PPT Overview

Monday, June 06, 2016 5:47 PM

Overview of API Test

Day1_UFT API Testing Overview_Intro.pptx
4/18/2016 10:36 AM, 436 KB

what is API test?----*Application Programming Interface*----Compared to simple test: no UI to test, concentrates on business logical layer

what to test?

- validate values of params,
- boundary values of params,
- boundary of results,
- overflow values of results,
- invalid values of params,
- null values of params,
- security

who to test?

- developers with unit test (white box, limited to given unit),
- who does black box test, integration test, capacity and load test.

challenges of api test?

1. •Need to test early, can't wait for UI
2. •QA needs to get more information from Dev
3. •If there's no UI, need to test service directly
4. •Services need to conform to standards
5. •Services are fragile, and can be broken easily
6. •Services are vulnerable, as anything can be passed in
7. •Applications need to be tested end-to-end

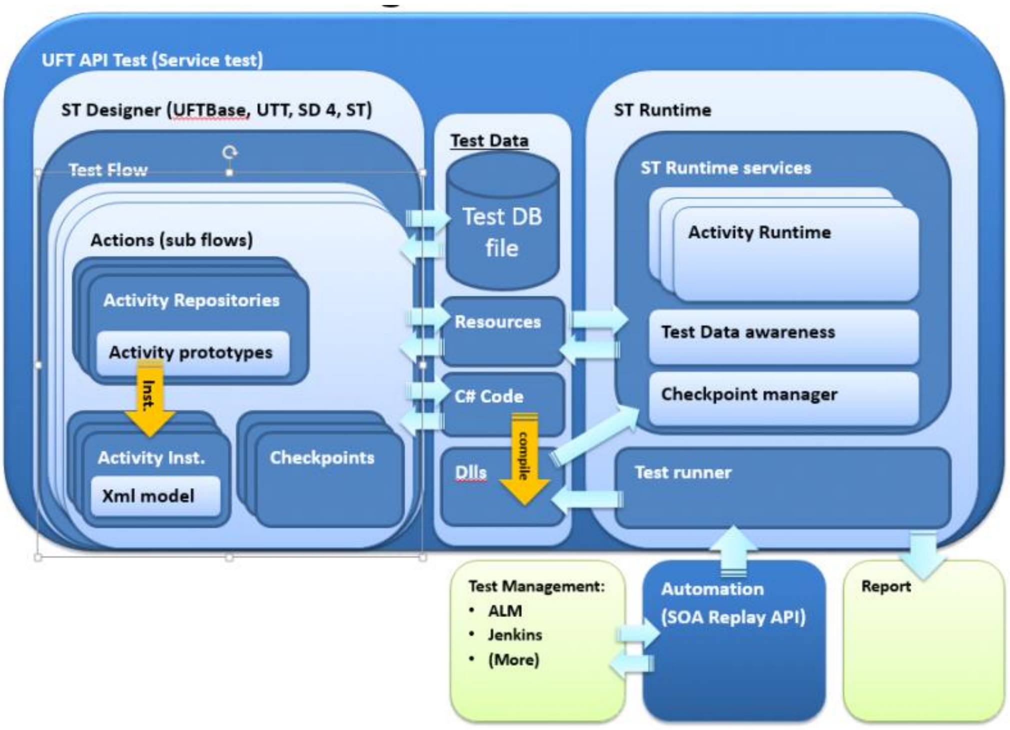
UFT API Test (former Service Test)

