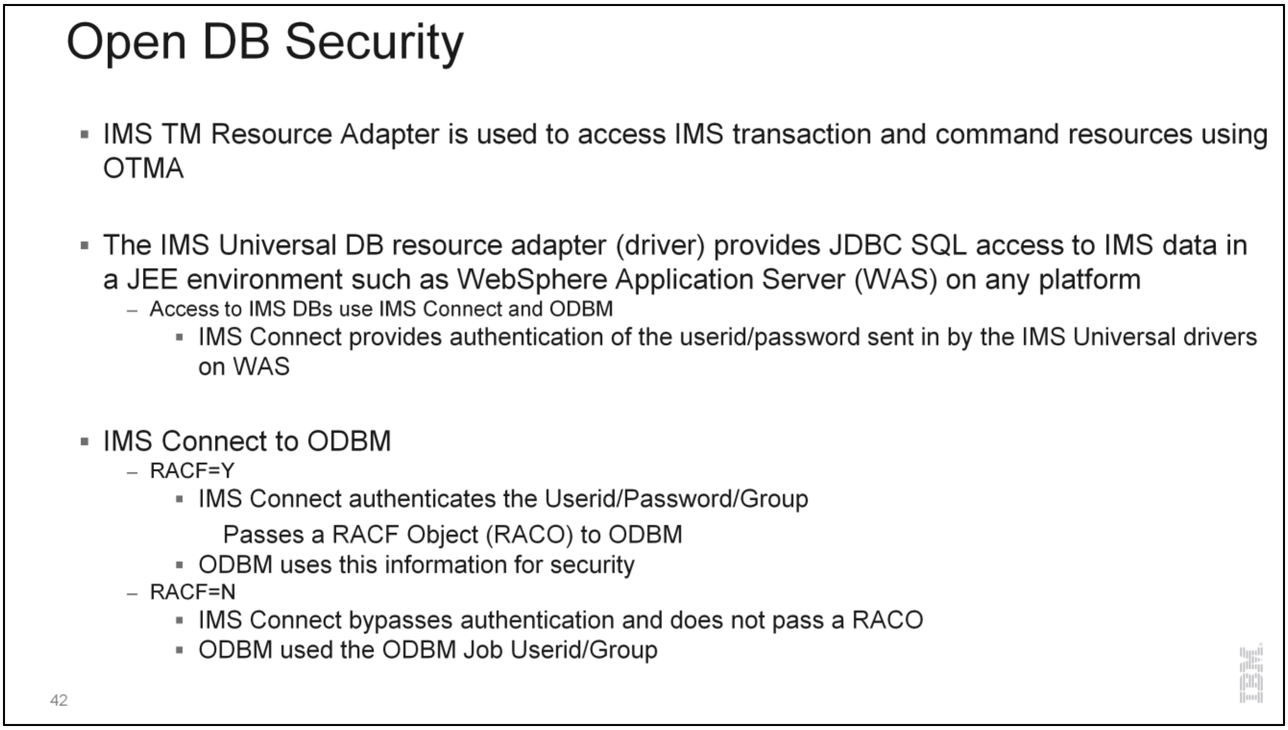
ISRT,ALT

Access to
PSB

ODBM

Access to DB

IMS



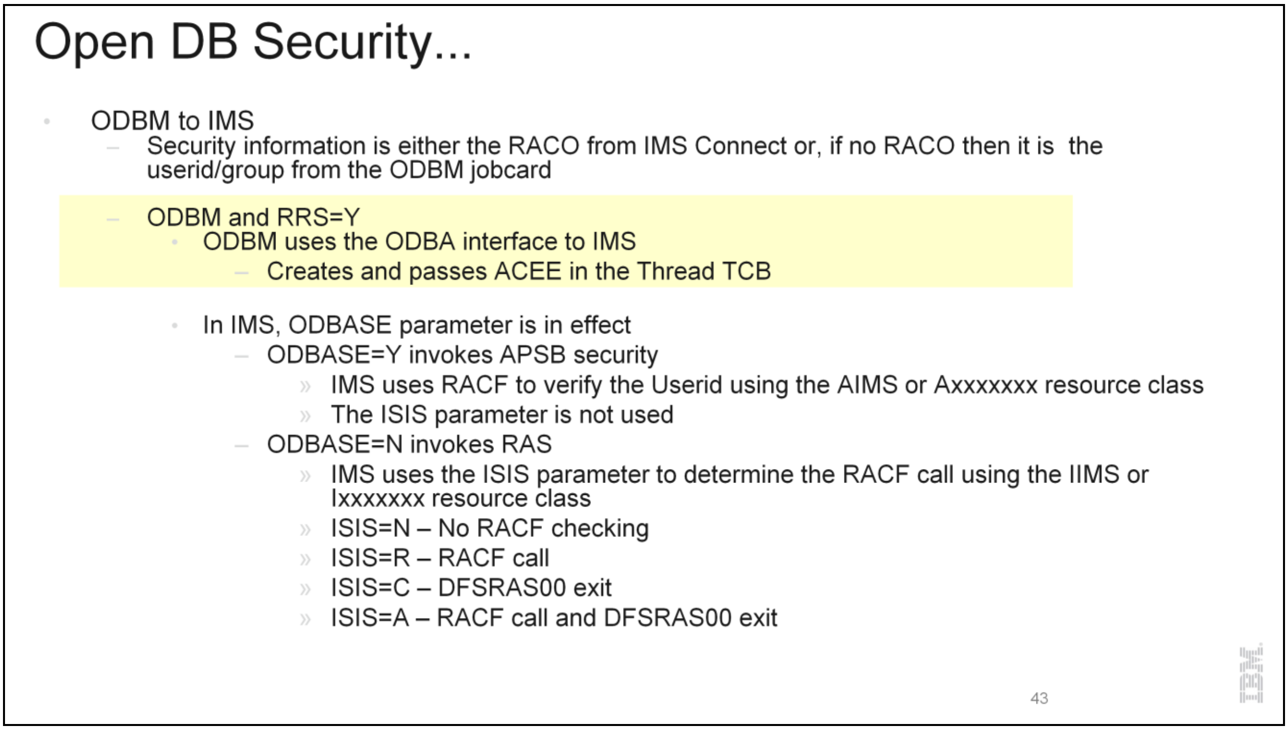
Open DB Security

- IMS TM Resource Adapter is used to access IMS transaction and command resources using OTMA

- The IMS Universal DB resource adapter (driver) provides JDBC SQL access to IMS data in a JEE environment such as WebSphere Application Server (WAS) on any platform
  - Access to IMS DBs use IMS Connect and ODBM
    - IMS Connect provides authentication of the userid/password sent in by the IMS Universal drivers on WAS

