



# JavaOne™

[java.sun.com/javaone](http://java.sun.com/javaone)

## Interoperable Business Web Services Using Project Metro and .NET 3.5

Harold Carr, Metro Lead Architect, Sun Microsystems  
Kevin Wittkopf, Senior Solutions Architect, Microsoft

TS-6128

**Microsoft®**



Learn how to architect and build  
interoperable business web services using  
Project Metro and .NET 3.5

A large, light blue graphic consisting of a stylized arrow pointing to the right, followed by the word "GOAL" in a large, bold, sans-serif font. The arrow and text are semi-transparent and overlaid on a darker blue background.

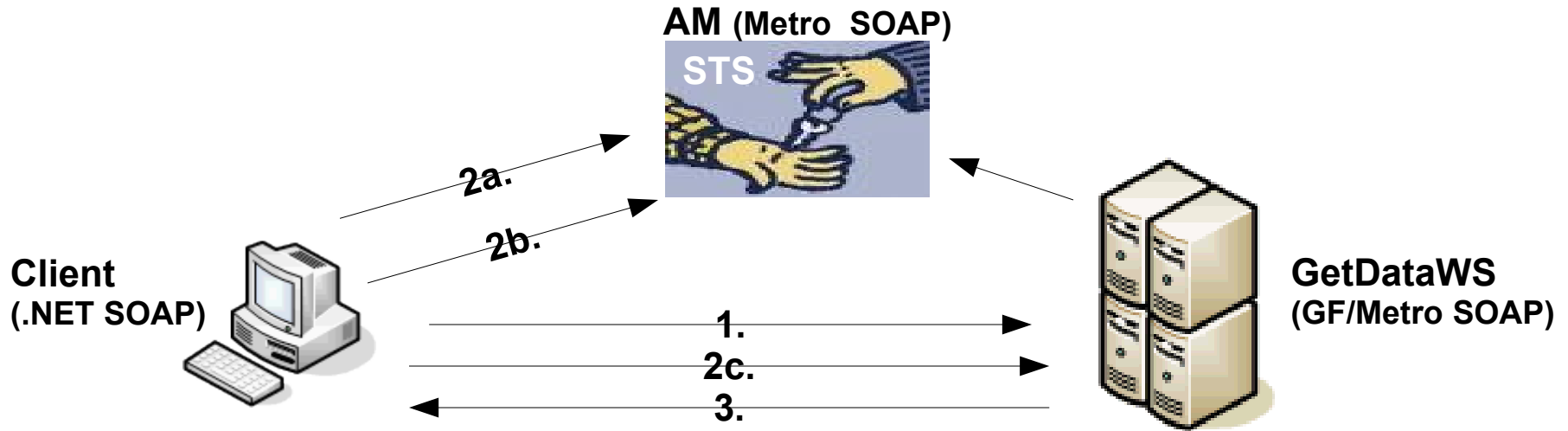
# Agenda

- Overview of Java™ environment/ .NET 3.5 Web Service Interoperability
- Demos
  - Token creation and validation
  - Token expiration
  - Identity and attribute extraction for Database search
  - Identity propagation thru multiple web apps & services
  - Brokered trust
- More info
- Q & A

# Java Environment / .NET 3.5 Web Service Interoperability

- .NET 3.5 framework
  - Windows Communication Foundation
- Java environment
  - Project Metro (aka JAX-WS RI + WSIT/Tango)
- GlassFish™ application server (Java Platform, Enterprise Edition ("Java EE platform"))
- Basic web services
  - WS-I: BP 1.1, BSP 1.0
- Enterprise web services
  - Oasis: WS-Security, WS-SecureConversation, WS-Trust, WS-SecurityPolicy, WS-ReliableMessaging, WS-AtomicTransactions, WS-Coordination
  - W3C: WS-Addressing, WS-Policy, WS-Transfer
  - WS-MetadataExchange

# Token Creation and Validation



1. HTTPS/MEX to get GetDataWS WSDL
- 1a. GetDataWS has WSDL that indicates SAML token required from STS
2. getTime called.
- 2a. HTTPS/MEX to get STS WSDL.
- 2b: HTTP/SAML security to do STS operation to get Token.
- 2c: Pass token w/Attribute inserted directly in token to GetDataWS
3. GetDataWS returns result when valid token received.

