JavaSi '14

20. oktober 2014

# WildFly in Oracle okolje

## *Integracija aplikacij z obstoječo infrastrukturo*

Predavatelj:

**Urh Srečnik** <urh.srecnik@abakus.si>

Software Architect @ Abakus Plus d.o.o.

ORACLE
Certified Associate
Oracle Database 11*g*
Administrator

AS
Abakus
As na disku.

ORACLE Gold Partner
Specialized
Oracle Database

# Abakus Plus d.o.o.

- Applications
  - Special
    - DB – Newspaper Distribution,
    - FIS – Flight Information System
    - DMS – Document Management System
  - ARBITER – the ultimate tool in audit trailing
  - APPM – Abakus Plus Performance and Monitoring Tool
  - Backup Server

- Services
  - DBA, OS administration, programming (MediaWiki, Oracle)
  - networks (services, VPN, QoS, security)
  - open source, monitoring (Nagios, OCS, Wiki)

- Hardware
  - servers, backup server, SAN storage, firewalls

GENERALI
Zavarovalnica

GENERALI
GROUP

Aerodrom Ljubljana

BANKA
SLOVENIJE
BANK OF SLOVENIA
EUROSYSTEM

DELO PRODAJA

KONTROLA
ZRAČNEGA
PROMETA
SLOVENIJE

Gorenjska B
Vse

GOODYEAR

HRANILNICA LON
Bančništvo na Ljubezniv Oseben Način

Mestna občina
Ljubljana

Mercator

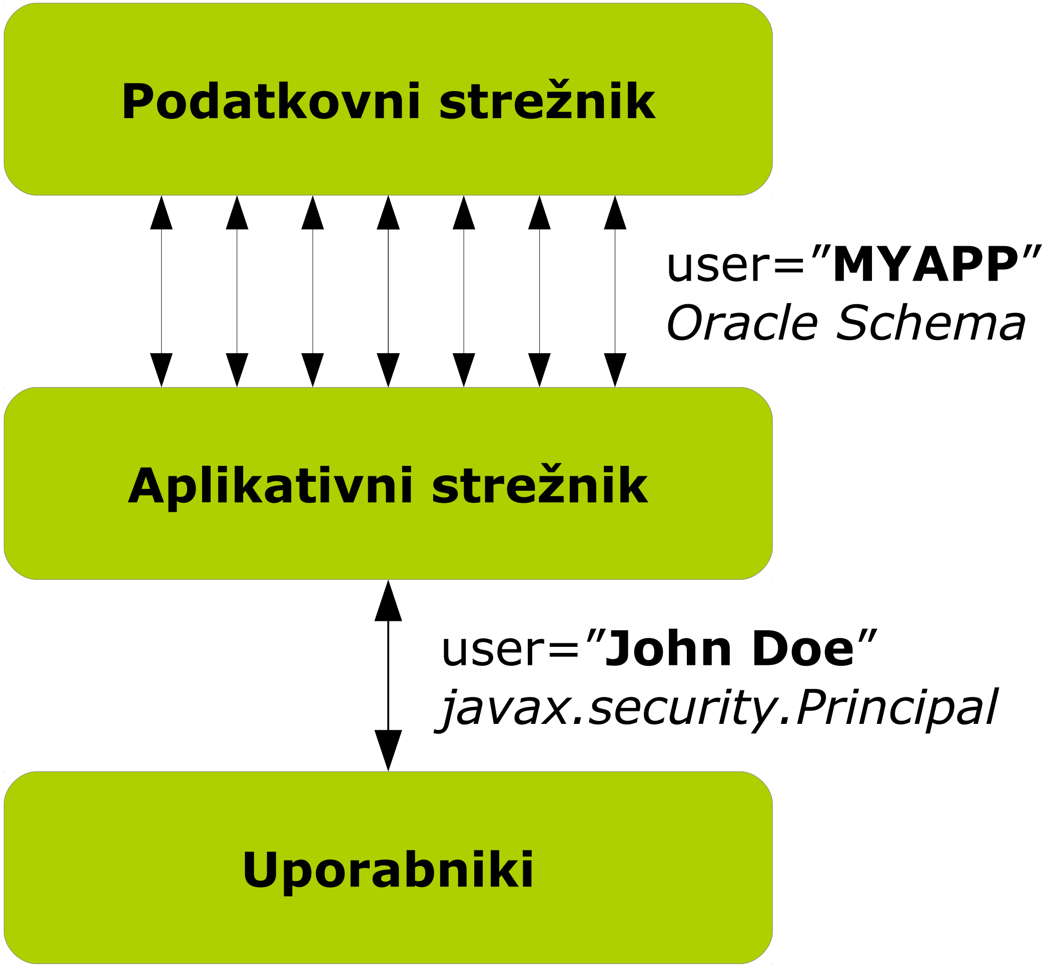
NLB Vita
Življenjska zavarovalnica

MESTNA OBČINA KOPER
COMUNE CITTÀ DI CAPODISTRIA

Sava

AS
Abakus
As na disku.

