



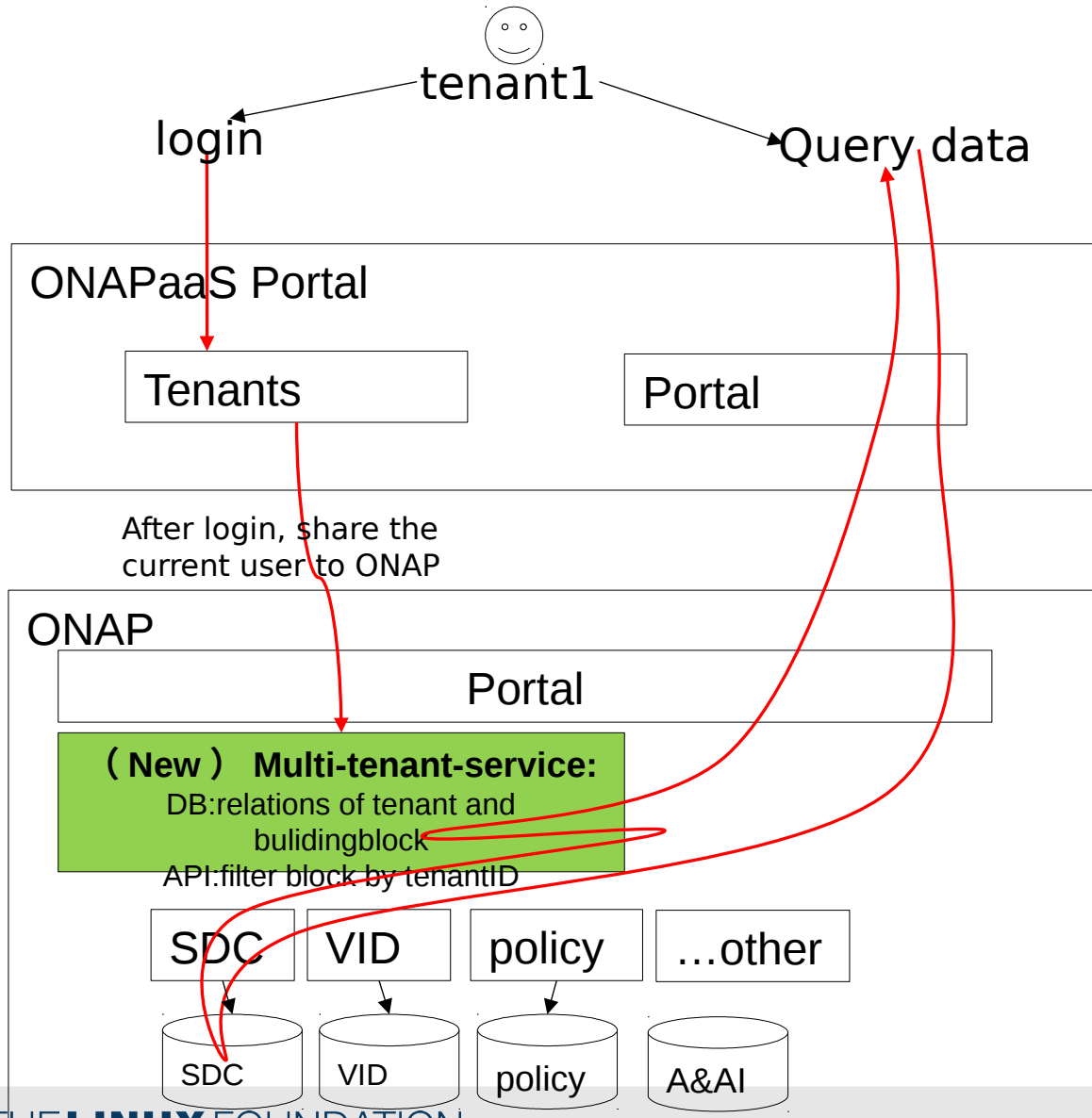
ONAP

OPEN NETWORK AUTOMATION PLATFORM

Multi-tenant for ONAP

Beiyuan Zhang
zhangbeiyuan@huawei.com
Seshu Kumar M
Seshu.kumar.m@huawei.com

Multi-tenant for ONAP



Merit	The change is thin and controllable, which does not affect the original functions.
Flaws	Basic data is not isolated and may conflict. For example, if a tenant creates a block with the same name, it will be identified as the same name in the ONAP system.

