



Metro Web Services, NetBeans, GlassFish and OpenSSO in Action with Amazon WS, Azure, and Office

S305138

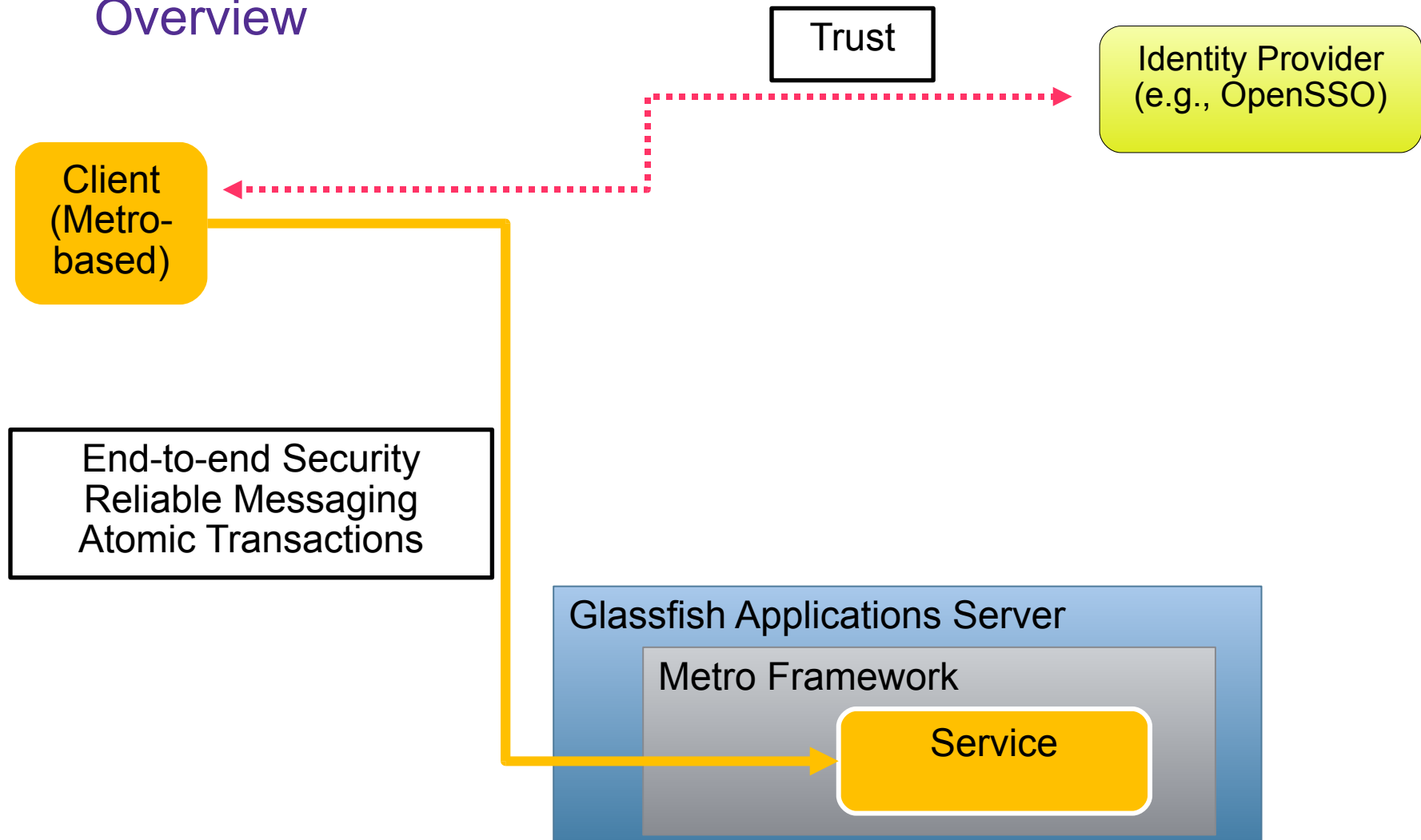
Harold Carr

Metro Architect, Sun Microsystems, Inc.

Learn how to use Metro-based web services and clients with Amazon, MS Office, OpenSSO using NetBeans and GlassFish

goal

Metro Web Services Overview



Agenda

- Metro with GlassFish and NetBeans
- Search Amazon with a Metro-based client
- Control Amazon EC2 with a Metro-based client
- Use Metro-based clients and services with 3rd party Identity Providers (e.g., OpenSSO)
- Use Metro-based clients and services with Windows Azure .NET Services
- Use Metro's faster transports and encodings
 - FastInfoset and SOAP/TCP (interoperates with .NET WCF)
- Save MS Office documents to Metro-based service
- Upcoming Metro features

Metro with NetBean And GlassFish

demo

Agenda

- Metro with GlassFish and NetBeans
- Search Amazon with a Metro-based client
- Control Amazon EC2 with a Metro-based client
- Use Metro-based clients and services with 3rd party Identity Providers (e.g., OpenSSO)
- Use Metro-based clients and services with Windows Azure .NET Services
- Use Metro's faster transports and encodings
 - FastInfoset and SOAP/TCP (interoperates with .NET WCF)
- Save MS Office documents to Metro-based service
- Upcoming Metro features

