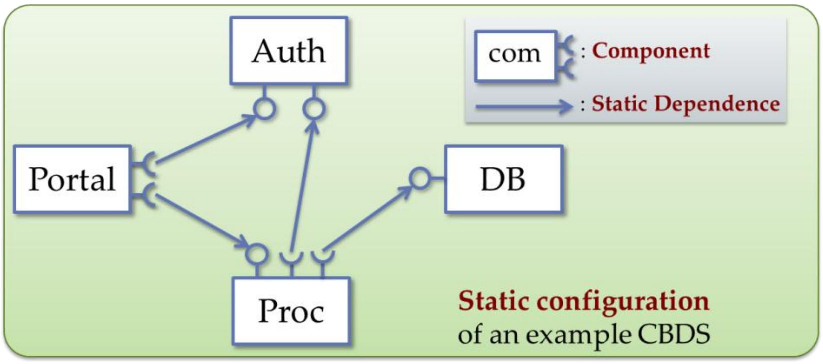
The following figure shows the static configuration of our component-based sample system:



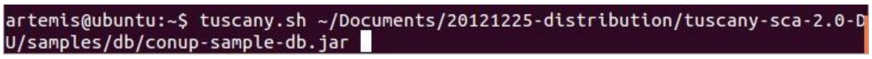
Static configuration of an example CBDS

To Run samples in our distribution (tuscany-sca-2.0-DU/samples) in your virtual machine, you need to make sure the following things:

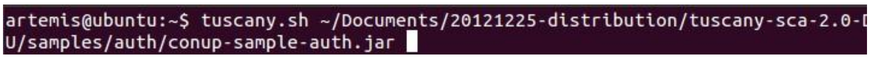
1. Setup your environment path

```
 set PATH=%PATH%:/home/xxxx/tuscany-sca-2.0-DU/bin
```

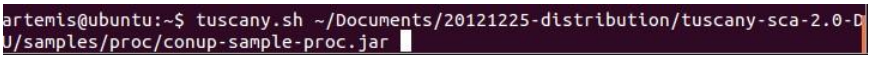
2. use NAT network, and make sure your virtual machine's ip address is 10.0.2.15, because our binding uri is this ip.
3. Start all four nodes in your terminals like the following figures.



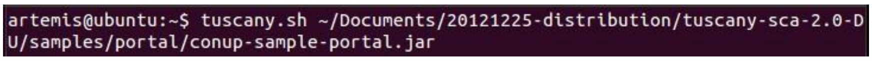
```
artemis@ubuntu:~$ tuscany.sh ~/Documents/20121225-distribution/tuscany-sca-2.0-D
U/samples/db/conup-sample-db.jar
```



```
artemis@ubuntu:~$ tuscany.sh ~/Documents/20121225-distribution/tuscany-sca-2.0-D
U/samples/auth/conup-sample-auth.jar
```

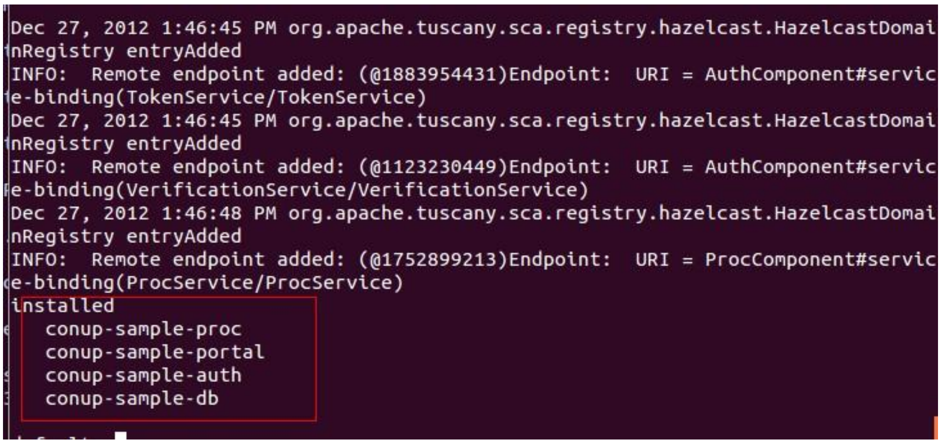


```
artemis@ubuntu:~$ tuscany.sh ~/Documents/20121225-distribution/tuscany-sca-2.0-D
U/samples/proc/conup-sample-proc.jar
```



```
artemis@ubuntu:~$ tuscany.sh ~/Documents/20121225-distribution/tuscany-sca-2.0-D
U/samples/portal/conup-sample-portal.jar
```

