S&T in TripCom in the 3rd Year
- WP4 -

Milano 2009/01

Agenda

- Deliverable Status Report
- S&T progress in the 3rd Year
  - T4.4 “Analysis of WS-* standards and how they map into a Triple Space Environment”
  - T4.6 “Integration and evaluation of Triple Space within WSMX”
- Showcase Presentation at the 3rd review

Sixth Framework Programme Priority 2
Information Society Technologies (IST)
Specific Targeted Research Project

Status report

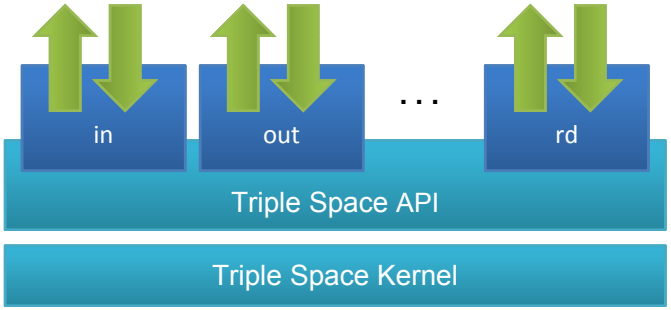
- Deliverables in the 3rd year
  - D4.4 “Methodology for augmenting Semantic Web Service with WS-* standards“ (USTUTT, M30)
    - Nature: Report
    - **Current status: finished, delivered M30**
  - D4.5 “Triple Space integration with respect to WSMX” (NUIG, M36)
    - Nature: Report/Prototype
    - **Current status: underway (due M36)**

D4.4

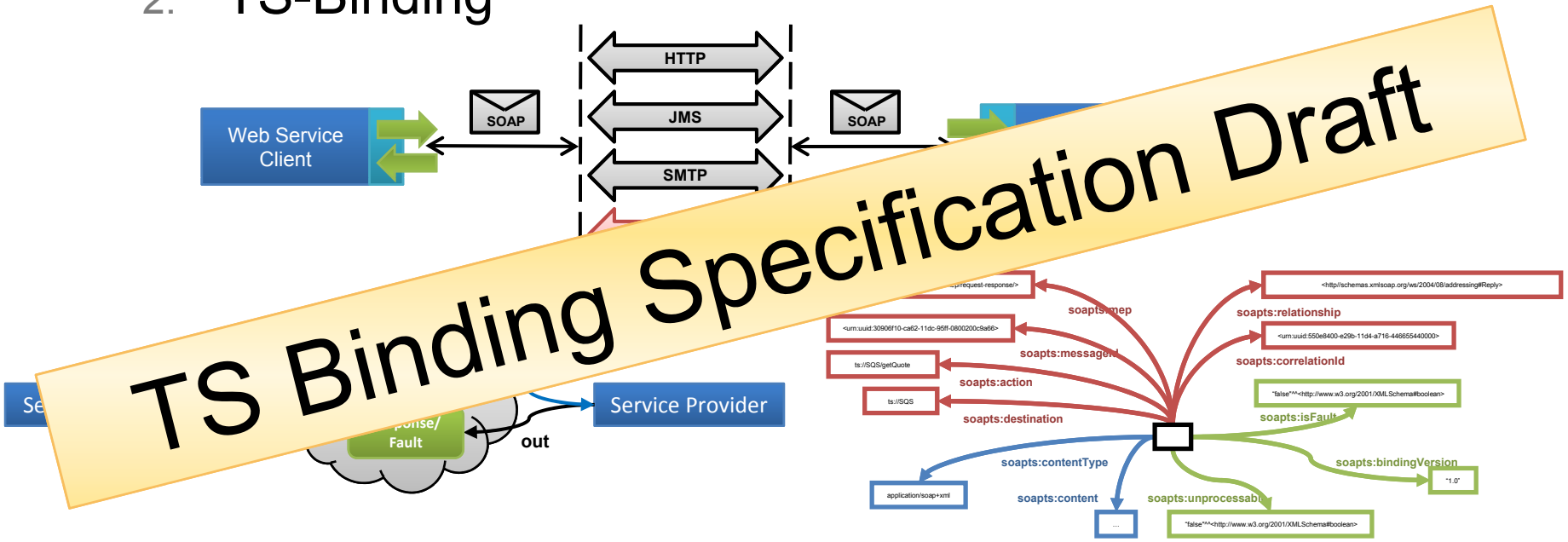
D4.4 – Summary

- Presentation of a number of common Web service standards from different areas
  - Messaging
  - Metadata
  - Security
  - Transactions
  - Resources
  - Management
  - Orchestration and Choreography
- Analysis and discussion of potential application within TripCom
- Special focus on standard's extensibility aspects
- Summarizes standard extension work already done within WP4 + goes beyond (e.g. BPEL ontology, WS-RM, WS-DM, …)

