KEYSTONE4J

ADAPT OPENSTACK KEYSTONE TO JAVA

PRESENTED BY:
PO-HSUN, HUANG
INFINITIESSOFT

PO-HSUN, HUANG (黃柏勳)

R&D Engineer
InfinitiesSoft, Inc. (數位無限軟體)

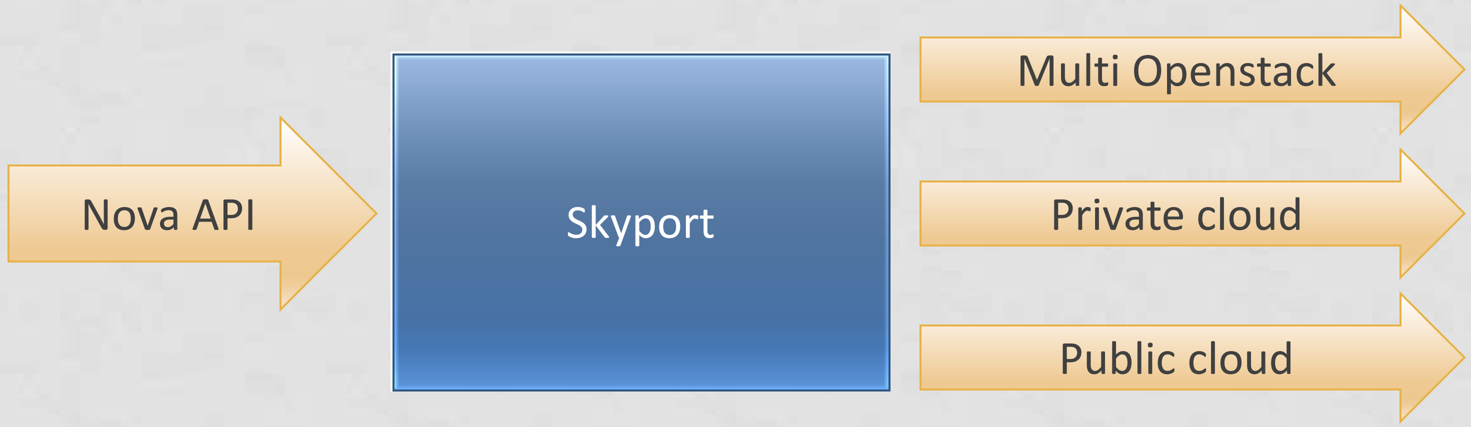
- Virtualization and Java guy
- Responsible for skyport, keystone4j development and wins3fs enhancement
- Keystone4j repository: https://bitbucket.org/infinities/keystone4j/
- E-mail: pohsun@infinitiessoft.com

AGENGA

- Subject of keystone4j project
- What is keystone?
- Keystone code structure
- How to adapt to java?
- Demo
- Q&A

SUBJECT OF KEYSTONE4J PROJECT

- Easier troubleshooting.
- Understand Openstack’s API design and access control .
- Integrated with Skyport.
- Promote Openstack in Java world.



NO RESTFUL API V.S. RESTFUL API

GET https://ec2.amazonaws.com/?Action=RunInstances &ImageId=ami-1a2b3c4d &MaxCount=1 &MinCount=1 &NetworkInterface.1.DeviceIndex=0 &NetworkInterface.1.AssociatePublicIpAddress=true &NetworkInterface.1.SubnetId=subnet-1a2b3c4d &AUTHPARAMS

Body: [empty]

⬇

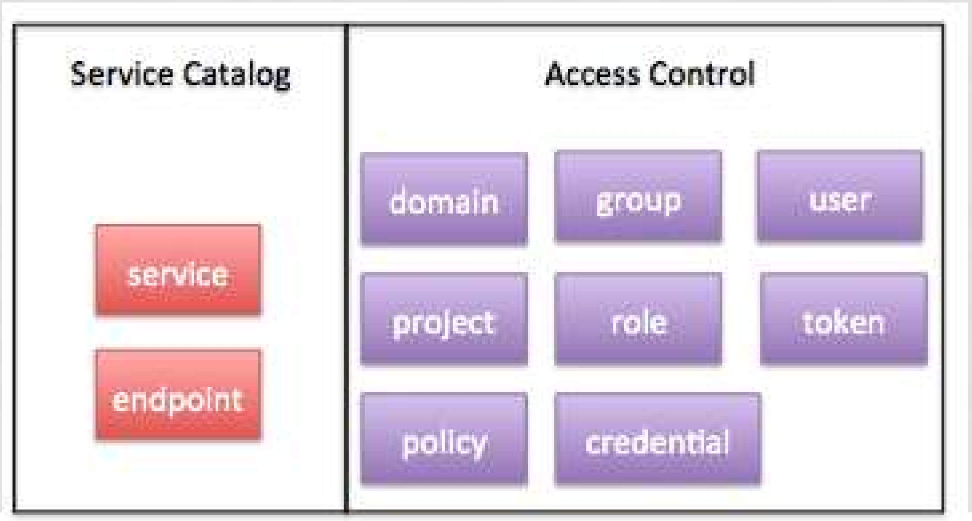
POST|GET|DELETE|PUT|HEAD https://openstack.com/v2/{tenant}/servers

Body:

{ "server": { "name": "server-test-1", "imageRef": "b5660a6e-4b46-4be3-9707-6b47221b454f", "flavorRef": "2", "max_count": 1, "min_count": 1, "networks": [{ "uuid": "d32019d3-bc6e-4319-9c1d-6722fc136a22" }], "security_groups": [{ "name": "default" }, { "name": "another-secgroup-name" }] } }

WHAT IS KEYSTONE

- An abstract layer
  - Backend: SQL、LDAP....
  - Authenticate: OAUTH、Password、Token....
- User management and Services Catalog
- Server-Client model : keystone, keystonemiddleware.auth_token
- WSGI model

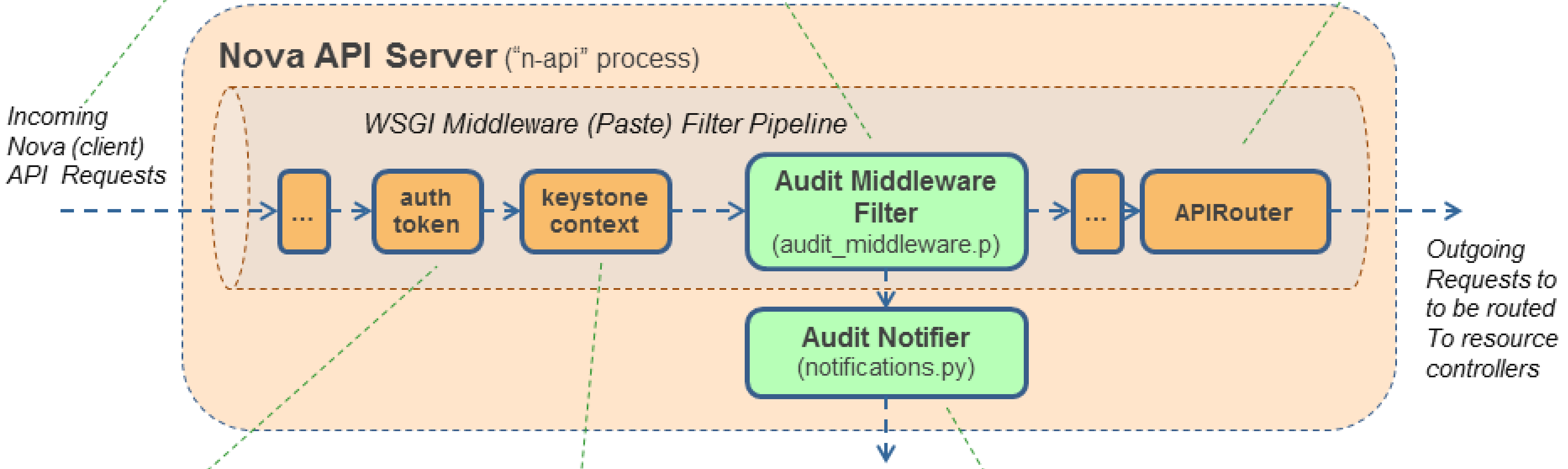


How the CADF Audit Middleware Filter Works in the Nova API Pipeline

1 The WSGI "Paste" framework handles Nova API Client requests and passes them to the Nova API Server "pipeline".

4 Audit Filter is added to pipeline after standard "keystone" processing. It uses URI and request data to map to CADF data model including data inserted by "authtoken" and "keystonecontext" and creates a CADF Event Record