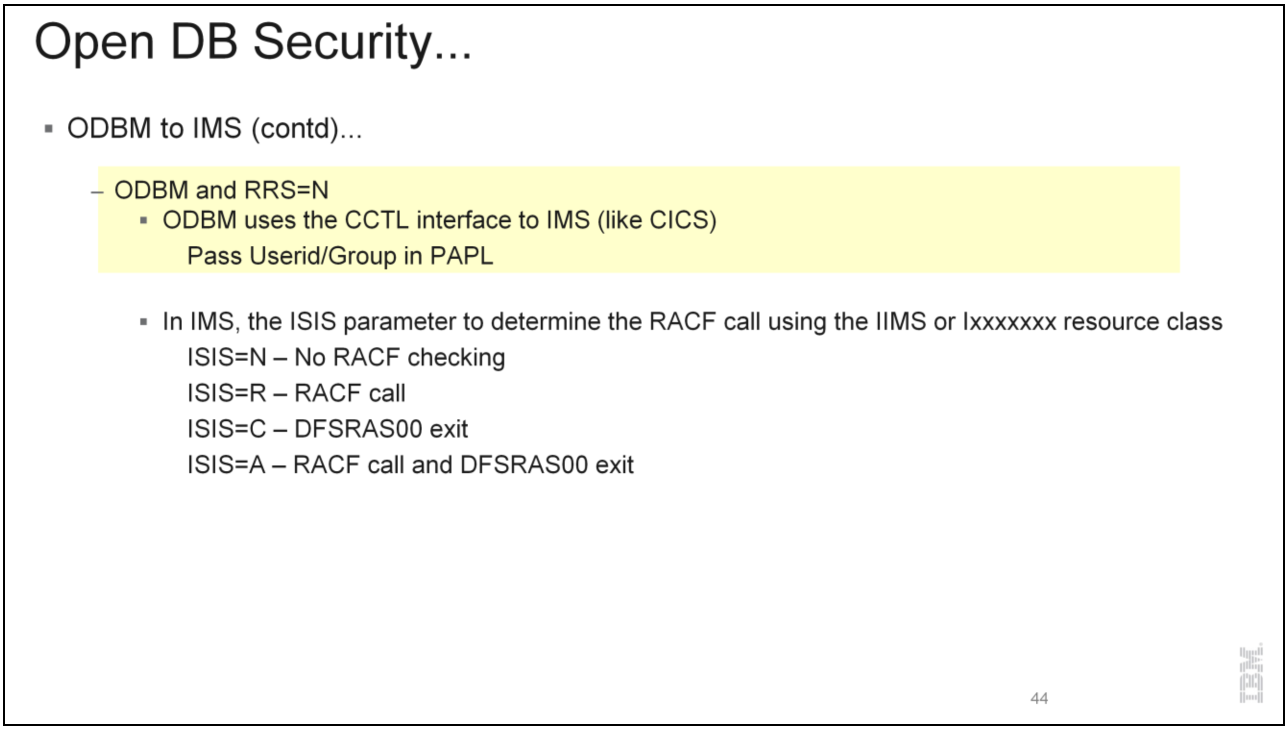
- IMS Connect to ODBM
  - RACF=Y
    - IMS Connect authenticates the Userid/Password/Group

      Passes a RACF Object (RACO) to ODBM
    - ODBM uses this information for security
  - RACF=N
    - IMS Connect bypasses authentication and does not pass a RACO
    - ODBM used the ODBM Job Userid/Group



Open DB Security...

- ODBM to IMS
  - Security information is either the RACO from IMS Connect or, if no RACO then it is the userid/group from the ODBM jobcard
  - ODBM and RRS=Y
    - ODBM uses the ODBA interface to IMS
      - Creates and passes ACEE in the Thread TCB
    - In IMS, ODBASE parameter is in effect
      - ODBASE=Y invokes APSB security
        - IMS uses RACF to verify the Userid using the AIMS or Axxxxxxx resource class
        - The ISIS parameter is not used
      - ODBASE=N invokes RAS
        - IMS uses the ISIS parameter to determine the RACF call using the IIMS or Ixxxxxxx resource class
        - ISIS=N – No RACF checking
        - ISIS=R – RACF call
        - ISIS=C – DFSRAS00 exit
        - ISIS=A – RACF call and DFSRAS00 exit



Open DB Security...

- ODBM to IMS (contd)...
  - ODBM and RRS=N
    - ODBM uses the CCTL interface to IMS (like CICS)

      Pass Userid/Group in PAPL

    - In IMS, the ISIS parameter to determine the RACF call using the IIMS or Ixxxxxxx resource class