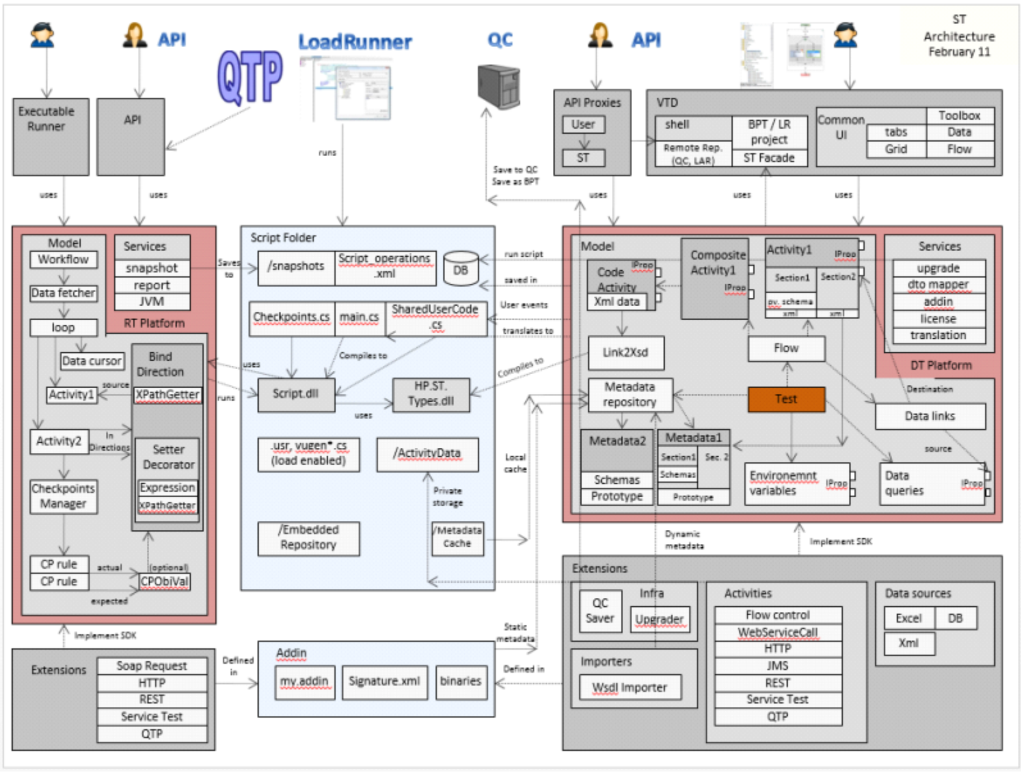
- •Drag and Drop UI
- •Extensible Framework
- •Powerful Data Handling
- •Validate with Checkpoints
- •Integration ALM/LR
- •Flexible Custom Codes

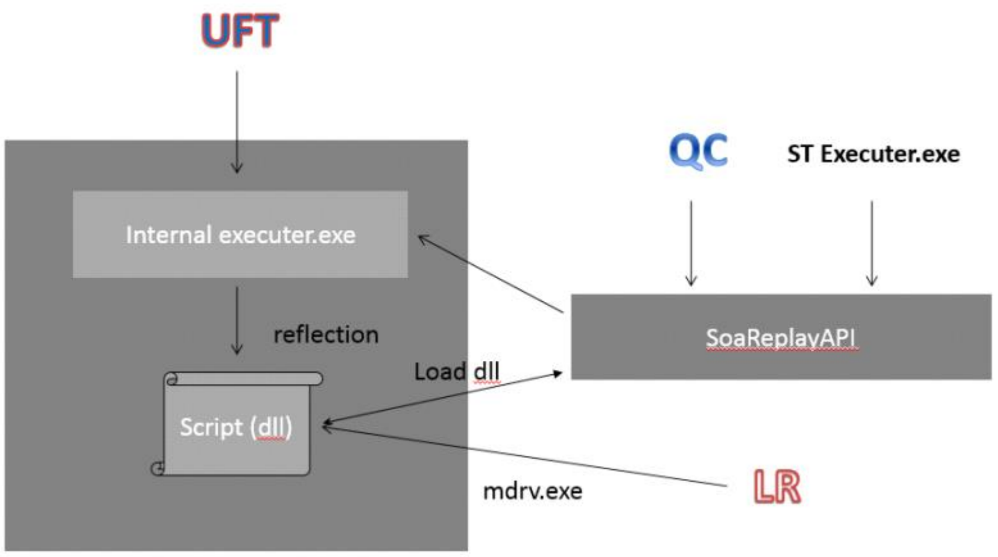
UFT API Testing Architecture

Day1_UFT_API Architecture Overview_part1.pptx
4/19/2016 1:37 PM, 62.9 KB

UFT API Testing Architecture


UFT API Test (Service test)
ST Designer (UFTBase, UTT, SD 4, ST)
Test Flow
Actions (sub flows)
Activity Repositories
Activity prototypes
Inst.
Activity Inst.
Xml model
Checkpoints
Test Data
Test DB file
Resources
C# Code
Dlls
compile
ST Runtime
ST Runtime services
Activity Runtime
Test Data awareness
Checkpoint manager
Test runner
Test Management:
• ALM
• Jenkins
• (More)
Automation
(SOA Replay API)
Report