- SOAP: Two applications within TripCom
  1. WS-enabled TS API (WP3)



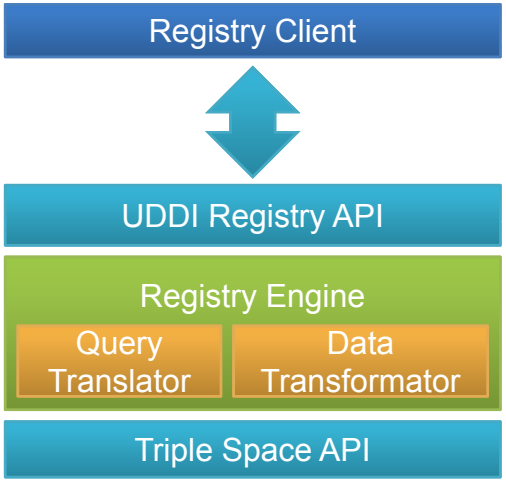
  2. TS-Binding



TS Binding Specification Draft

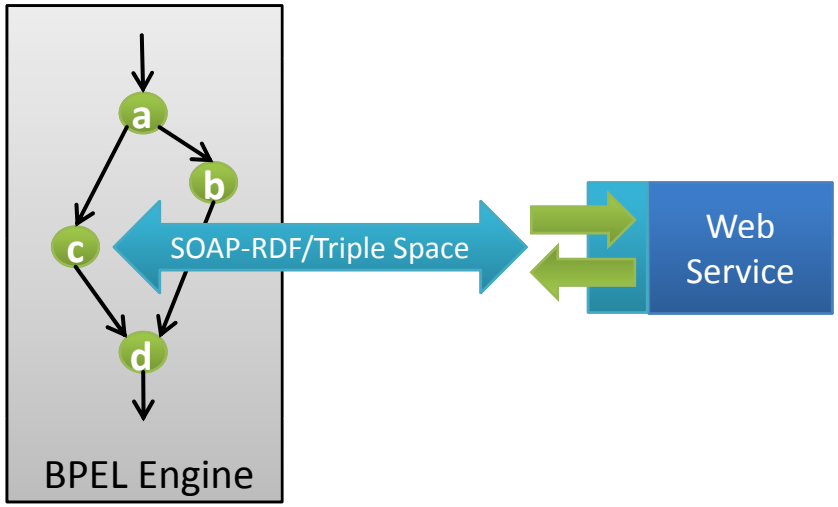
D4.4 – Metadata

- Metadata
  1. Description of functional service properties: WSDL
     - To support remote access to TS API operations WSDL can be used as is
     - To support description of Web services exposed over Triple Space a WSDL binding for Triple Space has been developed
     - To support storing WSDL descriptions in Triple Space the WSDL2.0-RDF-mapping has been implemented.
  2. Description of non-functional service properties: WS-Policy
     - The WS-Policy framework may be used as is
  3. Service registry
     - Development of a Triple Space-based UDDI registry



D4.4 – Business Processes (1)

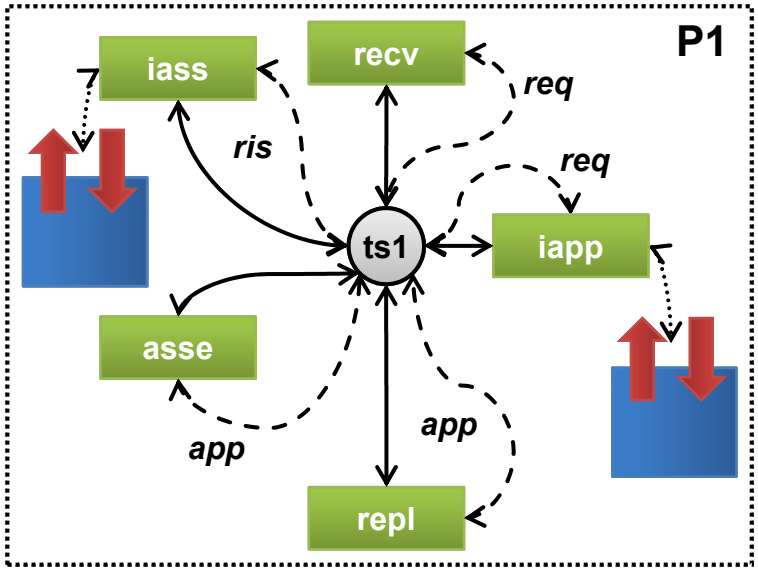
1. Invocation of Web services over Triple Space
   - Supported through the proposed Triple Space Web service binding