4. After all nodes are started, we can type installed in terminal to see which nodes we have started



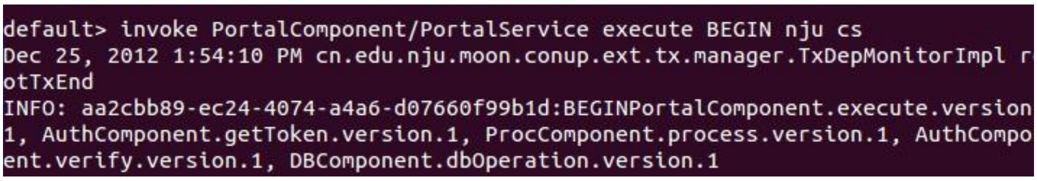
```
Dec 27, 2012 1:46:45 PM org.apache.tuscany.sca.registry.hazelcast.HazelcastDomai
nRegistry entryAdded
INFO:  Remote endpoint added: (@1883954431)Endpoint:  URI = AuthComponent#servic
e-binding(TokenService/TokenService)
Dec 27, 2012 1:46:45 PM org.apache.tuscany.sca.registry.hazelcast.HazelcastDomai
nRegistry entryAdded
INFO:  Remote endpoint added: (@1123230449)Endpoint:  URI = AuthComponent#servic
e-binding(VerificationService/VerificationService)
Dec 27, 2012 1:46:48 PM org.apache.tuscany.sca.registry.hazelcast.HazelcastDomai
nRegistry entryAdded
INFO:  Remote endpoint added: (@1752899213)Endpoint:  URI = ProcComponent#servic
e-binding(ProcService/ProcService)
installed
   conup-sample-proc
   conup-sample-portal
   conup-sample-auth
   conup-sample-db
```

5. Invoke services

a. Invoke PortalComponent's PortalService in portal node's terminal using the following command.

invoke PortalComponent/PortalService execute BEGIN nju cs

"execute" is the service method name; "BEGIN, nju, cs" are the three parameters that are required by this method. In our example, we will display all execution path in our result string like the following figure.



```
default> invoke PortalComponent/PortalService execute BEGIN nju cs
Dec 25, 2012 1:54:10 PM cn.edu.nju.moon.conup.ext.tx.manager.TxDepMonitorImpl r
otTxEnd
INFO: aa2cbb89-ec24-4074-a4a6-d07660f99b1d:BEGINPortalComponent.execute.version
1, AuthComponent.getToken.version.1, ProcComponent.process.version.1, AuthCompo
ent.verify.version.1, DBComponent.dbOperation.version.1
```

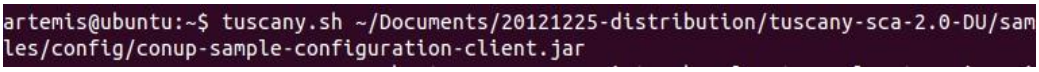
INFO: aa2cbb89-ec24-4074-a4a6-d07660f99b1d:BEGINPortalComponent.execute.version.1, AuthComponent.getToken.version.1, ProcComponent.process.version.1, AuthComponent.verify.version.1, DBComponent.dbOperation.version.1

The Green color string is root transaction id; the blue color string is our input parameter which is used to identify the start of execution; the black color string is our execution path, in the execution path we will also display the implementation's version

In every component implementation, we have added a variable which was used to identify implementation's version number. We use this number to test whether our algorithm is right or not.

b. Ondemand Setup

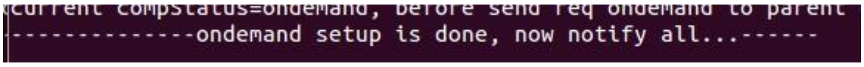
if you want to set up these dynamic dependencies, you can use our configuration client to set up.



```
artemis@ubuntu:~$ tuscany.sh ~/Documents/20121225-distribution/tuscany-sca-2.0-DU/sam
les/config/conup-sample-configuration-client.jar
```