6 The API request continues down filter pipeline and eventually is routed to the correct resources (controllers)



Nova API Server ("n-api" process)
WSGI Middleware (Paste) Filter Pipeline
Incoming Nova (client) API Requests
...
auth token
keystone context
Audit Middleware Filter (audit_middleware.p)
...
APIRouter
Audit Notifier (notifications.py)
Outgoing Requests to to be routed To resource controllers

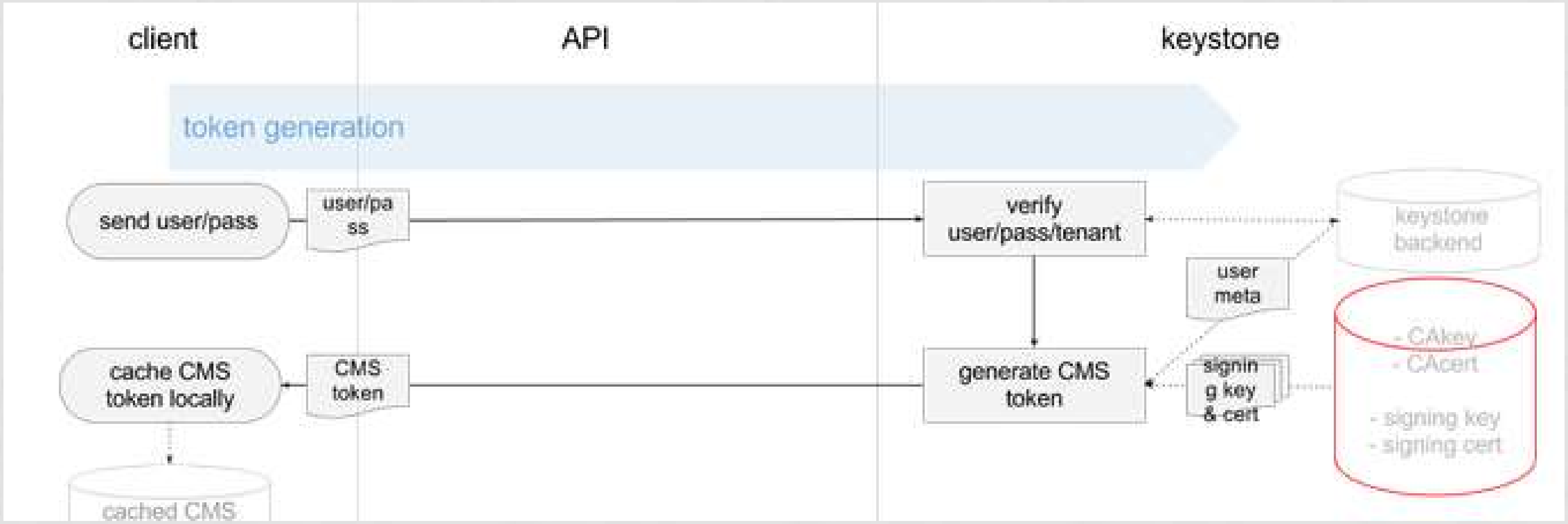
2 The (shared) Keystone Filter validates "tokens" and adds account, tenant and user data to request (e.g. user name and ID).

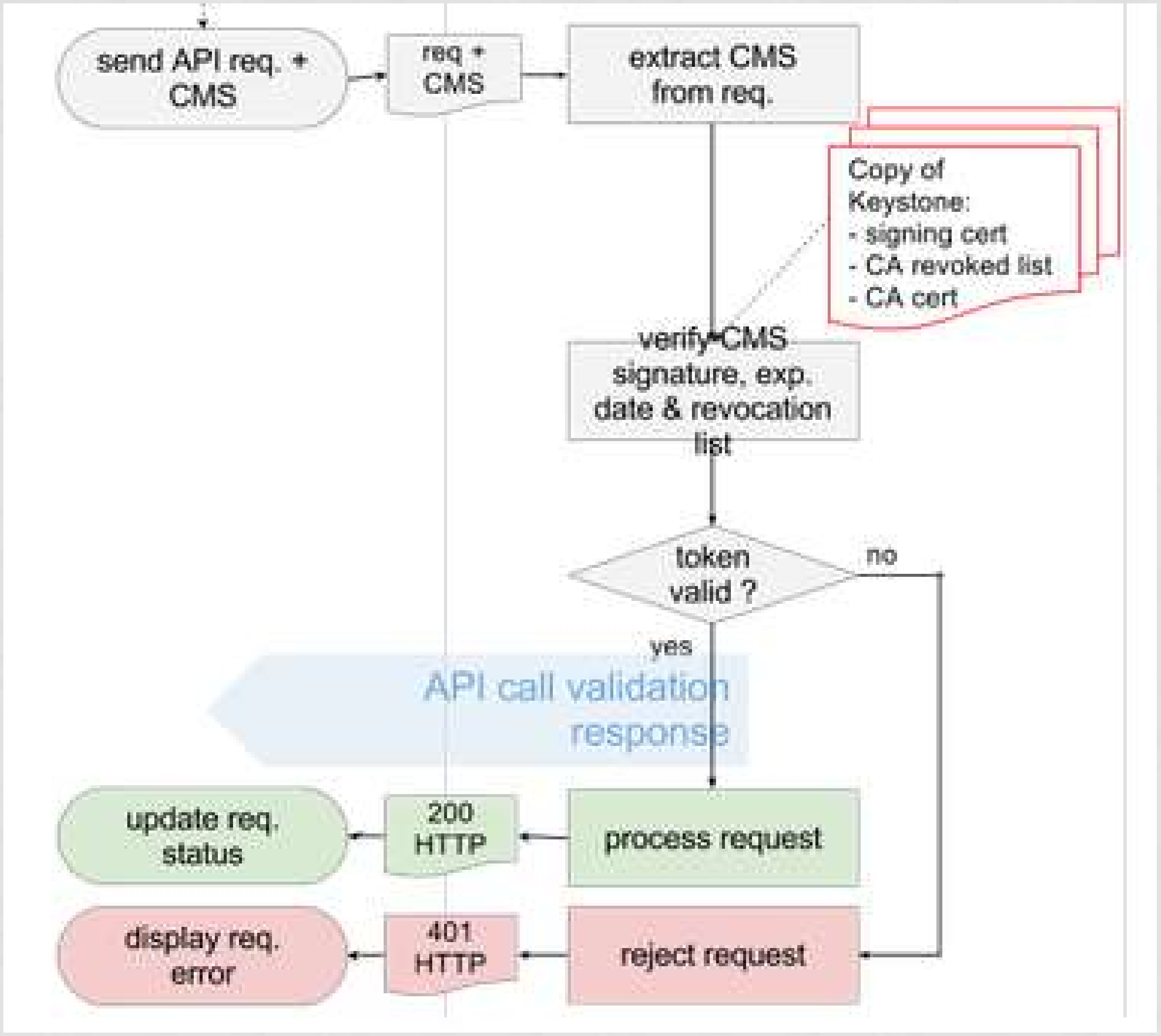
3 Builds a Keystone request context using the data inserted by "authtoken" filter and places it in the request environment (e.g. the "nova.context").

5 The Audit Filter invokes the Audit Notifier with the CADF Event Record which adds it to the standard message payload with the "audit" topic which the Ceilometer Event Listener is registered to receive.

Color Key

🟧	Nova Modules
🟩	Ceilometer Modules

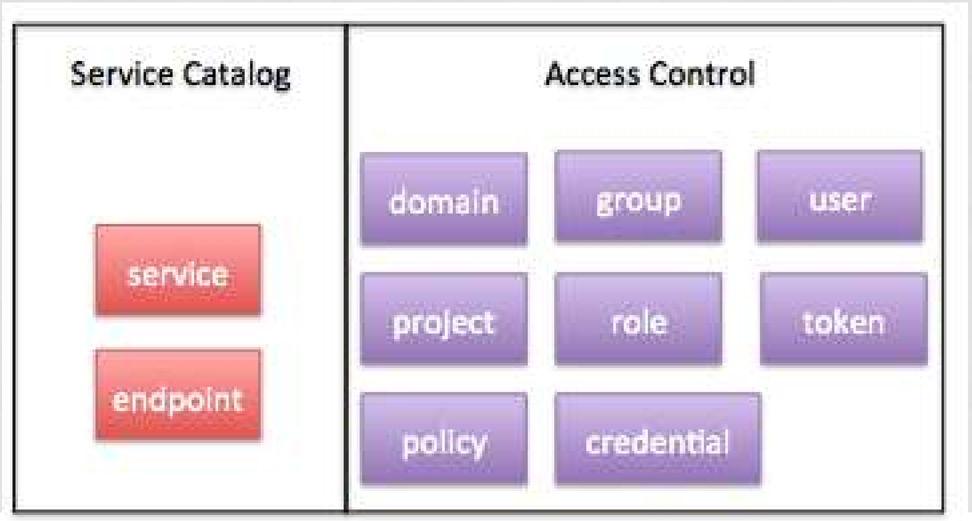

client
API
keystone
token generation
send user/pass
user/pass
verify user/pass/tenant
keystone backend
user meta
generate CMS token
signing key & cert
- CAkey
- CAcert
- signing key
- signing cert
CMS token
cache CMS token locally
cached CMS


send API req. + CMS
req + CMS
extract CMS from req.
Copy of Keystone:
- signing cert
- CA revoked list
- CA cert
verify CMS signature, exp. date & revocation list
token valid ?
no
yes
API call validation response
update req. status
200 HTTP
process request
display req. error
401 HTTP
reject request

```
{
    "access": {
        "metadata": {
            ....metadata goes here....
        },
        "serviceCatalog": [
            ....endpoints goes here....
        ],
        "token": {
            "expires": "2013-05-26T08:52:53Z",
            "id": "placeholder",
            "issued_at": "2013-05-25T18:59:33.841811",
            "tenant": {
                "description": null,
                "enabled": true,
                "id": "925c23eafe1b4763933e08a4c4143f08",
                "name": "user"
            }
        },
        "user": {
            ....userdata goes here....
        }
    }
}
```