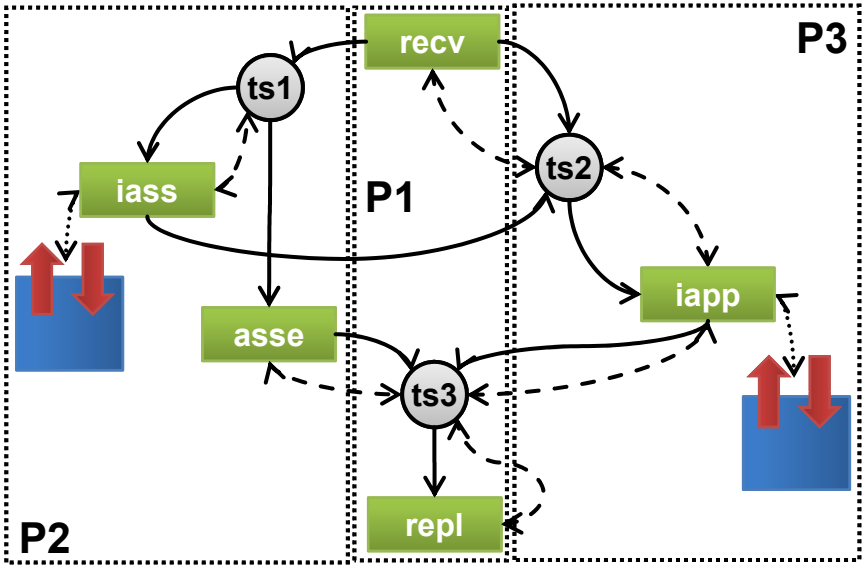
2. Orchestration of Triple Space API methods using WS-BPEL
   - Supported through the Triple Space Web service interface

D4.4 – Business Processes (2)

3. Triple Space as a storage mechanism for process models
   - Development of an RDF representation of BPEL processes
4. Decentralized navigation of BPEL processes
   - Process model for distributed execution of BPEL processes