http://aws.amazon.com/

wsimport

http://ecs.amazonaws.com/AWSECommerceService/AWSECommerceService.wsdl

```
<service name="AWSECommerceService">
  <port name="AWSECommerceServicePort"
        binding="tns:AWSECommerceServiceBinding">
    <soap:address
      location="https://ecs.amazonaws.com/onca/soap?
                Service=AWSECommerceService"/>
  </port>
</service>
```

```
AWSECommerceService service = new AWSECommerceService();
AWSECommerceServicePortType port =
    service.getAWSECommerceServicePort();
```

```
<xs:complexType name="ItemSearchRequest">
  <xs:sequence>
    <xs:element name="Keywords" type="xs:string"
      minOccurs="0"/>
    <xs:element name="SearchIndex" type="xs:string"
      minOccurs="0"/>
    . . .
  </xs:sequence>
</xs:complexType>
```

```
ItemSearchRequest request = new ItemSearchRequest();
request.setSearchIndex("Books");
request.setKeywords("Web Services");
```

```
<xs:element name="ItemSearch">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="MarketplaceDomain"
        type="xs:string" minOccurs="0"/>
      <xs:element name="AWSAccessKeyId" type="xs:string"
        minOccurs="0"/>
      <xs:element name="SubscriptionId" type="xs:string"
        minOccurs="0"/>
      <xs:element name="AssociateTag" type="xs:string"
        minOccurs="0"/>
      <xs:element name="XMLEscaping" type="xs:string"
        minOccurs="0"/>
      <xs:element name="Validate" type="xs:string"
        minOccurs="0"/>
      <xs:element name="Shared"
        type="tns:ItemSearchRequest" minOccurs="0"/>
      <xs:element name="Request" type="tns:ItemSearchRequest"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
ItemSearch search = new ItemSearch();
search.getRequest().add(request);
search.setAWSAccessKeyId(av[0]);
```

```
<portType name="AWSECommerceServicePortType">
  <operation name="ItemSearch">
    <input message="tns:ItemSearchRequestMsg"/>
    <output message="tns:ItemSearchResponseMsg"/>
  </operation>
  ...
</portType>
```

```
Holder<OperationRequest> operationRequest = null;
Holder<List<Items>> items = new Holder<List<Items>>();
```

```
port.itemSearch(search.getMarketplaceDomain(),
                search.getAWSAccessKeyId(),
                search.getSubscriptionId(),
                search.getAssociateTag(),
                search.getXMLEscaping(),
                search.getValidate(),
                search.getShared(),
                search.getRequest(),
                operationRequest,
                Items);
```

```
<xs:element name="Items">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tns:Item" minOccurs="0" maxOccurs="unbounded"/>
      ...
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="Item">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SalesRank" type="xs:string" minOccurs="0"/>
      <xs:element name="SmallImage" type="tns:Image" minOccurs="0"/>
      <xs:element ref="tns:ItemAttributes" minOccurs="0"/>
      ...
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="ItemAttributes">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Title" type="xs:string" minOccurs="0"/>
      ...
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
Items result = items.value.get(0);
List<Item> item_list = result.getItem();
for (Item item : item_list) {
    System.out.println(
        item.getItemAttributes().getTitle());
}
```

```
request.setSearchIndex("Books");  
request.setKeywords("Web Services");
```

Java Web Services: Up and Running

RESTful Web Services

SOA Using Java(TM) Web Services

Web Services: Principles and Technology

Microsoft .NET XML Web Services Step by Step (Step By Step (Microsoft))

Web Service Contract Design and Versioning for SOA (Prentice Hall Service-Oriented Computing Series from Thomas Erl)

Sams Teach Yourself Web Services in 24 Hours

Web Services Essentials (O'Reilly XML)

Programming Amazon Web Services: S3, EC2, SQS, FPS, and SimpleDB

Expert Service-Oriented Architecture in C# 2005, Second Edition

➔ <http://www.amazon.com/Java-Web-Services-Up-Running/dp/059652112X>

Agenda

- Metro with GlassFish and NetBeans
- Search Amazon with a Metro-based client
- Control Amazon EC2 with a Metro-based client
- Use Metro-based clients and services with 3rd party Identity Providers (e.g., OpenSSO)
- Use Metro-based clients and services with Windows Azure .NET Services
- Use Metro's faster transports and encodings
 - FastInfoset and SOAP/TCP (interoperates with .NET WCF)
- Save MS Office documents to Metro-based service
- Upcoming Metro features

Amazon EC2 with Metro

- <http://aws.amazon.com/ec2/>
- WSDL: <http://ec2.amazonaws.com/doc/2009-04-04/AmazonEC2.wsdl>
- SOAP requests sent over HTTPS
- Uses WS-Security: requires msgs to be hashed & signed for integrity & non-repudiation.
 - Uses BinarySecurityToken profile: X.509 certificate & RSA public key
- But security NOT specified in WSDL via WS-Policy
- Therefore requires Metro client-side security configuration
 - Kohsuke's ease-of-use layer: <https://jax-ws-commons.dev.java.net/ec2/>

Using EC2 common library

```
// Create a proxy that talks to the server.  
// Requires your AWS private key and X509 certificate.  
// Resulting port object is multi-thread safe
```

```
File privateKey = ...;  
File certificate = ...;  
AmazonEC2PortType p =  
    EC2.connect(privateKey, certificate);
```

```
// make an API call - for example:  
p.describeImages(  
    new DescribeImagesType().withOwnersSet(  
        new DescribeImagesOwnersType().withItem(  
            new DescribeImagesOwnerType()  
                .withOwner("amazon")  
            )))
```

More info on using EC2 commons with Metro

- > BOF-5261
- > Web Services in Practice
- > Bhakti Mehta and Fabian Ritzmann
 - Sun Microsystems, Inc
 - Metro engineers
- > Tuesday, June 02
- > 9:30 PM - 10:20 PM
- > Esplanade 304-306
- > Also covers
 - VirtualBox, SalesForces.com, eBay

Agenda

- Metro with GlassFish and NetBeans
- Search Amazon with a Metro-based client
- Control Amazon EC2 with a Metro-based client
- Use Metro-based clients and services with 3rd party Identity Providers (e.g., OpenSSO)
- Use Metro-based clients and services with Windows Azure .NET Services
- Use Metro's faster transports and encodings
 - FastInfoset and SOAP/TCP (interoperates with .NET WCF)
- Save MS Office documents to Metro-based service
- Upcoming Metro features

Using an STS

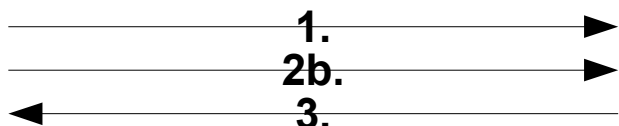
(e.g. OpenSSO)



Client
(Metro)