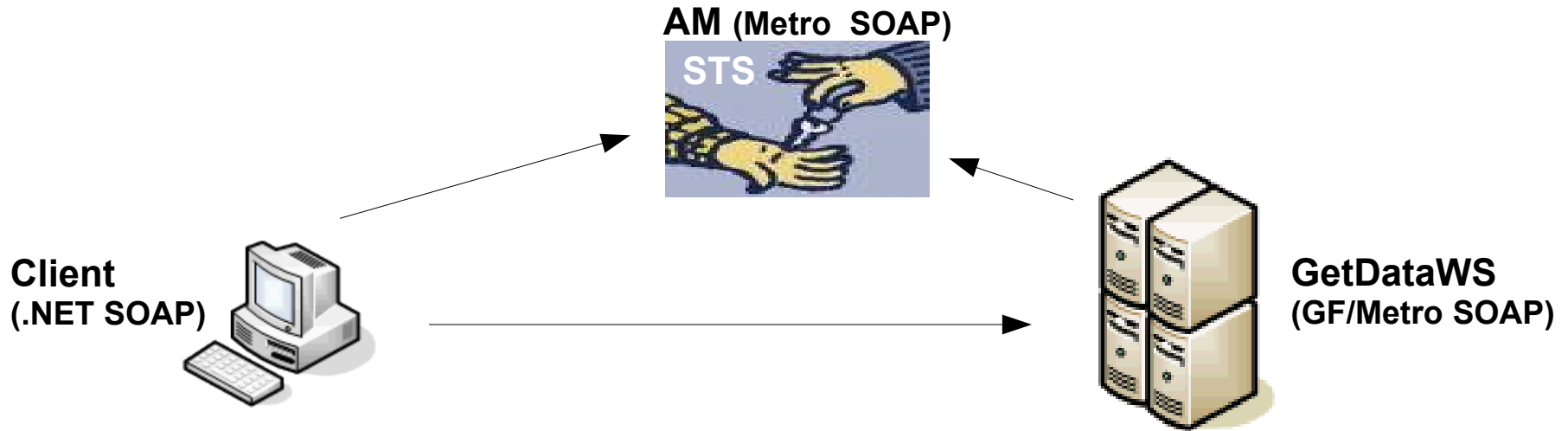
# Protocols used in Token Creation scenario

- **WS-Transfer/WS-Metadata Exchange**
  - Used to obtain service and STS WSDLs
- **WS-Trust**
  - Used by client to obtain security token from STS
- **WS-Security**
  - Used to sign/encrypt messages between client and service
- **STS = Secure Token Service**
  - Sun Java System Access Manager in this example
  - Uses SAML tokens
  - More on STS and SAML in subsequent slides

# Security Token Creation and Validation

DEMO

# Token Expiration



1. Same setup / interaction as previous slide.
1. Change token expiration on STS to 5 seconds.
2. After getting token from STS have client sleep 10 seconds then call getTime. Should receive “invalid token” fault
3. Change token expiration on STS to 15 seconds.
4. After getting token from STS have client sleep 10 seconds then call getTime. Should now receive valid result.



# STS used in SAML Token Creation scenario

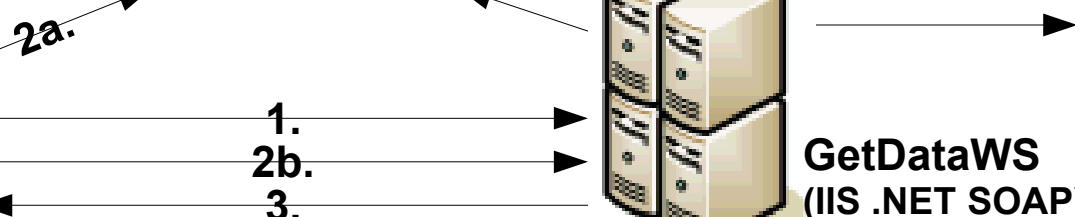
- STS == Secure Token Service
  - STS in this example is Java System Access Manager
- SAML == Security Assertion Markup Language
- SAML tokens generated by STS specify details ('claims') about client to server
  - Tokens have predefined elements & attributes
  - Token can include user-defined claims
- Token includes 'expires' element
  - STS (in this example) sets 'expires' to 15 seconds

# Security Token Expiration

DEMO

# Identity and attribute extraction for Database Search

Active Directory (.NET SOAP)



1. HTTPS/MEX to get GetDataWS WSDL.
- 1a. GetDataWS has WSDL that indicates SAML token required from STS
2. User A (permission to SOME data) logs in and calls getData.
- 2a. HTTPS/MEX and HTTP/SAML STS interaction.
- 2b. Pass token w/Attribute inserted directly in token to GetDataWS
3. Use token to determine user role.  
Result should be a subset of data (e.g., 5 rows).

