

Information systems modeling

Tomasz Kubik



UDDI specification

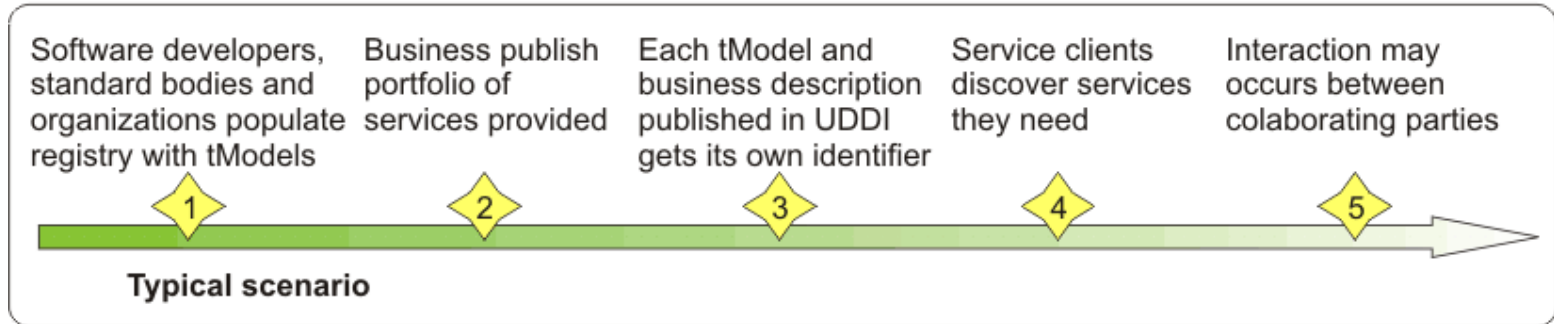
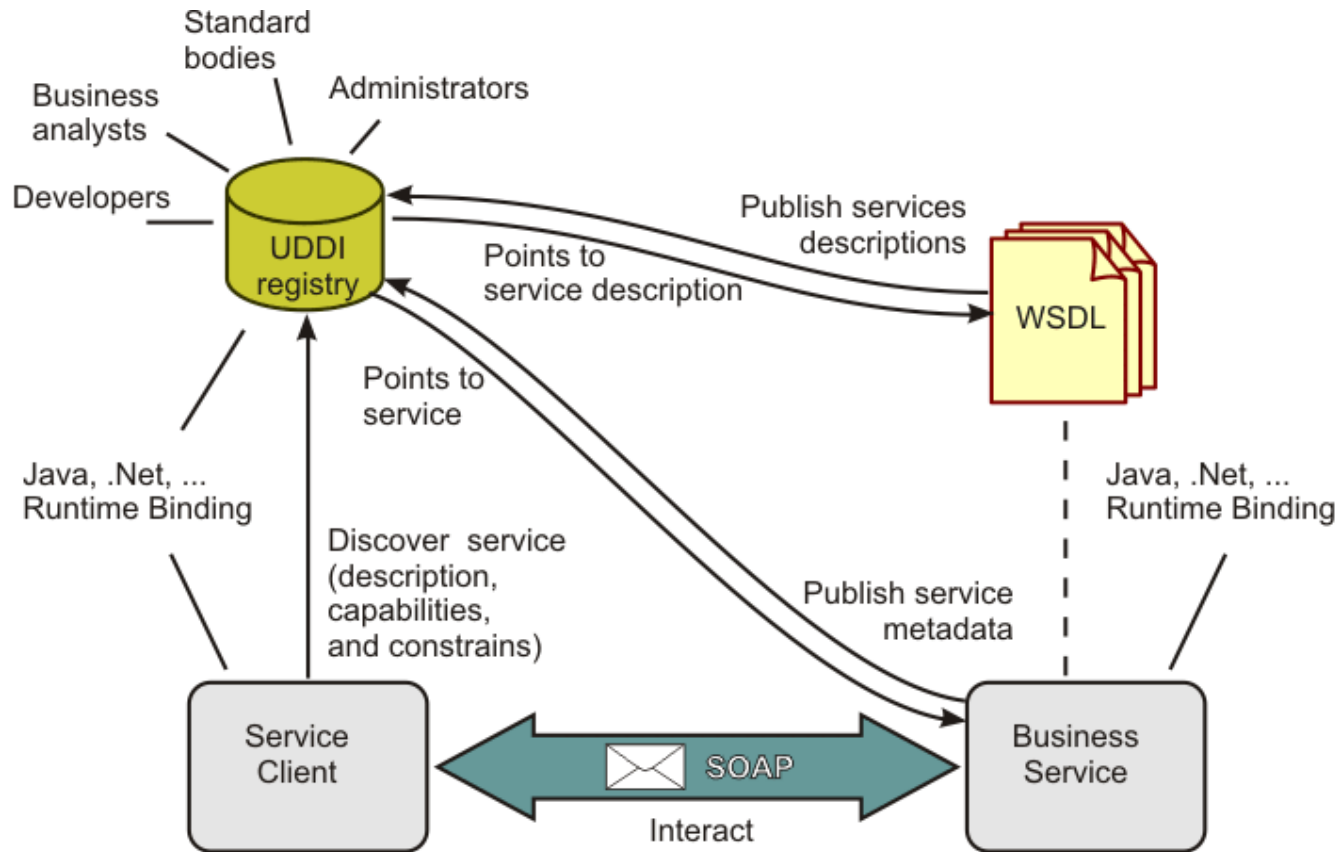
- Maintained by OASIS
- Based upon several other established industry standards
 - HTTP, XML, XML Schema, SOAP and WSDL.
- Includes:
 - Data model
 - allows creation and storing of information on businesses and services
 - each core data structure gets in UDDI an unique identifier (UDDI key)
 - API specification
 - defines a set of operations on UDDI registry as searching for data, publishing data, and other operations
 - applies management of rights
 - Registry
 - enables companies to advertise their business
 - enables customers to find potential partners.