# DBA_USERS in aplikativni uporabniki



**Podatkovni strežnik**

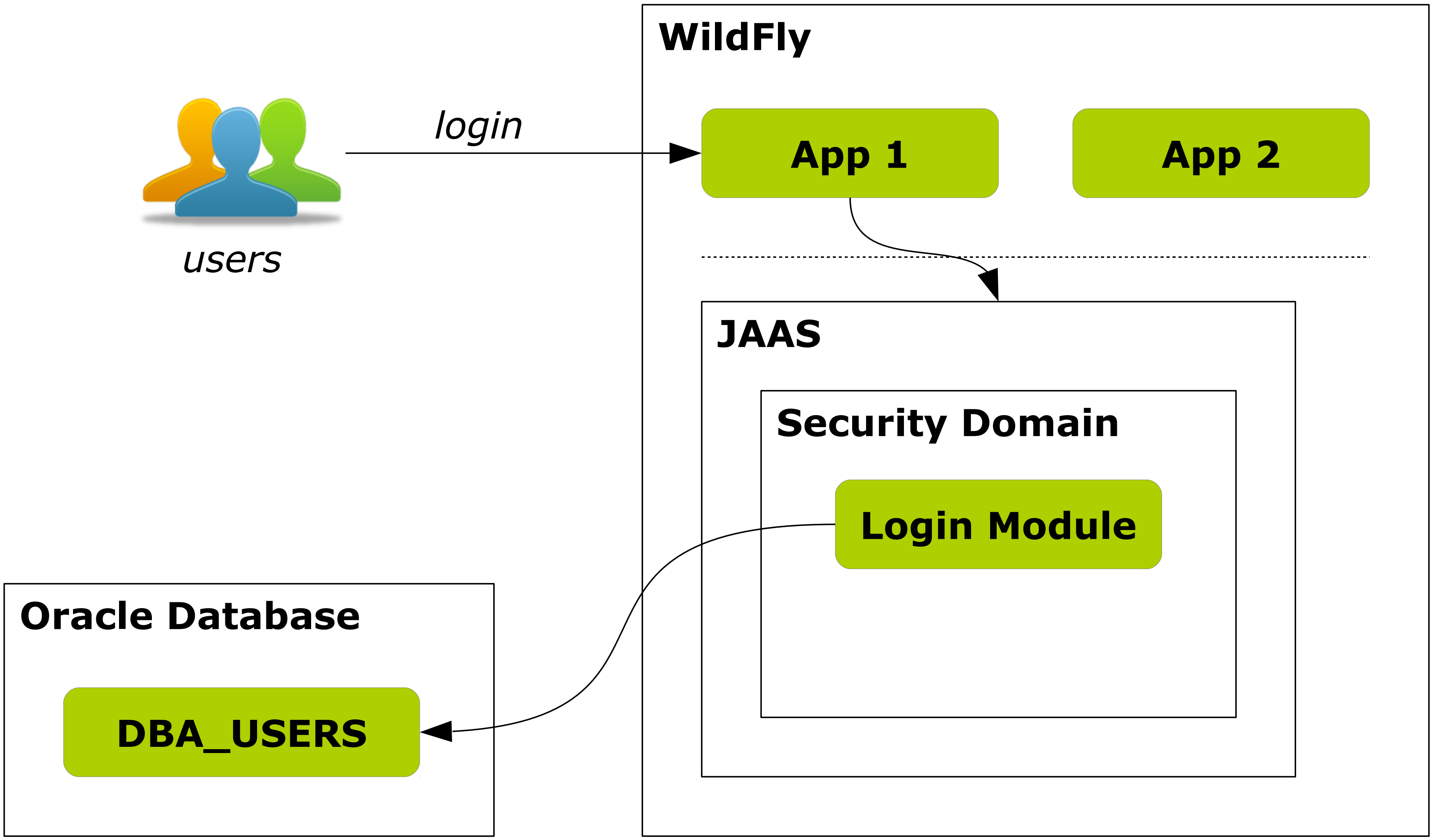
user=”**MYAPP**”