Change 1: The Multi-tenant-service records the home tenants of all data and provides interfaces for adding, deleting, modifying, and querying data.

Change 2: When data is accessed on all pages, data is obtained based on the original process and then filtered based on tenantID.

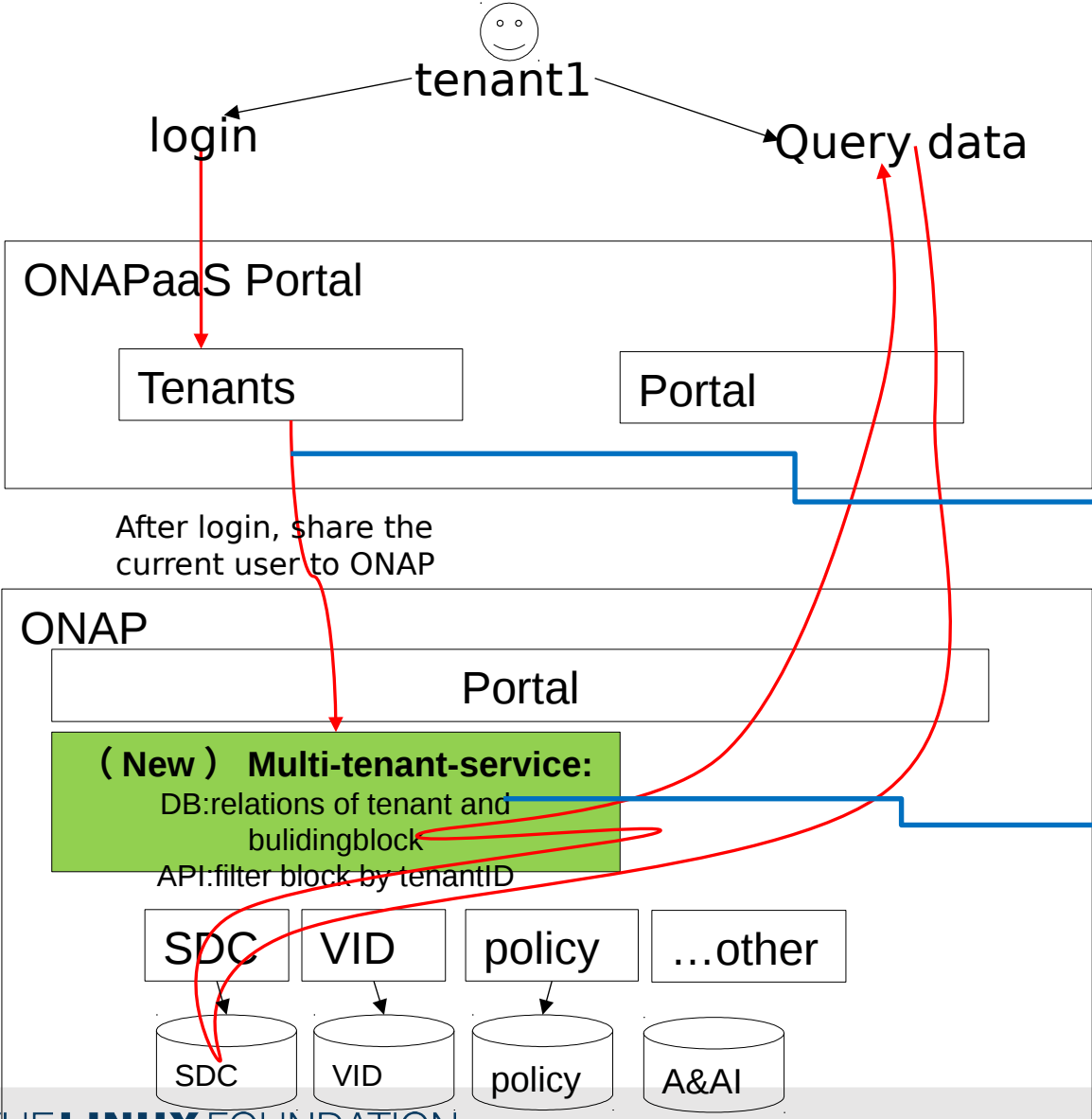
Change 3: When data is added or deleted, the Multi-tenant-service interface needs to be invoked to update the relationship between data and tenants.