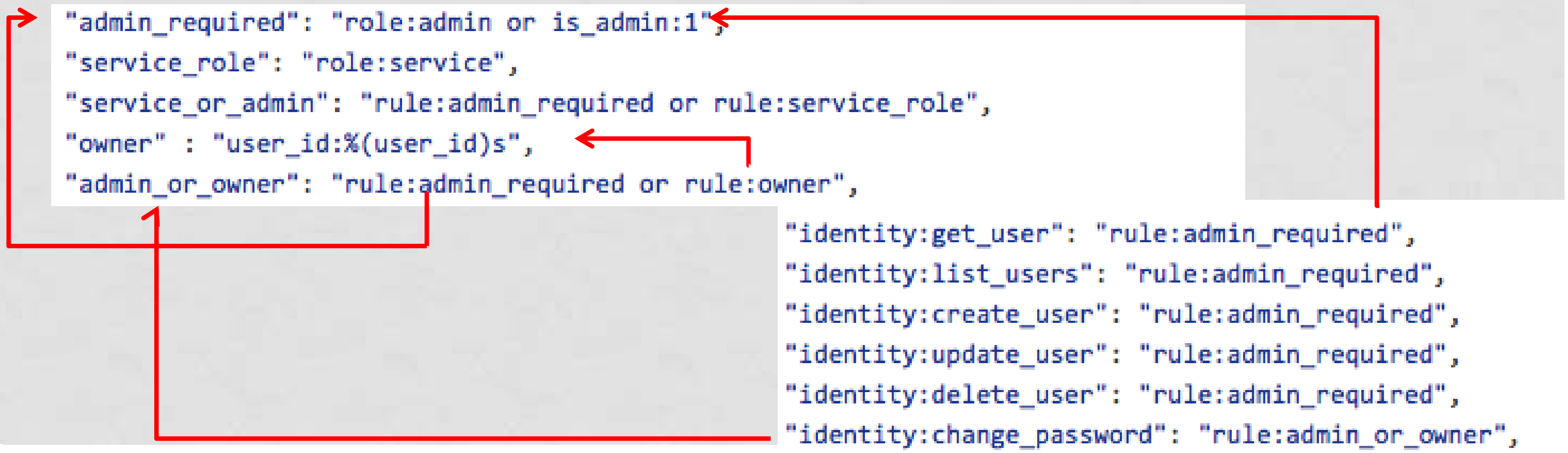
WHAT IS KEYSTONE

- An abstract layer
  - Backend: SQL、LDAP....
  - Authenticate: OAUTH、Password、Token....
- User management and Services Catalog
- Server-Client model : keystone, keystonemiddleware.auth_token
- WSGI model



WHERE TO BEGIN – CONFIG FILES

- Keystone repository: https://github.com/openstack/keystone
- Analysis configuration files
  - policy.json: API access control rule
  - keystone.conf : application options
  - keystone-paste.ini : delopyment

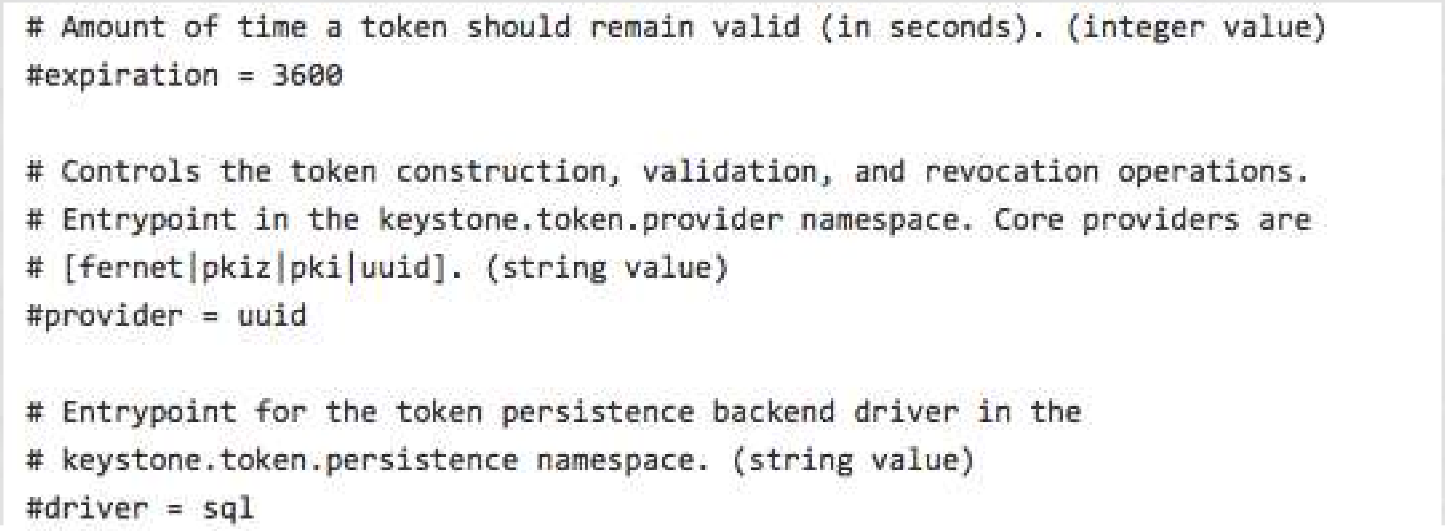


```
"admin_required": "role:admin or is_admin:1",
"service_role": "role:service",
"service_or_admin": "rule:admin_required or rule:service_role",
"owner" : "user_id:%(user_id)s",
"admin_or_owner": "rule:admin_required or rule:owner",
```

```
"identity:get_user": "rule:admin_required",
"identity:list_users": "rule:admin_required",
"identity:create_user": "rule:admin_required",
"identity:update_user": "rule:admin_required",
"identity:delete_user": "rule:admin_required",
"identity:change_password": "rule:admin_or_owner",
```

WHERE TO BEGIN – CONFIG FILES

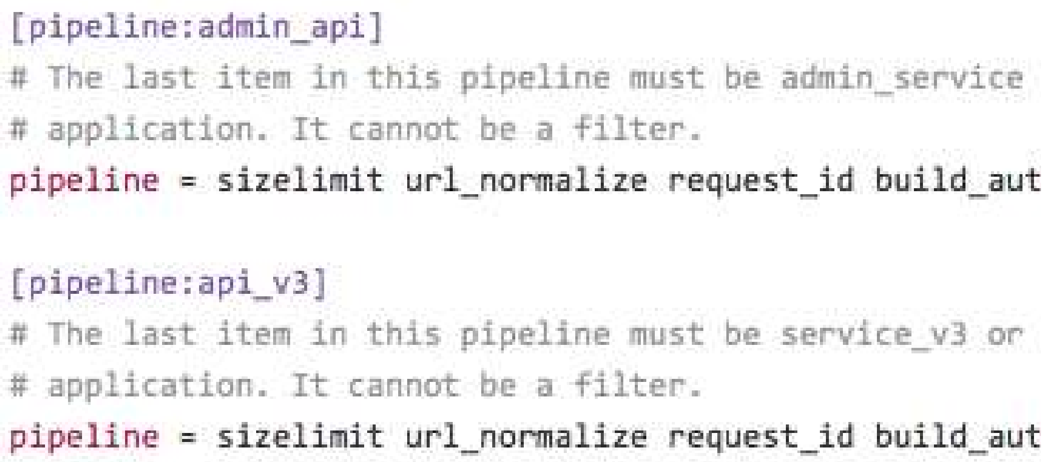
- Keystone repository: https://github.com/openstack/keystone
- Analysis configuration files
  - policy.json: API access control rule
  - keystone.conf : application options
  - keystone-paste.ini : delopyment



