Unit 6 Service Oriented Architecture

Unit Outcomes. Here you will learn

- How different WS can be coordinated to work together in systematic, standard ways.
- Why the concept of service architecture is interesting for developing large business DSs.
- How various WS standards extend the basic WS to facilitate important aspects of large business DSs such as security, resources, addressing and notification, in particular:
  - how UDDI helps in the discovery and management of services

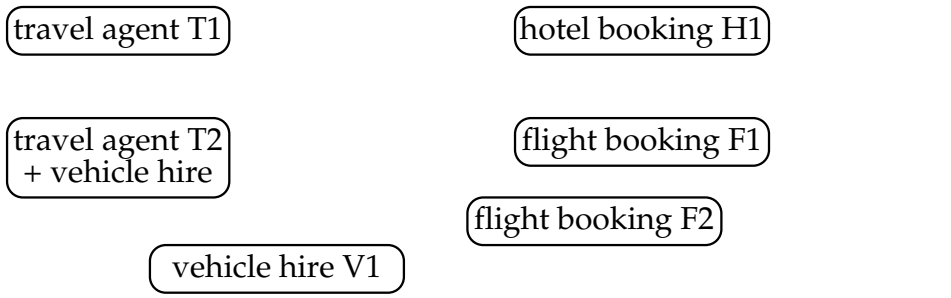
Contents

1 Introduction to SOA
- Example
- Definition
- Business benefits of SOA
- Standard service descriptions
- Overview of WS standards

2 WS registry services — UDDI
- Uses
- Data model description
- Data model class diagram

Introduction to SOA
Example

- WS intended for large DSs spanning multiple organisations



Definition

- SOA = large DSs communicating solely through services
- SOA usually built using SOAP WS but not always (REST)
- SOA service:
  - long-term available (on demand)
  - *concisely* yet *fully* described in a standard way, consequently
    - easy to connect to and use for all programmers
    - having confined and predictable effects
    - using standard formats for data exchange
  - accessed in a *stateless* request-response manner

Business benefits of SOA

- why SOA and not distributed objects or ad hoc RPC?
  - services *reusable* for multiple purposes;
  - can *integrate* new and *legacy* systems;
  - applications *adaptable* to changing business environment and available technologies;
  - cheap and flexible electronic links with other businesses (*e-business*).

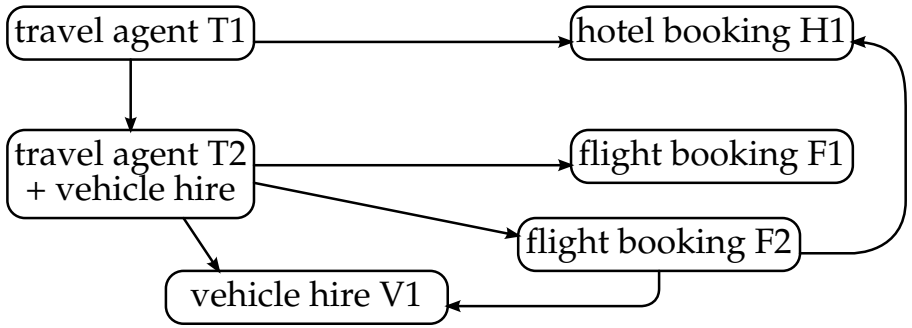
Standard service descriptions

- for RESTful services:
  Web Application Description Language (WADL)
- for RPC and similar Web services:
  Web Service Description Language (WSDL)
- such descriptions can be used to:
  - auto-generate parts of code for server and client
  - recognise functionally identical services

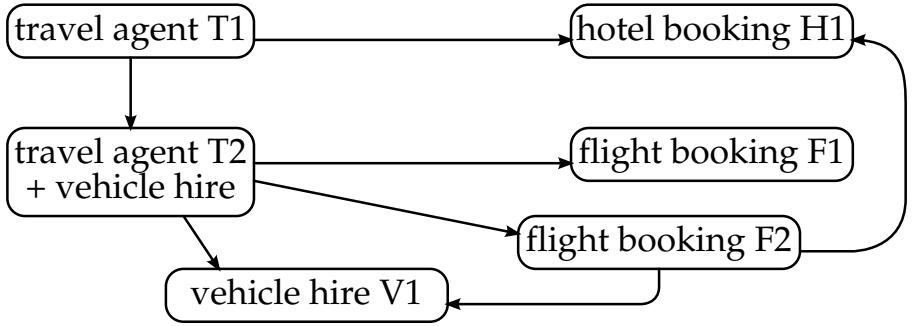
Overview of WS standards

- standardise common WS patterns (WSDL, SOAP not enough)
  - service *publishing* and *discovery*
  - service *choreography*, ie describing sequences of service invocations
  - management of shared stateful *resources*
  - *transactions*, ie coordinated update of multiple resources
  - *notifications* of changes in stateful resources
  - service *life-cycle* management (eg deploying, upgrading, decommissioning) and *monitoring*
  - *reliability* of messaging beyond TCP/IP
  - *security* (eg authentication, encryption, permissions management)
  - service *usage contracts* (eg payment, performance, booking, penalties for failures)