*Oracle Schema*

**Aplikativni strežnik**

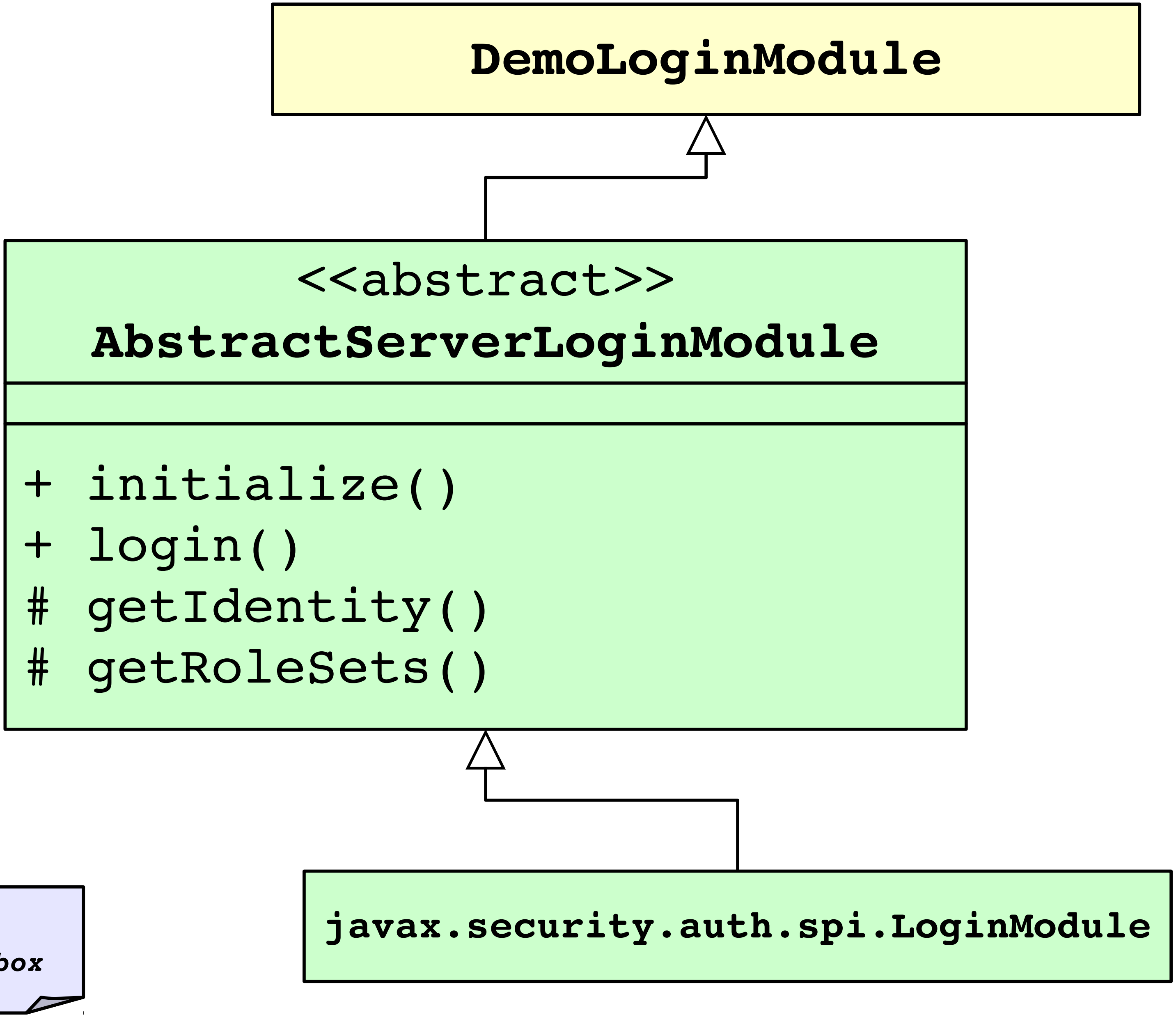
user=”**John Doe**”
*javax.security.Principal*