API
QTP
LoadRunner
QC
API
ST
Architecture
February 11
Executable Runner
API
runs
Save to QC
Save as BPT
API Proxies
User
ST
VTD
shell
BPT / LR project
Remote Rep. (QC, LAR)
ST Facade
Common UI
tabs
Grid
Toolbox
Data
Flow
uses
uses
uses
uses
uses
Model
Workflow
Data fetcher
loop
Data cursor
Activity1
source
Activity2
In Directions
Checkpoints Manager
CP rule
CP rule
actual
expected
(optional)
CPObjVal
Bind Direction
XPathGetter
Setter Decorator
Expression
XPathGetter
Services
snapshot
report
JVM
RT Platform
Saves to
uses
runs
Script Folder
/snapshots
Script_operations .xml
DB
Checkpoints.cs
main.cs
SharedUserCode .cs
Compiles to
Script.dll
HP.ST. Types.dll
uses
.usr, vugen*.cs (load enabled)
/ActivityData
Private storage
/Embedded Repository
/Metadata Cache
run script
saved in
User events
translates to
Compiles to
Local cache
Static metadata
Defined in
Model
Code Activity
IProp
Xml data
Composite Activity1
IProp
Activity1
IProp
Section1
Section2
pv. schema
xml
xml
Link2Xsd
Flow
Metadata repository
Test
Metadata2
Schemas
Prototype
Metadata1
Section1
Sec. 2
Schemas
Prototype
Environemnt variables
IProp
Services
upgrade
dto mapper
addin
license
translation
DT Platform
Destination
Data links
source
Data queries
IProp
Dynamic metadata
Implement SDK
Implement SDK
Extensions
Soap Request
HTTP
REST
Service Test
QTP
Defined in
Addin
my.addin
Signature.xml
binaries
Extensions
QC Saver
Infra
Upgrader
Importers
Wsdl Importer
Activities
Flow control
WebServiceCall
HTTP
JMS
REST
Service Test
QTP
Data sources
Excel
DB
Xml
run time



Technology-- •Net 4.5 •C# •VS 2012 •SQLite •WinForms •Infragistics / DevExpress •.....

ST Design time ≠ ST Runtime

1. •Different model (xml model)
2. •Design time is part of UFT UI
3. •Runtime is CommandLine& COM
4. •Runtime allows multiple instances

Xml data model & XmlGrid

1. •The XmlModel is a big part of ST
2. •Xsd schema defines the structure of the data
3. •Schema particles = prototypes for data items
4. •Xml data populates the model
5. •The XmlGrid is the UI representation of this model
6. •XmlGrids are a part of every property tab
7. •A single activity may use up to 10 xml models

SQlite db file in design time

1. •A database file, used to store the test info
2. •Only used for loading the test in designer
3. •When saving, code is generated and compiled
4. •Runtime only uses the compiled dll

Backward compatible tests

1. •A new UFT version is able to open old API tests
2. •Upgraders are used to avoid multiple test versions
3. •Upgraders convert the test to a new UFT format
4. •Only design time (db + code) is upgraded
5. •Runtime keeps working since it is already compiled

code structure

Day1_API Test Code Structure_part3.pptx
4/19/2016 1:37 PM, 58.8 KB

Most of the projects in ST branch

- ◦ ST: *https://csvn1-pro.isr.hp.com:19181/svn/tsg-bto-apps-st*

Few projects:

1. ◦ UTT: *https://csvn1-pro.isr.hp.com:19181/svn/tsg-bto-apps-utt/trunk/src/Infra/app/ST*

- Common projects, can be used for other applications (as LR)
- WsSecurity: all processes related to security of web service
- XmlGrid: building Grid input and output properties based on XML for API test
- Utilities: some utilities, especially is StHttpRequest.cs: all processes for building http request and response
- JsonXmlConverter: convert Jsonto xml to display in XML Grid

1. ◦ UFT-Base: https://csvn1-pro.isr.hp.com:19181/svn/tsg-bto-apps-uftbase/trunk/src/UFT/Projects/Services/BatchRunner

- Integration with UFTBatchRunner to run many API tests in batch

WCF in LR:

- *https://csvn1-pro.isr.hp.com:19181/svn/tsg-bto-apps-lt-lr/trunk/app/Protocols/WebServicesProtocol/WebServices/SOAWcfRouter/app/WcfRouter*
- ◦ WCF processing in Web service (binding, security,...)