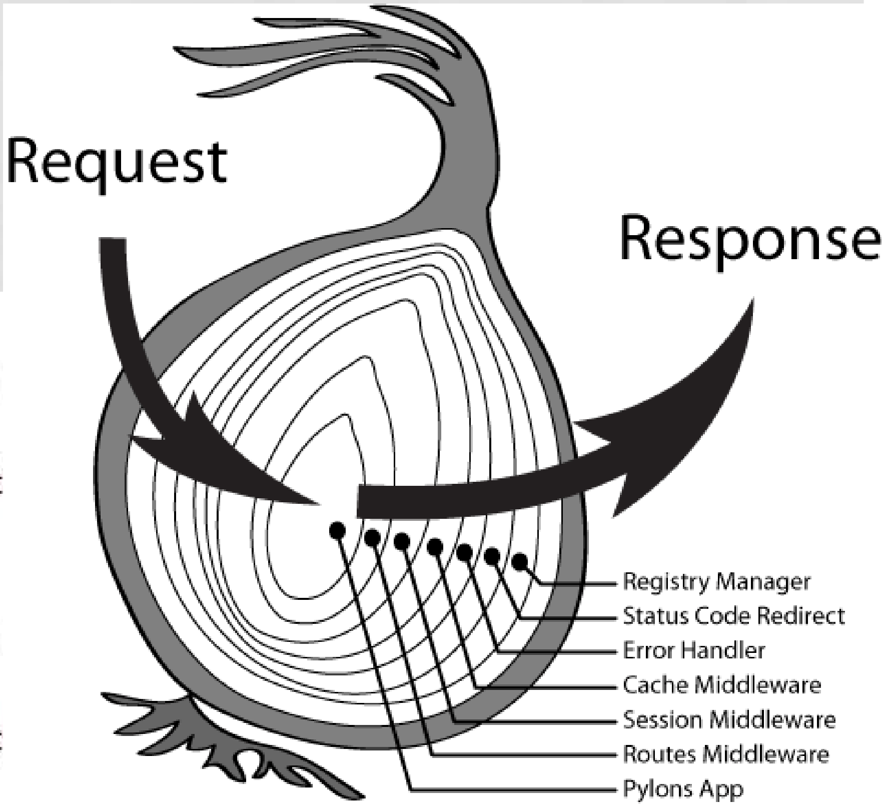
```
# Amount of time a token should remain valid (in seconds). (integer value)
#expiration = 3600

# Controls the token construction, validation, and revocation operations.
# Entrypoint in the keystone.token.provider namespace. Core providers are
# [fernet|pkiz|pki|uuid]. (string value)
#provider = uuid

# Entrypoint for the token persistence backend driver in the
# keystone.token.persistence namespace. (string value)
#driver = sql
```

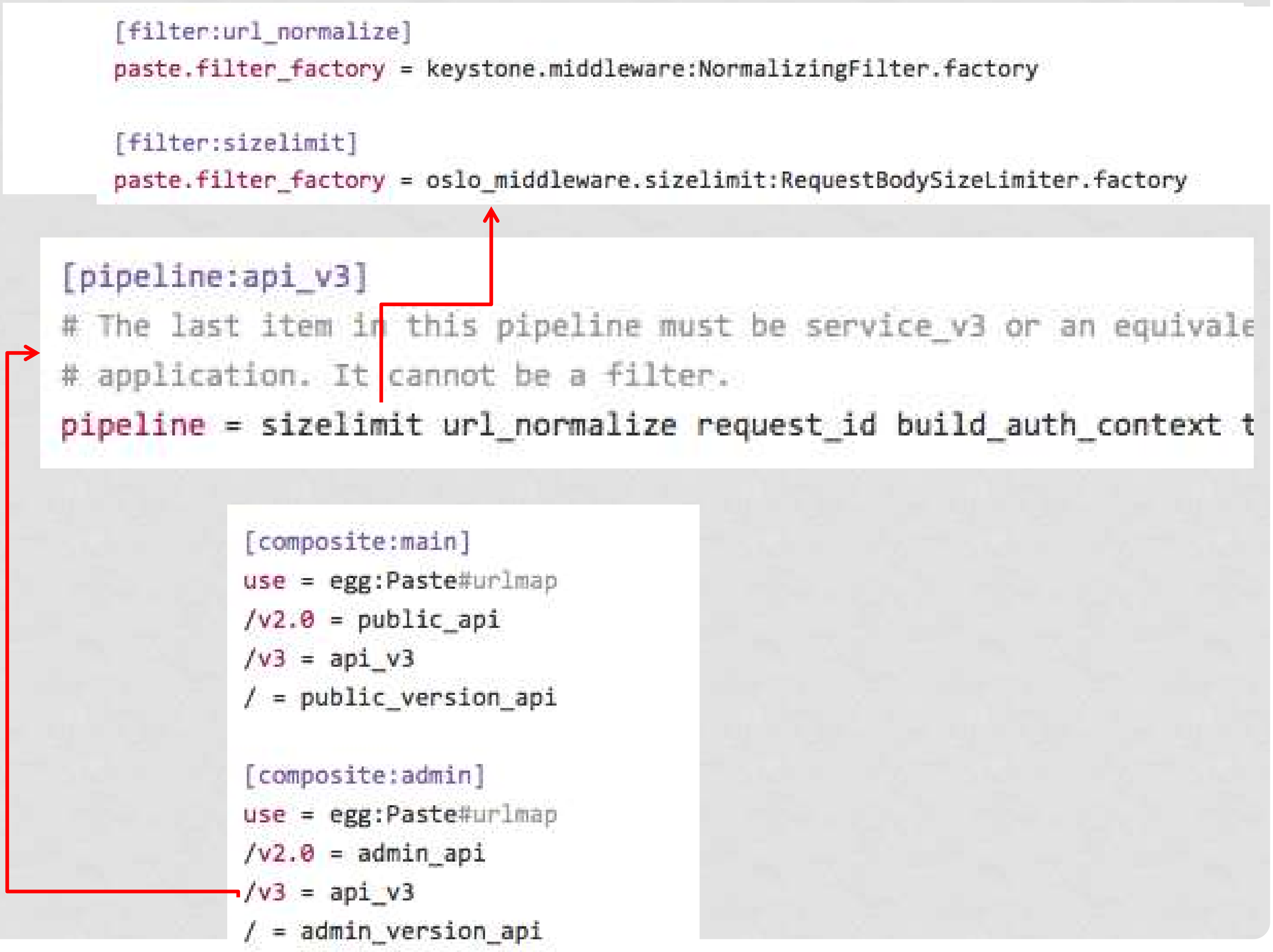
WHERE TO BEGIN – CONFIG FILES

- Keystone repository: https://github.com/openstack/keystone
- Analysis configuration files
  - policy.json: API access control rule
  - keystone.conf : application options
  - keystone-paste.ini : delopyment



```
[pipeline:admin_api]
# The last item in this pipeline must be admin_service
# application. It cannot be a filter.
pipeline = sizelimit url_normalize request_id build_aut

[pipeline:api_v3]
# The last item in this pipeline must be service_v3 or
# application. It cannot be a filter.
pipeline = sizelimit url_normalize request_id build_aut
```