Centralized Execution

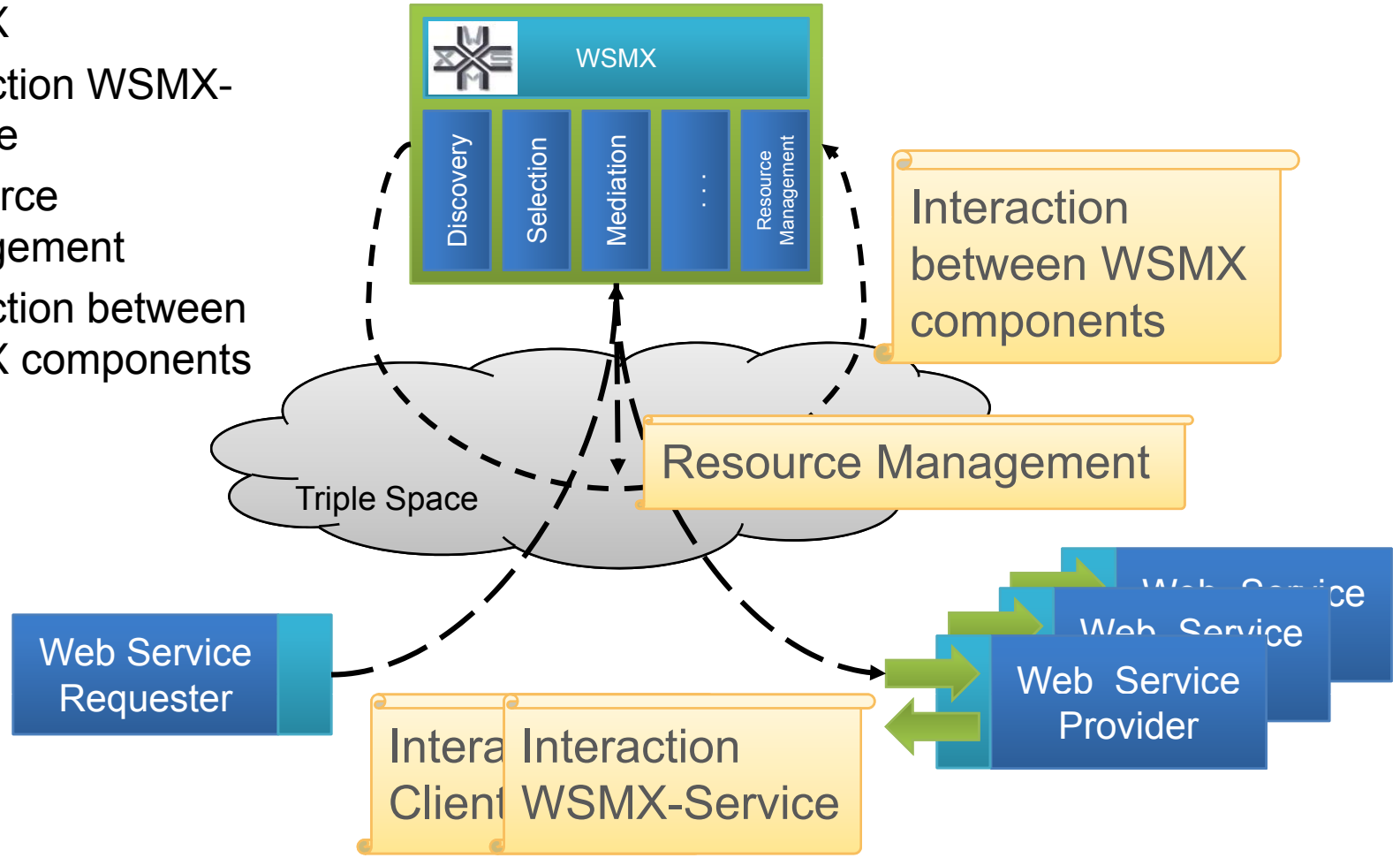


De-centralized Execution

D4.5
(Prototype)

Integration of Triple Space and WSMX

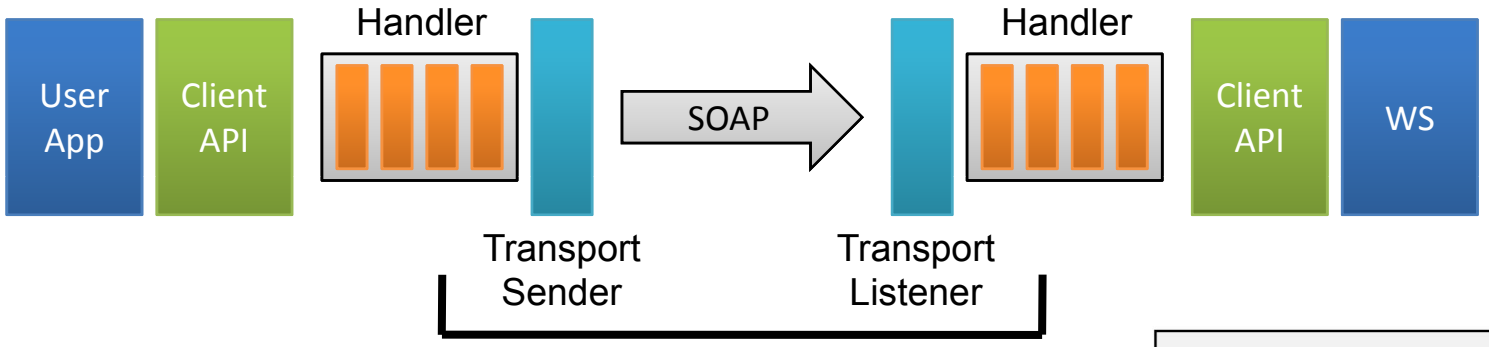
1. Interaction Client-WSMX
2. Interaction WSMX-Service
3. Resource Management
4. Interaction between WSMX components



