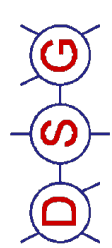


SECURE SERVICE INVOCATION IN A PEER-TO- PEER ENVIRONMENT USING JXTA-SOAP

MobiSec 2009, Turin June 3-5



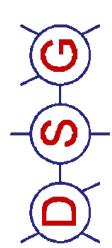
M. C. Laghi, M. Amoretti, G. Conte
**Distributed System Groups - Università degli
Studi di Parma**



Outline

2-21

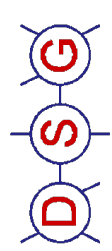
- Introduction
 - Jxta e jxta soap
 - P2psecurity issues
 - Tls vs message security
- Conclusions and future work
- Jxta pipes
- Sicurezza standard
- Sicurezza mobile
- Mikey
- conclusioni



Introduction

- P2P enables highly scalable decentralized applications based on resource sharing
- Service oriented peer to peer systems
- Services offered by peers are usually *consumable resources*, i.e. resources that cannot be acquired (by replication) once discovered, but may only be directly used upon contracting with their hosts.

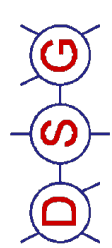
- Convergence of SOA and P2P
 - Requires standards for overlay network construction and maintenance
 - Message routing among peers
 - Service advertising, discovery and interaction
 - Security



Introduction

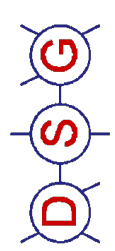
- A **lightweight middleware** is required for systems that include
 - high performance nodes
- but also
 - resource constrained devices

The goal is to allow a vast class of networked devices (PCs, notebooks, PDAs, smart-phones, but also different kind of sensors) to communicate and collaborate seamlessly in a decentralized fashion



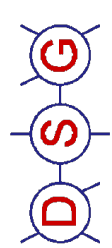
Peer-to-peer security

- Security attacks in p2p systems are:
 - Passive
 - The attacker just monitors the activity and maintains an inert state (eavesdropping, traffic analysis..)
 - Active
 - Communications are disrupted by the deletion, modification or insertion of data (spoofing, man in the middle, playback, local data alteration...)
- Security policies are adopted at the overlay peer-to-peer network level
 - Key management, authentication, admission control, authorization



Transport-level vs message-level security

- Transport-level security
 - Create a secure pipe between two web servers
 - Authentication when the pipe is created
 - Confidentiality and integrity while the message is on the pipe
 - The client and the server negotiate encryption algorithm and cryptographic keys
- Message level security
 - Intermediary services may be involved in the message path
 - End-to-end protection
 - WS-Security bridge gaps between different security



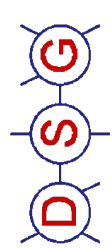
JXTA (by Sun Microsystems)

7-21

- A set of open peer-to-peer protocols
- A framework and a middleware (J2SE, J2ME, C/C++/C#)
- Supporting interactions among peers:
 - Discover each other
 - Self-organize into peer groups
 - Advertise and discover *network services*
 - Communicate with each other
 - Monitor each other