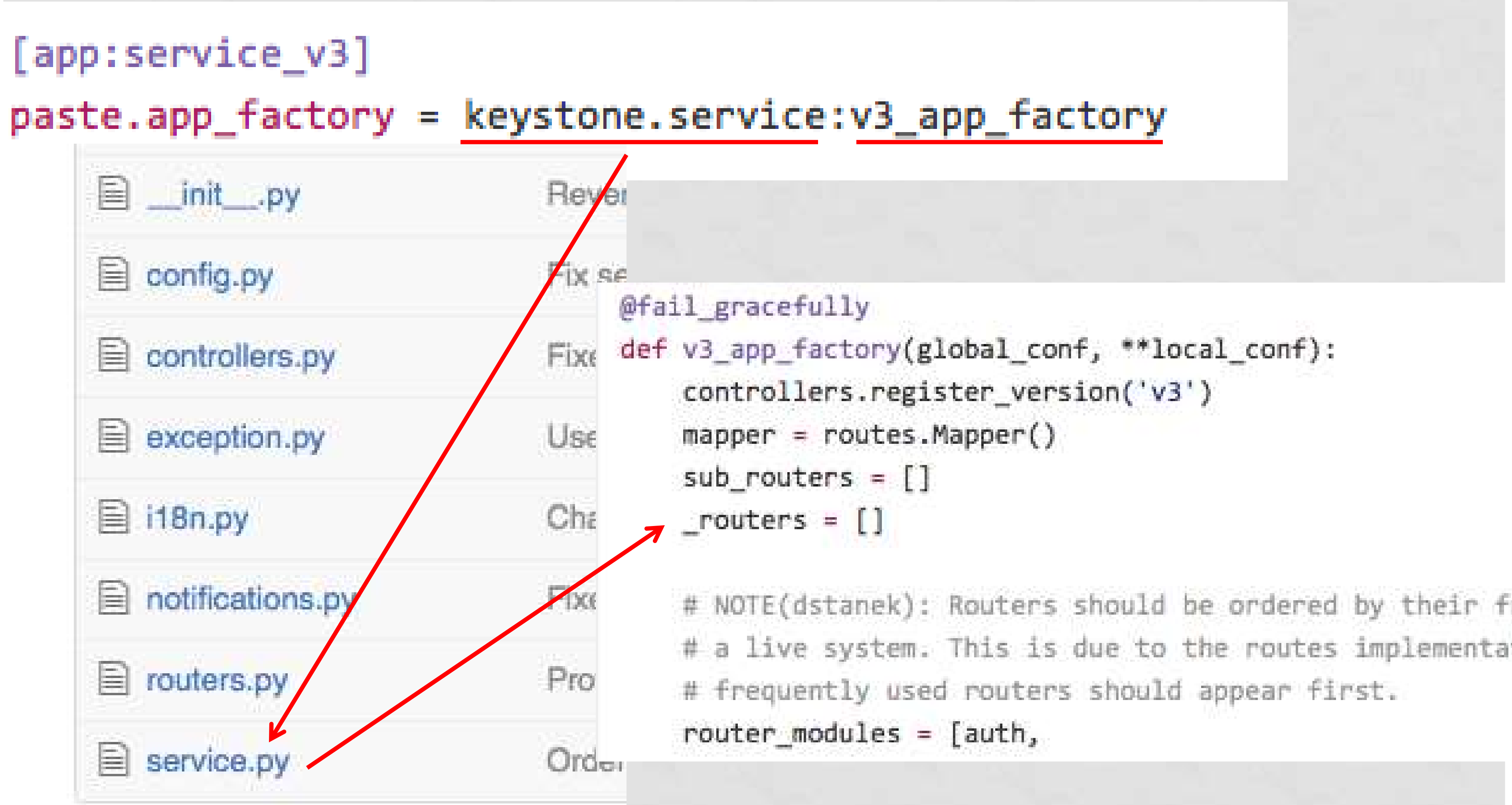


```
[filter:url_normalize]
paste.filter_factory = keystone.middleware:NormalizingFilter.factory

[filter:sizelimit]
paste.filter_factory = oslo_middleware.sizelimit:RequestBodySizeLimiter.factory
```

```
[pipeline:api_v3]
# The last item in this pipeline must be service_v3 or an equivale
# application. It cannot be a filter.
pipeline = sizelimit url_normalize request_id build_auth_context t
```

```
[composite:main]
use = egg:Paste#urlmap
/v2.0 = public_api
/v3 = api_v3
/ = public_version_api

[composite:admin]
use = egg:Paste#urlmap
/v2.0 = admin_api
/v3 = api_v3
/ = admin_version_api
```

FROM CONFIG FILES TO CODE

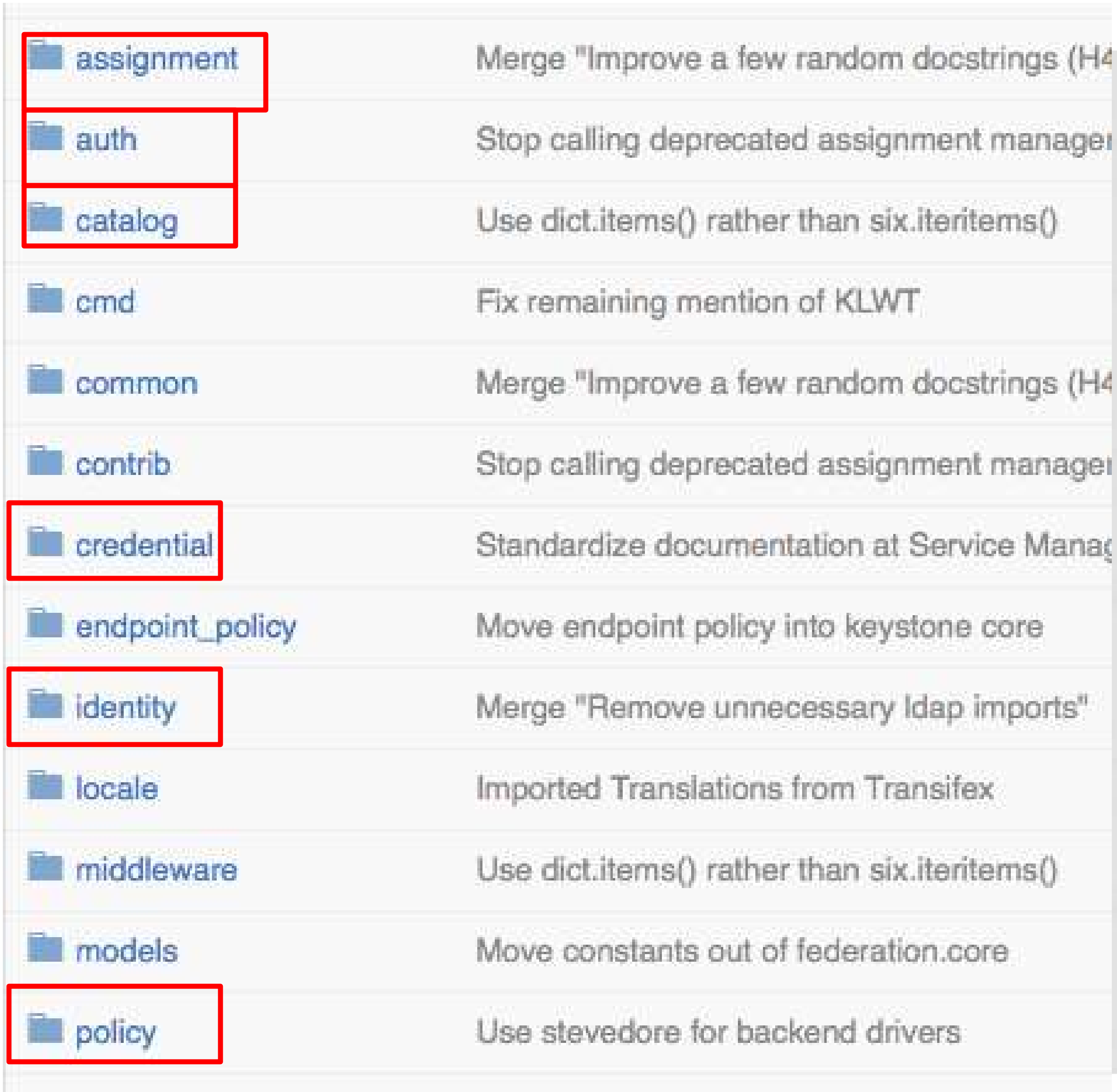


```
[app:service_v3]
paste.app_factory = keystone.service:v3_app_factory
```

__init__.py	Rever
config.py	Fix se
controllers.py	Fixe
exception.py	Use
i18n.py	Cha
notifications.py	Fixe
routers.py	Pro
service.py	Orde

```
@fail_gracefully
def v3_app_factory(global_conf, **local_conf):
    controllers.register_version('v3')
    mapper = routes.Mapper()
    sub_routers = []
    _routers = []

    # NOTE(dstanek): Routers should be ordered by their fr
    # a live system. This is due to the routes implementat
    # frequently used routers should appear first.
    router_modules = [auth,
```