**Uporabniki**

# Container Managed Authentication



# WildFly Login Module Implementation



**Maven Coordinates:**
org.picketbox:***picketbox***

# WildFly Module Deployment

```
$WILDFLY_HOME/
  `- modules/
     `- system/
        `- layers/
           `- base/
              `- mycompany/
                 `- mymodule/
                   `- main/
```

**mymodule.xml**

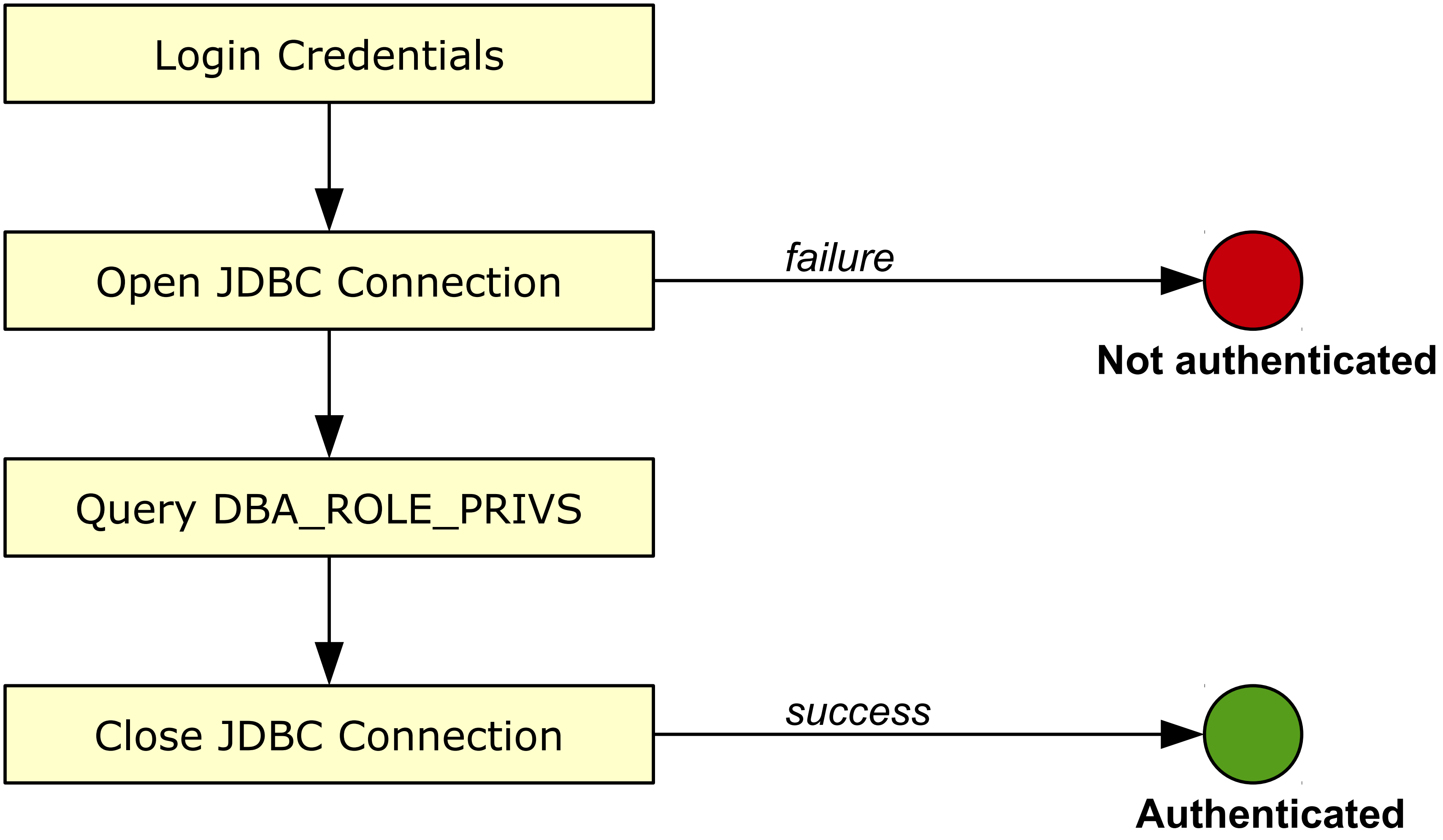
**mymodule.jar**

```
$ ./jboss-cli.sh --connect
[standalone@localhost:9990 /] module add \
> --name=mycompany.mymodule \
> --resources=mymodule.jar \
> --dependencies=org.picketbox
```