User B (permission to ALL data) logs in and does SAME query.  
Result should be all data (e.g., 10 rows).

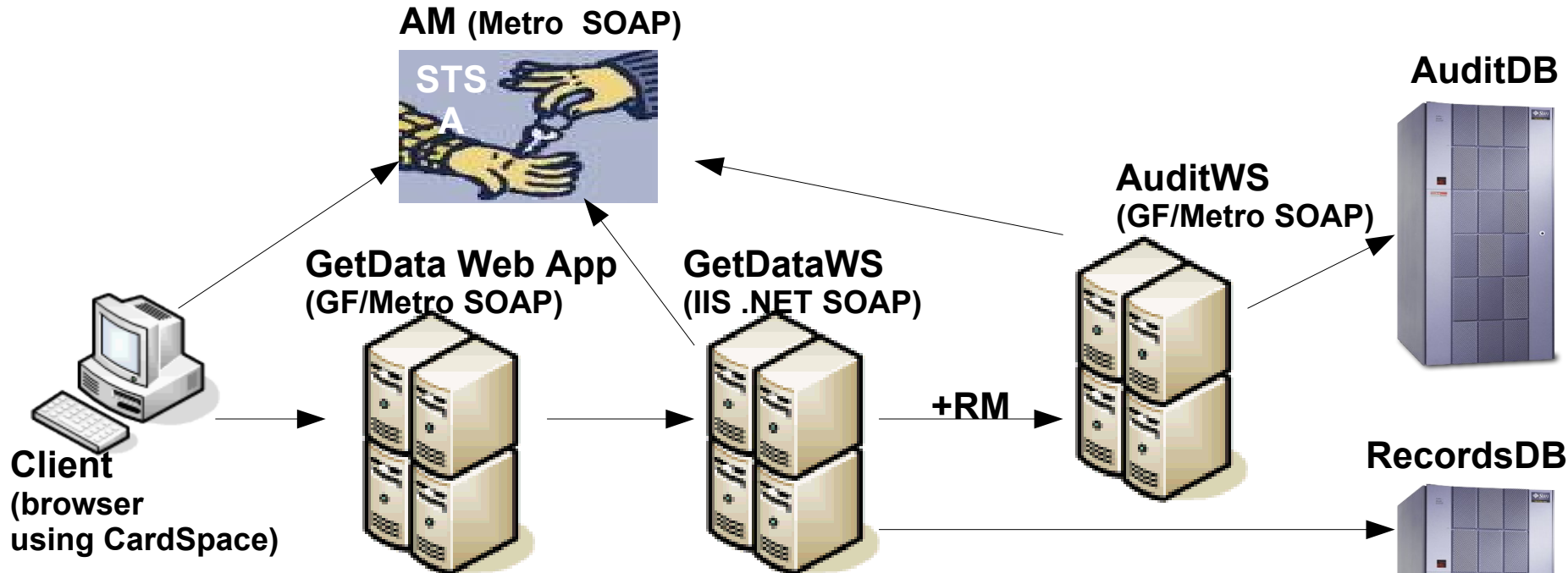
# STS used in DB search scenario

- STS in this example is backed by Active Directory (AD)
- User supplies credentials to authenticate to Active Directory (username/password, X.509, etc)
- STS issues SAML token with claims regarding user
  - Identity
  - STS inserts additional claim regarding the users ROLE
    - (as defined in AD)
- GetDataWS verifies SAML token issued by trusted STS
- Role extracted from SAML token
  - Used in DB access

# Identity for Role-based DB Access

DEMO

# Identity Propagation thru multiple web apps & services



1. Browser-based client authenticates via CardSpace + AM
2. Client does call on GetData Web Application.
3. GetData WA calls GetDataWS.getData.
- 3a. GetDataWS will get data from RecordsDB.
- 3a. GetDataWS will also call AuditWS.audit. Will use WS-RM.

Validate: record must be retrieved correctly and AuditDB verified.

Audit record should show User A, time, Application, GetDataWS and RecordsDB.

NOTE: GetDataWs and AuditWS also secured using initial client token.