A phone book metaphor

- *The UDDI registry contains:*
 - *green pages - a technical description of the service and its URL reference (by assumption the service described do not necessarily have to be a Web service);*
 - *white pages - identification, addresses and other contact details of companies;*
 - *yellow pages - a list of companies arranged by industrial classification.*

Facts

- Version 1.0, released in 2000
 - a foundation for the registry of Internet-based business services
- Version 2.0, released in 2001 and ratified as an OASIS Standard in 2003,
 - aligned the specification with emerging Web services standards and provided a flexible service taxonomy.
- Version 3.0, released in 2004 and ratified as an OASIS Standard in 2005
 - supported secure interaction of private and public implementations as major element of service-oriented infrastructure.
- The current version is 3.0.2.
- Standards are published on a web page:
<http://www.oasis-open.org/specs/index.php>

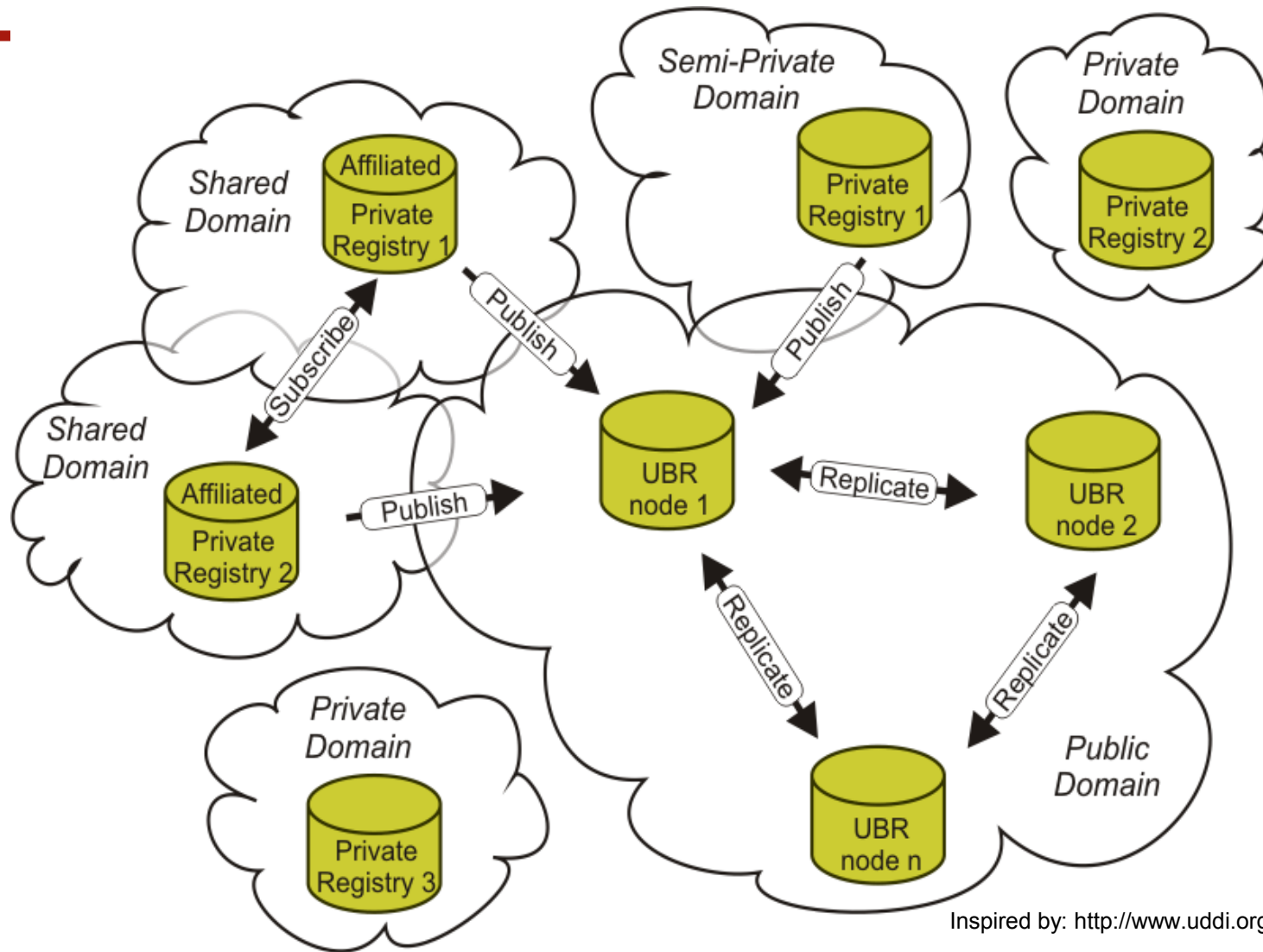


Some concepts

- *node*
 - a server that supports at least the minimum set of functionality
 - a member of exactly one UDDI registry
- *registry*
 - is composed of one or more nodes.
 - provides complete set of functionality
- *affiliated registry*
 - implements policy-based sharing of information with another registry
 - share a common namespace for UDDI keys that uniquely identify data records

The types of UDDI servers

- private
 - internal registry deployed behind firewall,
 - isolated from the public network
 - offers restricted access to administrative functions and data
 - data is not shared with other registries
- affiliated
 - deployed within controlled environment
 - offers an access for authorized clients
 - administrative tasks may be delegated to trusted parties
 - data may be shared with other registries in a controlled manner
- public (UDDI Business Registry, UBR)
 - service in a cloud
 - administrative functions usually secured
 - access to data is essentially open and public
 - data may be shared or transferred among other registries,
 - content may or may not be moderated



Inspired by: <http://www.uddi.org/pubs/uddi-tech-wp.pdf>

General rules

- UDDI registry **MUST** have at least one node that offers a Web service compliant Inquiry API set.
- UDDI registry **SHOULD** have at least one node that offers a Web service compliant with the Publication, Security, and Custody and Ownership Transfer API sets.
- If a UDDI registry has multiple nodes, all nodes **SHOULD** offer Web services that are compliant with the Replication API set.
 - data supplied to one of the registry nodes can be replicated to all other nodes (replication occurs every 24 hours).
- The Subscription and Value Set API sets are **OPTIONAL** for all nodes and all registries.
- A registry **MUST** make a policy decision for each policy decision point. It **MAY** choose to delegate policy decisions to nodes.