# WildFly Create Security Domain

```
./subsystem=security/security-domain=demo-security-domain:add(cache-type="default")

cd ./subsystem=security/security-domain=demo-security-domain

./authentication=classic:add(\
  login-modules=[ { \
    code="com.mypackage.MyDemoModule", \
    flag="required", \
    module-options={ \
      option="value" \
    } \
  }])
```

# Login Logic


Login Credentials
Open JDBC Connection
failure
Not authenticated
Query DBA_ROLE_PRIVS
Close JDBC Connection
success
Authenticated

# Oracle Users and Roles (example)

```
create user app_schema
  identified by app_schema
  account lock;

create user app$proxy identified by app$proxy;

create user app$user_a identified by user_a;

create user app$user_b identified by user_b;

create user app$user_c identified by user_c;

create role apr$admin;

create role apr$user;
```

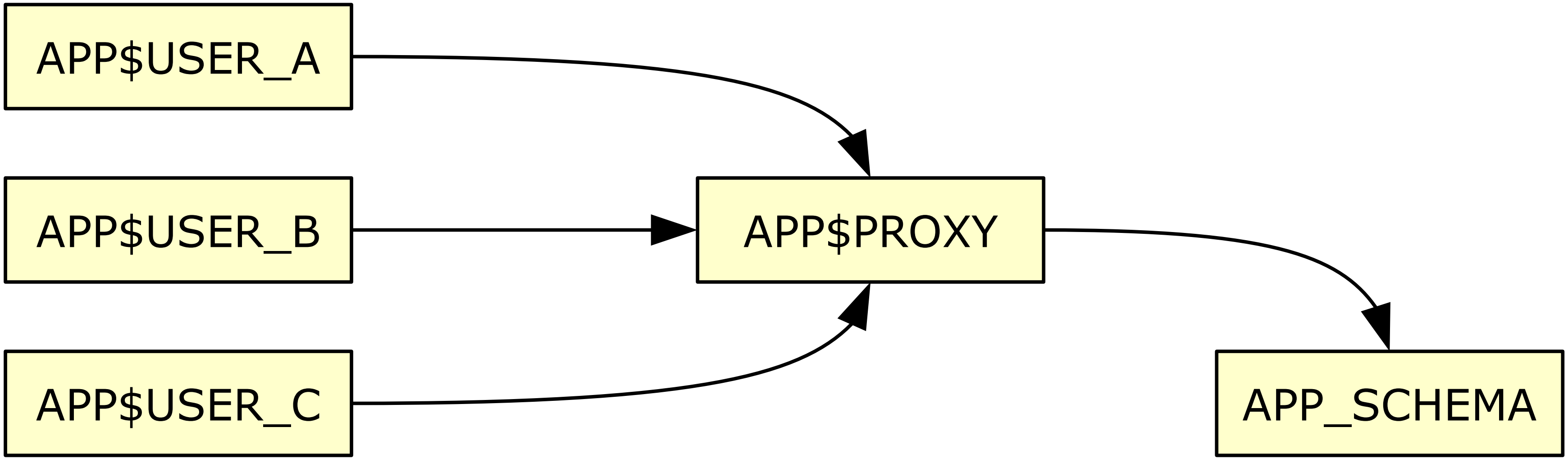
# Oracle Proxy Users (grants)

```
grant create session to app$proxy;

grant create session to apr$user;

alter user app$user_a
    grant connect through app$proxy;

alter user app$user_b
    grant connect through app$proxy;

alter user app$user_c
    grant connect through app$proxy;
```