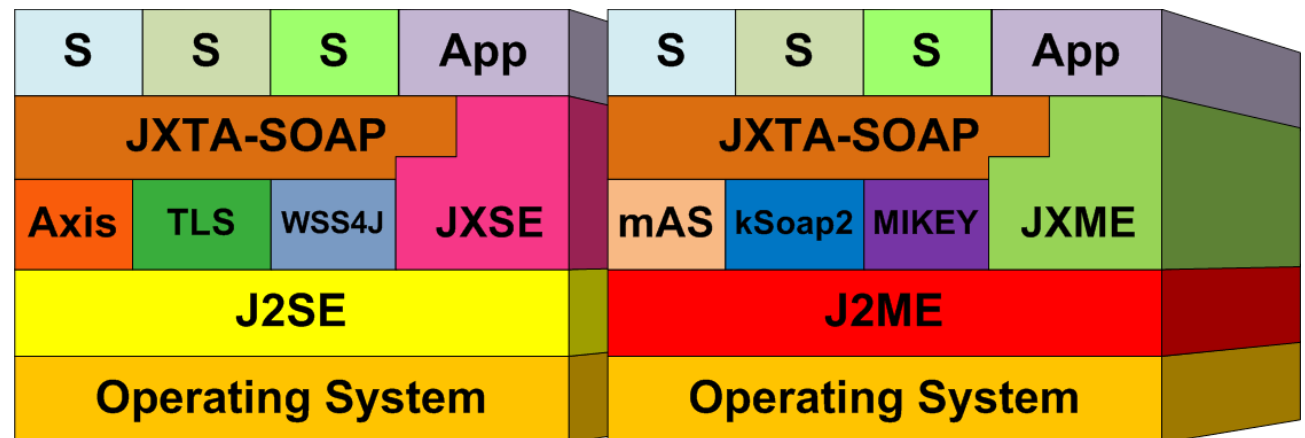
<https://jxta.dev.java.net>

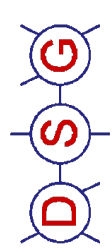


JXTA-SOAP

8-21

- JXTA-SOAP is a component which extends the JXTA middleware (Java version).
- Design goals:
 - Wrapping Web Services in JXTA services.
 - Using JXTA for Web Service discovery and SOAP message transport.
 - Allowing a vast class of networked devices to communicate and collaborate

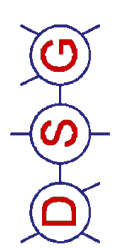




JXTA pipes

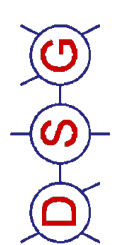
- JXTA peers use pipes to exchange messages and access resources
- JXTA messages are XML documents that
 - May contain any type of payload
 - Are the basic data exchange unit
- Pipes provide asynchronous, unidirectional and unreliable channel
- JXTA pipes have a unique ID and are published in the network (like the resources)
- Input and output pipes

