

WP8A Use Case Implementation

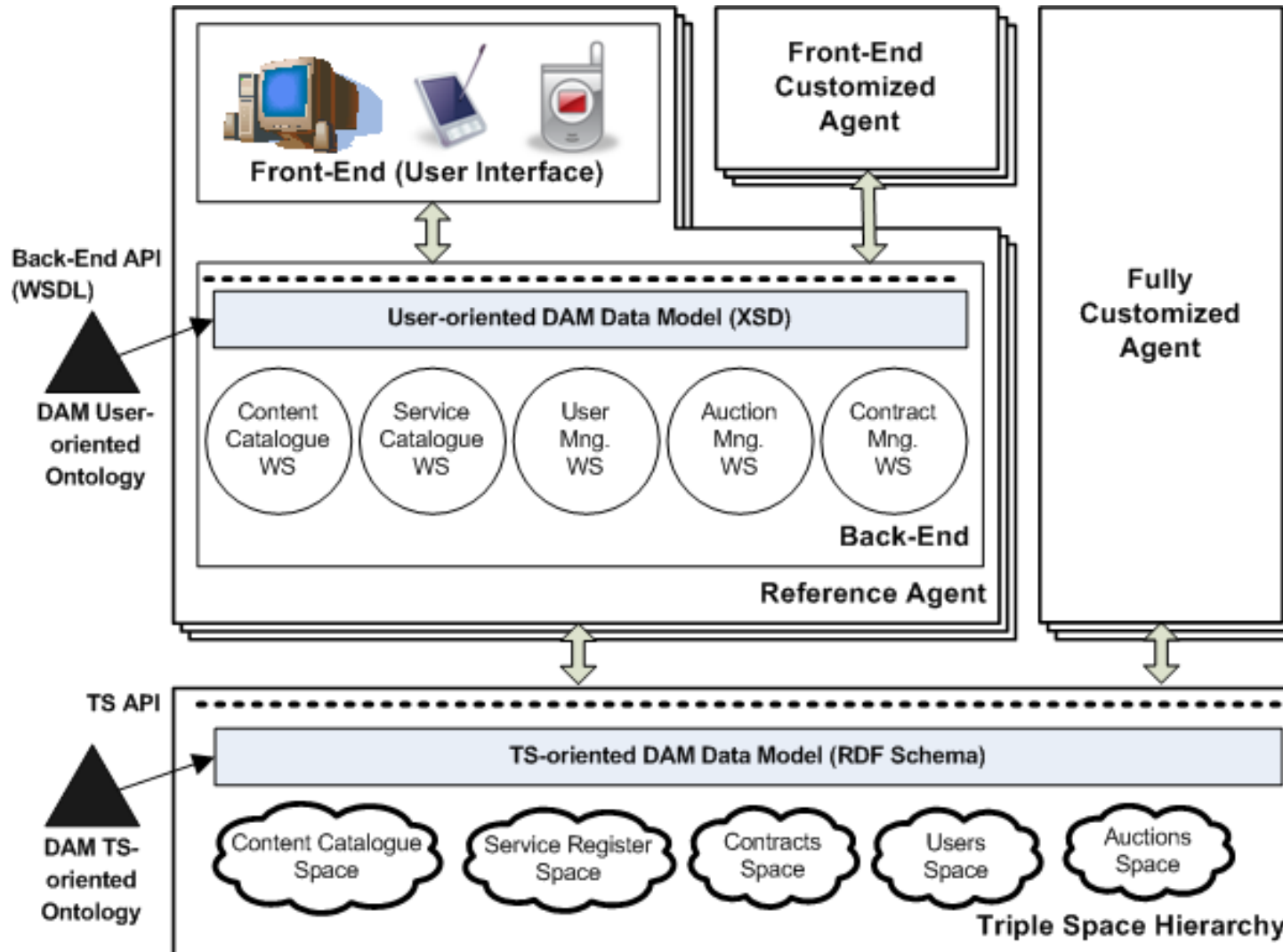


David de Francisco
Telefónica I+D
Innsbruck, January 2008



- Use Case Architecture
- Ontologies
- Interactions Between Actors
- Spaces Hierarchy
- Implementation Plan & Requirements

Use Case Architecture Design



- Offers a **user interface** to actors
- Captures catalogue's data and validates it
- Captures bid's and auctions data
- Presents the notifications to actors

