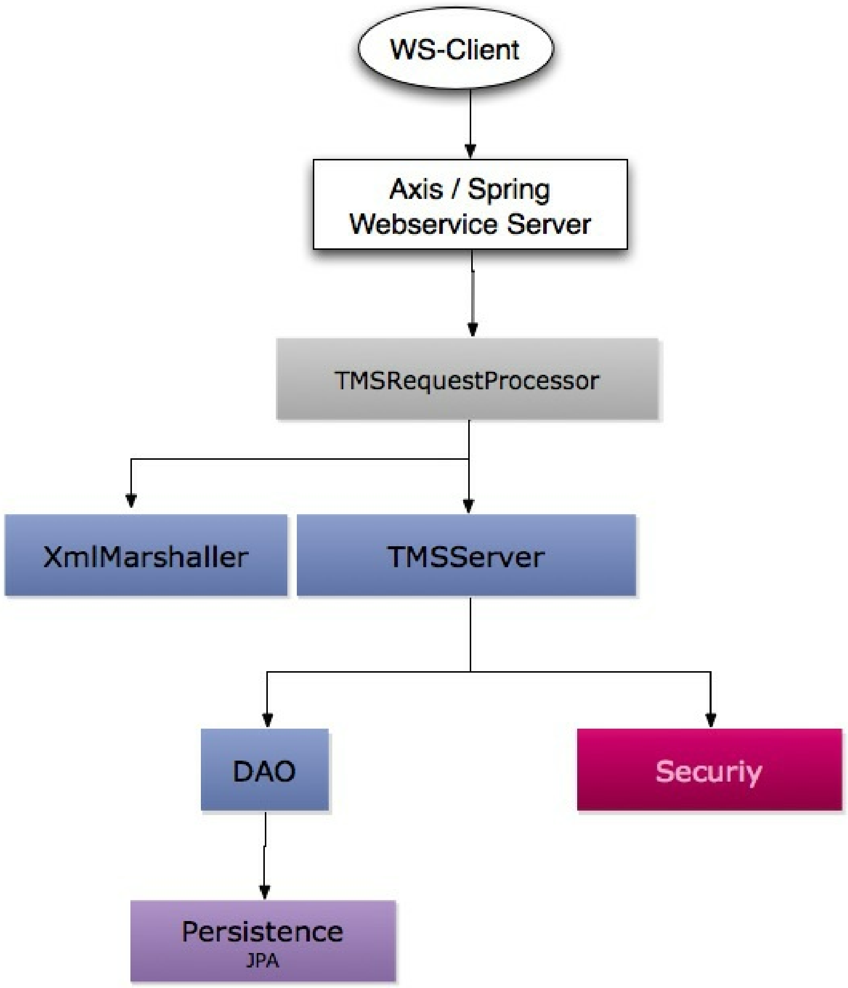
Tempo

INTALIO Tempo b4p branch document

Added by Ark Xu, last edited by Ark Xu on Aug 20, 2009

Architecture:



- TMSRequestProcessor
  TMSRequestProcessor is implementation of B4P Human Task Web service interface, which is strictly compliant with WS-HT spec version 1.0.(Except for xpath function)
  TMSRequestProcessor is deployed in tempo-tms.xml as "tms.serviceImplementation", and runing on axis2.
  All ws-interface implemented by TMSProcessor was declared in tms/src/main/axix2/service.xml
  What one interface impl of TMSRequestProcessor typically does is:
  1. check particitant token (please see limitation)
  2. umarshal ws request
  3. check request, throw IllegalArgumentException if failed
  4. call TMSServer to process request (operate task data, change status ...)
  5. marshal response
  6. convert to xml and return to client
- TMSServer
  Compared to TMSRequestProcessor that is respossible for processing request, TMSServer is responsible for implementing task operation logic.
  What one TMSServer method does:
  1. do authetication, get user/role info form participant token.
  2. check permisstion for specific task operation, according to spec.
  3. check current task status, throw IllegalStateException if it's not relevant to specifi task operation.
  4. call DAO, to update task data including task status. Throw B4PPersistException if failed.
- XmlMarshaller
  Located in tms-axis, generated by XMLBean tech.
  Responsible for unmarshal webservice request and marshal response.
- DAO
  Located in tms-service, responsible for operating task data.
  It will call openjpa entity class for task data.
- JPA
  JPA entity classes for data structure used in WS-HT, including task, attachment, status definition, comment and user/group OrgnizationEntity...

Scope and Limitation

1. Last phase of WS-HT dev implemented the task engine side functionality, defined in "Web Services Human Task (WS-HumanTask), Version 1.0 spec", including

1. Administrative operaton
2. Participant operation, (Except for getRenderingTypes and getRendering)
3. Simple query operation
4. Advanced query operation
   Tempo plays as Human Task Provider in constellation 4, described in b4p spec chapt 4.
   Tempo tms b4p dosen't substitute old tms task ws interface, it still works fine.

2. WS-Security
The WS-HT and b4p specification does not stipulate the authentication mechanism. We simply put participant token in soap header. Participant token is generated by tempo security.

2. ODE side
B4p extends the BEPEL schema to call human task, which needs to change ODE. This change dosn't happen in this version of tempo.

3. xpath function
xpath function is ODE related development, so is not done yet in this version.

4. Interoperable Protocol
Interoperable Protocol is responsible for communication between application and task engine, including creating task, skiping task ...
Interoperable Protocol is not implemented yet.
To work around, we use specific operation "create" to create task.

Build Tempo b4p branch

1. Get branch source code from http://github.com/intalio/tempo/tree/b4p
2. Type
   >buildr package

Test Tempo b4p ws-interface

1. buildr will run all unit test for b4p
2. After you deploy current tempo to bmpserver., you can test ws-ht interface through http://locahost:8080/ui-fw/b4ptest/test.html.

Comments