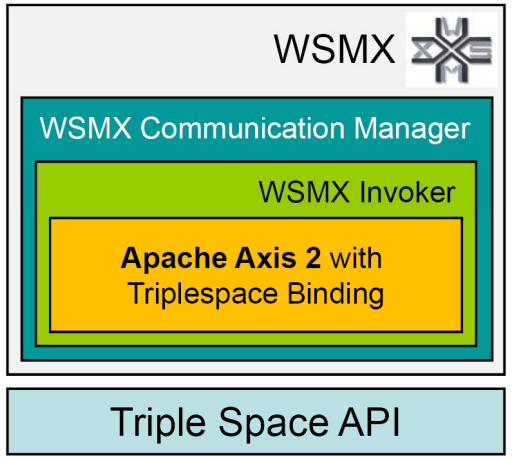
Sixth Framework Programme Priority 2
Information Society Technologies (IST)
Specific Targeted Research Project

Interaction WSMX-Service

- Interaction: WSMX – Service
  - Based on the implementation of the Web service binding for Triple Space
    - Based on *Apache Axis 2*



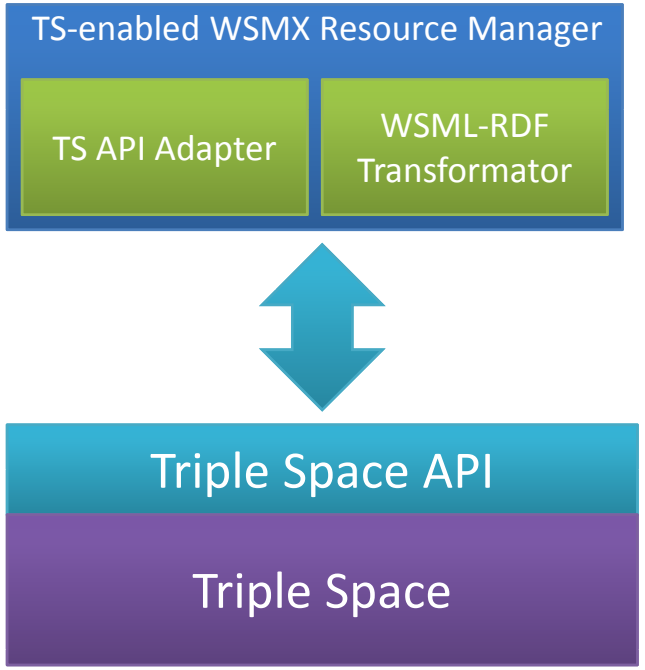
Triple Space-based implementation



  - Extension of the Invoker component of the WSMX communication manager

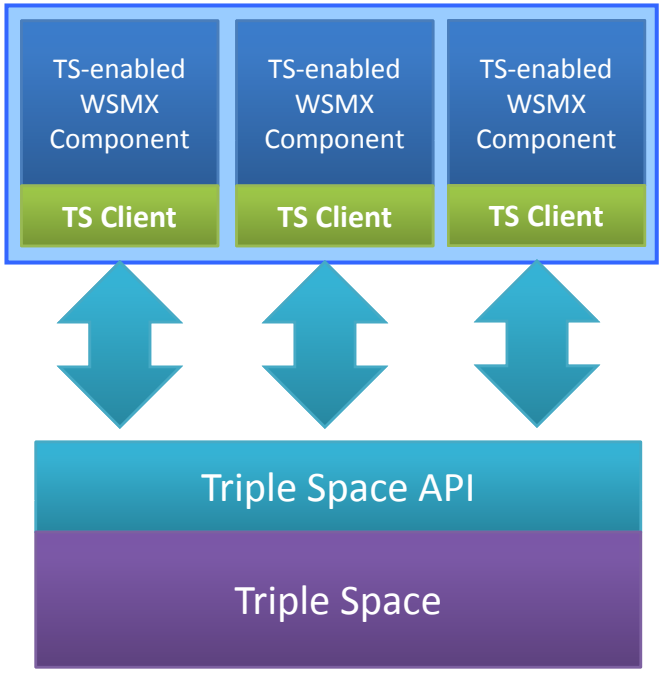
Resource Management

- Implementation of a WSMX Resource manager that uses Triple Space for data storage and retrieval
- Conceptually includes
  - a TS API adapter that maps RM operations to TS API operations
  - WSML-RDF translator for mapping data (i.e. Goals, Web Service descriptions, Mediators, Ontologies, event information) to RDF