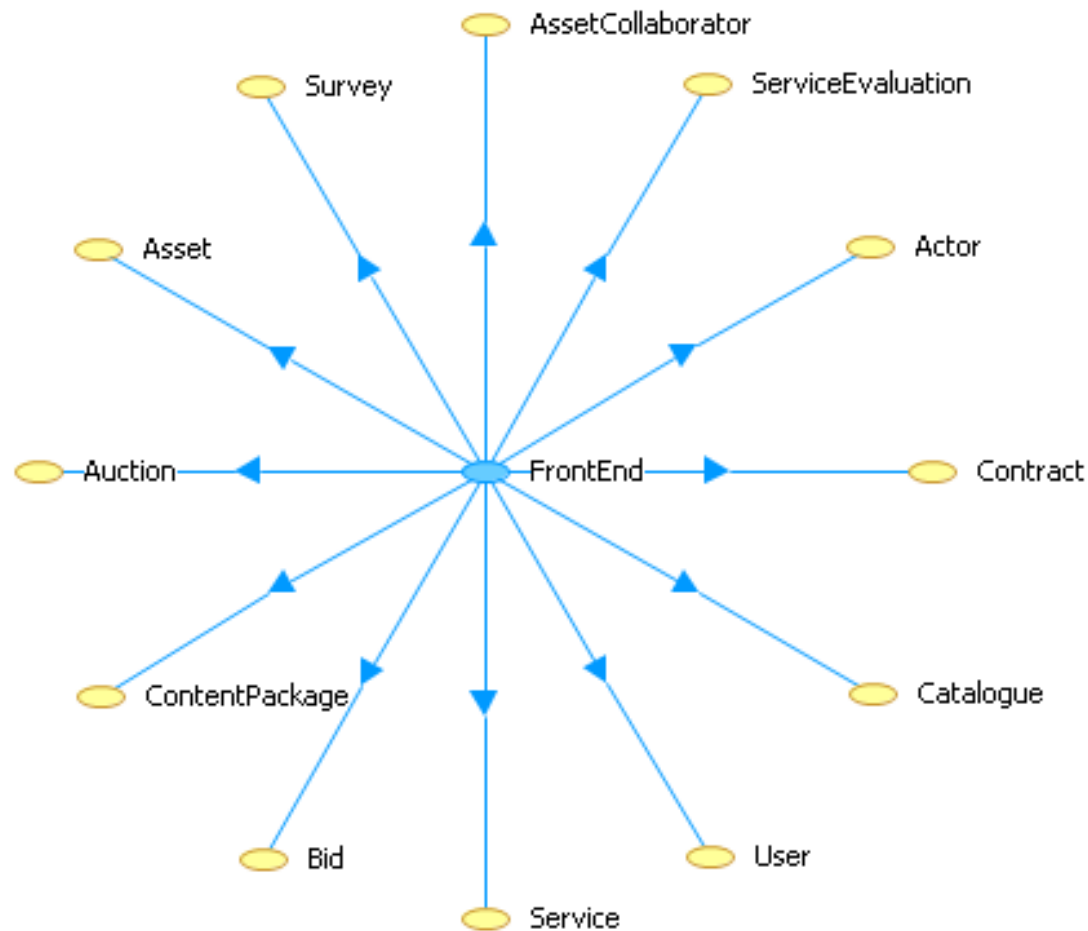
- **Abstracts the TS behaviour** to users
 - Allows changing the communication infrastructure
 - Makes TS dialogue transparent
- Contains the **business logic** (i.e: bid validation, auction rules,...)
 - Catalogue management (content and services)
 - User management
 - Auction management
 - Contract management