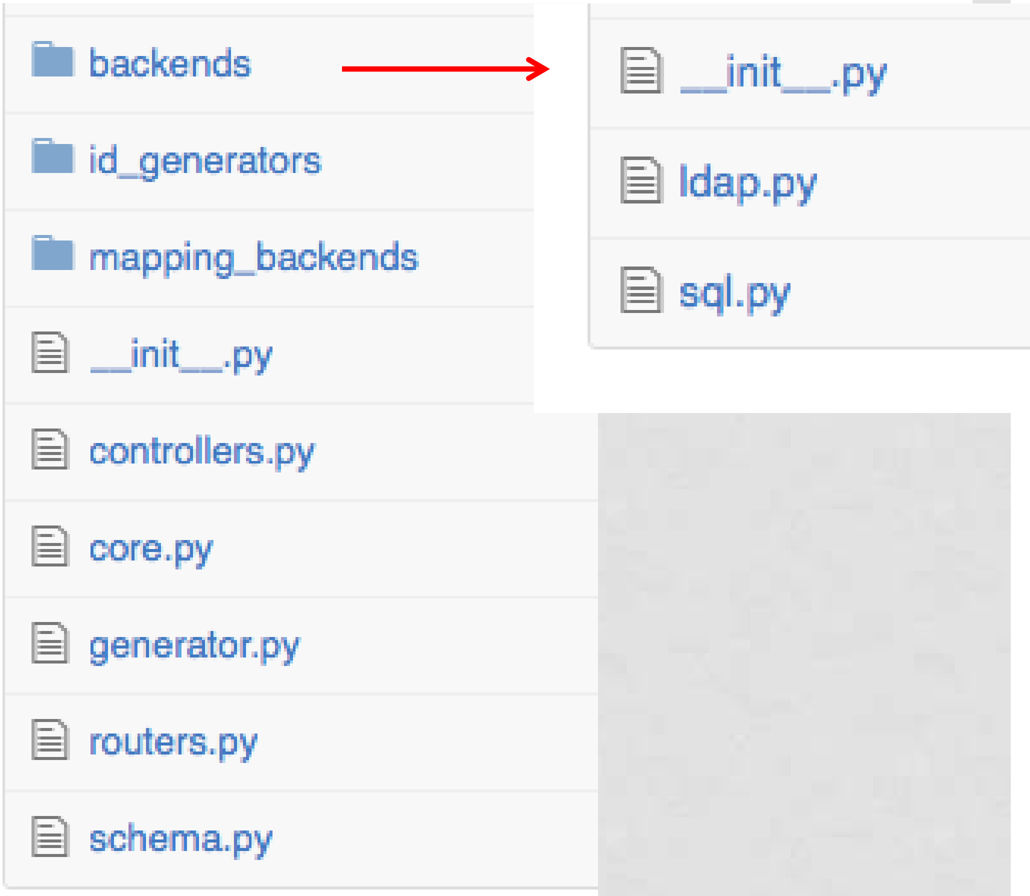
WHERE TO BEGIN – CODE



assignment	Merge "Improve a few random docstrings (H4
auth	Stop calling deprecated assignment manage
catalog	Use dict.items() rather than six.iteritems()
cmd	Fix remaining mention of KLWT
common	Merge "Improve a few random docstrings (H4
contrib	Stop calling deprecated assignment manage
credential	Standardize documentation at Service Mana
endpoint_policy	Move endpoint policy into keystone core
identity	Merge "Remove unnecessary ldap imports"
locale	Imported Translations from Transifex
middleware	Use dict.items() rather than six.iteritems()
models	Move constants out of federation.core
policy	Use stevedore for backend drivers

CODE STRUCTURE

- Router(routers.py)
- Controller(controllers.py)
- Manager(core.py)
- Driver
  - Interface(core.py)
  - Implementation(backends)

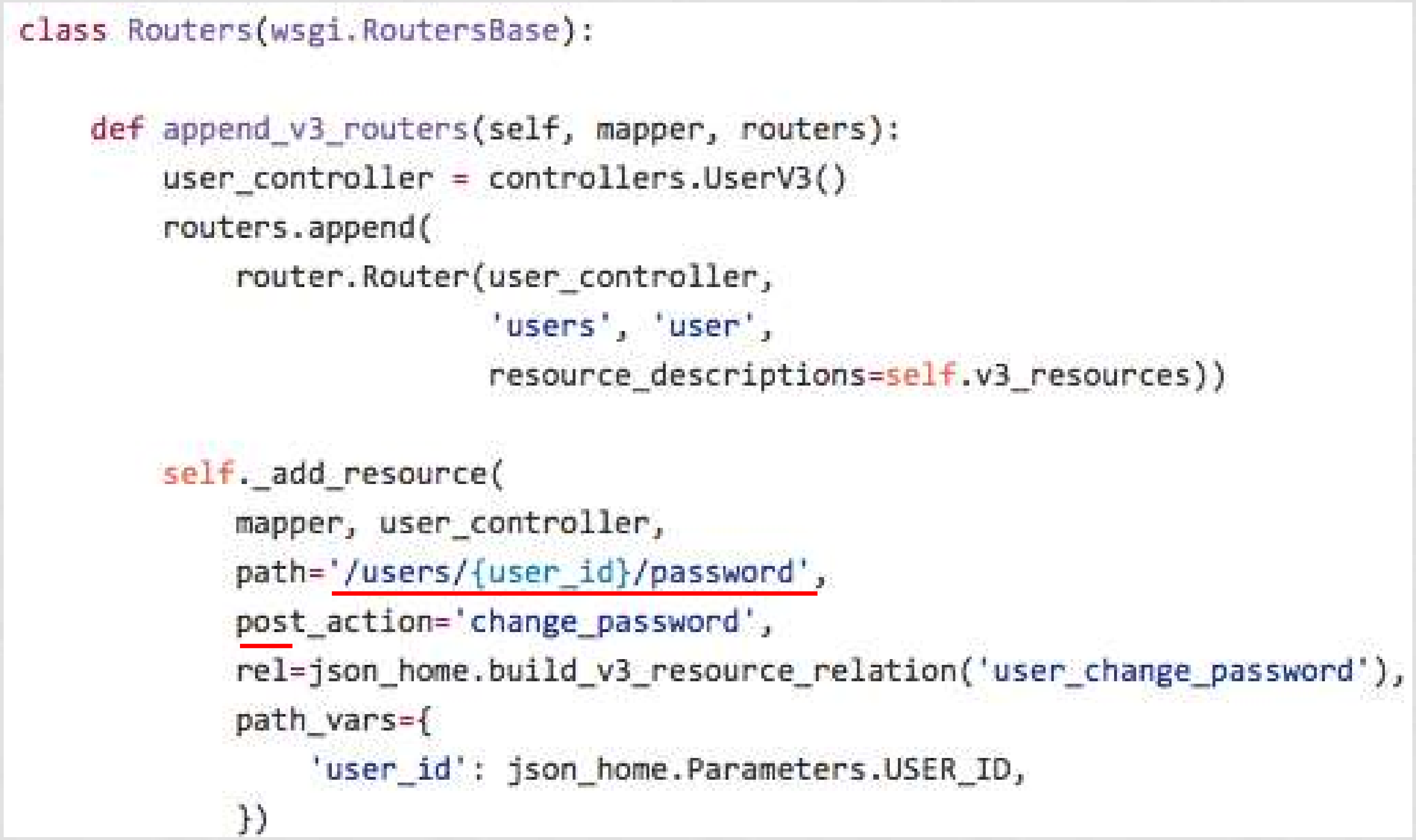


backends
id_generators
mapping_backends
__init__.py
controllers.py
core.py
generator.py
routers.py
schema.py

__init__.py
ldap.py
sql.py

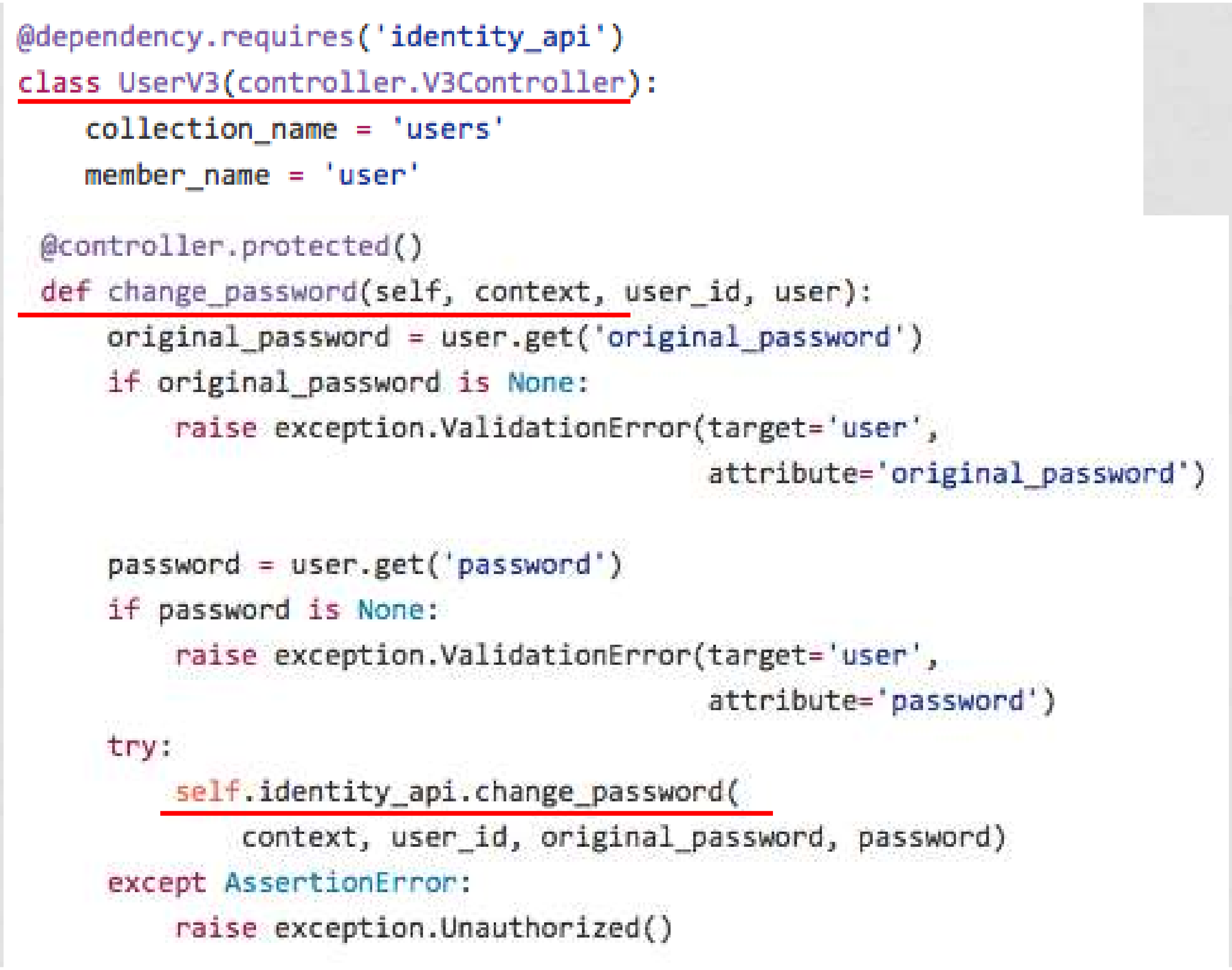
- Router →Controller→Manager(API)→Driver