Registry Affiliation in the UDDI specification

- Supports variety of network/infrastructure topologies
 - hierarchical, peer-based, delegated, ...
- Allows to match up the structure of a UDDI registry with the underlying business processes
- Provides guidance to help facilitate the maintenance and mapping of UDDI keys and records across registries
 - facilitate a wide range of business scenarios, but do not defines them

UDDI Version 3.0.2 XML Schema files

(available at <http://uddi.org/schema/>)

Description	XML Schema file
API Schema	uddi_v3.xsd
Custody Schema	uddi_v3custody.xsd
Subscription Schema	uddi_v3subscription.xsd
Subscription Listener Schema	uddi_v3subscriptionListener.xsd
Replication Schema	uddi_v3replication.xsd
Value Set Validation Schema	uddi_v3valueset.xsd
Value Set Caching	uddi_v3valuesetcaching.xsd
Policy	uddi_v3policy.xsd
Policy Instance Parameters	uddi_v3policy_instanceParms.xsd

UDDI Version 3.0.2 WSDL Service Interface Descriptions files

(available at <http://uddi.org/wsd/>)

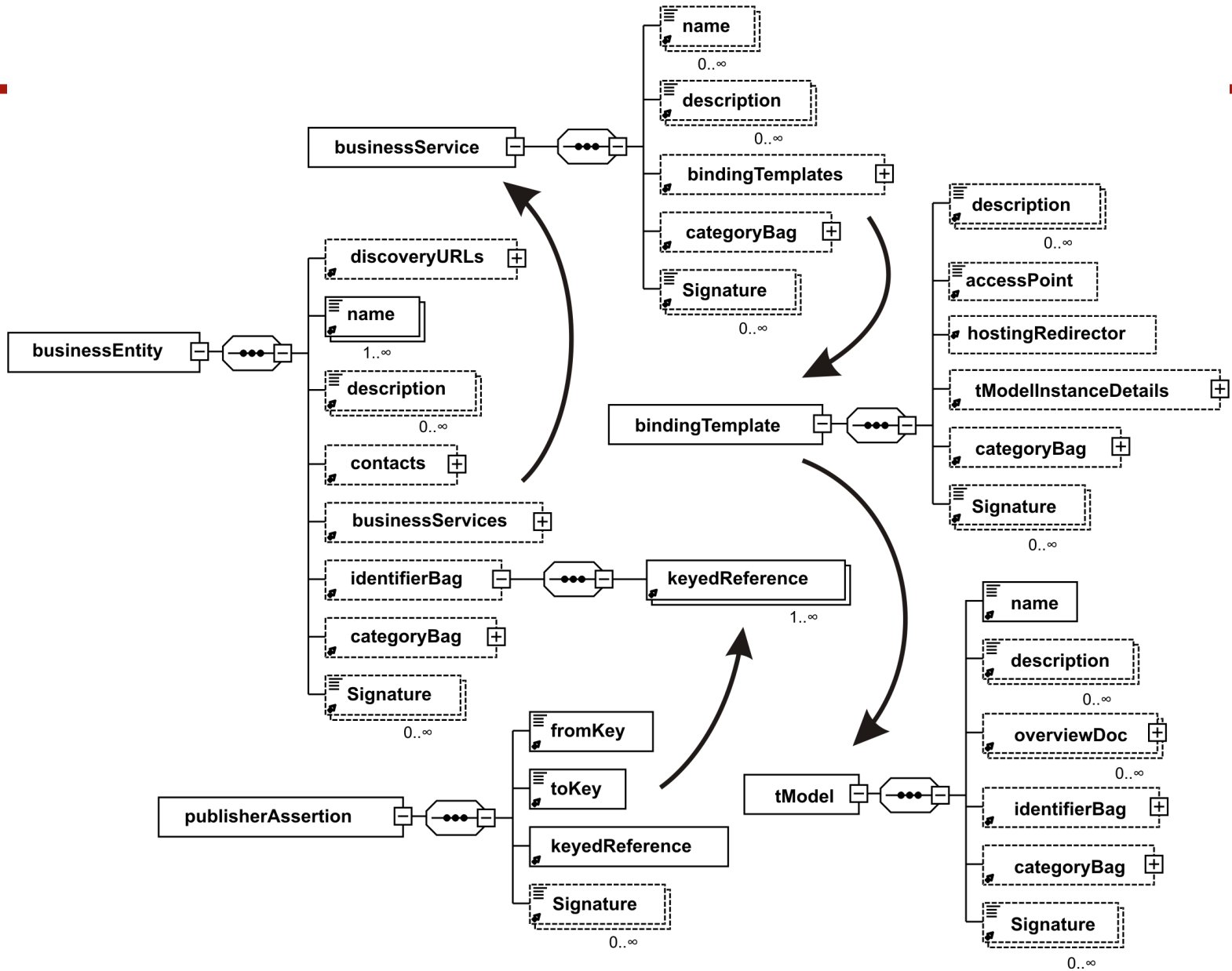
Description	WSDL file	Description	WSDL file
API Binding	uddi_api_v3_binding.wsdl	Subscription Port Type	uddi_sub_v3_portType.wsdl
API Port Type	uddi_api_v3_portType.wsdl	Subscription Listener Binding	uddi_subr_v3_binding.wsdl
Custody Binding	uddi_custody_v3_binding.wsdl	Subscription Listener Port Type	uddi_subr_v3_portType.wsdl
Custody Port Type	uddi_custody_v3_portType.wsdl	Value Set Validation Binding	uddi_vs_v3_binding.wsdl
Replication Binding	uddi_repl_v3_binding.wsdl	Value Set Validation Port Type	uddi_vs_v3_portType.wsdl
Replication Port Type	uddi_repl_v3_portType.wsdl	Value Set Caching Binding	uddi_vscache_v3_binding.wsdl
Subscription Binding	uddi_sub_v3_binding.wsdl	Value Set Caching Port Type	uddi_vscache_v3_portType.wsdl