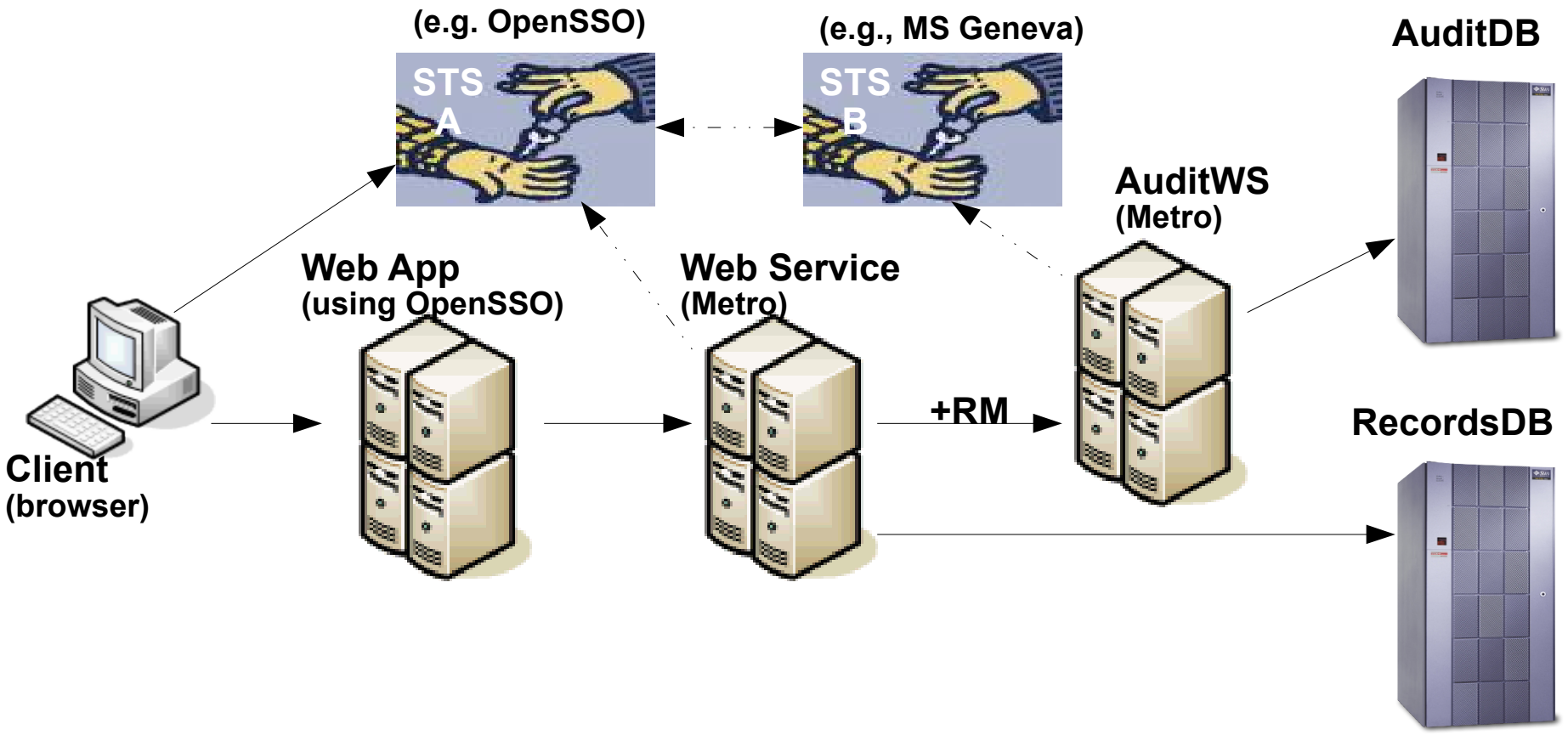
2a.



Web Service
(Metro)

1. HTTPS/MEX to get web service WSDL.
WSDL indicates SAML token required from STS
2. User A (permission to SOME data) logs in and calls WS operation.
 - 2a. HTTPS/MEX and HTTP/SAML STS interaction.
 - 2b. Pass token w/Attribute to web service
3. Use token attributes to determine user role

Federated Trust

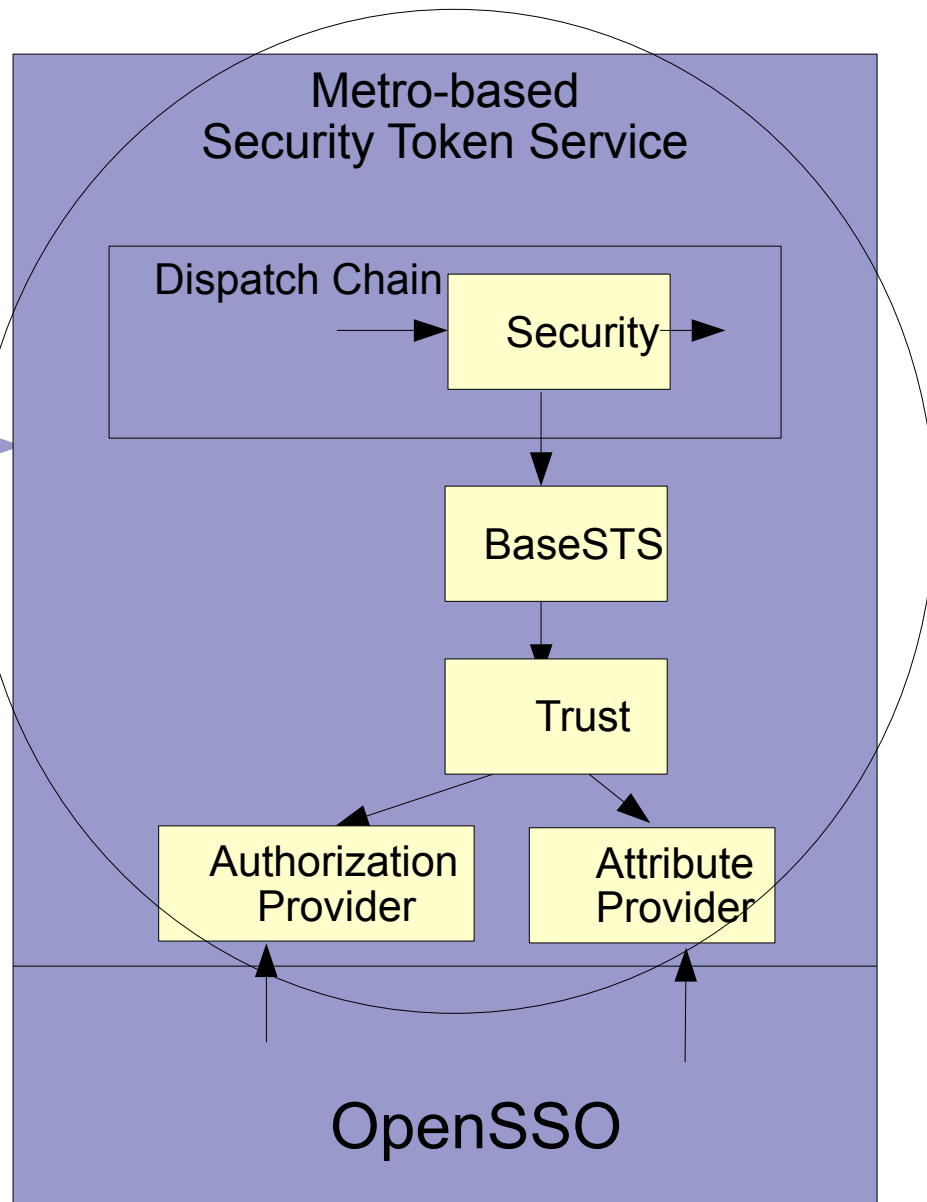


Secret: OpenSSO: “Metro Inside”