JXTA-SOAP uses pipes for consumer-to service requests and service responses



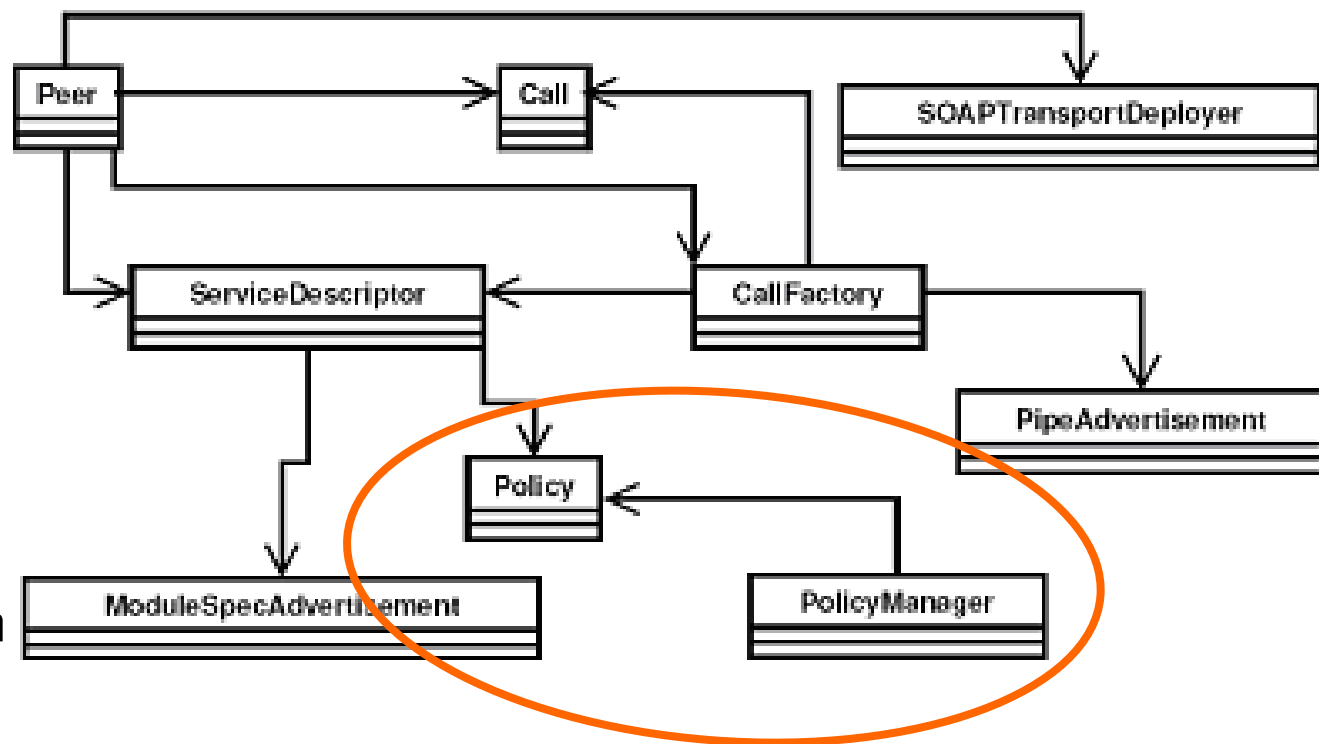
JXTA-SOAP security

- JXTA-SOAP supports secure service invocation by means of two orthogonal mechanisms:
 - *transport-level security* (TLS)
 - X509 certificates are exchanged through JXTA pipes
 - *message-level security* (based on WSS)
 - SOAP messages sent by service consumers contain security parameters (tokens) which are extracted by service providers to check for consumers' compliance with the security policy of the invoked service.