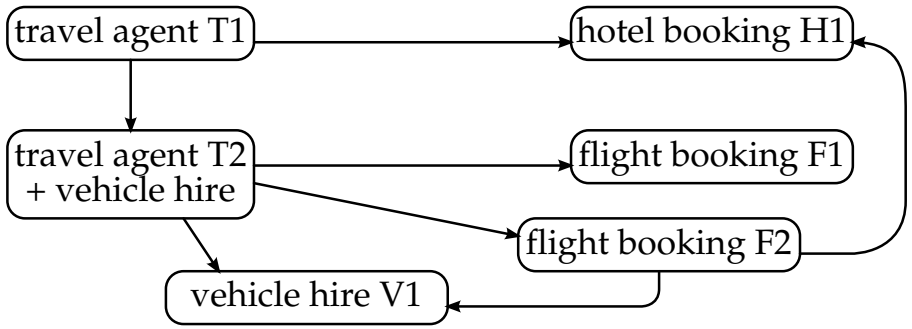
Overview of WS standards — discovery



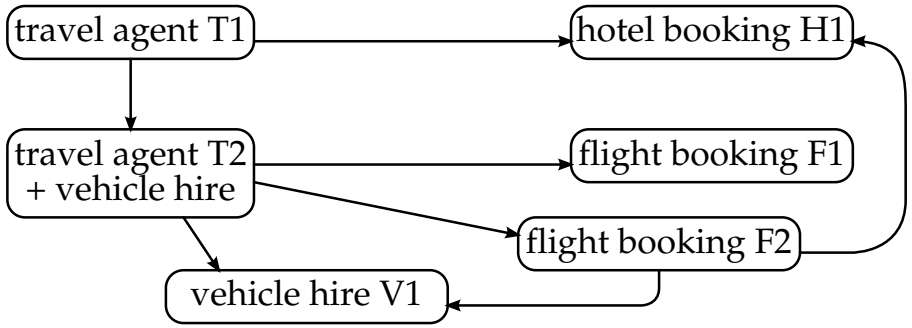
Overview of WS standards — choreography



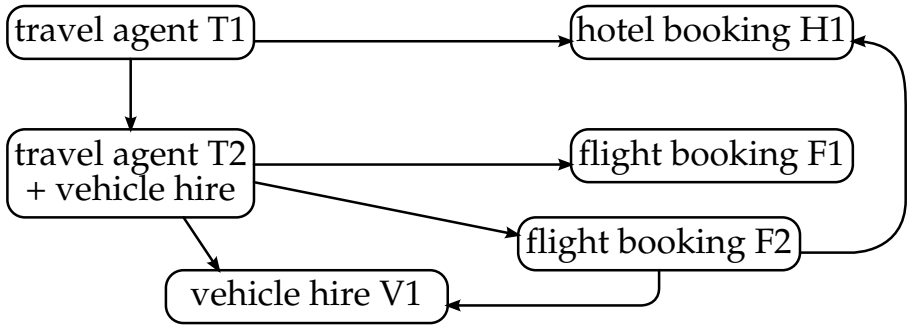
Overview of WS standards — resources, transactions, notification



Overview of WS standards — management and monitoring



Overview of WS standards — reliability, security, contract



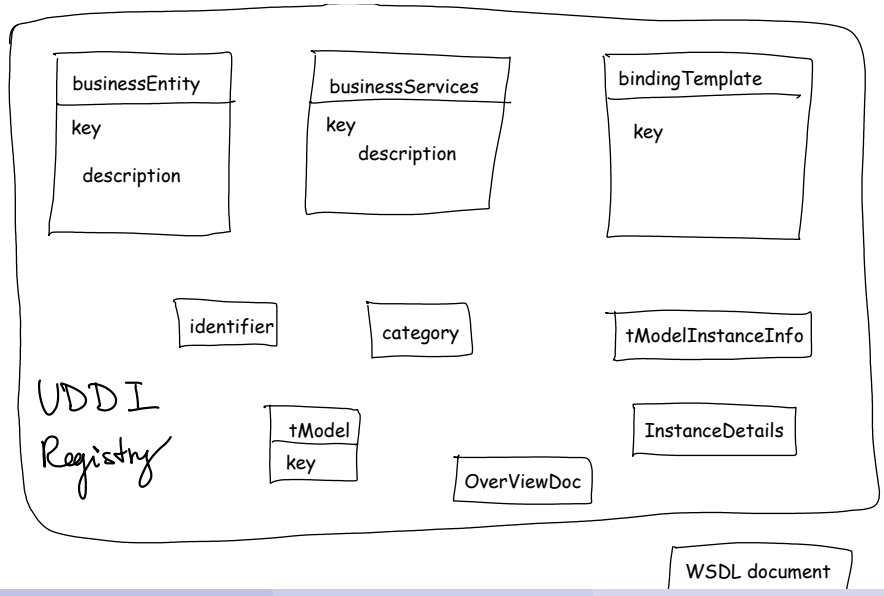
WS registry services — UDDI
Uses

- UDDI = Universal Description Discovery and Integration
- service providers
  - advertise services
  - relate their services to industry standards and taxonomies
- service users:
  - locate suitable services: manually or automatically
    - indexed by unique identifiers: "white pages"
    - indexed by standard categories: "yellow pages"
  - get links to technical specification of services ("green pages")

Data model description

- *business entity* descriptions
  - mainly for human reading and keyword searching
  - indexed by unique identifiers and categories
  - contains:
- *business service* descriptions
  - human-friendly description of a family of similar services
  - contains:
- *binding templates*
  - description of concrete ports for service
  - contains:
    - human-friendly descriptions
    - URLs to technical descriptions (usually XML documents) — instance of technical model (eg WADL, WSDL)

Data model class diagram



Learning Outcomes

Learning Outcomes. You should now be able to

- Using examples, describe the importance of various WS standards for security, resources, orchestration, addressing and notification in developing open, widely applicable services.
- Describe the UDDI mechanism for automated publishing and discovering of Web Services and argue its strengths and weaknesses.