- **Translates** from user model to TS model

- Use Case Architecture
- **Ontologies**
- Interactions Between Actors
- Spaces Hierarchy
- Implementation Plan & Requirements

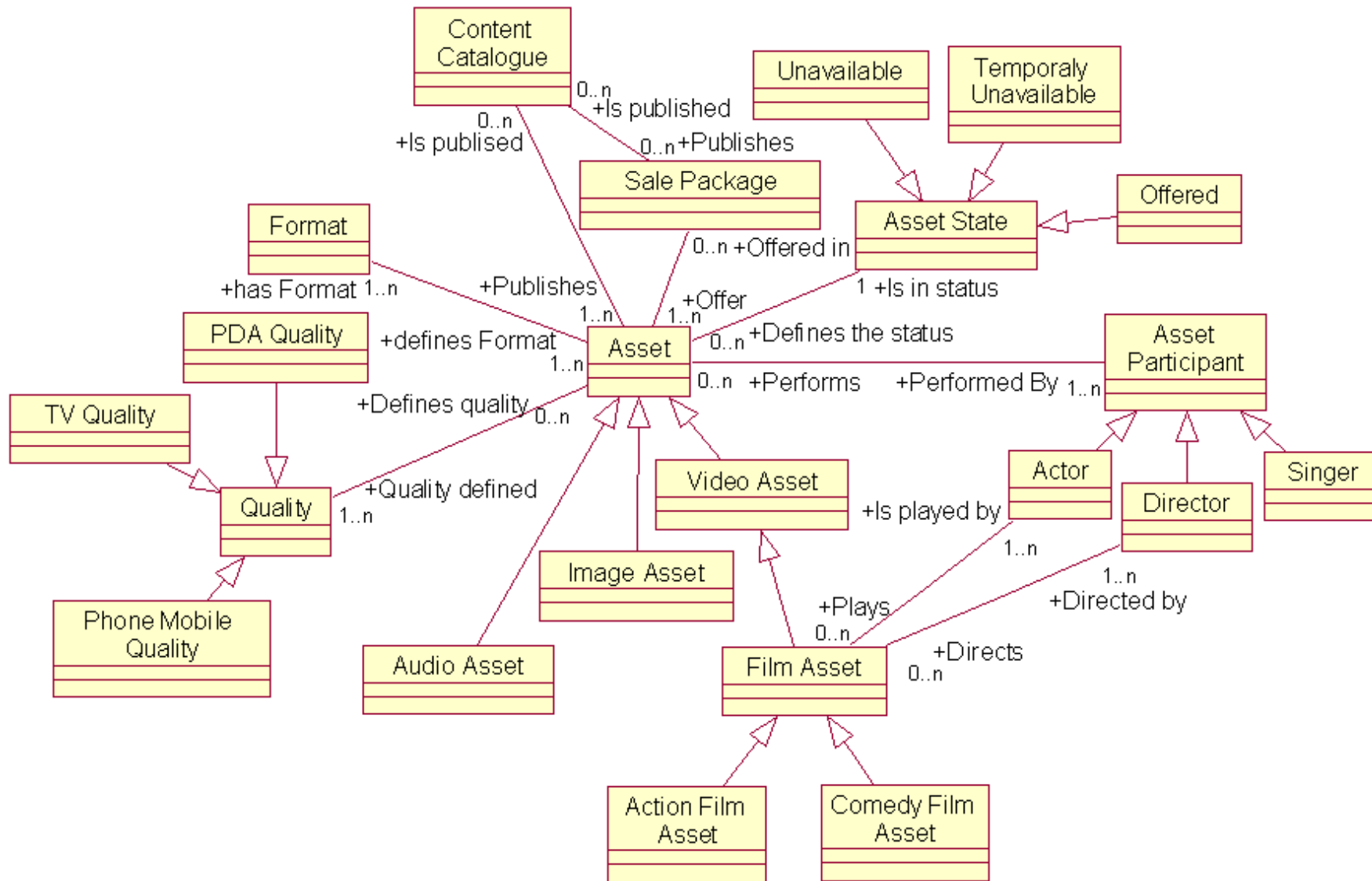
- User Model
 - Translation of the XSD supported by the Back-End's Web services
 - Simplified view of the model
 - Defined for mediation and coherence purposes