JXTA-SOAP security

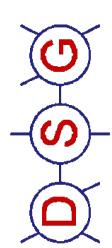
In order to deploy its services, a peer has to instantiate and configure the related *SOAPService* objects (one for each hosted service), and to advertise the service interfaces in the network. A service can



Associations among classes which are involved in the service invocation task, performed by a generic Peer

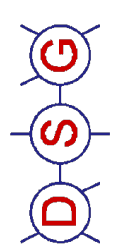
associated

be



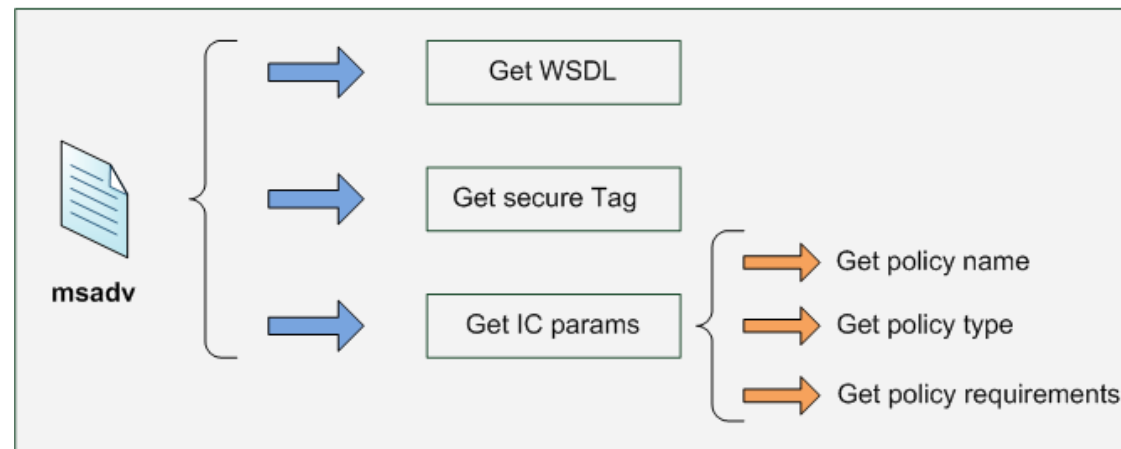
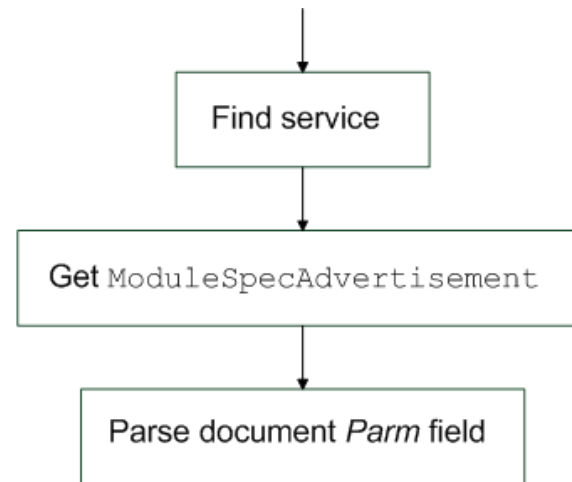
Secure service invocation