Accessing UDDI programmatically

- Formal definitions of UDDI APIs and core data model are provided in XML Schemas
- Service operations invocation through messages
- API sets defined
 - Node API Sets:
 - UDDI Inquiry, UDDI Publication, UDDI Security, UDDI Custody Transfer, UDDI Subscription, UDDI Replication
 - Client API Sets
 - UDDI Subscription Listener, UDDI Value Set
- The most commonly used APIs
 - Inquiry, Publication and Security APIs.

API Schema (core data model)

- **businessEntity**
 - represents descriptive information about the business or provider and about the services it offers, such as: contact information, business category, business identifiers, the list of services provided. It includes one or more **businessService** structures containing service descriptions and technical information.
- **businessService**
 - represents collection of related network services offered by a service provider defined by **businessEntity**. It includes information about particular service's binding, type, and category through one or more **bindingTemplates**.
- **bindingTemplate**
 - represents technical information necessary to use a particular Web service. It contains references to **tModels**.
- **tModel**
 - represents a technical model of reusable concept, such as a Web service type, a protocol used by Web services, or a category system. The use of **tModel** structure eliminates duplication of the same information in different places (e.g. in a situation where many services offers the same interface).
- **publisherAssertion**
 - A **publisherAssertion** structure links two or more business entities represented by **businessEntity** structure.



businessEntity attributes

- businessKey
 - The value of is UUID type,
 - MUST be omitted if the publisher wants the registry to generate a key during registration.
 - MUST be present, if a businessEntity is retrieved from a UDDI registry.

businessEntity elements

- discoveryURLs
 - a list of URLs that point to alternate, file based service discovery mechanisms;
- name
 - a name of a businessEntity (required, non-empty, can occur multiple times);
- description
 - textual information about the businessEntity (optional, can occur multiple times);
- contacts
 - a simple list of single contact information (optional, can occur multiple times);
- businessServices
 - a list of business services provided by a businessEntity;
- identifierBag
 - contains a list of identifiers used for other purposes than a businessKey, valid in their own identifier systems (like tax identifier);
- categoryBag
 - contains a list of business categories which a businessEntity can be associated with
- Signature
 - contains a digital signature created in accordance with the XML-Signature specification (optional, but MUST be provided if a businessEntity was digitally signed).