SOATestDesigner

- SOATestDesigner: *ST\app\VTD\Framework\DesignerFramework\app\SOATestDesigner.sln*
- ◦ All actions related to UI: open test, save test, edit test,... on UI
- ◦ Model classes: Test, Activity, Link,...v.v

SOA2BasicFWK:

- *ST\app\VTD\Framework\BasicFramework\app\SOA2BasicFWK.sln*
- ◦ SOAReplayAPI: Entry point for Replay Service
- ◦ InternalExecuter.exe: replay API test
- ◦ ServiceTestExecuter.exe: run API test by Command Line

SOA2TechImplEXT:

- *ST\app\VTD\Extensibility\TechnologyImplementationExtensibility\app\SOA2TechImplEXT.sln*
- ◦ Extensibility Technology
- ◦ Defines all activities via Toolbox
- Design time: provide UI elements and model (What should I look like?)
- Runtime: what should I behave?

SOA2TechRTFWK:

- *ST\app\VTD\Framework\TechnologiesRuntimeFramework\app\SOA2TechRTFWK.sln*
- ◦ Runtime Framework
- ◦ Process data binding (VTDBindingImpl)
- ◦ Process checkpoint rules
- ◦ Create report (ReporterFWK)

SOA2TechRTFWKCpp

- *ST\app\VTD\Framework\TechnologiesRuntimeFramework\app SOA2TechRTFWKCpp.sln*
- ◦ C++ class
- ◦ JavaInfraproject: responsible for loading and unloading JVM

TechImplExtCpp

- *ST\app\VTD\Extensibility\TechnologyImplementationExtensibility\app\TechImplExtCpp.sln*
- ◦ JMS code
- ◦ Java framework works with JNI

SharedComponents

- *ST\app\VTD\Utils\Shared\app\SharedComponents.sln*
- ◦ Few common projects
- ◦ WsdlImporter: import wsdl
- ◦ Utilities: some utilities

Code structure of script

- <APITest_name>.csproj
- Main.cs
- Checkpoints.cs
- TestUserCode.cs
- EnvironmentProfiles.xml
- TestInputParameters.xml
- TestOutputParameters.xml
- DataQueries: test data driven
- EmbeddedResources: WSDL files, Rest prototype

Debugging Session with main features

Day2_Debugging Session with main features_WS....
4/19/2016 1:38 PM, 157 KB

Attention for API test project debug:

- copy necessary dll references or configuration in the according property by adding the folder you find by Everything tool to the folder workDir/out/bin to ensure all references is ok
- compile the code you want to debug and get the compiled dll, then copy it to the UFT according directory to overwrite it
- attach the UFT process when debug in VS
- set the main breakpoints to watch it.
- If you want debug your own code in UFT, just change the mode to allow debug. In *Tools/Options/API test/*click allow debug mode

Attention for ALM project Debug:

- When connecting ALM, Server Url: http://16.153.233.15:8080/qcbin, username: sa, password: [blank], domain: Phuong, Project: APITest_NoVersion
- above setting which you can open it by IE, then watch the ALM project that you have created by GUI.
- above setting , you can manage create domain and project for your own in site Administration.