# Oracle Role Grants

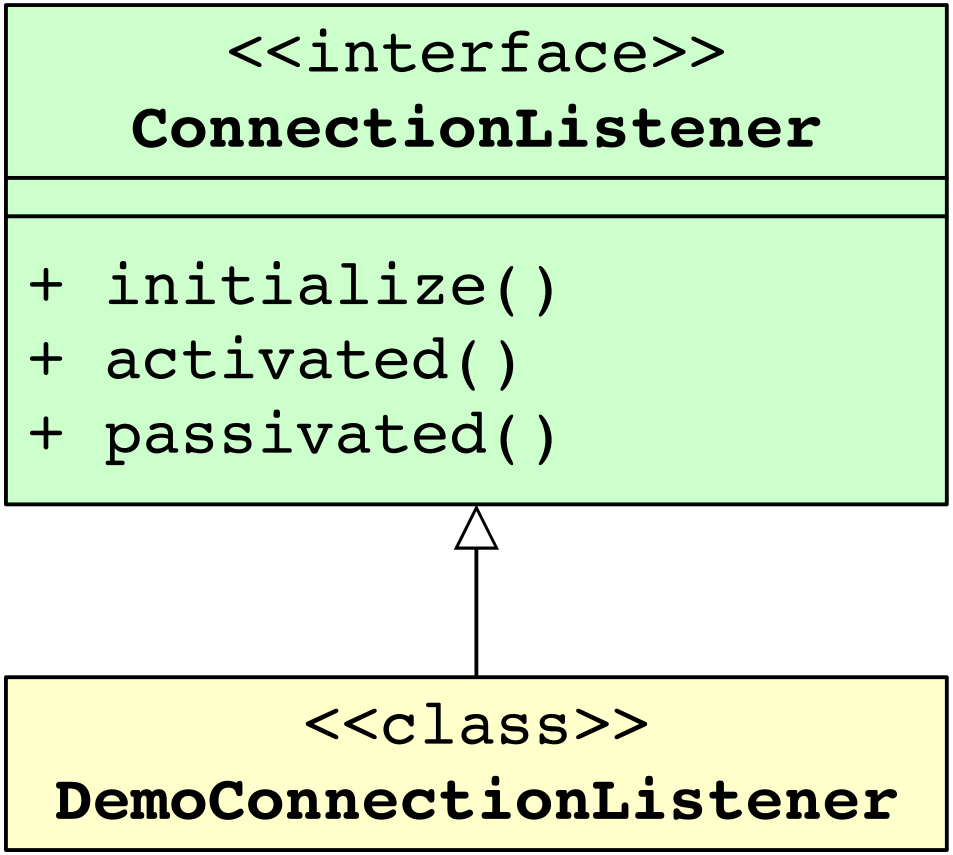
```
grant apr$user to apr$admin;

grant apr$admin to app$user_a;

grant apr$user to app$user_b;

grant apr$user to app$user_c;
```

# Oracle Setup Overview



```
$ sqlplus app$proxy[app$user_a]/app$proxy
SQL> alter session set current_schema=app_schema;
```

# JDBC Connection Listener Implementation



<<interface>>
**ConnectionListener**

+ initialize()
+ activated()
+ passivated()

<<class>>
**DemoConnectionListener**

**Maven Coordinates:**
org.jboss.ironjacamar:ironjacamar-jdbc

# JDBC Connection Listener Deployment

```
data-source add \
 --name=MyDemoDataSource \
 --jndi-name=java:jboss/datasources/MyDemoDataSource \
 --driver-name=oracle \
 --connection-url= \
   jdbc:oracle:thin:@//your.host.com/service \
 --user-name=app\$proxy \
 --password=my_proxy_pass \
 --connection-listener-class=\
   com.abakus.lib.oraproxy.OraProxyConnectionListener \
 --connection-listener-property={\
   "currentSchema"=>"MY_APP" \
 }
```

# JDBC: Oracle Proxy Sessions

```
oracle.jdbc.OracleConnection conn;

conn.openProxySession()

conn.close(
    OracleConnection.PROXY_SESSION
);
```