      ISIS=N – No RACF checking

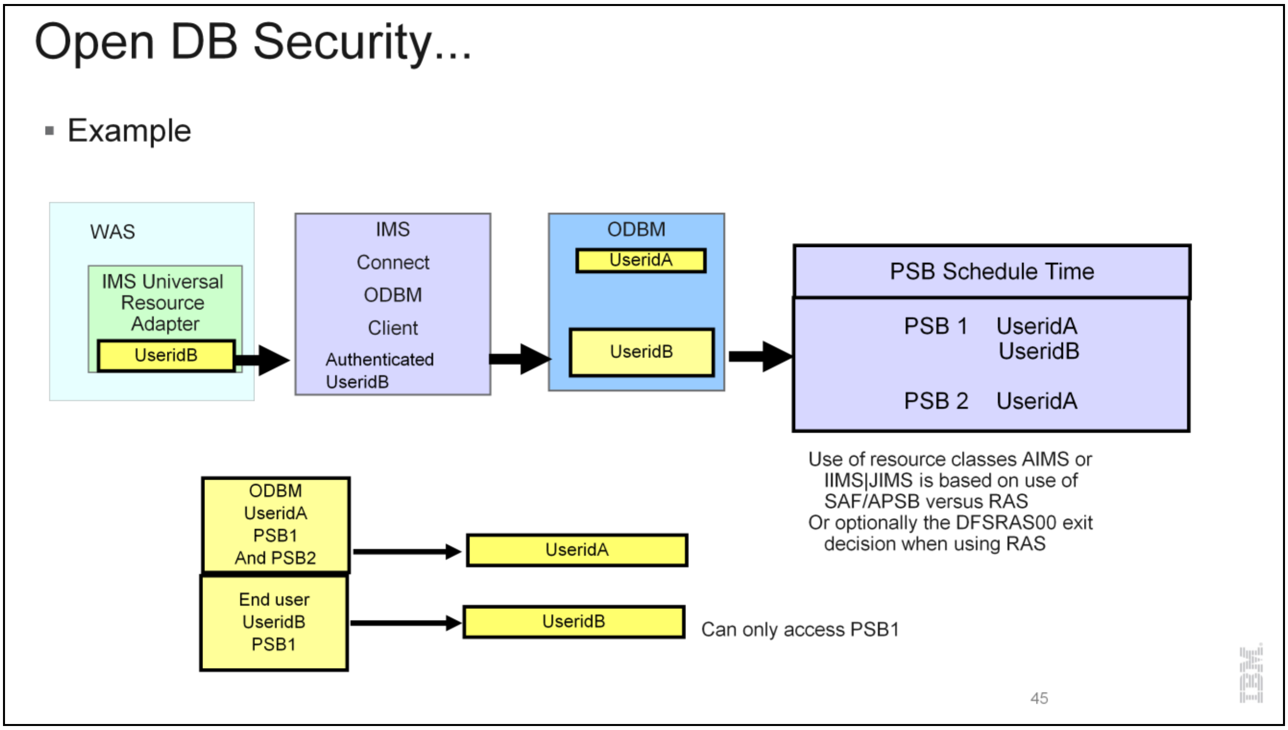
      ISIS=R – RACF call

      ISIS=C – DFSRAS00 exit

      ISIS=A – RACF call and DFSRAS00 exit

Open DB Security...

- Example



WAS

IMS Universal Resource Adapter

UseridB

IMS Connect ODBM Client

Authenticated UseridB

ODBM

UseridA

UseridB

PSB Schedule Time

PSB 1	UseridA UseridB
PSB 2	UseridA

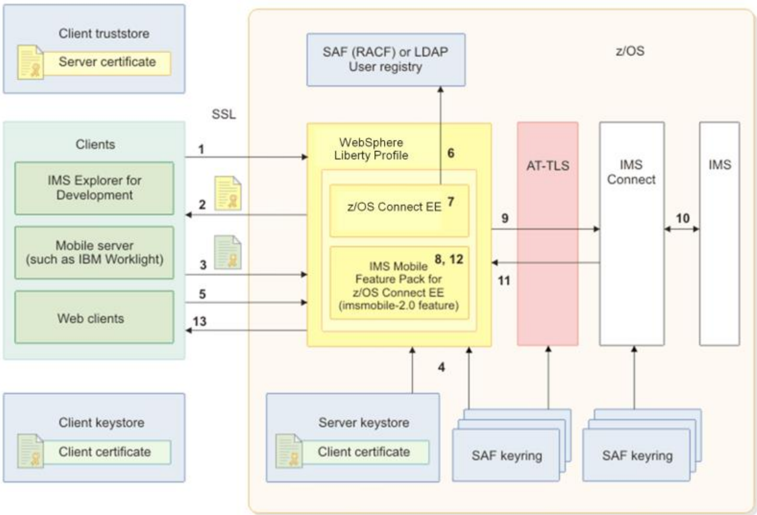
Use of resource classes AIMS or IIMS|JIMS is based on use of SAF/APSB versus RAS
Or optionally the DFSRAS00 exit decision when using RAS