MS-based
client

WS-Trust

Metro provides foundation for OpenSSO to act as an Identity Provider for clients that only speak WS-Trust (e.g., MS: .NET/WCF)



More info on using OpenSSO (with Metro)

> BOF-5275

- Using and Participating in the OpenSSO Project
- Tuesday, June 02; 9:30 - 10:20 PM; Esplanade 300

> TS-5295

- Designing and Building Security into REST Applications
- Wednesday, June 03; 2:50 - 3:50 PM; Esplanade 307-310

> LAB-6727

- Web Application Security with OpenSSO: From Simple Log-In to Single Sign-On to Federation
- Thursday, June 04; 1:30 - 3:10 PM; Hall E 130-131

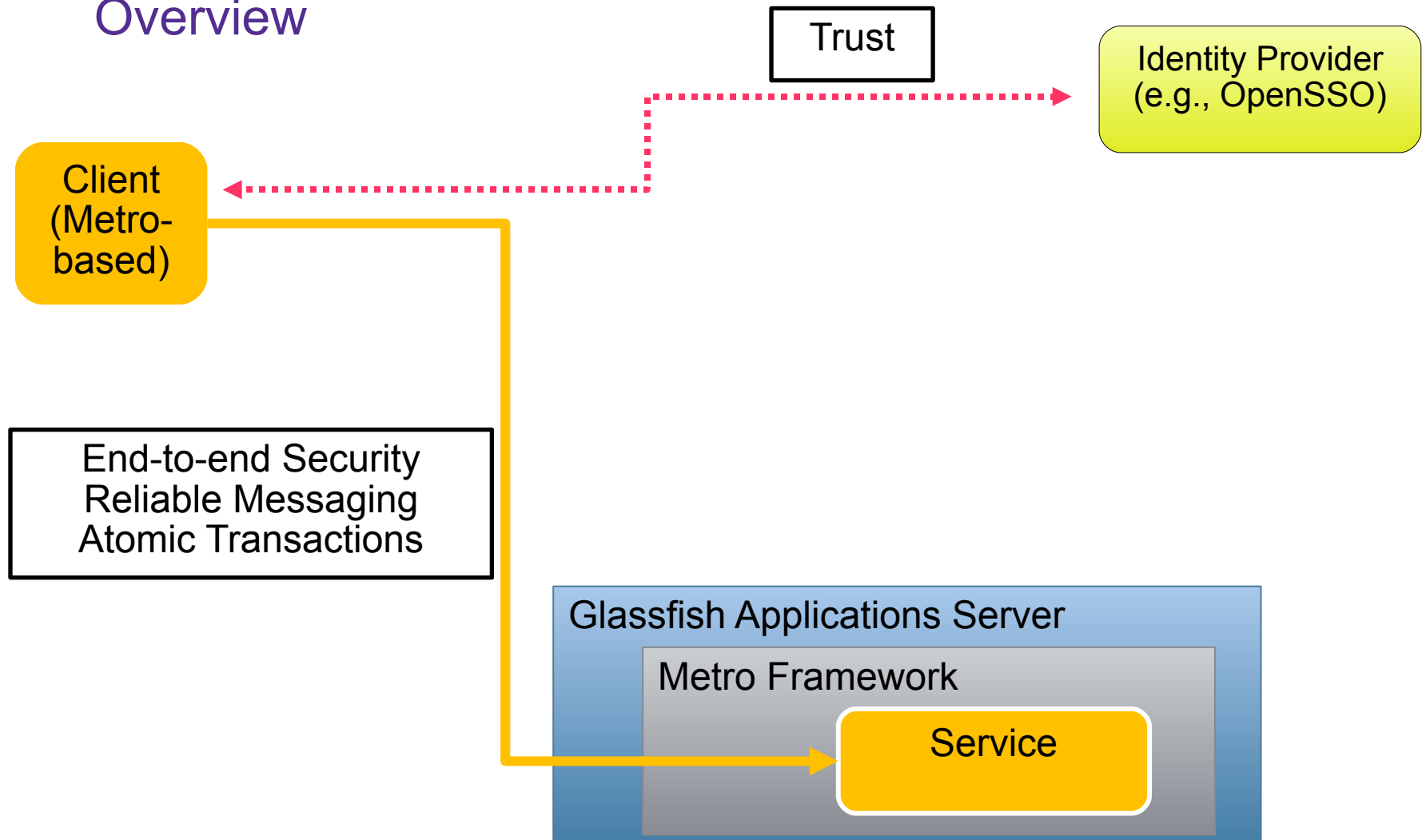
> Metro Users Guide

- https://metro.dev.java.net/guide/Example_Applications.html#ahiey

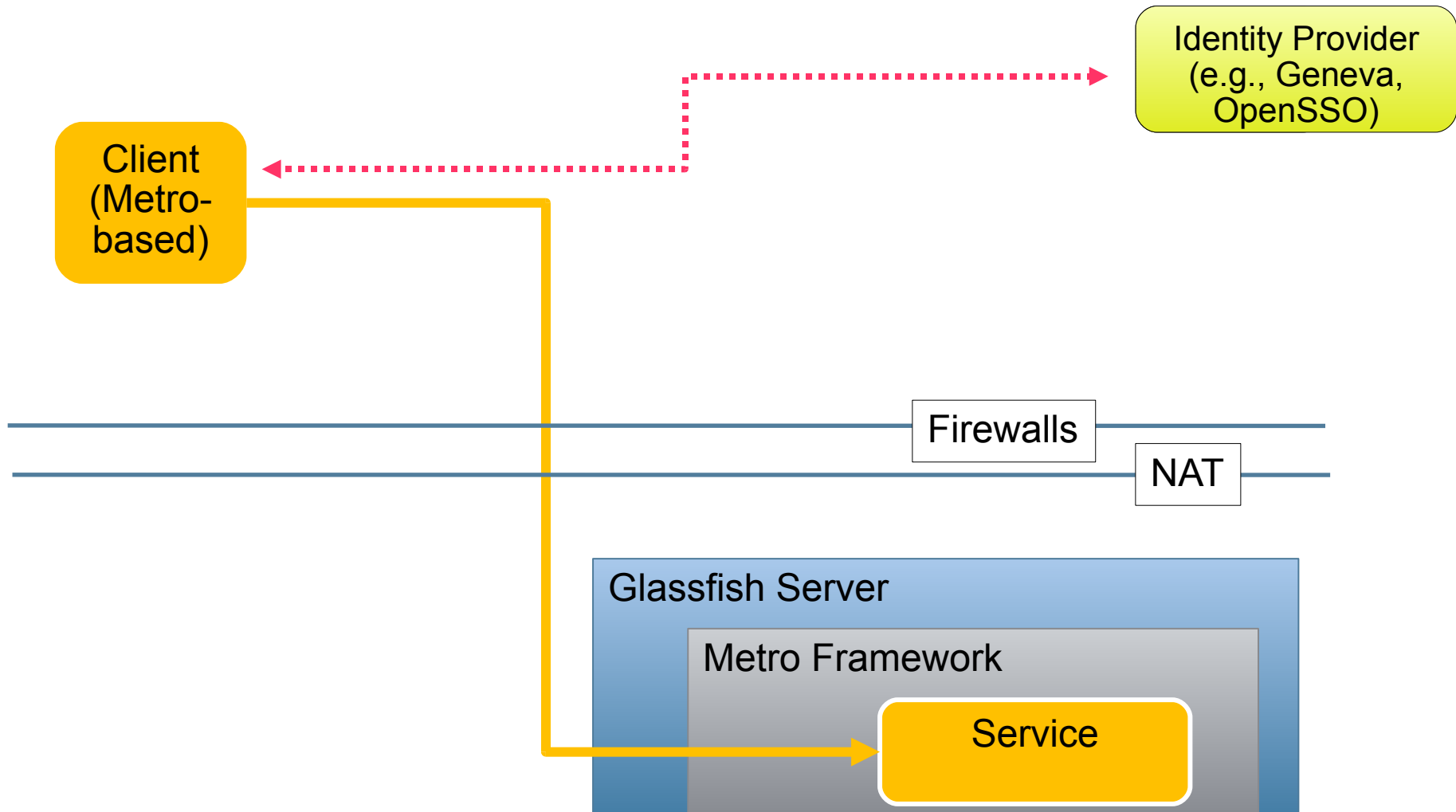
Agenda

- Metro with GlassFish and NetBeans
- Search Amazon with a Metro-based client
- Control Amazon EC2 with a Metro-based client
- Use Metro-based clients and services with 3rd party Identity Providers (e.g., OpenSSO)
- Use Metro-based clients and services with Windows Azure .NET Services
- Use Metro's faster transports and encodings
 - FastInfoset and SOAP/TCP (interoperates with .NET WCF)
- Save MS Office documents to Metro-based service
- Upcoming Metro features