Key Point :

1. Add the `tenant_id` to session, let ONAP to know who is login.
2. Add a table to record the relationship of `tenant_id` and `buildingblock_id`, so 2 step should be done:
 - step1: when create a buildingblock, you should record the relationship into the table;
 - step2: when you want to get the buildingblocks from interface, and then use the relationships to filter the data what you got from DB

We used SDC and AAI to verify the scheme

On SDC: Two points should be done



Insert test code in two positions

1, Simulate tenant user login:
In the login interface of the simulate, add tenant to the cookie, **and need to write the session in the future**

2, insert test API
Find the query data interface in the SDC code, and filter the data according to the tenant-Id in the cookie, and need to be transplanted into the multi-tenant component in the future;
The relationship between Element and tenant is temporarily saved in memory and needs to be written to the database in the future.

On SDC: Insert tenant_id with simulation

SDC can simulate the login by simulation, modify the original login interface, add the tenant attribute and save it to the cookie.

← → ↻ ⓘ 不安全 | 192.168.2.173:8285/login

Webseal simulator

Login:

Tenant id:

User id:

Password:

Quick links:

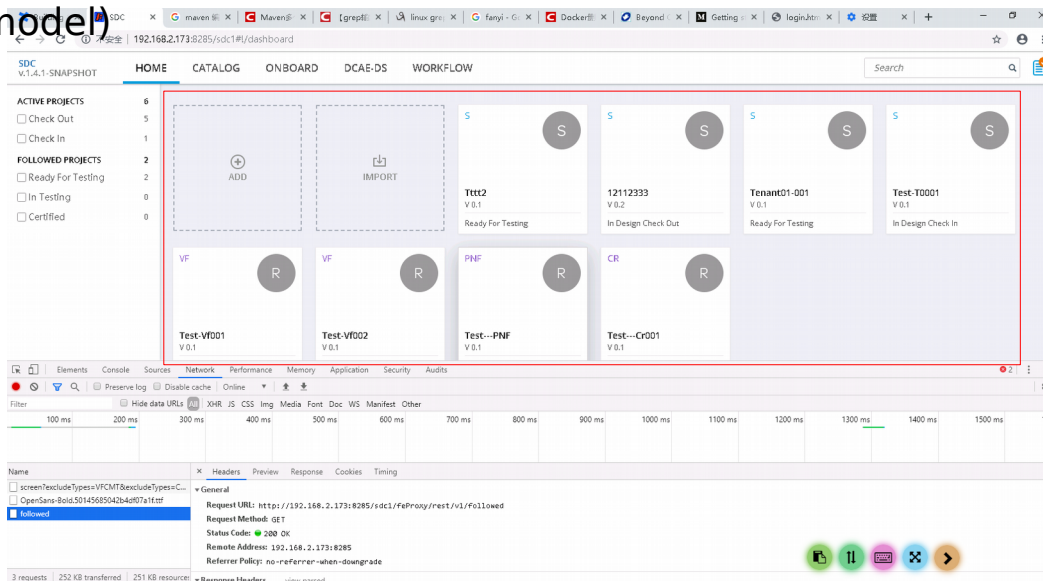
full name	user id	role	action
Jimmy Hendrix	jh0003	Admin	create
Carlos Santana	cs0008	Designer	create
Johnny Depp	jm0007	Tester	create
Aretha Franklin	op0001	Ops	create
David Shadmi	gv0001	Governor	create
Create All			

Login by **Tenant Id** + User Id: Could operate the data of the tenant only
User Id:cs0008 or jm0007
Password:123123a

Login by User Id: Could operate all of data

SDC Home Data Filtering

Step1: find the API: rest/v1/followed
(Use Chrome to open the SDC page, and open developer-model)



Step2: To find the code of the API from SDC code by the path.

```
rest/v1/followed that code is catalog-be-ElementServlet.java  
ConsumerServlet 472 // retrieve all followed resources and services  
CsarBuildServlet 473  
DistributionServiceS 474 @GET  
ElementServlet 475 @Path("/followed")  
GenericArtifactBrow 476 @Consumes(MediaType.APPLICATION_JSON)  
GroupEndpoint 477 @Produces(MediaType.APPLICATION_JSON)  
GroupServlet 478 @ApiOperation(value = "Retrieve all followed", httpMethod = "GET", notes = "Retrieve all followed", res  
GroupTypesEndpoi 479 @ApiResponses(value = { @ApiResponse(code = 200, message = "Returns followed Ok"), @ApiResponse(code =  
InputsServlet 480 @ApiResponse(code = 500, message = "Internal Server Error") })  
InterfaceOperatio 481  
LifecycleServlet 481 public Response getFollowedResourcesServices(@Context final HttpServletRequest request, @HeaderParam(v
```