businessService attributes

- **serviceKey**
 - uniquely identifies a given businessService in all UDDI registries. The value of serviceKey follows the same rules as the value of businessKey attribute of businessEntity (it MUST be omitted when registering an entity and MUST be present when retrieving an entity).
- **businessKey**
 - represents logical parent-child relationship between a businessEntity and a businessService. Every businessService has exactly one businessEntity parent, uniquely identified by businessKey attribute. However, the value of a businessKey attribute of a businessService may differ from the value of a businessKey attribute of a parental businessEntity. When it happens, it indicates a service projection.

businessService elements

- name
 - a name of a businessEntity (required except when indicating a service projection, non-empty, can occur multiple times);
- description
 - textual information about the businessService (optional, can occur multiple times);
- bindingTemplates
 - a list of technical descriptions for the Web services provided;
- categoryBag
 - contains a list of business categories which a businessService can be associated with (like industry, product category or geographic region);
- Signature
 - contains a digital signature created in accordance with the XML-Signature specification (optional, but MUST be provided if a businessService was digitaly signed).

bindingTemplate attributes

- **bindingKey**
 - uniquely identifies a bindingTemplate in all UDDI registries. The value of bindingKey follows the same rules as the value of businessKey attribute of businessEntity (it MUST be omitted when registering an entity and MUST be present when retrieving an entity).
- **serviceKey**
 - uniquely identifies the businessService that contains the bindingTemplate (each bindingTemplate is the child of a single businessService which is referenced by a serviceKey). The value of serviceKey follows similar rules as the value of bindingKey attribute (it MAY be omitted when registering a bindingTemplate entity and this entity is a part of a fully expressed businessService element, and MUST be present when retrieving a bindingTemplate).

bindingTemplate elements

- description
 - textual information about the bindingTemplate (optional, can occur multiple times, potentially in multiple languages);
- accessPoint
 - a string used to convey the network address suitable for invoking the Web service being described (typically an URL, but may be any other locator, as an e-mail address, a telephone number, etc.);
- hostingRedirector
 - a deprecated element, mutually exclusive with access point
- tModelInstanceDetails
 - a structure containing a list of one or more tModelInstanceInfo elements, each with tModelKey attribute. All they form a kind of stamp that can be used to identify compatible services;
- categoryBag
 - a container for categories which a bindingTemplate can be associated with (as, for example, „test” or „production”).
- Signature
 - contains a digital signature created in accordance with the XML-Signature specification (optional, but MUST be provided if a bindingTemplate was digitally signed).

tModel attributes

- tModelKey
 - uniquely identifies a tModel entity in all UDDI registries. The value of tModelKey follows the same rules as the value of businessKey attribute of businessEntity (it MUST be omitted when registering an entity and MUST be present when retrieving an entity).
- deleted
 - appears in the retrieved tModel data as an information-only field. It indicates whether tModel was deleted from a registry or not. Two allowed values for this attribute are "true" and "false".

tModel elements

- name
 - a name of a tModel (required, non-empty, can occur multiple times, SHOULD be formatted as a URI, and the xml:lang attribute SHOULD NOT be used);
- description
 - short, textual information about the tModel (optional, can occur multiple times and in multiple languages);
- overviewDoc
 - contains an URL reference to remote descriptive information or instructions related to the tModel (optional, can occur multiple times)
 - overviewURL element, included in overviewDoc, with optional useType, holds URL ("text" value of useType tells that overviewURL refers to additional textual information, and "wsdlInterface" - that to a WSDL interface document).
- identifierBag
 - contains a list of identifiers used for other purposes than a tModelKey (valid in their own identifier systems).
- categoryBag
 - a container for categories which a tModel can be associated with.
- Signature
 - contains a digital signature created in accordance with the XML-Signature specification (optional, but MUST be provided if a tModel entity was digitally signed).

publisherAssertion

- used in a case when there are some associations, such as the manufacturer - supplier, contractor - subcontractor, etc. between these entities.
- To make these associations visible in the registry (for customers who wish to find information about cooperating businesses), these structures should be registered by the both entities involved.
- publisherAssertion structure consists of three elements:
 - fromKey (a key of the first entity),
 - toKey (a key of the associated entity) and
 - keyedReference (a reference that defines the type of association in terms of pairs: KeyName, keyValue inside a tModel referenced by tModelKey).