- this certificate below is used to security about WCF web service. If you don't configure it , server may refuse you web service request. *C:\Users\jzhang4\Work\app\VTD\Extensibility\TechnologyImplementationExtensibility\app\WebServicesTechnologyRTIntegrationTests\Certificates* to
- when newing api test, select location from ALM

Open Test – Load Test

- –*ST\app\VTD\Framework\DesignerFramework\app\DesignerModel\ServiceTestFactory.cs*
- –public IServiceTest Load(ShellTest shellTest)---Load test data from test.db TestPersistence.LoadTest(db, internalTest);

Save Test

- –*..\..\ServiceTestFactory.cs : SaveTest(…)*
- –*ST\app\VTD\Framework\DesignerFramework\app\DesignerModel\Test.cs*
- •public virtual void *Save*()， Save test to its own database: TestPersistence.SaveTest(db, this);

Save as Test

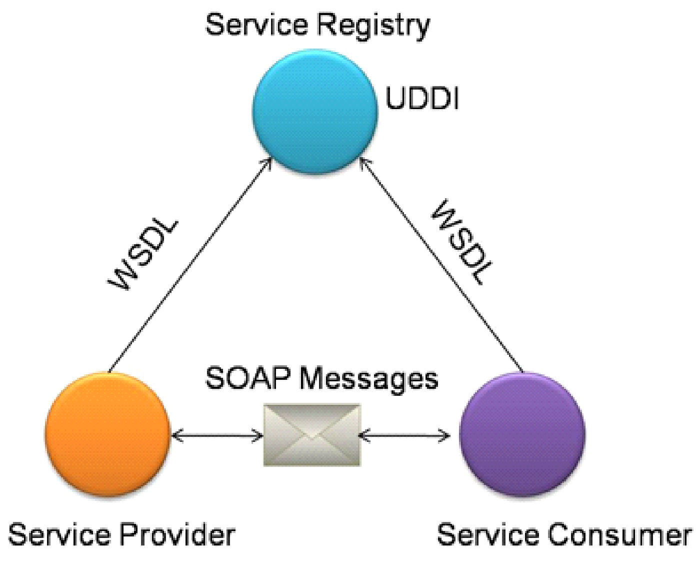
- –*ServiceTestProject.cs ， SaveAs(….)*

Debug RT Script Project:

- •Open CS project of the test in VS
- •Open Properties -> Debug tab -> Start external program

1. –Enter: C:\Program Files (x86)\HP\Unified Functional Testing\bin\HP.ST.Fwk.InternalExecuter.exe
2. –Command line argument: -test "<test-folder>“
3. –Add class we need to debug and place breakpoint

Web-Service



- •XML: format the data
- •*SOAP*: transfer the data
- •*WSDL*: describing the services available
- •*UDDI*: listing what services are available
- Web Services is a mechanism to provide a standard means of communication between various applications operating on similar or heterogeneous platforms. The World Wide Web Consortium (*W3C*) is the official body for maintaining web services standards.
- In the context of Service Oriented Architecture(*SOA*), Web Services are used to facilitate communication between service providers and service consumers. Special adapters are used for applications that don’t support web services.
- *Krawler* employs web services standards that enable our Service Oriented Architecture to offer functionalities such as location neutrality and technology/platform independence.There are many web services standards used in the SOA framework, including WS-Security, WS-Transaction, WS-Reliable Messaging and WS-Policy. But the underlying principals of web services are publication, discovery and communication:
- *Publish* – Service Producers register their service in the SOA registry. Web Services Description Language (*WSDL*) is used to describe a service.
- *Discovery* – Service Consumers make a request for a service in the SOA registry. Universal Description, Discovery and Integration (*UDDI*) standard is used for locating a service in the registry.
- *Communication* – Simple Object Access Protocol (*SOAP*) is used for facilitating communication between Service Providers and Consumers.

Import WSDL

- •Import mechanism structure
- –ST\app\VTD\Extensibility\TechnologyImplementationExtensibility\app\WebServicesTechnologyDesignTime\WebServiceImporter.cs
- •ST\app\VTD\Utils\Shared\app\WsdlImporter\Implementation\Importer\NetworkImporter.cs
- •ST\app\VTD\Utils\Shared\app\WsdlImporter\Implementation\Importer\CustomImporter.cs

- •ST\app\VTD\Utils\Shared\app\WsdlImporter\Implementation\Importer\FuzzyImporter.cs
- •WSDL importers' types
  —Default
  —Fuzzy (if schemas have errors, use Fuzzy mode to try to fix them)
- Debug

WS - Security

WS-Security.pptx
4/20/2016 3:14 PM, 334 KB

Attention:

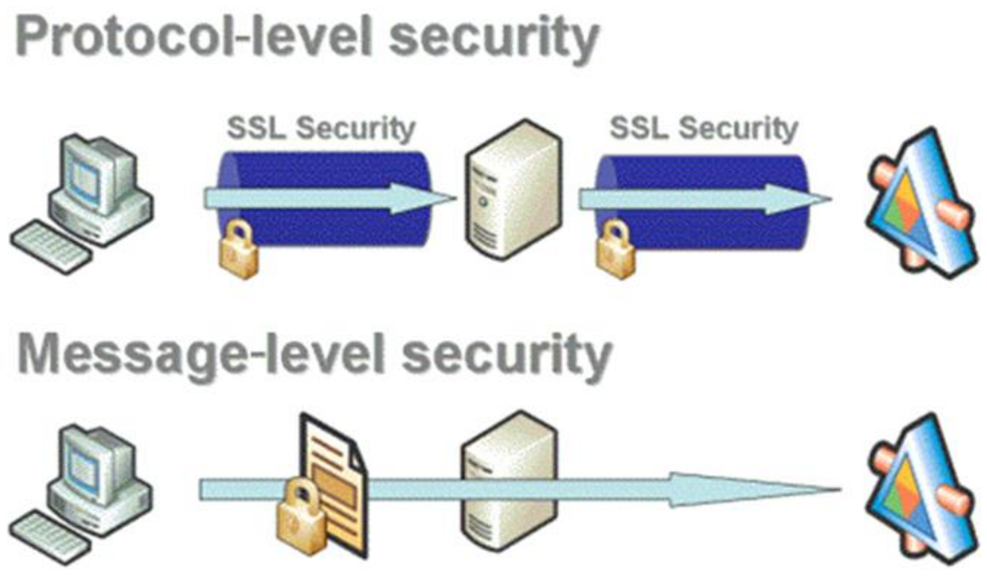
- before you want test web service security, you should install wsdl by unified function test's sample HP Flights
- order is important in security. E.g. token must before signature and encryption. Different composition may cause different result.
- security: element with body signature not ok, signature must be with reference if you want to use reference.
- make sure the WS addressing version when debug

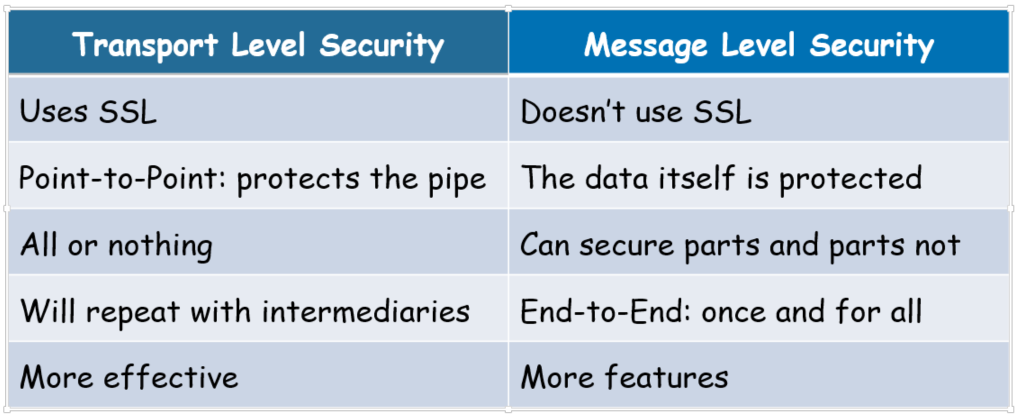
What is Security

- •Confidentiality: keeping message secret from unintended listeners during the transit. -- •Encryption
- •Integrity: the message was not corrupted in the way. -- •use signature
- •Authentication(&Authorization): solves questions as "Who is the other party?" and "How does he prove his identity?" -- use Authorization