# InfoCard

- **CardSpace – Microsoft’s identity metasytem**
  - Supports multiple identity systems
  - based on standards (e.g., WS-Security, WS-Trust, WS-MetadataExchange, WS-SecurityPolicy)
- **Users download cards from identity providers**
  - their bank/etc, or create their own self-issued cards
- **Cards used to convey any info from identity provider to relying party that makes sense to both of them**
- **CardSpace allows the user to select a card that provides identity and required claims to STS**
- **Java System Access Manager supports InfoCard using its own identity system**
- **SAML token returned by STS includes identity is propagated and verified by Metro and .NET based services**

# WS-ReliableMessaging

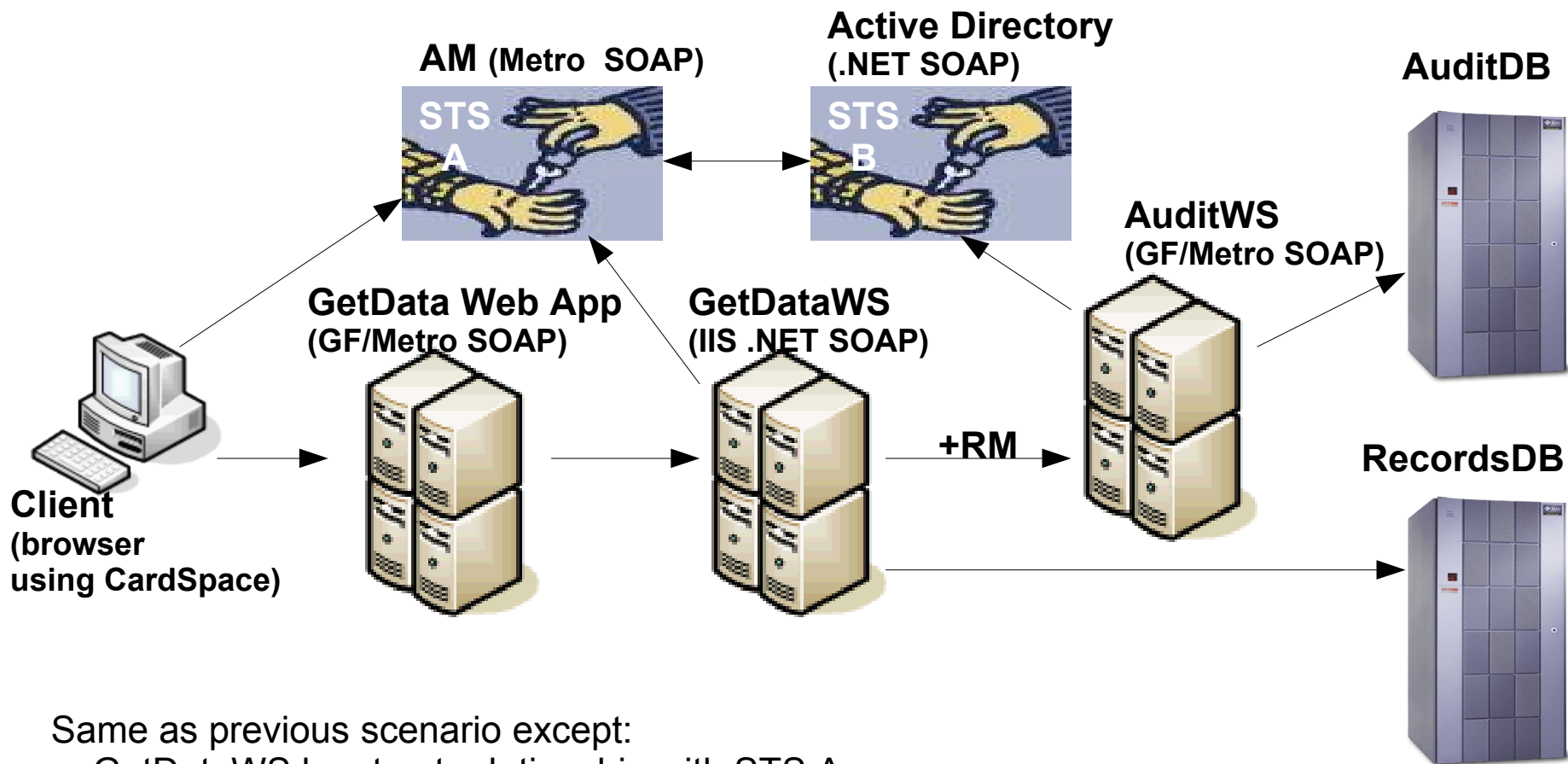
- Used between GetDataWS and AuditWS
- To ensure audit trail
- Ensures all messages sent are received



# Identity Propagation through multiple Web Applications and Services

DEMO

# Brokered Trust



Same as previous scenario except:  
 GetDataWS has trust relationship with STS A,  
 AuditWS has trust relationship with STS B.  
 STS A and B trust each other.