Node API Sets

- UDDI Inquiry
 - contains operations for querying the registry for details on registered entities
- UDDI Publication
 - contains operations for publishing entities into the registry
- UDDI Security
 - contains operations for authentication handling
- UDDI Custody Transfer
 - contains operations for transferring ownership and custody of entities
- UDDI Subscription
 - contains operations for retrieving information on entities in a timely manner using a subscription format
- UDDI Replication
 - contains operations related to data replication between registry nodes

Client API Sets

- **UDDI Subscription Listener**
 - contains operations for receiving subscription results
- **UDDI Value Set**
 - contains operations related to keyed reference values validation

Inquiry API Set, search requests

- `find_binding` – returns a list of network services bindings matching the criteria specified in the input arguments (on technical information);
- `find_business` – returns a list of business entities matching the search criteria;
- `find_relatedBusinesses` – discovers related business compounds.
- `find_service` – returns a list of network services matching the search criteria;
- `find_tModel` – returns a list of tModel structures matching the search criteria;

Inquiry API Set, drilling demands

- `get_bindingDetail` – returns the complete information about the `service bindingTemplate` structure for all values of `bindingKey` listed in the request;
- `get_businessDetail`– returns the information in the `businessEntity` structure for all business entities whose values are `bindingKey` listed in the request ;
- `get_businessDetailExt` – returns extended information on a business entity in a `businessDetailExt` structure;
- `get_operationalInfo` – returns the information in the `operationalInfos` structure (containing data such as time and date of creation of a structure, the date and time Last modified, node identifier, the entity which has been published, the service provider identifier) for all entities whose values are `entityKey` listed in the request.
- `get_serviceDetail` – returns the details in the structure of `businessService` for services for which the value of `serviceKey` listed in the request;
- `get_tModelDetail` – returns the information for `tModel` entities identified by the values of `tModelKey` listed in the request;

Publication API Set (add, get, save)

- `add_publisherAssertions` – adds one or more relationship assertions to the existing assertion collection;
- `get_assertionStatusReport` – provides administrative support for determining the status of all assertions made involving any `businessEntity` controlled by the requesting publisher account. The status is included in the `assertionStatusReport` returned.
- `get_publisherAssertions` – gets a list of all relationship assertions associated with a specific publisher account in `publisherAssertions` structure;
- `get_registeredInfo` – returns an abbreviated synopsis of all information currently managed by a given individual;
- `save_binding` – create or updates `bindingTemplate` information;
- `save_business` – create or update `businessEntity` information;
- `save_service` – create or update complete information about a `businessService` exposed by a specified `businessEntity`;
- `save_tModel` – create or update complete information about a `tModel`;

Publication API Set (delete, set)

- `delete_binding` – removes an existing `bindingTemplate` from a `bindingTemplates` collection that is part of a specified `businessService` structure;
- `delete_business` – removes the registered `businessEntity` information from the registry;
- `delete_publisherAssertions` – removes one or more assertion from the assertion set managed by a particular publisher account.
- `delete_service` – removes the existing `businessService` from the `businessServices` collection that is part of a specified `businessEntity`;
- `delete_tModel` – hides registered information about a `tModel` (`models` can not be deleted normally, except by administrative petition);
- `set_publisherAssertions` – manages all relationship assertions for an individual publisher account (replaces any existing assertions, and causes any old assertions that are not reasserted to be removed from the registry);

Security Policy API Set

- `discard_authToken` – is used to inform a node that an authentication token it has obtained is no longer required and should be considered as invalid (equivalent to logout from the system).
- `get_authToken` – is used to request an authentication token potentially required for performing operations included in the Inquiry API Set, Publication API Set, Custody and Ownership Transfer API Set, and Subscription API Set. The requested token issued from a UDDI node has the form of an `authInfo` element (equivalent to login to the system).