ODBM UseridA PSB1 And PSB2 → UseridA

End user UseridB PSB1 → UseridB

Can only access PSB1

And most recently…. Mobile access



And most recently…. Mobile access

1. The client initiates an HTTPS call to IBM WebSphere Liberty Profile
2. z/OS Connect is configured with SSL client authentication and a fallback to basic authentication
3. The client sends a client certificate. (Note: IMS Explorer does not send in the client certificate. A valid user ID (registered in the RACF® or LDAP user registry) must be specified in IMS Explorer for Development when you use the provided wizard to create and publish a mobile service. With this user ID properly configured, IMS Explorer for Development is considered a trusted client)
4. WebSphere Liberty Profile verifies the client certificate with the previously imported client certificate that is stored in the sever truststore or keyring. If the client certificate is missing, basic authentication is applied against the user registry that was configured (SAF or LDAP).
5. The client starts transmitted data over a secure connection
6. For a service request, WebSphere Liberty Profile authenticates the user credential. Then z/OS Connect authorizes the user by using a SAF call to validate that the group names in the service configuration matches one of the group names associated with the user ID in the request subject.
7. After authentication and authorization, z/OS Connect passes the request to the IMS Mobile feature for transforming the data from JSON to bytes. If authentication and authorization fail, an error is returned to the client.
8. The IMS Mobile feature transforms the incoming request from JSON to bytes. The user ID is extracted from the request subject from z/OS Connect
    - ★ If a user name and a password are specified in IMS Explorer for Development (V3.2 or later is required) when the connection profile is created, they are used for SAF authentication with IMS Connect.
        - If no user name is specified in the connection profile: The password that is specified in the connection profile is ignored.
    - *Otherwise*, the user ID is extracted from the request subject. WebSphere® Liberty z/OS® Connect must be configured with SAF registry authentication and the subject must be 8 bytes or less in order for the IMS Mobile feature to retrieve the user ID from the request subject. The IMS technical password is the SAF password for the user.
        - If the request subject is more than 8 bytes, or authentication is disabled on z/OS Connect, the IMS Mobile feature retrieves the user ID from the technical ID, an IMS mobile global element that is specified during initial installation and setup. The IMS technical password is the SAF password for the user.
            - If the technical ID is left blank, the IMS Mobile feature uses the z/OS Connect started job user ID. The IMS technical password, if specified, is the SAF password for the user.
9. The IMS Mobile feature initiates a request to send the input bytes array and RACF information to IMS Connect. The request triggers SSL handshake via AT-TLS, if it is configured, to protect the communication between WebSphere Application Server Liberty Profile and IMS Connect.
10. IMS Connect flows the request with the RACF user ID to IMS. IMS might perform additional authorization, depending on the setting. IMS transaction runs. IMS returns response (bytes) to IMS Connect.IMS Connect returns response (bytes) to the IMS Mobile feature.
11. The IMS Mobile feature transforms the response from bytes to JSON.
12. The response is returned to the client.

★ Still under consideration

The Bottom Line

- Multiple levels of security
  - OTMA
    - Validates whether an OTMA member (IMS Connect) can communicate with IMS
    - Implements transaction and command security
      - Userid that flows in on a message against the IMS resource
    - Supports callout to web services
  - ODBM
    - Passes security information to IMS for database access
  - IMS Connect
    - Supports the authentication of userids, groups, passwords and passes the utoken to IMS with the message
    - Additionally extends the security authentication
      - PassTicket support
      - Trusted User support
  - Network – connection security and encryption
    - SSL – TLS
    - AT-TLS