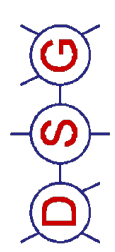
- Policy manager class allows to associate a service with a security policy
- Policy interface provides methods that are implemented by all policy classes
 - Allows to extract authentication parameters from client request messages
 - Authentication parameters are stored in a vector of **SOAPService** class, used to keep the list of authenticated peer
 - DefaultTLSPolicy is based on JXTA UnicastSecure a secure pipe
 - DefaultWSSPolicy uses WSS4J APIs to provide security headers and fill them with tokens



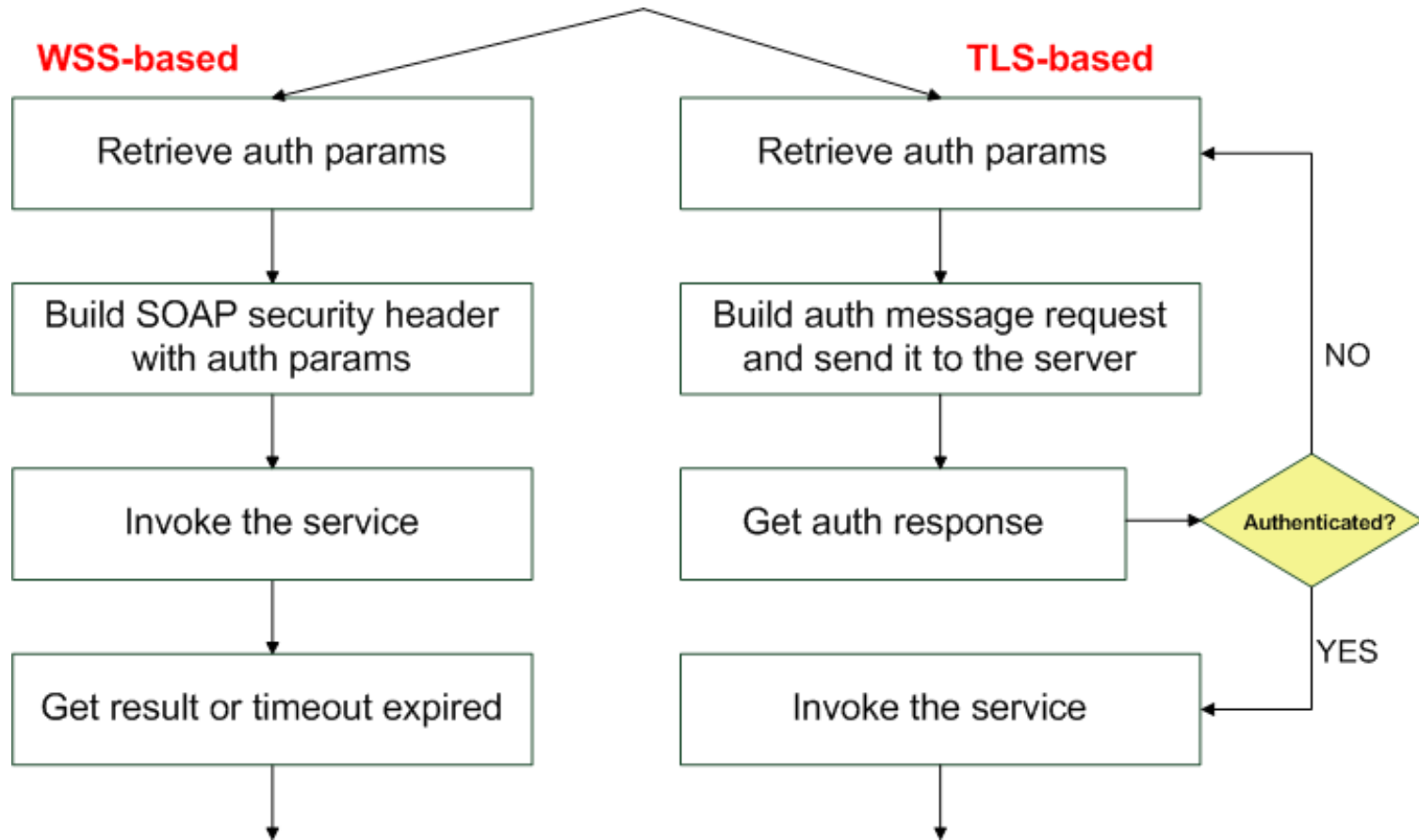
Secure service invocation

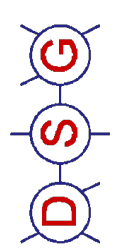
Policies are associated to services by means of two extensions of <Parm> field in ModuleSpecAdvertisement





Secure service invocation



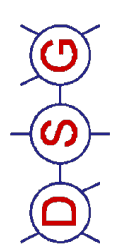


Default Transport-level security

- Messages between client and server are sent through a TLS channel that requires previous authentication
- Client sends *PeerAdvertisement* with the peer self-signed certificate created by JXTA
- Server updates the list of authenticated peers and associates a TTL with the authentication of peer
- PSE keystore is used for storing/retrieving certificates

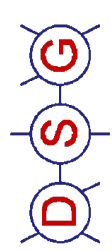
- Client and server have to authenticate to

PSEMembershipService



Default message-level security

- A security header is built with WS-Security classes and included in all messages and invocation requests
- During the authentication phase the client attaches the X.509 certificate and digitally signs the message body
- WSS-based policy does not use PSE keystore but
- Requires that both client and server generate a couple of private/public keys and use them to create a self-signed X.509 certificate whose integrity and privacy are granted by means of a password.



Mobile security

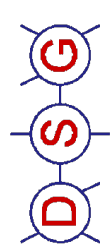
- PSE Membership classes not available for JXME
 - Need to import and modify them in order to authenticate a peer within a peergroup
- J2ME does not support TLS
 - Impossible to use JXTA secure pipes

Implementation of a new pipe:

JXTAUnicastCrypto

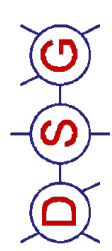
new security policy for connected device

configuration and personal profile



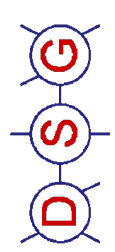
MIKEY

- Multimedia Internet Keying (MIKEY) to create key pair and parameters for encryption/decryption
- Encryption remains a resource-intensive task for resource constrained devices
- Mikey can be used in real-time and p2p applications
 - Developed to minimize latency when exchanging cryptographic keys between small interactive groups in heterogeneous networks
- Implementation with RSA-R algorithm