- TS Model
 - The “real” ontology
 - Designed for simple reasoning (i.e: genres as separate concepts).
 - Richer model for SPARQL queries

User Ontology

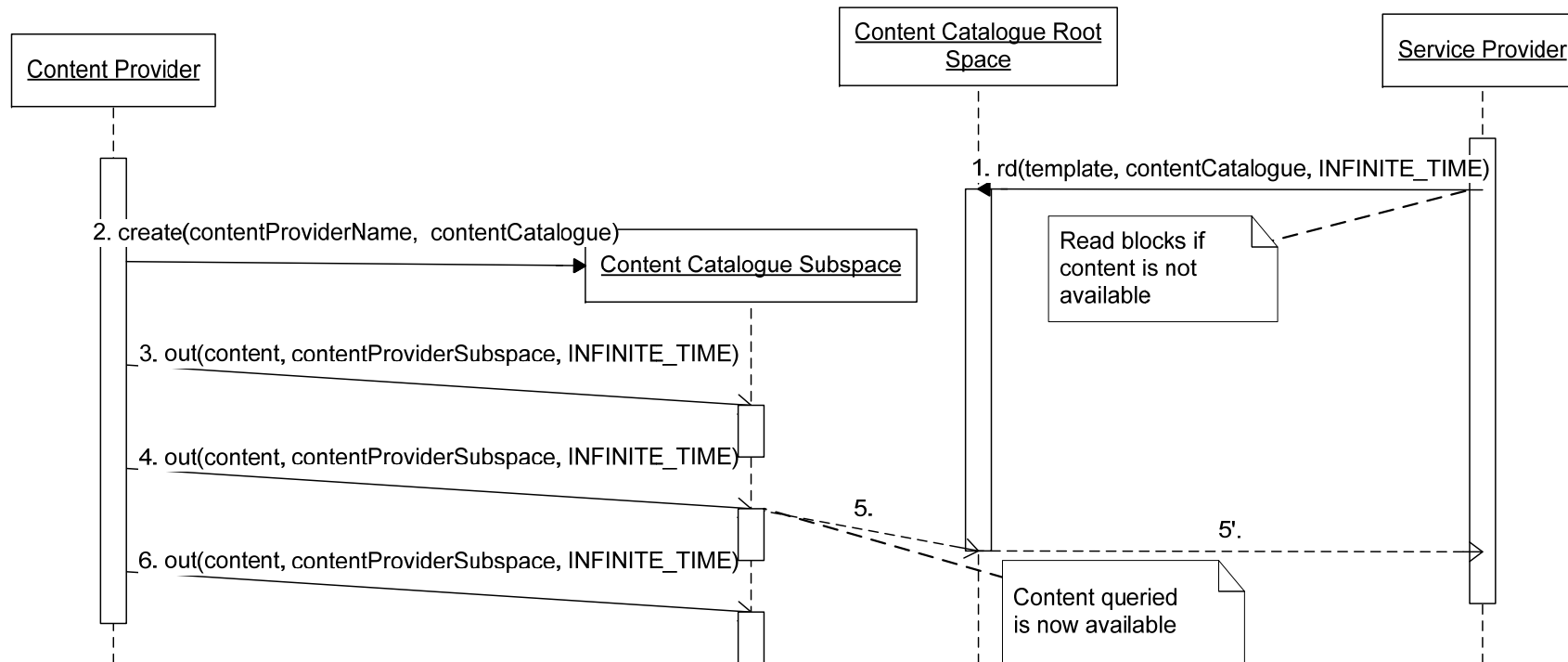


DAM Ontology Model

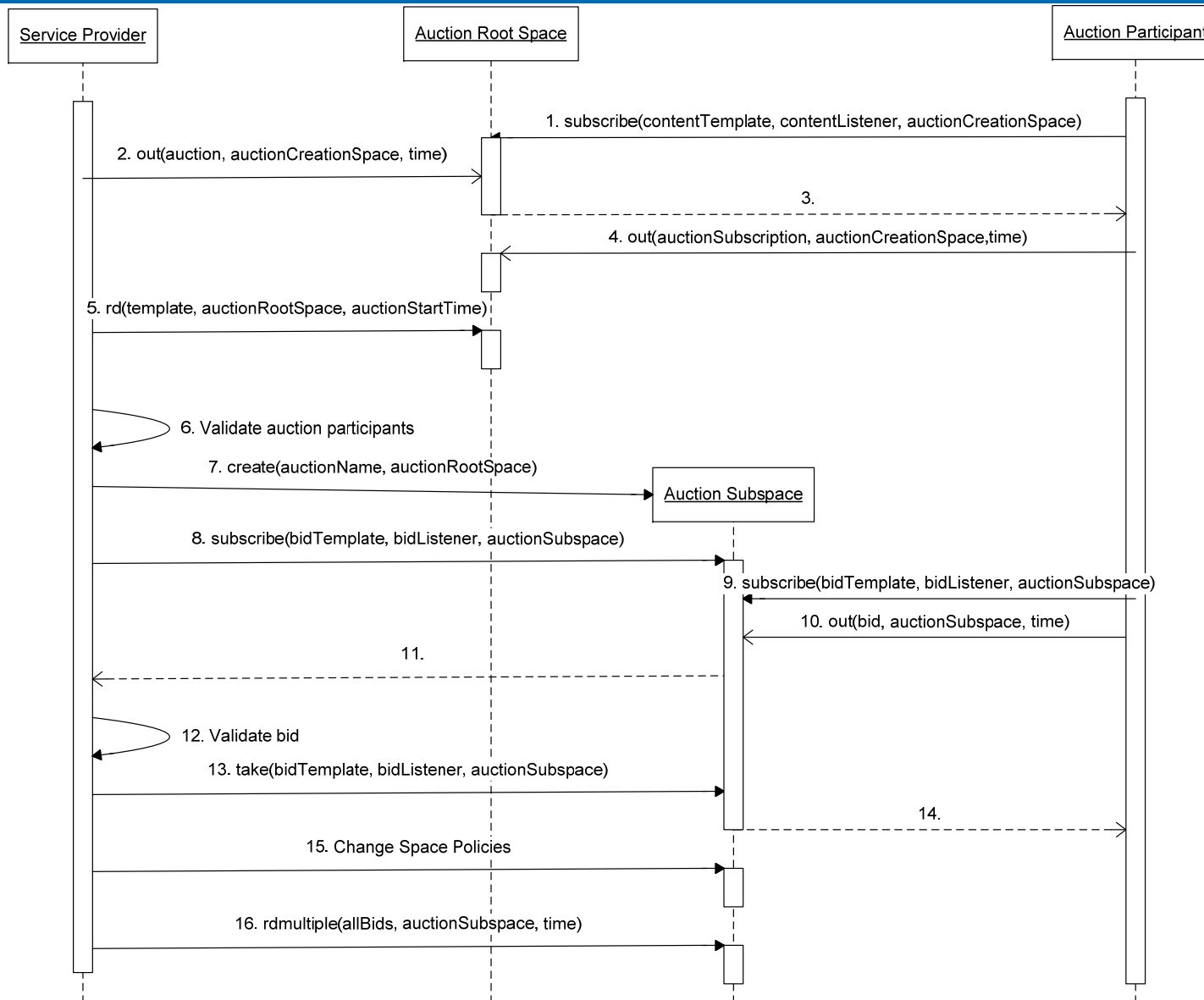


- Use Case Architecture
- Ontologies
- Interactions Between Actors
- Spaces Hierarchy
- Implementation Plan & Requirements