Step3: Insert this code to filter the result(detail in next page)

```
496  
497 if (either.left().value() != null) {  
498     userData = either.left().value();  
499     Either<Map<String, List<? extends Component>>, ResponseFormat> followedResourcesServices = getElementBL(request)  
500     if (followedResourcesServices.isRight()) {  
501         Log.debug("failed to get followed resources services");  
502         return buildErrorResponse(followedResourcesServices.right().value());  
503     }  
504     // TODO test start: to filter resource by tenant, and tenant id is saved in session.  
505     followedResourcesServices = filterElementByTenantId(request, followedResourcesServices);  
506     // TODO test end.  
507     Object data = RepresentationUtils.toRepresentation(followedResourcesServices.left().value());  
508     res = buildOkResponse(getComponentsUtils().getResponseFormat(ActionStatus.OK), data);  
509 } else {  
510     res = buildErrorResponse(getComponentsUtils().getResponseFormat(ActionStatus.GENERAL_ERROR));  
511 }
```

SDC Code modification to achieve data filtering

GenericServlet. getFollowedResourcesServices()

```
496 if (either.left().value() != null) {
497     userData = either.left().value();
498     Either<Map<String, List<? extends Component>>, ResponseFormat> followedResourcesServices = getElementBL(request);
499     if (followedResourcesServices.isRight()) {
500         Log.debug("failed to get followed resources services ");
501         return buildErrorResponse(followedResourcesServices.right().value());
502     }
503     // TODO test start: to filter resource by tenant, and tenant id is saved in session.
504     followedResourcesServices = filterElementByTenantId(request, followedResourcesServices);
505     // TODO test end.
506     Object data = RepresentationUtils.toRepresentation(followedResourcesServices.left().value());
507     res = buildOkResponse(getComponentsUtils().getResponseFormat(ActionStatus.OK), data);
508 } else {
509     res = buildErrorResponse(getComponentsUtils().getResponseFormat(ActionStatus.GENERAL_ERROR));
510 }
```

GenericServlet. filterElementByTenantId()

```
549 // TODO test start: when get date from DB, to filter by tenant id
550 protected Either<Map<String, List<? extends Component>>, ResponseFormat> filterElementByTenantId(@Context HttpServletRequest request,
551     Either
552
553     String tenantId = getTenantIdFromCookie(request);
554     if (tenantId == null || tenantId.isEmpty())
555     {
556         // if no tenant id, that will return all of data
557         return resources;
558     }
559     MultiTenantService multiserver = new MultiTenantService();
560     Either<Map<String, List<? extends Component>>, ResponseFormat> filtered = multiserver.filerResultByTenantId(resources, tenantId);
561     return filtered;
562 }
```

MultiTenantService.filerResultByTenantId()

```
21 @ public Either<Map<String, List<? extends Component>>, ResponseFormat> filerResultByTenantId(Either<Map<String, List<? extends Component>>,
22     ResponseFormat> resources, String tenantId)
23 {
24     if (resources.isRight())
25     {
26         return resources;
27     }
28     for (Map.Entry<String, List<? extends Component>> entry : resources.left().value().entrySet())
29     {
30         String key = entry.getKey();
31         List<? extends Component> values = entry.getValue();
32         if (values == null || values.isEmpty())
33         {
34             continue;
35         }
36         Iterator<? extends Component> iterator = values.iterator();
37         while (iterator.hasNext())
38         {
39             Object obj = iterator.next();
40             Component component = (Component)obj;
41             String uuid = component.getUUID();
42             if (!isComparedWithId(uuid, tenantId))
43             {
44                 iterator.remove();
45             }
46         }
47     }
48     return resources;
49 }
```

Filter data by tenant id

Test Result: Using two tenant-id to create services

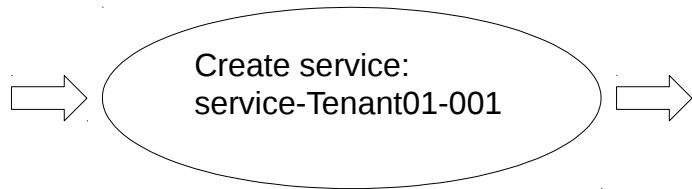
Login with **tenant01** + cs0008:

Login:

Tenant id:

User id:

Password:



SDC v.1.4.1-SNAPSHOT HOME CATALOG ONBOARD DCAE-DS WORKFLOW

ACTIVE PROJECTS 1

- Check Out 1
- Check In 0

FOLLOWED PROJECTS 0

- Ready For Testing 0
- In Testing 0
- Certified 0

ADD IMPORT

Service-Tenant01-001 V 0.1 In Design Check Out

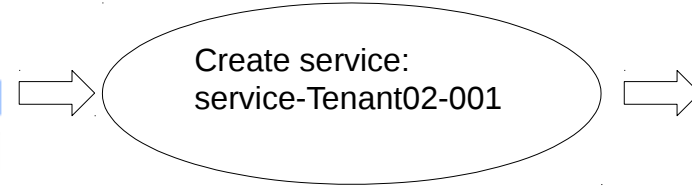
Login with **tenant02** + cs0008:

Login:

Tenant id:

User id:

Password:



SDC v.1.4.1-SNAPSHOT HOME CATALOG ONBOARD DCAE-DS WORKFLOW

ACTIVE PROJECTS 1

- Check Out 1
- Check In 0

FOLLOWED PROJECTS 0

- Ready For Testing 0
- In Testing 0
- Certified 0

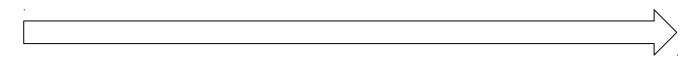
ADD IMPORT

Service-Tenant02-001 V 0.1 In Design Check Out

Login with cs0008:

Quick links:

full name	user id	role	action
Jimmy Hendrix	jh0003	Admin	create
Carlos Santana	cs0008	Designer	create
Johnny Depp	jim0007	Tester	create
Aretha Franklin	op0001	Ops	create
David Shadmi	gv0001	Governor	create