Interaction between WSMX components

- Enable interaction of WSMX components by publishing messages to and retrieving from Triple Space
- Conceptually, each component includes a TripCom client that
  - transforms messages into their RDF representation and back
  - uses TS API operations to
    - subscribe to events directed to the component
    - publish messages directed to other components

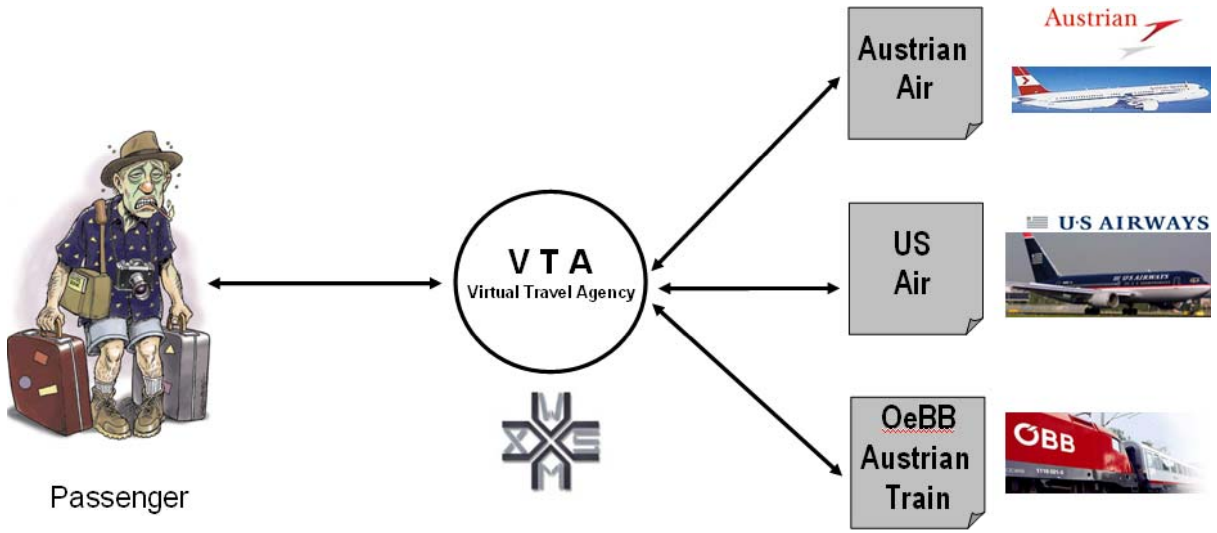


3rd Project Review
Demo Scenario

Scenario (1)

- “Virtual Travel Agency”
  - acts as a broker between passengers and different airlines as well as train services
  - allows passengers to specify their priorities regarding source and destination locations, preference about time and cost etc.
  - based on the information provided by the passengers, the VTA comes up with different travel options for the passengers
- Service types taken into account
  - Airlines
  - Trains
  - Hotels

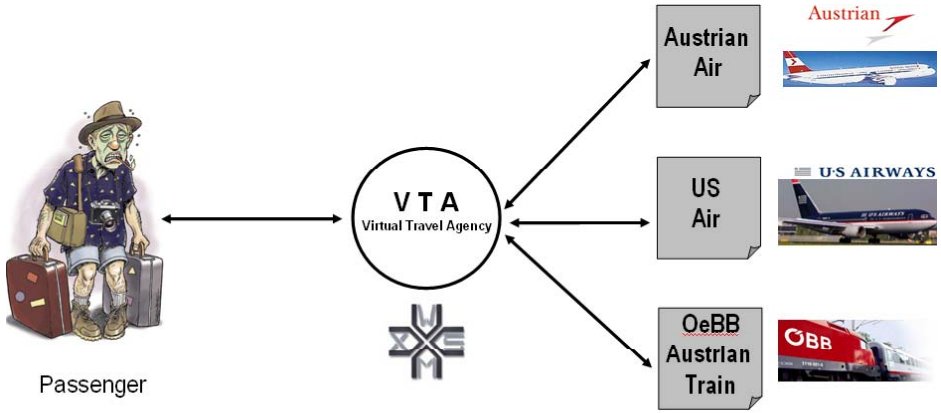
Scenario (2)