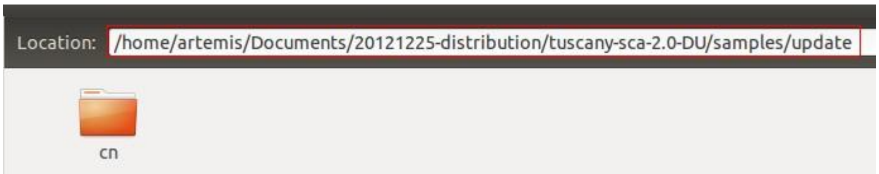
invoke ConfigComponent/ConfService ondemand

For simplicity, here we send ondemand command to AuthComponent. When finish ondemand setup, the following figure will be display in your terminal.



```
current compStatus=ondemand, before send req ondemand to parent
--------------ondemand setup is done, now notify all...------
```

c. Update your component. in paper example, we update AuthComponent, we put our new version of the component implementation in one folder like following figure:

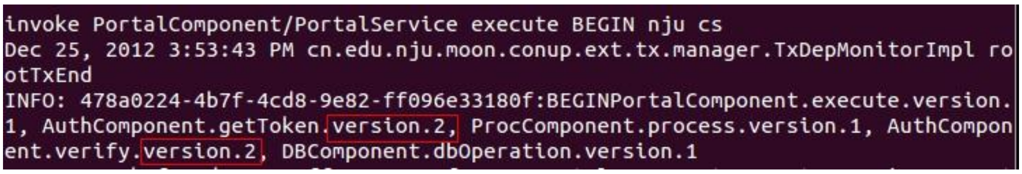


Location: /home/artemis/Documents/20121225-distribution/tuscany-sca-2.0-DU/samples/update

Attention: here the new version of the component implementation is in the form of a .class file with its package name sub-folder.

invoke ConfigComponent/ConfService update /home/artemis/Documents/20121225-distribution/tuscany-sca-2.0-DU/samples/update

For simplicity, here we send update command to AuthComponent. when you finish updating, you can invoke Portal service to test whether your update is right.

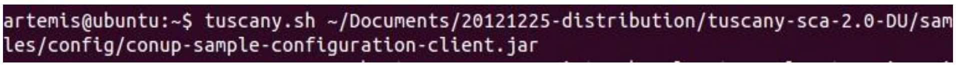


```
invoke PortalComponent/PortalService execute BEGIN nju cs
Dec 25, 2012 3:53:43 PM cn.edu.nju.moon.conup.ext.tx.manager.TxDepMonitorImpl ro
otTxEnd
INFO: 478a0224-4b7f-4cd8-9e82-ff096e33180f:BEGINPortalComponent.execute.version.
1, AuthComponent.getToken.version.2, ProcComponent.process.version.1, AuthCompon
ent.verify.version.2, DBComponent.dbOperation.version.1
```

6. We have implement a service access tool to help you make some tests on our algorithm.

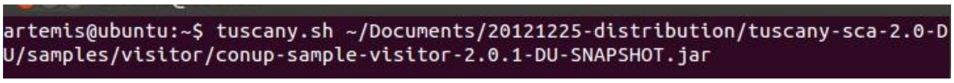
   a. Start four nodes: portal, proc, auth, db

   b. Start config node:



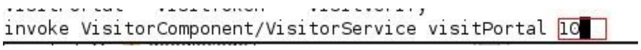
```
artemis@ubuntu:~$ tuscany.sh ~/Documents/20121225-distribution/tuscany-sca-2.0-DU/sam
les/config/conup-sample-configuration-client.jar
```

   c. Start service visitor node:



```
artemis@ubuntu:~$ tuscany.sh ~/Documents/20121225-distribution/tuscany-sca-2.0-D
U/samples/visitor/conup-sample-visitor-2.0.1-DU-SNAPSHOT.jar
```

After start this node, you can specify the time of accessing specify service like the following figure: the last parameter is the times that you want to access.



```
invoke VisitorComponent/VisitorService visitPortal 10
```

During the service invoking, you can use config node to send update request to AuthComponent.

If you want to use different algorithm and strategy, you can change them in Conup.xml which locates in bin folder in our distribution.