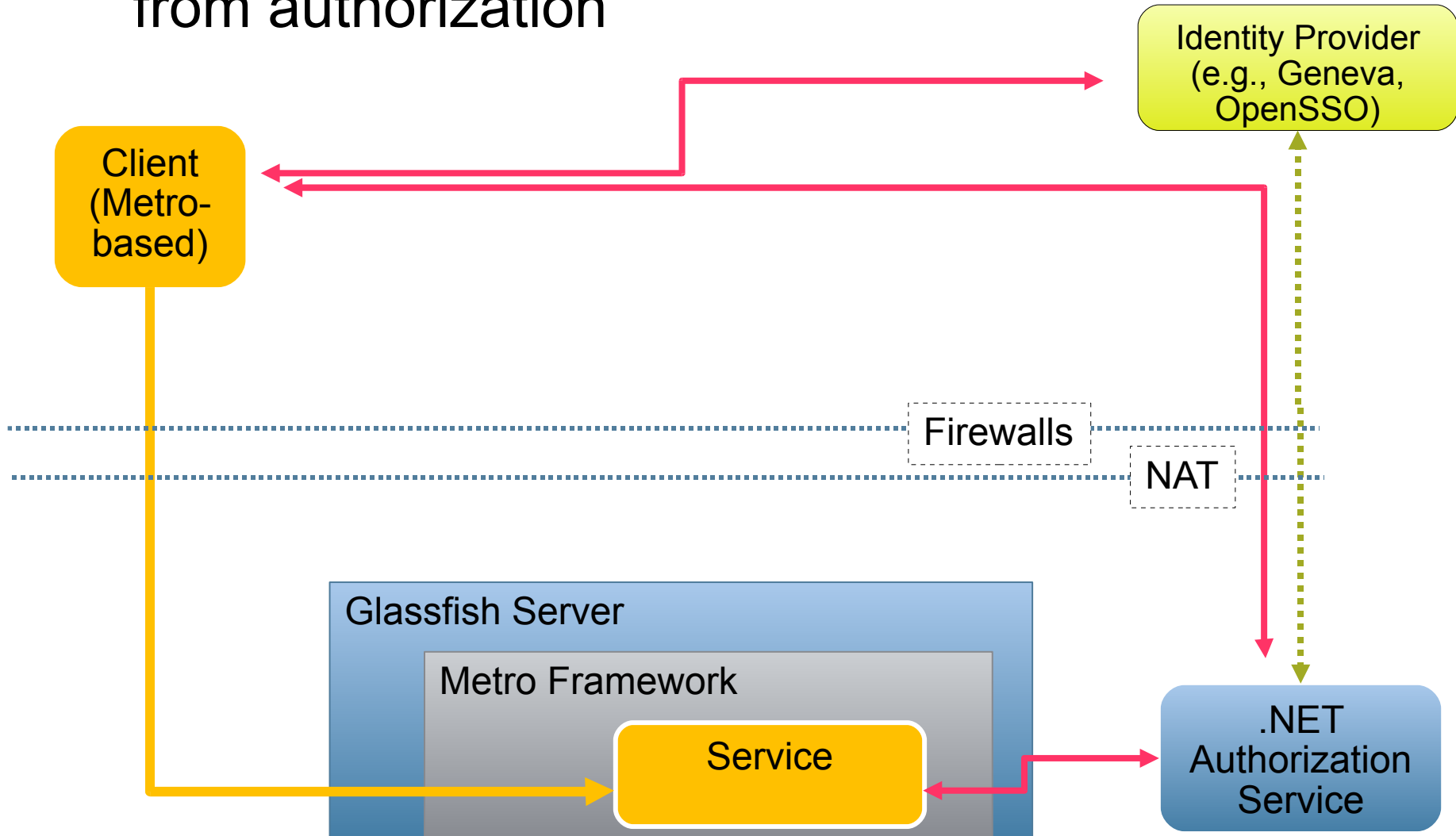
Metro Web Services Overview



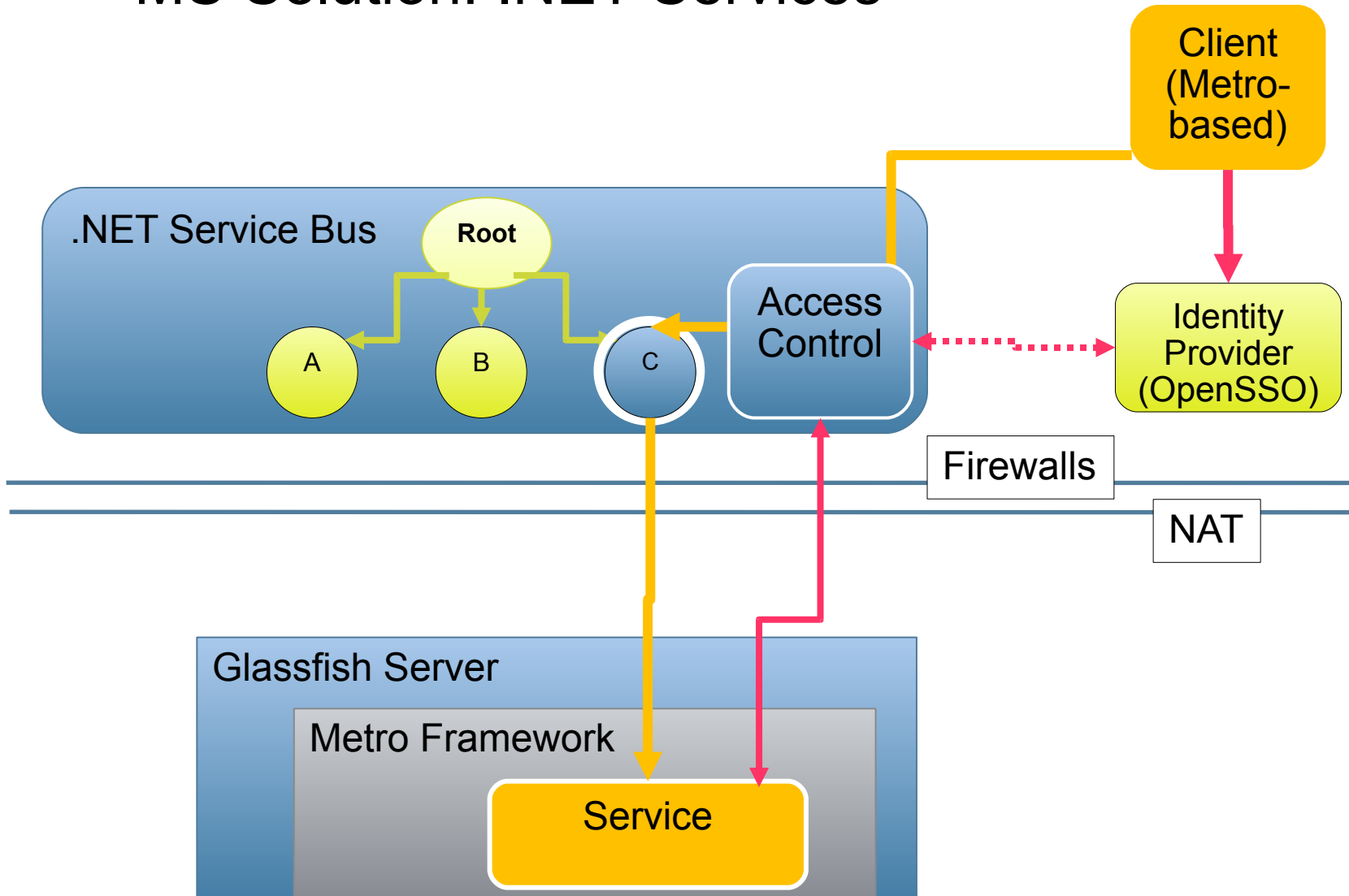
Problem: getting thru *all* the network



Problem: separating authentication from authorization



MS Solution: .NET Services



More info on using Metro with Windows Azure .NET Services

- TS-4617
- Using Java™ Technology in the Windows Azure Cloud via the Metro Web Services Stack
- Harold Carr
 - Sun Microsystems, Inc.
- Clemens Vasters
 - Microsoft
- Wednesday, June 03
- 11:05 AM - 12:05 PM
- Esplanade 300

Agenda

- Metro with GlassFish and NetBeans
- Search Amazon with a Metro-based client
- Control Amazon EC2 with a Metro-based client
- Use Metro-based clients and services with 3rd party Identity Providers (e.g., OpenSSO)
- Use Metro-based clients and services with Windows Azure .NET Services
- Use Metro's faster transports and encodings
 - FastInfoset and SOAP/TCP (interoperates with .NET WCF)
- Save MS Office documents to Metro-based service
- Upcoming Metro features