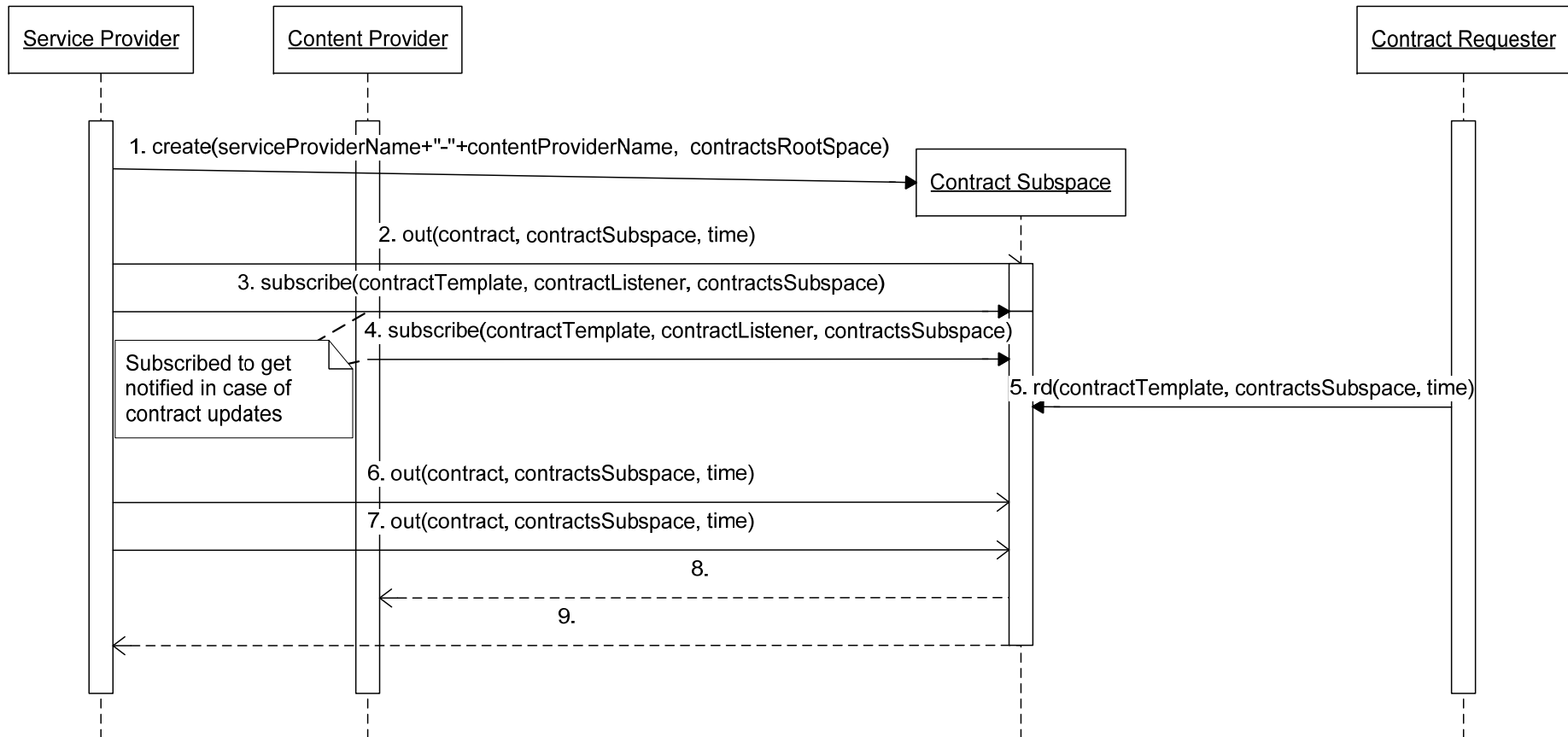
Content Catalogue Mng



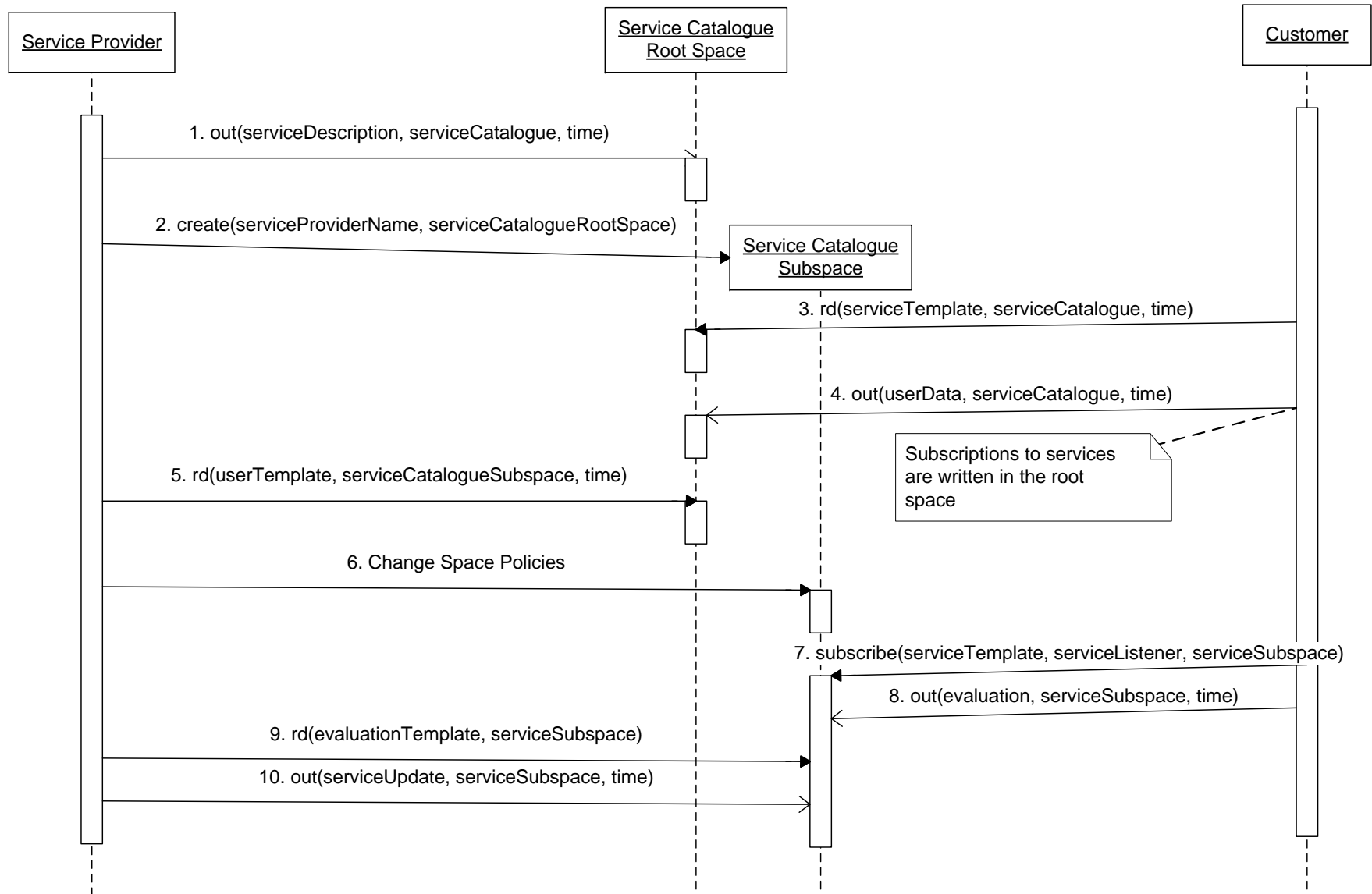
Auction Mng



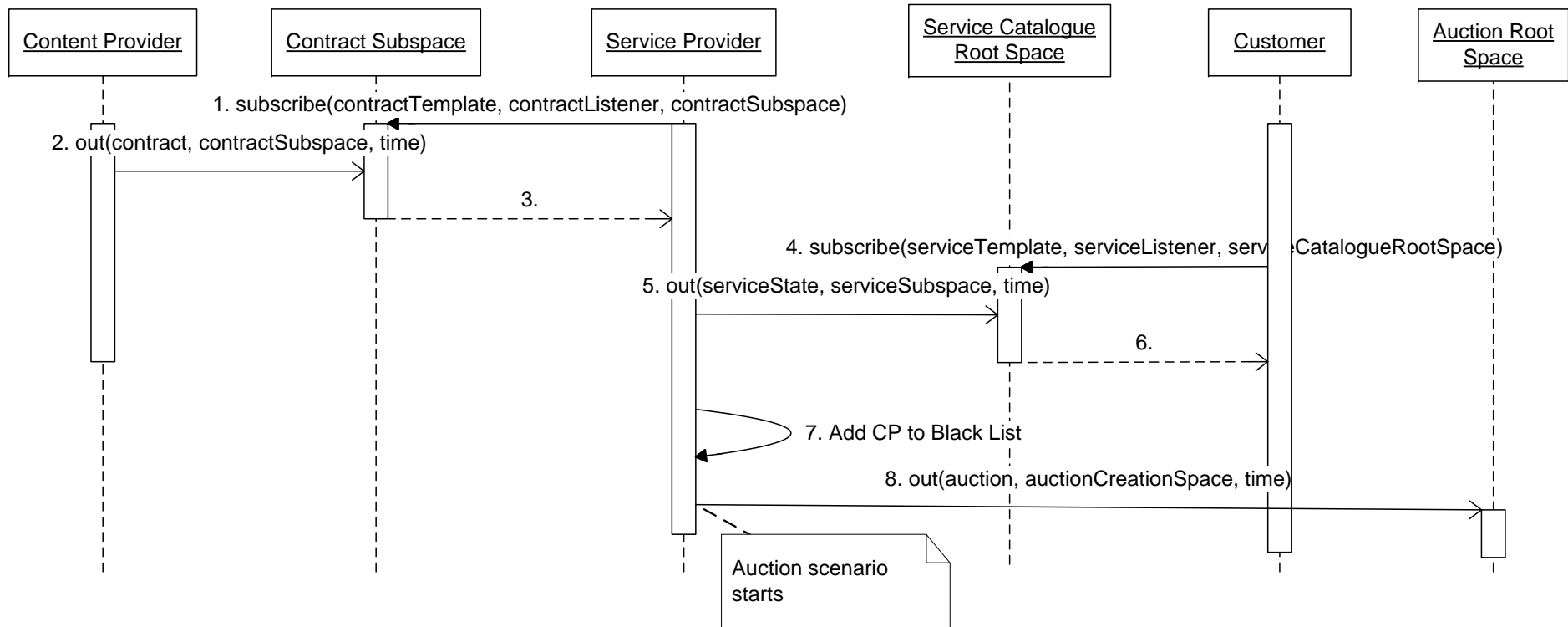