Custody and Ownership Transfer

API Set

- `discard_transferToken` – is used to discard a `transferToken` obtained through the `get_transferToken` API at the same node;
- `get_transferToken` – is used to initiate the transfer of custody of one or more `businessEntity` or `tModel` entities from one node to another;
- `transfer_entities` – is used by publishers to whom custody is being transferred to actually perform the transfer. The recipient publisher must have an unexpired `transferToken` that was issued by the custodial node for the entities being transferred;
- `transfer_custody` – is used by the custodial node to ensure that permission has been granted to transfer custody of the entities that the target publisher has requested (is invoked by the target node in response to `transfer_entities`). The `transfer_custody` API is in the replication namespace since it is sent from one node to another node in a registry using replication.

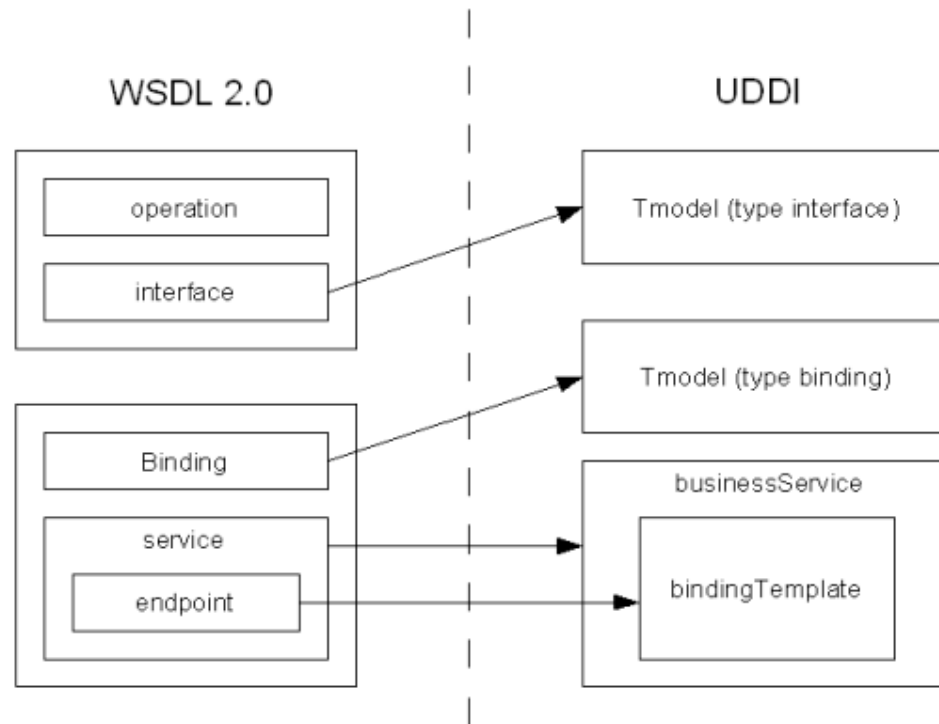
Subscription API Set

- `save_subscription` – creates a new subscription, changes an existing subscription or renew an existing subscription;
- `delete_subscription` – cancels one or more subscription;
- `get_subscriptions` – returns a list of existing subscriptions previously saved by the applicant;
- `get_subscriptionResults` – returns the registry data pertaining to the specific subscription within a specified time;
- `notify_subscriptionListener` – optional method implemented by a subscriber and pre-subscribed in UDDI, called by the UDDI node in order to notify subscriber about data changes (about new, modified or deleted data matching the subscription).

Value Set API Set

- `validate_values` – used by nodes to allow external network service providers to validate value sets (to determine whether `keyedReferences` or `keyedReferenceGroups` are correct). Returns a `dispositionReport` structure;
- `get_allValidValues` – used by nodes to obtain the set of valid values from cacheable checked value sets (applies to nodes supporting caching of valid values). Returns an empty message or a `dispositionReport` structure.

Simplified representation of the correspondences between WSDL 2.0 and UDDI structures



Simplified representation of the correspondences between WSDL 1.1 and UDDI V2 structures