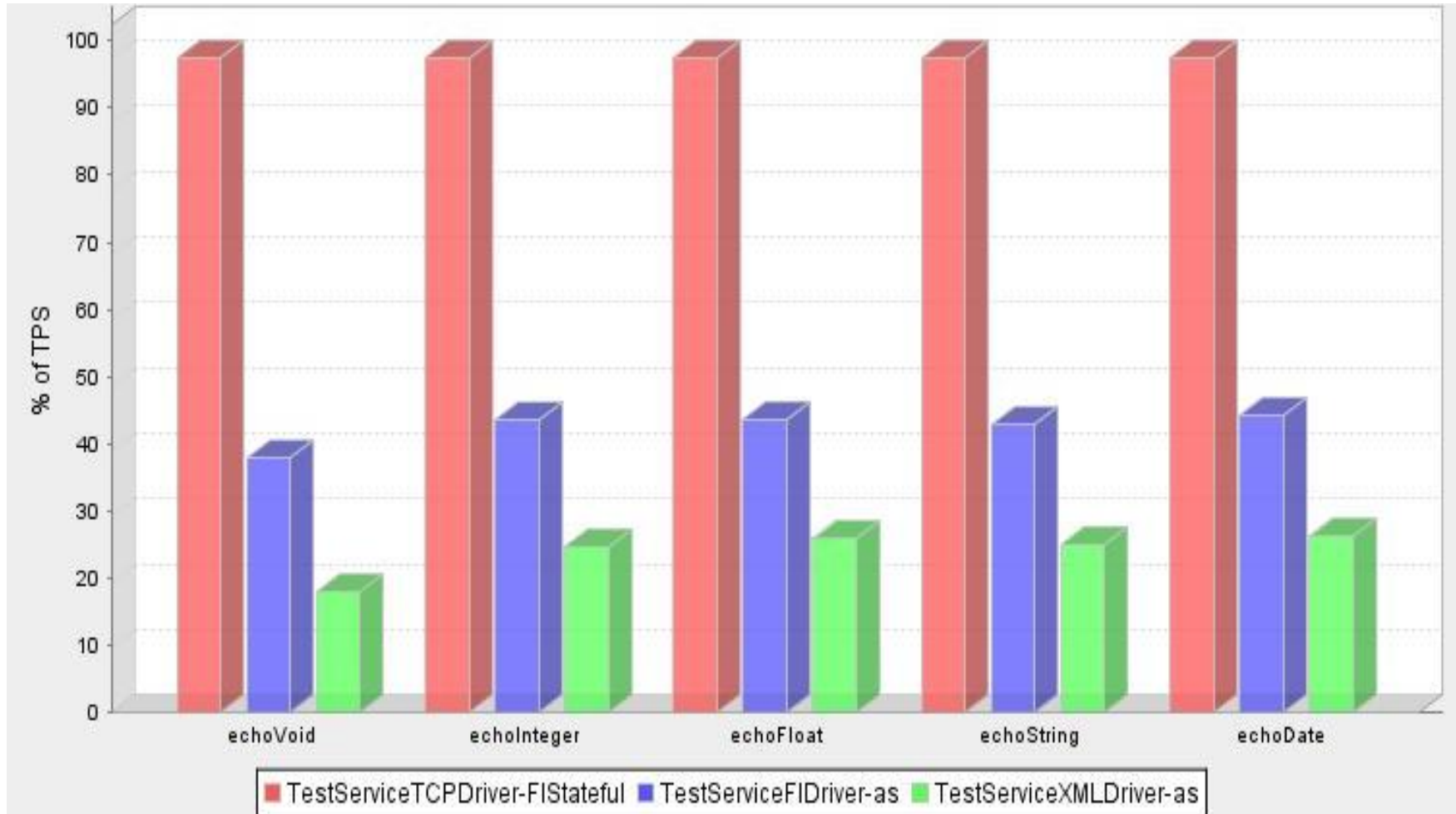
Metro SOAP/TCP and FastInfoset

Smaller and faster

- Fast Infoset message encoding
 - ITU-T and ISO/IEC standard
 - <http://www.itu.int/rec/T-REC-X.891/>
 - encoding of XML Information Set
 - more compact than text, MTOM and .NET Binary
- SOAP/TCP transport
 - open specification : protocol for WS messages over TCP
 - <http://java.sun.com/webservices/reference/apis-docs/>
 - works with message security and transport security
 - stateful connections + FastInfoset vocabulary reuse
 - even better performance when used together
- Built into Metro
- Also available for .NET WCF interoperability with Metro