CODE STRUCTURE - ROUTER



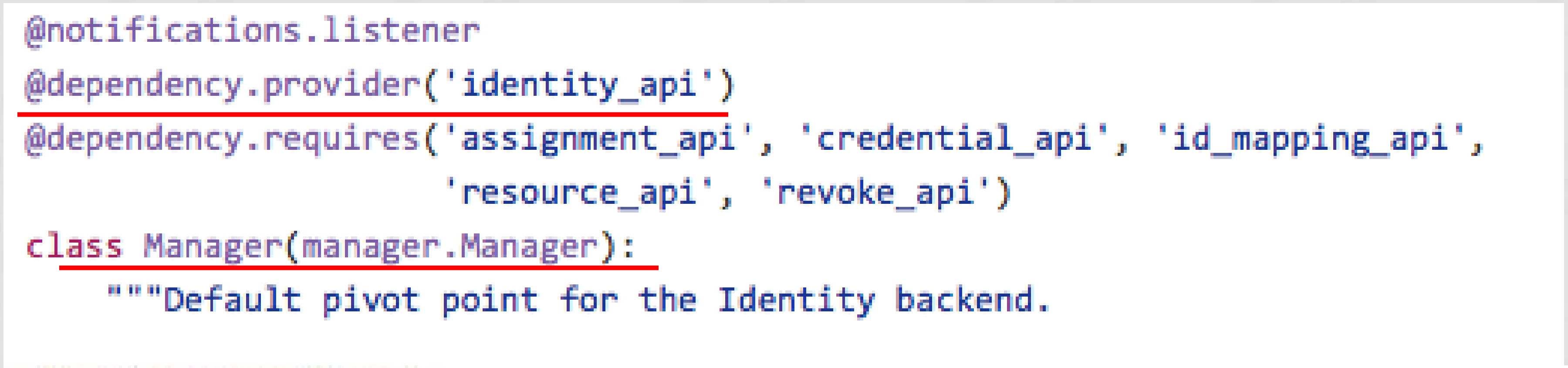
```
class Routers(wsgi.RoutersBase):

    def append_v3_routers(self, mapper, routers):
        user_controller = controllers.UserV3()
        routers.append(
            router.Router(user_controller,
                          'users', 'user',
                          resource_descriptions=self.v3_resources))

        self._add_resource(
            mapper, user_controller,
            path='/users/{user_id}/password',
            post_action='change_password',
            rel=json_home.build_v3_resource_relation('user_change_password'),
            path_vars={
                'user_id': json_home.Parameters.USER_ID,
            })
```

CODE STRUCTURE - CONTROLLER

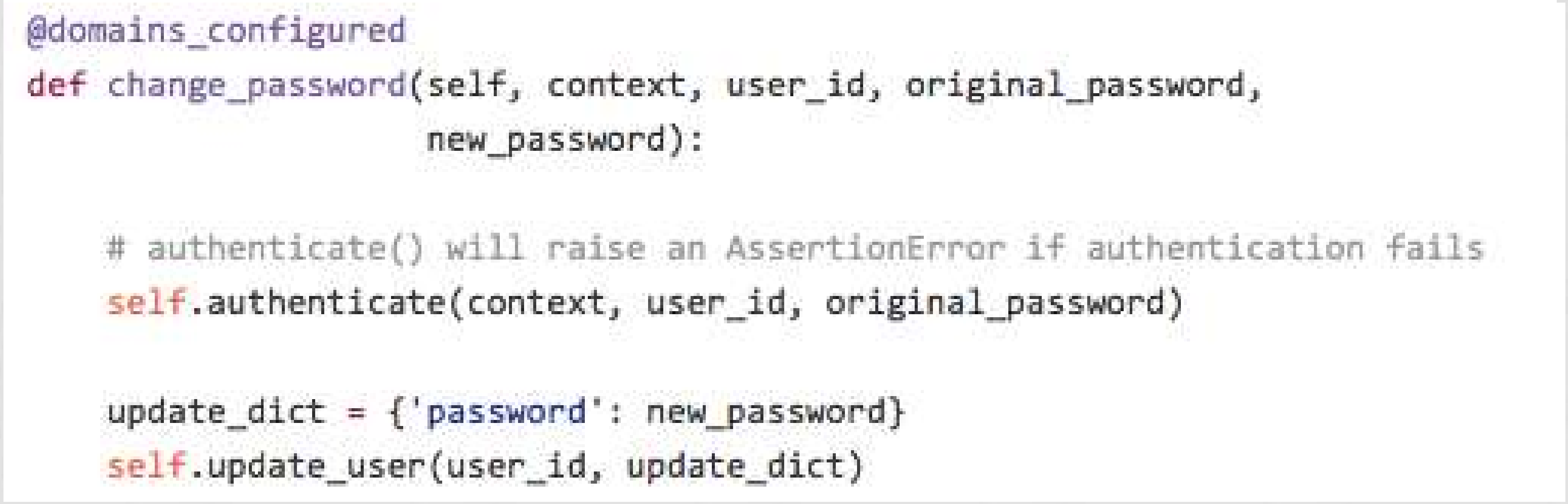


```
@dependency.requires('identity_api')
class UserV3(controller.V3Controller):
    collection_name = 'users'
    member_name = 'user'

  @controller.protected()
  def change_password(self, context, user_id, user):
      original_password = user.get('original_password')
      if original_password is None:
          raise exception.ValidationError(target='user',
                                          attribute='original_password')

      password = user.get('password')
      if password is None:
          raise exception.ValidationError(target='user',
                                          attribute='password')
      try:
          self.identity_api.change_password(
              context, user_id, original_password, password)
      except AssertionError:
          raise exception.Unauthorized()
```