Contracts Mng



Services Mng

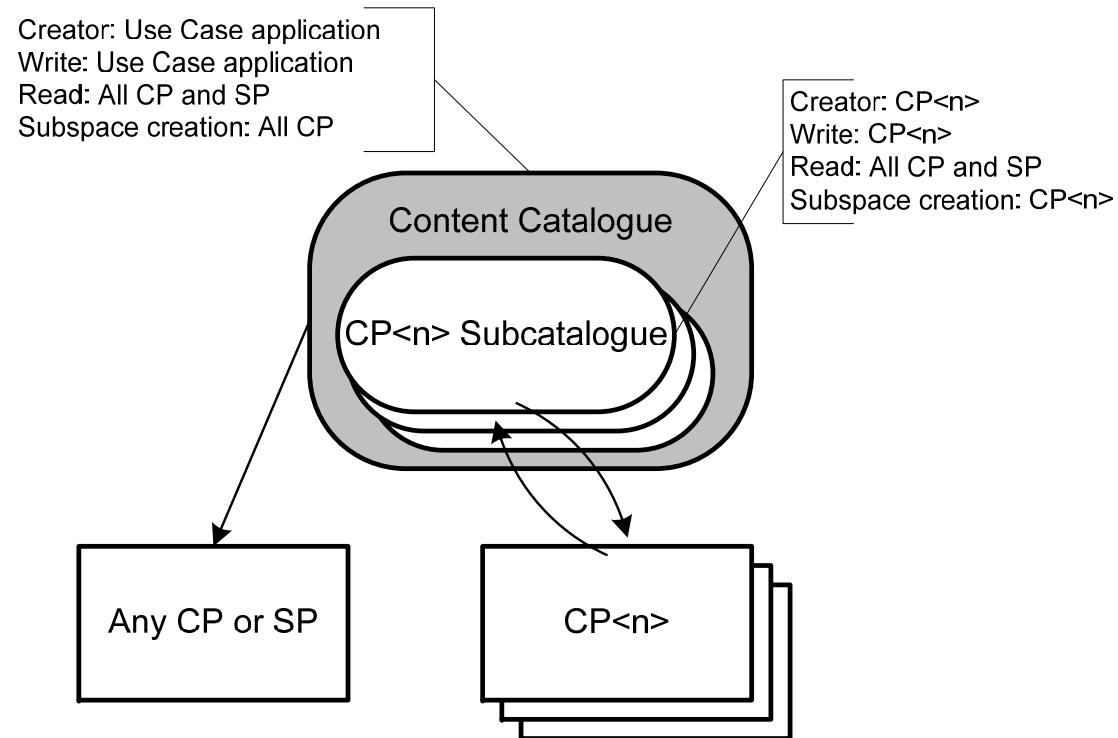


Content Unavailability

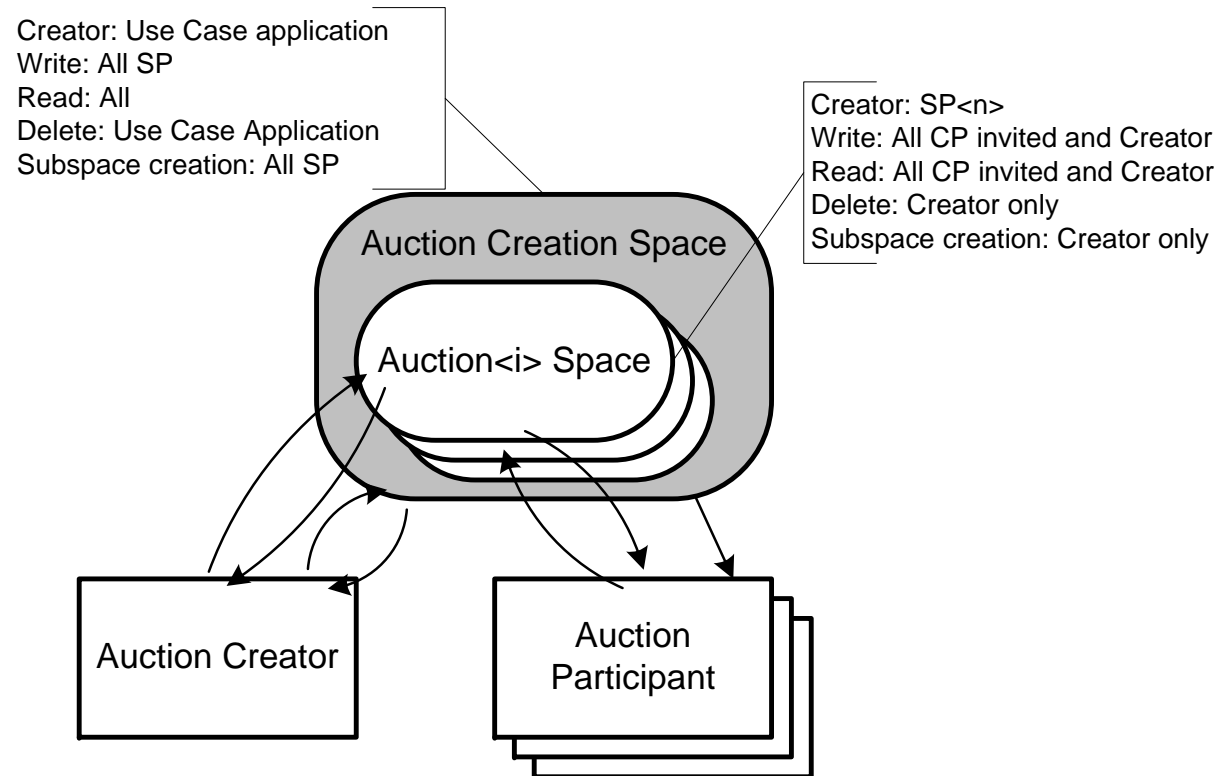


- Use Case Architecture
- Ontologies
- Interactions Between Actors
- Spaces Hierarchy
- Implementation Plan & Requirements