# Security Context

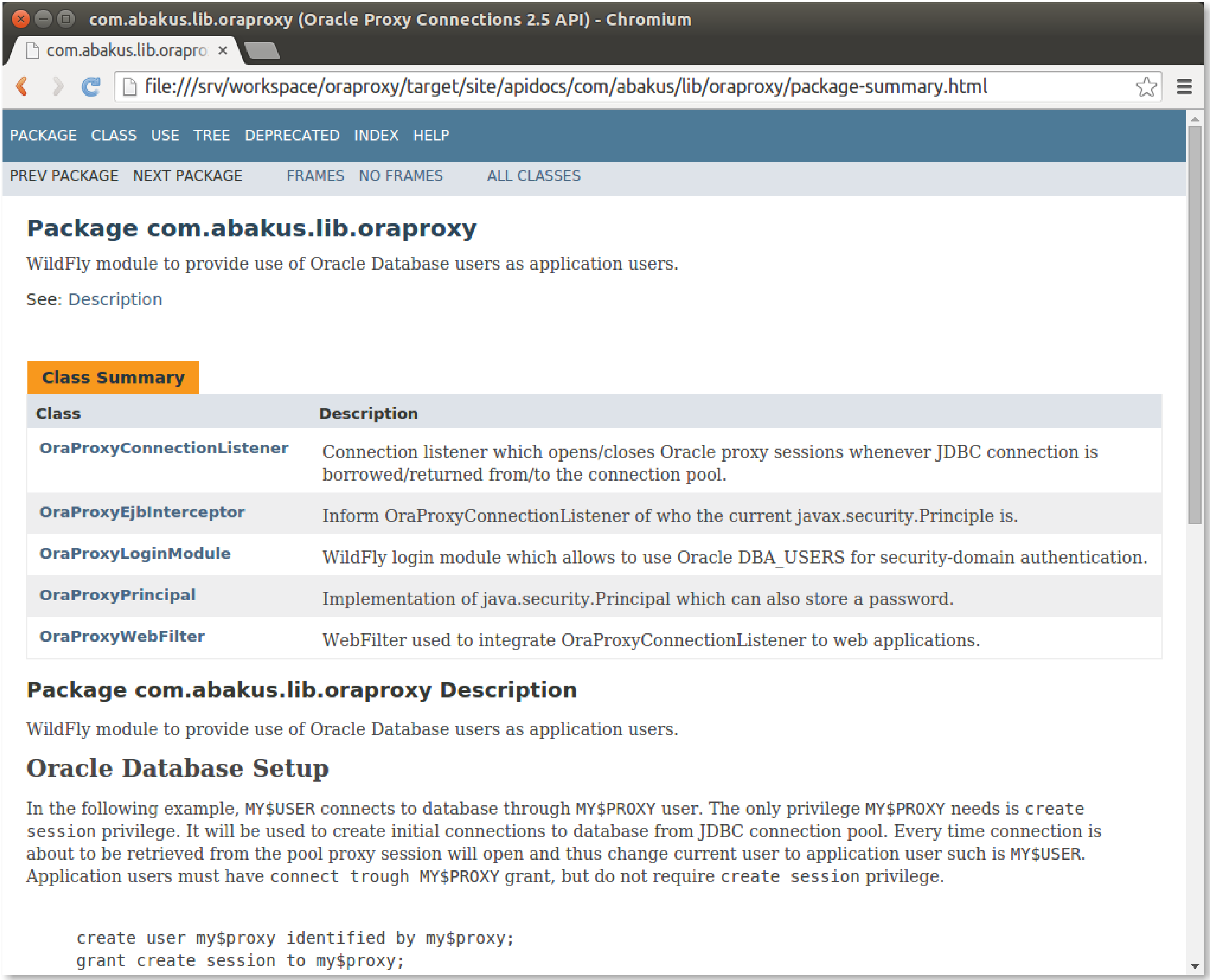
- How to obtain javax.security.Principal in JDBC Connection Listener?!
- **`ThreadLocal<Principal>`**

*Does that really work?*

- Sure
  - `@WebFilter`
  - `@AroundInvoke`
- Uhm.. What about
  - `@Asynchronous`
  - `@Timeout`
  - …

# What About “AUTHENTICATION REQUIRED” ?

- Wrap javax.security.Principal to include password.


com.abakus.lib.oraproxy (Oracle Proxy Connections 2.5 API) - Chromium
com.abakus.lib.orapro ×
file:///srv/workspace/oraproxy/target/site/apidocs/com/abakus/lib/oraproxy/package-summary.html
PACKAGE CLASS USE TREE DEPRECATED INDEX HELP
PREV PACKAGE NEXT PACKAGE FRAMES NO FRAMES ALL CLASSES

# Package com.abakus.lib.oraproxy

WildFly module to provide use of Oracle Database users as application users.

See: Description

**Class Summary**

| Class | Description |
|---|---|
| OraProxyConnectionListener | Connection listener which opens/closes Oracle proxy sessions whenever JDBC connection is borrowed/returned from/to the connection pool. |
| OraProxyEjbInterceptor | Inform OraProxyConnectionListener of who the current javax.security.Principle is. |
| OraProxyLoginModule | WildFly login module which allows to use Oracle DBA_USERS for security-domain authentication. |
| OraProxyPrincipal | Implementation of java.security.Principal which can also store a password. |
| OraProxyWebFilter | WebFilter used to integrate OraProxyConnectionListener to web applications. |