CODE STRUCTURE – MANAGER

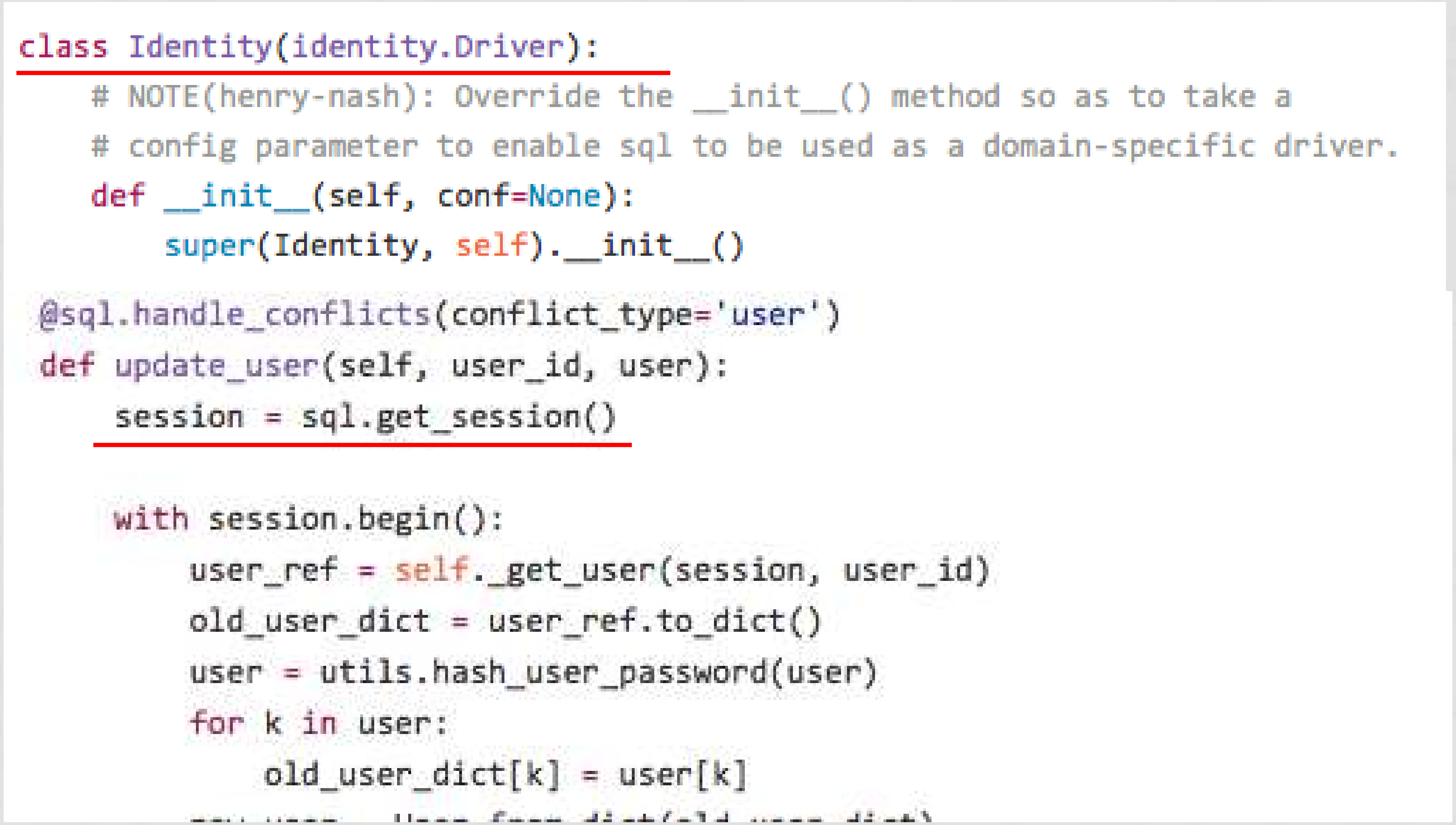


```
@notifications.listener
@dependency.provider('identity_api')
@dependency.requires('assignment_api', 'credential_api', 'id_mapping_api',
                     'resource_api', 'revoke_api')
class Manager(manager.Manager):
    """Default pivot point for the Identity backend.
```



```
@domains_configured
def change_password(self, context, user_id, original_password,
                    new_password):

    # authenticate() will raise an AssertionError if authentication fails
    self.authenticate(context, user_id, original_password)

    update_dict = {'password': new_password}
    self.update_user(user_id, update_dict)
```

CODE STRUCTURE – DRIVER



```
class Identity(identity.Driver):
    # NOTE(henry-nash): Override the __init__() method so as to take a
    # config parameter to enable sql to be used as a domain-specific driver.
    def __init__(self, conf=None):
        super(Identity, self).__init__()
```

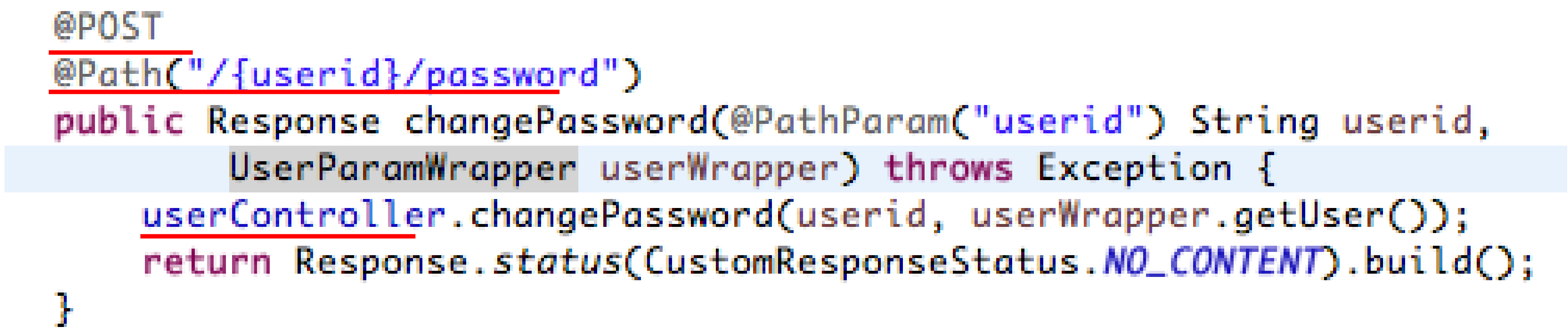
```
@sql.handle_conflicts(conflict_type='user')
def update_user(self, user_id, user):
    session = sql.get_session()

    with session.begin():
        user_ref = self._get_user(session, user_id)
        old_user_dict = user_ref.to_dict()
        user = utils.hash_user_password(user)
        for k in user:
            old_user_dict[k] = user[k]
```

KEYSTONE4J

- All Identity API v3 and partial v2.0(token)
- SQL driver only, no LDAP
- Support password, token authentication
- Configured using keystone.conf, policy.json
- JRE(Java Runtime Environment) 7
- Jersey framework(RESTful Web Service framework)
  - https://jersey.java.net/

HOW TO ADAPT TO JAVA

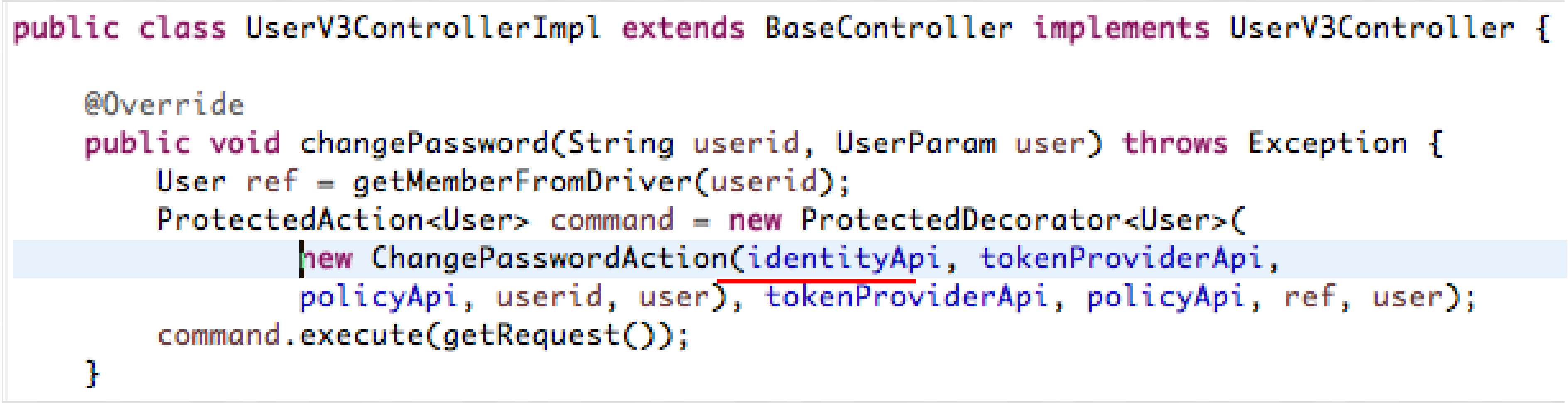
Router



```
@POST
@Path("/{userid}/password")
public Response changePassword(@PathParam("userid") String userid,
        UserParamWrapper userWrapper) throws Exception {
    userController.changePassword(userid, userWrapper.getUser());
    return Response.status(CustomResponseStatus.NO_CONTENT).build();
}
```

HOW TO ADAPT TO JAVA

Controller



```
public class UserV3ControllerImpl extends BaseController implements UserV3Controller {

    @Override
    public void changePassword(String userid, UserParam user) throws Exception {
        User ref = getMemberFromDriver(userid);
        ProtectedAction<User> command = new ProtectedDecorator<User>(
                new ChangePasswordAction(identityApi, tokenProviderApi,
                policyApi, userid, user), tokenProviderApi, policyApi, ref, user);
        command.execute(getRequest());
    }
```

Demo

Q&A