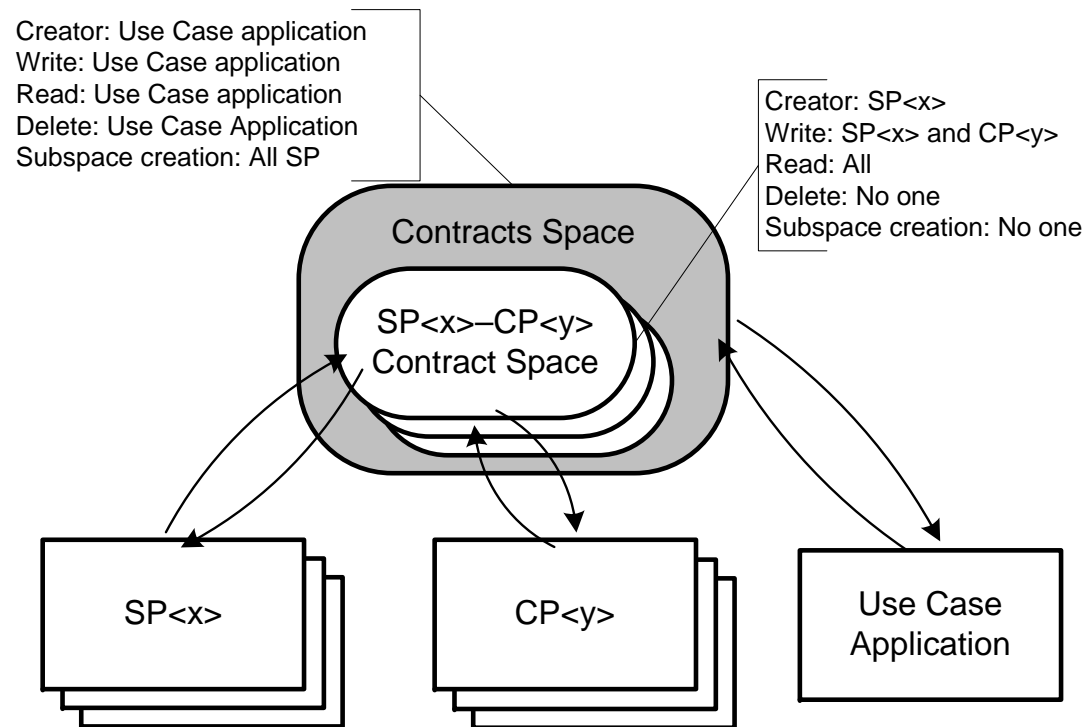
Content Catalogue Space Hierarchy



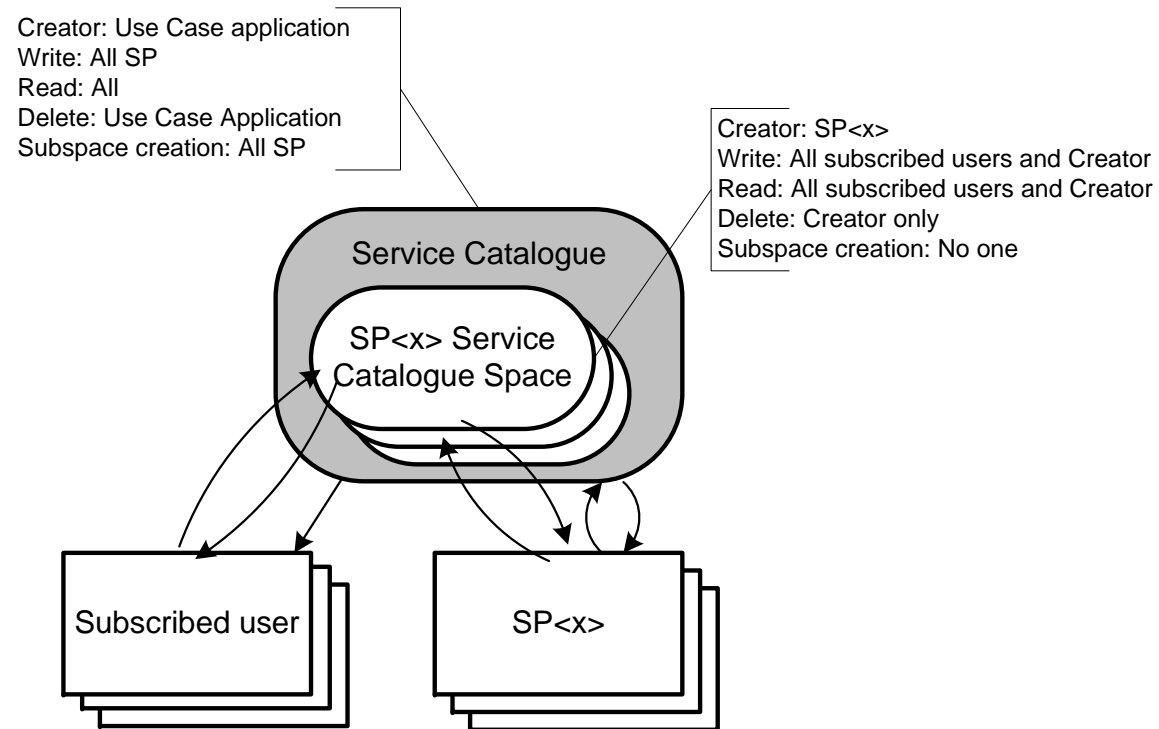
Auction Space Hierarchy



Contracts Space Hierarchy



Service Catalogue Space Hierarchy



- Use Case Architecture
- Ontologies
- Interactions Between Actors
- Spaces Hierarchy
- Implementation Plan & Requirements

- We have a set of DB for different content types
- For actors we will create data from the scratch (not a great amount of data)
- 2nd Version of the ontology available
- Ontotext to generate instances
 - Onto to provide sample queries
 - TID has to provide mappings from DB schemas to ontology attributes (document)
 - Onto to build and execute queries to generate instances?
- Deadline: End of January

- M28 First Implementation
 - Front-End Component
 - First Prototype ready by middle of January
 - Web portal to access functionalities
 - Other interfaces planned if possible (i.e: mobile)
 - Back-End Component
 - WSDL of Web services defined
 - Translations Java-RDF for WS invocation to be scheduled (inside WP4?)
 - Invocations to TS API to be defined (policies change as well)
 - Ontologies
 - First version available: User and DAM versions
 - NUIG will refine them and relate to EDI ontologies
 - Ontotext is generating instances from sample DB

- End of the project
 - Refinement of Use Case functionalities
 - Application for deployment purposes (M28 prototype refined)
 - Thin clients for validation purposes
 - Simulation of a real scenario
 - Performance and scalability tests
 - Objective: to derive some conclusions about TripCom's infrastructure which can be useful for exploitation purposes

- 2nd year review
 - A demo can't be focused on TS functionalities
 - Out and in functionalities won't allow us to show underlying business logic
 - Will SPARQL API will be ready?
 - We will be able to show part of the interface and planned functionality (we can simulate it)
 - Documentation or instructions to integrate first prototype with the use case implementation

- **Functionality that will be called by the Use Case**
 - **TS Functionality**
 - Extended querying mechanisms
 - Creation / deletion of spaces
 - Multiple reading/writing
 - Subscribe/notify
 - Completeness in a subspace is needed (i.e: return of all valid bids is critical)
 - Transactions would be desirable for realistic scenario
 - **Security**
 - API to define and change policies