SDC v.1.4.1-SNAPSHOT HOME CATALOG ONBOARD DCAE-DS WORKFLOW

ACTIVE PROJECTS 2

- Check Out 2
- Check In 0

FOLLOWED PROJECTS 0

- Ready For Testing 0
- In Testing 0
- Certified 0

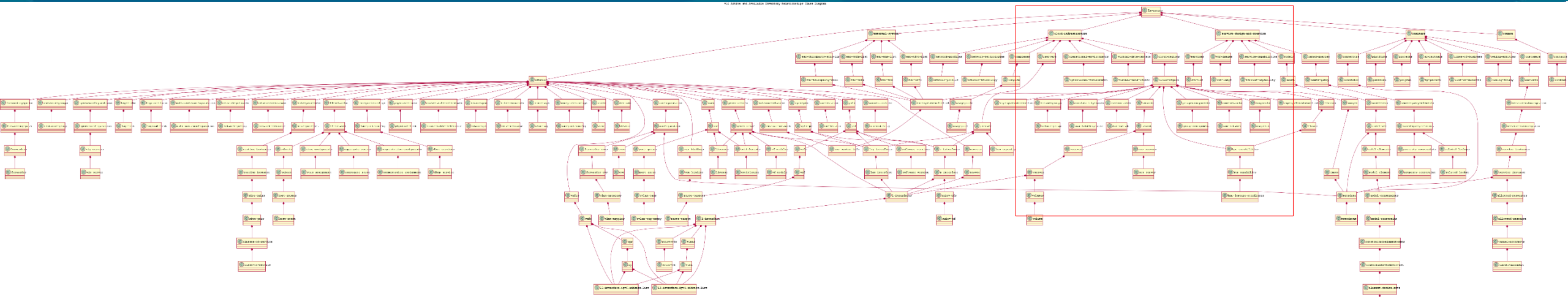
ADD IMPORT

Service-Tenant01-001 V 0.1 In Design Check Out

Service-Tenant02-001 V 0.1 In Design Check Out

So, SDC is easy to do, but AAI is very difficult to change, To be continued

AAI code analysis ---resource relationship



The main code is in aai-resource:
aai-oxm-v16.xml: define all API interfaces

DbedgeRules_v16.json define the relationship between nodes

```
<xml-element container-type="java.util.Array...
</java-attributes>
<xml-properties...
<xml-property name="maximumDepth" value="0"/...
</xml-properties>
</java-type>
<java-type name="CloudRegion">
  <xml-root-element name="cloud-region"/>
  <java-attributes...>
  <xml-properties...>
  </java-type>
<java-type name="VolumeGroups">
  <xml-properties...>
  <xml-property name="description" value="Coll...>
  </xml-properties>
  <xml-root-element name="volume-groups"/>
  <java-attributes...>
  <xml-element container-type="java.util.Array...>
  </java-attributes>
</java-type>
```

```
"description":""
},
{
  "from": "tunnel-xconnect",
  "to": "allotted-resource",
  "label": "org.onap.relationships.inventory.BelongsTo",
  "direction": "OUT",
  "multiplicity": "ONE2ONE",
  "contains-other-v": "!${direction}",
  "delete-other-v": "!${direction}",
  "prevent-delete": "NONE",
  "default": "true",
  "description":""
},
{
  "from": "availability-zone",
  "to": "complex",
  "label": "org.onap.relationships.inventory.LocatedIn",
  "direction": "OUT",
  "multiplicity": "MANY2ONE",
  "contains-other-v": "NONE",
```

AAI - @get interface analysis

Interface entry

```
@GET
@Path("/{uri: .+}")
@Consumes({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
@Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
public Response getLegacy (String content, @DefaultValue("-1") @QueryParam("resultIndex") String resultIndex, @QueryParam("depthParam") String depthParam, @QueryParam("cleanUp") String cleanUp, @QueryParam("header") String header) {
    return runner(AAIConstants.AAI_CRUD_TIMEOUT_ENABLED,
        AAIConstants.AAI_CRUD_TIMEOUT_APP,
        AAIConstants.AAI_CRUD_TIMEOUT_LIMIT,
        headers,
        info,
        HttpMethod.GET,
        (AaiCallable) () -> {
            return getLegacy(content, versionParam, uri, depthParam, cleanUp, header);
        });
}
```

Get tenant_id from header:

```
261 */
262 @ public Response getLegacy(String content, String versionParam, String uri, String
263 String sourceOfTruth = headers.getRequestHeaders().getFirst( k: "X-FromAppId");
264 String transId = headers.getRequestHeaders().getFirst( k: "X-TransactionId");
265 String realTime = headers.getRequestHeaders().getFirst( k: "Real-Time");
266 String tenantId = headers.getRequestHeaders().getFirst( k: "Tenant-Id");
267 Response response = null;
268 TransactionalGraphEngine dbEngine = null;
269 Loader loader = null;
```

To query and filter data :

```
301 DBRequest request =
302     new DBRequest.Builder(HttpMethod.GET, uriObject, uriQuery, obj, headers, info, transId).build();
303 List<DBRequest> requests = new ArrayList<>();
304 requests.add(request);
305 if (resultIndex != null && resultIndex != "-1" && resultSize != null && resultSize != "-1") {
306     traversalUriHttpEntry.setPaginationIndex(Integer.parseInt(resultIndex));
307     traversalUriHttpEntry.setPaginationBucket(Integer.parseInt(resultSize));
308 }
309 Pair<Boolean, List<Pair<URI, Response>>> responsesTuple = traversalUriHttpEntry.process(requests, sourceOfTruth);
310 response = responsesTuple.getValue1().get(0).getValue1();
311
312 } catch (AAIException e) {
313     response = consumerExceptionResponseGenerator(headers, info, HttpMethod.GET, e);
314 } catch (Exception e) {
315     AAIException ex = new AAIException("AAI_4000", e);
```

The result returned by this interface is a response structure, and the actual return value is a JSON or XML result in string format. But because we don't know the specific object in here, **we can't automatically identify the only key value, so we can't filter the data very well.**

Summary

1. This is just a simulated multi tenant scheme, because it does not achieve real data isolation
2. SDC can be implemented through simple changes, but AAI is very difficult to do this, requiring major changes or other better solutions
3. Other modules have yet to be verified
4. There are defects in the scheme itself, such as the inability to create a buildingBlock with the same name, but the single user cannot perceive the error

Next step

1. Does the community has plans to implement a multi-tenant solution? And how to do it
2. Welcome everyone to join us to explore a feasible multi-tenant implementation
3. We hope that more people in the community can join us, and discuss how to achieve multi-tenancy.