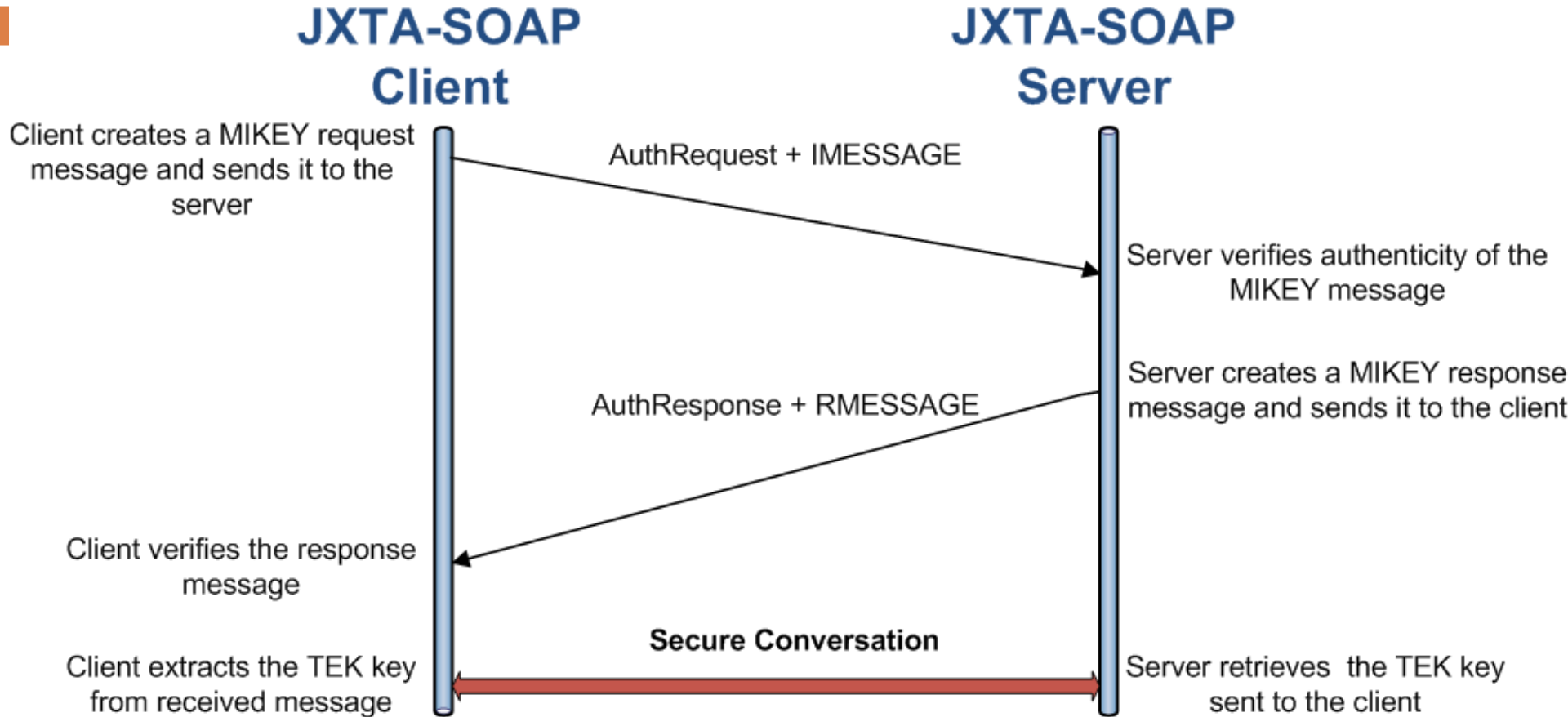


MIKEY

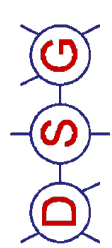
- Three methods supported:
 - Pre-shared secret (PSK)
 - Public key encryption (PKE)
 - Diffie hellmann key exchange (DH)
- The same approach is used in sending and receiving messages
- The message attributes are different
- Mikey defines several payloads to support the methods and the corresponding architectural scenarios (p2p, one to many, many to many...)



Mikey interaction



- The main characteristic is that MIKEY minimizes message exchange operations

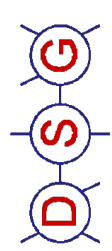


Conclusions and future work

21
-21

- JXTA-SOAP as a powerful solution for building service-oriented, peer-to-peer ubiquitous applications (standard edition and mobile edition)
- Mechanisms for secure service invocation in a peer-to-peer network

- Support for WS-Security in the J2ME-based version of JXTA-SOAP component
 - Sign and verify SOAP messages through security headers
 - End-to-end security when client and server are resource constrained devices



Questions?

22
-23

THANK YOU!

Contacts:

laghi@ce.unipr.it

www.ce.unipr.it/people/laghi