# Brokered Trust

- User supplies credentials to authenticate to STS A (Java System Access Manager)
- Identity is propagated through multiple web apps/services
- AuditWS does not know/trust STS A (Java System Access Manager)
- AuditWS trusts STS B (AD)
- STS B has a trust relationship with STS A (via WS-Trust)
- STS B can use STS A to validate identity

# Identity Propagation through multiple Web Applications and Services + Brokered Trust

DEMO

# What we did not cover

- **WS-AtomicTransactions & WS-Coordination**
  - All operations in TX boundary succeed or rollback
  - Same TX as in EJBs and COM+/Serviced Components
  - Now supported in web services
- **WS-SecureConversation**
  - Optimization when multiple secure messages exchanged

# Summary

## ➤ Web Service Interoperability

- Java technology-based web services using Metro and GlassFish application server
- .NET 3.5 web services using Windows Communication Foundation

## ➤ Identity

- Java System Access Manager
- Microsoft Active Directory
- WS-Trust
- SAML
- InfoCard

## ➤ Security

- SAML, WS-Security, WS-SecureConversation, WS-SecurityPolicy

# Related Sessions

## > TS-6373

- Next-Generation Web User Experience Interoperability with Java™ Platform, Enterprise Edition (Java EE Platform) Technology and Silverlight

## > TS-6658

- GlassFish Project Web Services Stack “Metro”: Easy to Use, Robust, and High-Performance

## > BOF-5590

- Java Technology for Web Services Secure Exchange: New WS-SX Standards in Action

## > LAB-3410

- Metro: Try Out Simple and Interoperable Web Services

# Relevant Blogs

- WCF blog:
  - [http://wcf.netfx3.com/blogs/wcf\\_team\\_bloggers/](http://wcf.netfx3.com/blogs/wcf_team_bloggers/)
- InfoCard blogs:
  - [http://netfx3.com/blogs/cardspace\\_blogs/](http://netfx3.com/blogs/cardspace_blogs/)
  - <http://blogs.sun.com/main/tags/infocard>
- GlassFish project and Metro blogs:
  - <http://blogs.sun.com/theaquarium/>
  - <http://feeds.feedburner.com/MetroBlogs>
- Java System Access Manager and OpenSSO blogs:
  - <http://planets.sun.com/OpenSSO/>
  - <http://developers.sun.com/identity/>



# For More Information

- GlassFish project, Metro, Java System Access Manager, OpenSSO
  - <http://glassfish.java.net/>
  - <https://metro.dev.java.net/>
  - [http://www.sun.com/software/products/access\\_mgr/index.jsp](http://www.sun.com/software/products/access_mgr/index.jsp)
  - <http://opensso.org/>
- Windows Communications Foundation & .NET 3.5:
  - <http://msdn2.microsoft.com/en-us/netframework/aa663324.aspx>
  - <http://netfx3.com/default.aspx>
- WCF Interoperability Guide & Plugfests:
  - <http://msdn2.microsoft.com/en-us/library/ms734776.aspx>
  - <http://msdn2.microsoft.com/en-us/webservices/aa740612.aspx>
- Windows CardSpace:
  - <http://msdn2.microsoft.com/en-us/netframework/aa663320.aspx>
- J+N (Java technology + .NET) Program:
  - <http://www.microsoft.com/windowsserver/jplusn/default.msp>

# THANK YOU



Harold Carr, Metro Lead Architect, Sun Microsystems  
Kevin Wittkopf, Senior Solutions Architect, Microsoft

TS-6128



# EXTRA SLIDES


**in case InfoCard demo doesn't work**

Windows CardSpace


Select a card to preview

To see or edit card data before you send it, select a card, and then click Preview. To create a new card, click Add a card and then click Add.


Your cards:




**My Default Web Card**



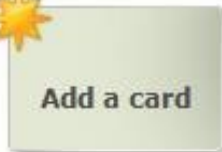
**My Detailed Web Card**



**My Personal Web Card**



**My Private Web Card**



Tasks

---

Duplicate card

Delete card

---

Add a card

Back up cards

Restore cards


Preferences

---


Delete all cards

---

Help



You created this card on 11/2/2006.



2008 JavaOne<sup>SM</sup> Conference | [java.sun.com/javaone](http://java.sun.com/javaone) | 28