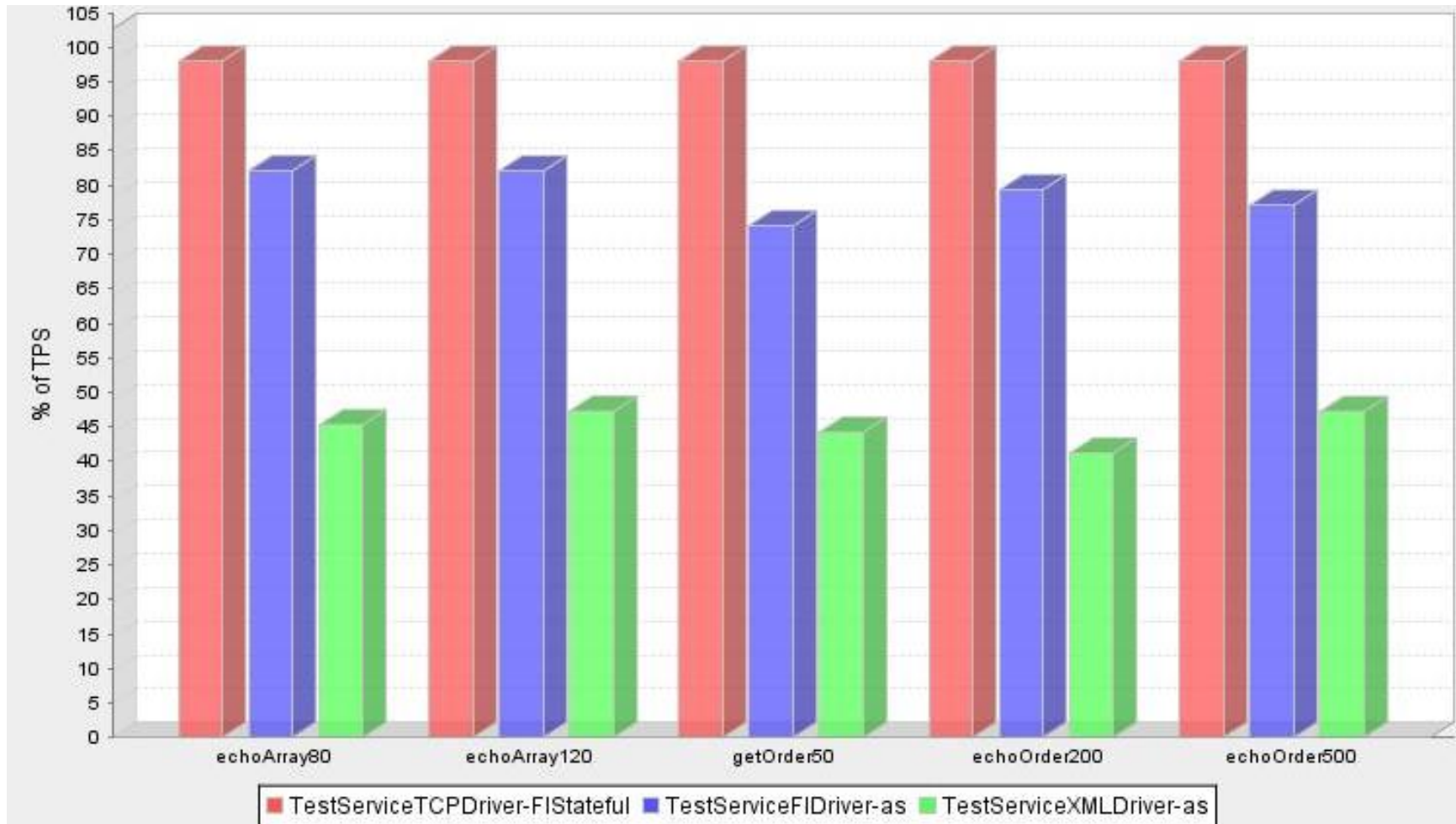
FastInfoset and SOAP/TCP

Small messages



FastInfoset and SOAP/TCP

Large messages



More info on SOAP/TCP & FastInfoset

- <https://metro.dev.java.net/guide/>
 - Section 10 covers SOAP/TCP in Metro
- <https://metro.dev.java.net/guide/FastInfoset.html>
 - How to use FastInfoset in Metro
- http://blogs.sun.com/oleksiys/entry/soap_tcp_makes_web_services
 - Oleksiy's blog on performance tests
- <http://www.itu.int/rec/T-REC-X.891/>
 - FastInfoset standard
- <http://java.sun.com/webservices/reference/apis-docs/>
 - SOAP/TCP open specification
- <http://www.noemax.com/products/wcfx/features.html>
 - Interoperable versions for .NET WCF