Two Levels of Security

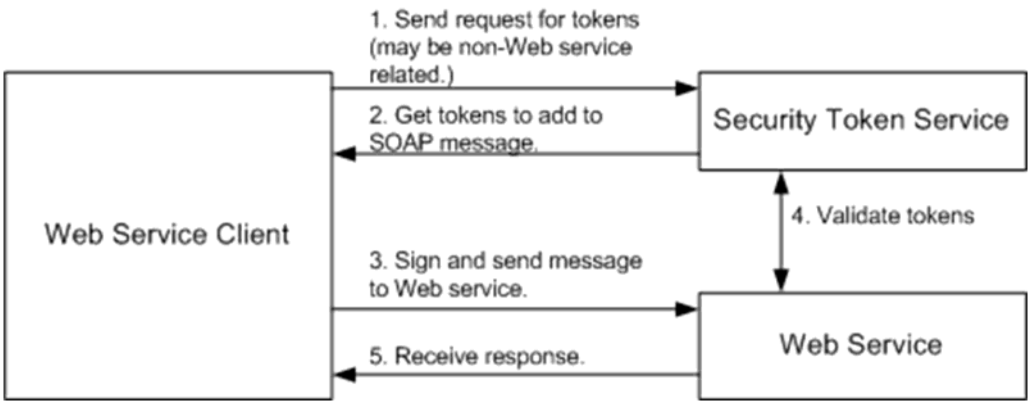
- —Transport Level Security (*TLS*)
- —Message Level Security
  •Security Tokens
  •Timestamp
  •Message Signature
  •Data Encryption





Transport Level Security	Message Level Security
Uses SSL	Doesn't use SSL
Point-to-Point: protects the pipe	The data itself is protected
All or nothing	Can secure parts and parts not
Will repeat with intermediaries	End-to-End: once and for all
More effective	More features

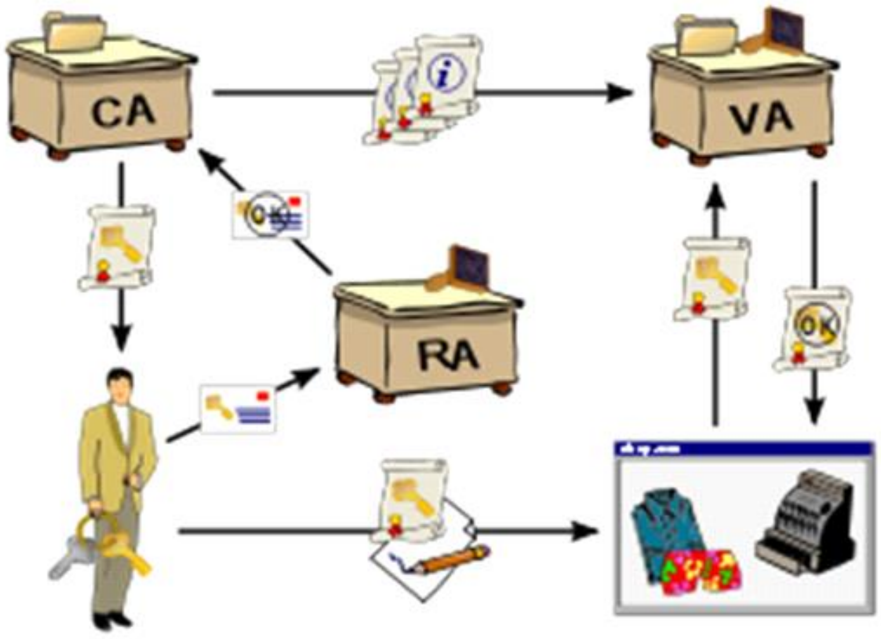
(*SSL*) secure socket layer


1. Send request for tokens (may be non-Web service related.)
Security Token Service
2. Get tokens to add to SOAP message.
Web Service Client
4. Validate tokens
3. Sign and send message to Web service.
Web Service
5. Receive response.

•All security configuration in Message Level Security will be under the Security header

Security Tokens

- UserNameToken:
    1. •<Password>(Optional): contains the password.
    –Type: the type is a URL that implies the type of the password: Text or Digest (hashed)Nonce
    1. •<Nonce> (Optional): uniquerandom number, it is used to ban replay attacks – no two requests have the same nonce value.
    –EncodingType: default is Base64
    1. •<Created> (Optional): specifies the creation time.
    2. •Nonce and created are used to create the hash function when sending the password in 64 encoding.
    3. •An additional way used to prevent replay attacks is to save the nonces used latest in the server
- X509 PKI (Public Key Infrastructure)
    •CA: certificate authority
    •RA: Registration authority
    •VA: Validation authority
    •SSO: single sign on : if you sign to one service you are signed to all