- A passenger decides to travel from San Francisco to New York City, then New York City to Vienna (Austria), and then from Vienna to Innsbruck
- The VTA should discover three services for the passenger, i.e. US airways, Austrian airways and OeBB train service

Realization with WSMX/TripCom

- Realization of the scenario comprises four steps
  - Step 1: Client accessing WSMX
  - Step 2: WSMX internal execution and component management
  - Step 3: WSMX storing or accessing data through the Resource Manager
  - Step 4: WSMX invoking external Web Services (i.e. Web Services of airlines or hotels)



Step 1: Client accessing WSMX

- "The VTA application acting as a client of WSMX submits a goal to WSMX, while communicating with it via Triple Space"

- The demo will show
  - the original goal (in WSML) that was prepared based on passenger's preferences
  - the conversion of the goal in WSML to RDF and submitting it to Triple Space
  - the WSMX communication manager receiving the notification and the contained goal

Step 2: Internal execution

- “After WSMX receives a goal, it has to use the discovery component in order to perform the matchmaking of the goal provided by the user and the semantic descriptions of known Web Services”

- The demo will show
  - the messages (i.e. WSMX-internal events) exchanged via Triple Space between the WSMX manager and its internal components

Step 3: Storing/accessing data

- “During goal processing, WSMX stores and accesses semantic descriptions of Web Services”

- The demo will show
  - the data being stored and retrieved to and from Triple Space by the WSMX Resource Manager
  - the TS API operations being used

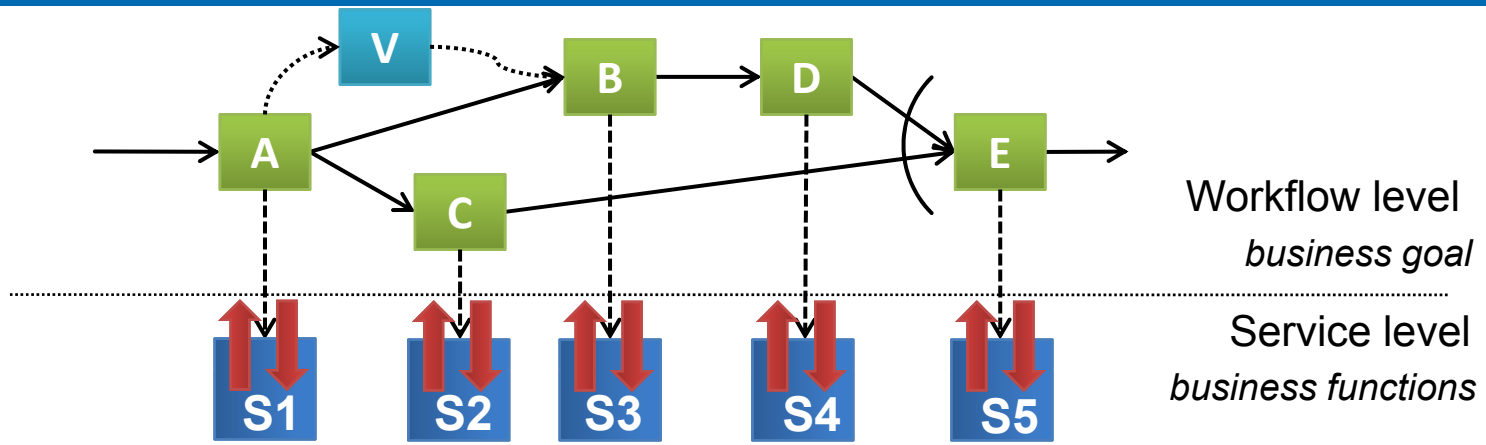
Step 4: External service invocation

- “After having discovered some Web service, WSMX uses the WSMX invoker to interact with the target Web service over Triple Space”

- The demo will show
  - the invocation message sent from the WSMX invoker to the Web service using the Triple Space WS Binding
  - and the received response message

End of Document

Business Process Execution Language



- Based on the “Two-level Programming” paradigm
- Orchestrated services (B,C,D) and the process itself (A,E) are Web services
- Basic activities
  - Interaction with services, data manipulation, …
- Structured activities
  - Sequential activity execution, branching, iteration, parallel execution with synchronization
- Typed variables and means for fault handling and compensation