Agenda

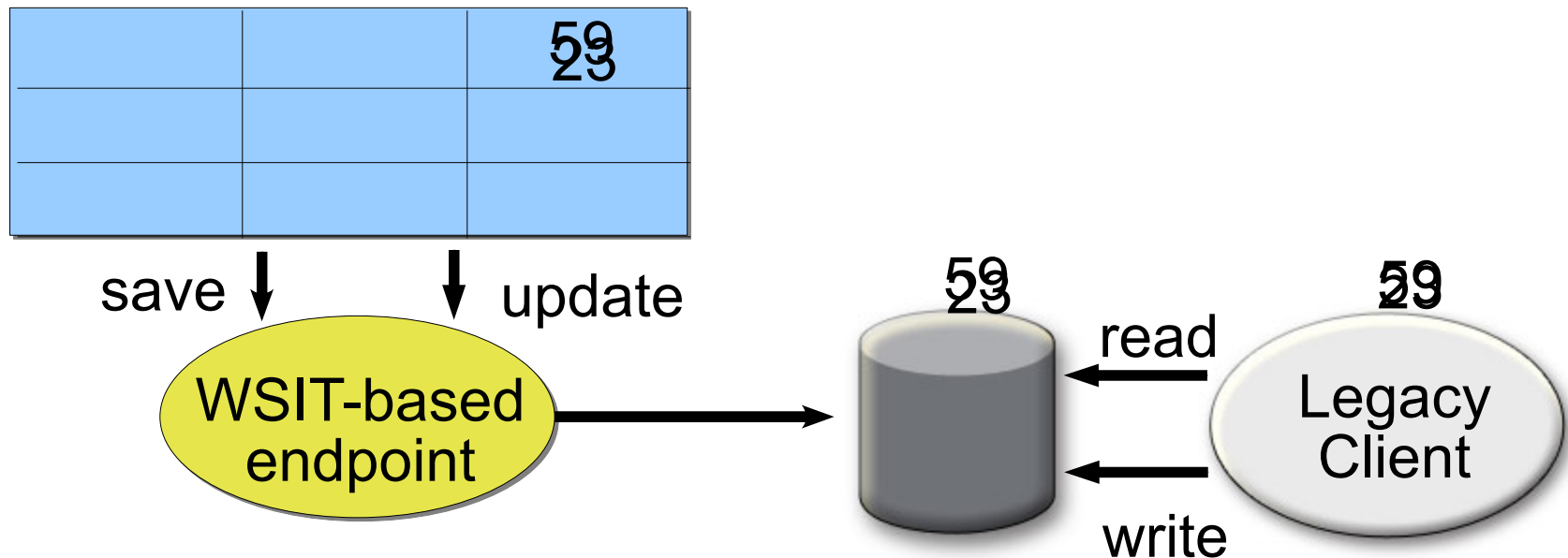
- Metro with GlassFish and NetBeans
- Search Amazon with a Metro-based client
- Control Amazon EC2 with a Metro-based client
- Use Metro-based clients and services with 3rd party Identity Providers (e.g., OpenSSO)
- Use Metro-based clients and services with Windows Azure .NET Services
- Use Metro's faster transports and encodings
 - FastInfoset and SOAP/TCP (interoperates with .NET WCF)
- Save MS Office documents to Metro-based service
- Upcoming Metro features

MS Office + Metro-based services

- Write .NET Windows Communication Foundation client that sends and/or receives data from a Metro-based web service
- Embed WCF client in Office as “macro”
- Add buttons for “save” and “update”

Save MS Office documents to Metro-based service

- Excel 2007 on Vista; Spreadsheet has Save/Update buttons
- A WSIT-enabled Web service endpoint that communicates with back-end system
- Legacy client using other technology to access DB



More info on MS Office + Metro

- http://blogs.sun.com/arungupta/entry/excel_using_wsit_metro_and
 - Source code
- <http://download.java.net/javaee5/screencasts/wsit-excel-demo/>
 - Screencast

Agenda

- Metro with GlassFish and NetBeans
- Search Amazon with a Metro-based client
- Control Amazon EC2 with a Metro-based client
- Use Metro-based clients and services with 3rd party Identity Providers (e.g., OpenSSO)
- Use Metro-based clients and services with Windows Azure .NET Services
- Use Metro's faster transports and encodings
 - FastInfoset and SOAP/TCP (interoperates with .NET WCF)
- Save MS Office documents to Metro-based service
- Upcoming Metro features

Upcoming Metro Features

- Centralized runtime policy management
- Durable Reliable Messaging
- Exchange credentials at runtime
- Issued Token Caching, Sharing, Renewal, Cancel
- Password-derived keys
- Interoperability with .NET 4.0
- Monitoring
- WS-MakeConnection,
- WS-I WS-RSP



Metro and GlassFish

Open Source and Enterprise Ready

- **GlassFish v3 Preview Available now!**

- Java EE 6 reference implementation
- Modular OSGi architecture – easy to develop & deploy
- Runs in-process and easy to extend
- Support for Ruby-on-Rails, Groovy and Grails, Python and Django

- **GlassFish v2 – Production Ready**

- Best price/performance open source App server with Clustering, High Availability, Load Balancing
- Secure, Reliable, Transactional, .NET-interop Web svcs
- Support for Ajax and Comet

- **GlassFish ESB**

- SOA and Business Integration platform

- **GlassFish Communications App Server**

- SIP servlet technology for converged services

glassfish.org

- **24x7 Enterprise and Mission Critical Support**

- sun.com/appserver

- **Tools Integration**

- NetBeans and Eclipse

- **Pavilion booth numbers: 550, 566, 567**

- **Meet Java EE spec leads and experts at Ancillary Event & Booth**

metro.dev.java.net



Metro Web Services, NetBeans, GlassFish and OpenSSO in Action with Amazon WS, Azure, and Office

S305138

Harold Carr, Metro Architect, Sun Microsystems
weblogs.java.net/blog/haroldcarr