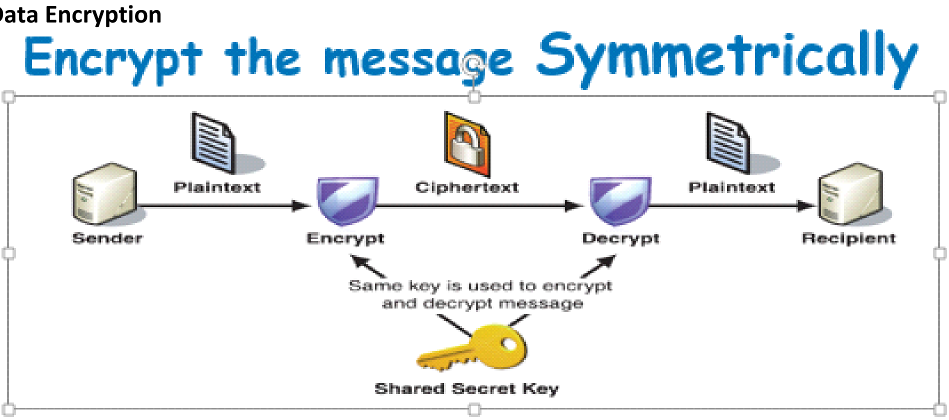
- •Kerberos 1
- •Kerberos 2
- •SAML

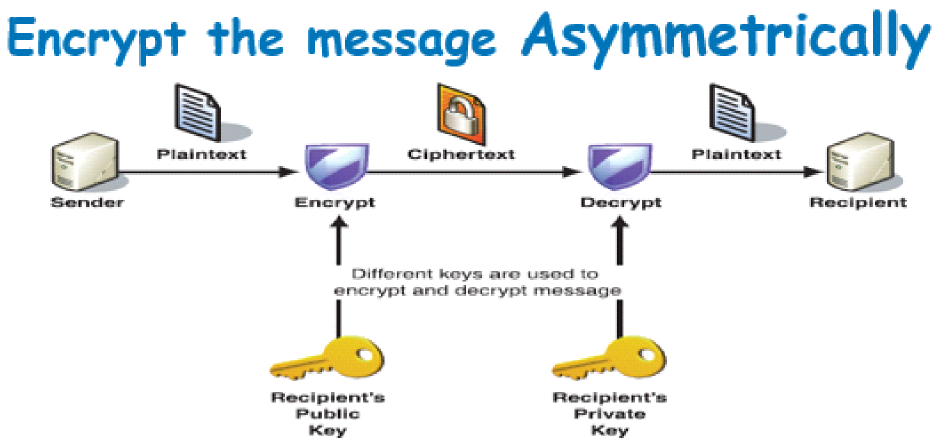
Timestamp Element: •Each element can be added alone to any other element

Message Signature

- •Based on Security Token
- •Integrity
- •Authentication (sometimes)
- •May sign some parts and some not.
- Will add ID attribute to signed parts
- xpath means that you can sign more dom in the XML more than body, timestamps, addressing
- •<signedInfo>: the information actually assigned.
- •<Transforms>: a list of <Transform>, each transform defines a processing step
- •<KeyInfo>: contains the key details by <SecurityTokenReference> element

Data Encryption


Encrypt the message Symmetrically
Plaintext
Ciphertext
Plaintext
Sender
Encrypt
Decrypt
Recipient
Same key is used to encrypt and decrypt message
Shared Secret Key



JMS (Java Message Service)

Day3_JMS.pptx
4/20/2016 3:16 PM, 65.1 KB

- •JMS (Java Messaging Service) is a peer-to-peer messaging system for java programs to send and receive messages.
- •Java API that allows applications to create, send, receive, and read messages
- •**Asynchronous**: A JMS provider can deliver messages to a client as they arrive; a client does not have to request messages in order to receive them.
- •**Reliable**: The JMS API can ensure that a message is delivered once and only once.

JMS Servers supported in UFT API

- •IBM MQ
- •IBM Websphere
- •Weblogic
- •Tibco
- •Apache ActiveMQ
- •Sonic MQ
- •Systinet SSJ