## Package com.abakus.lib.oraproxy Description

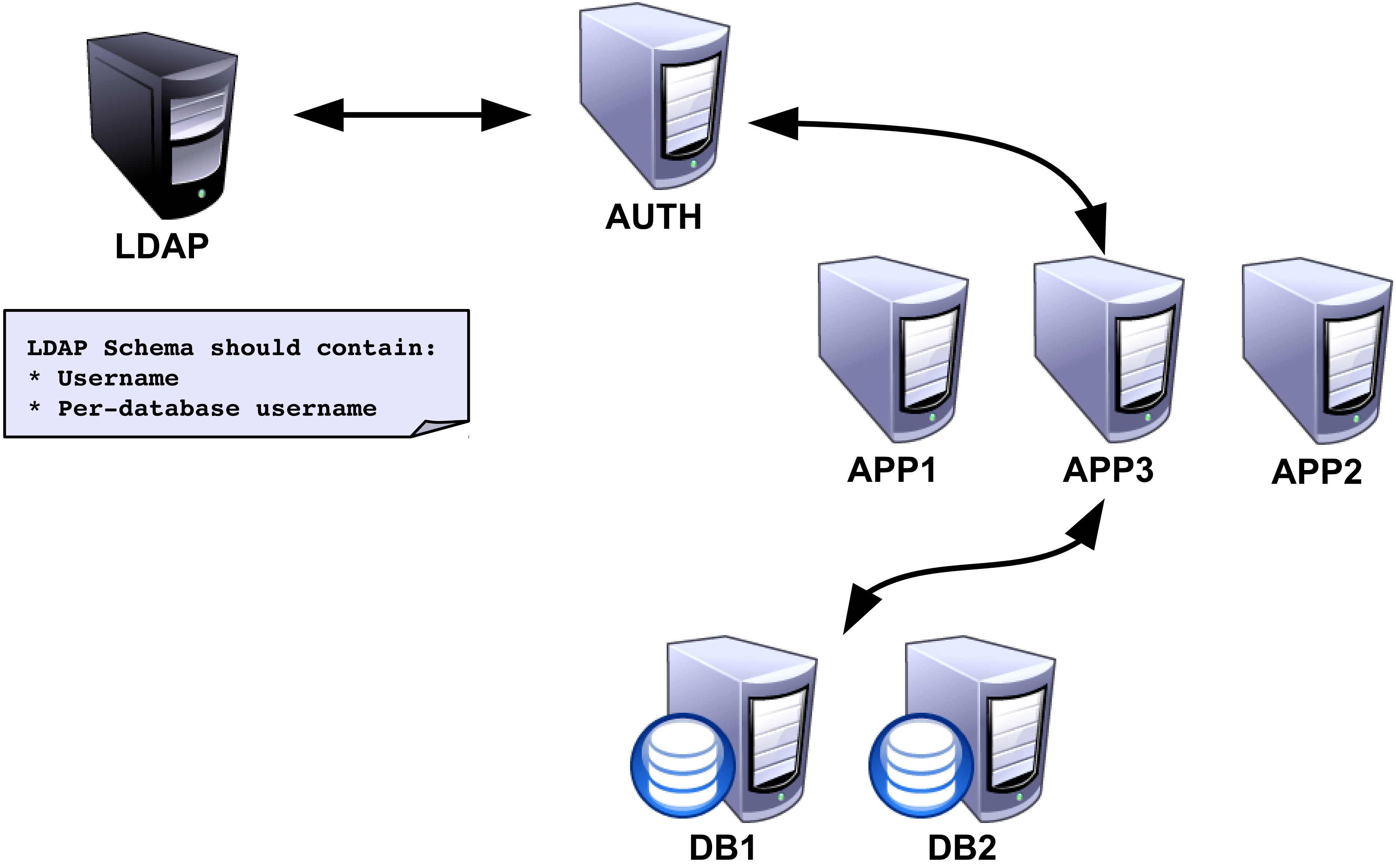
WildFly module to provide use of Oracle Database users as application users.

## Oracle Database Setup

In the following example, `MY$USER` connects to database through `MY$PROXY` user. The only privilege `MY$PROXY` needs is `create session` privilege. It will be used to create initial connections to database from JDBC connection pool. Every time connection is about to be retrieved from the pool proxy session will open and thus change current user to application user such is `MY$USER`. Application users must have `connect trough MY$PROXY` grant, but do not require `create session` privilege.

```
    create user my$proxy identified by my$proxy;
    grant create session to my$proxy;
```

# Single Sign On

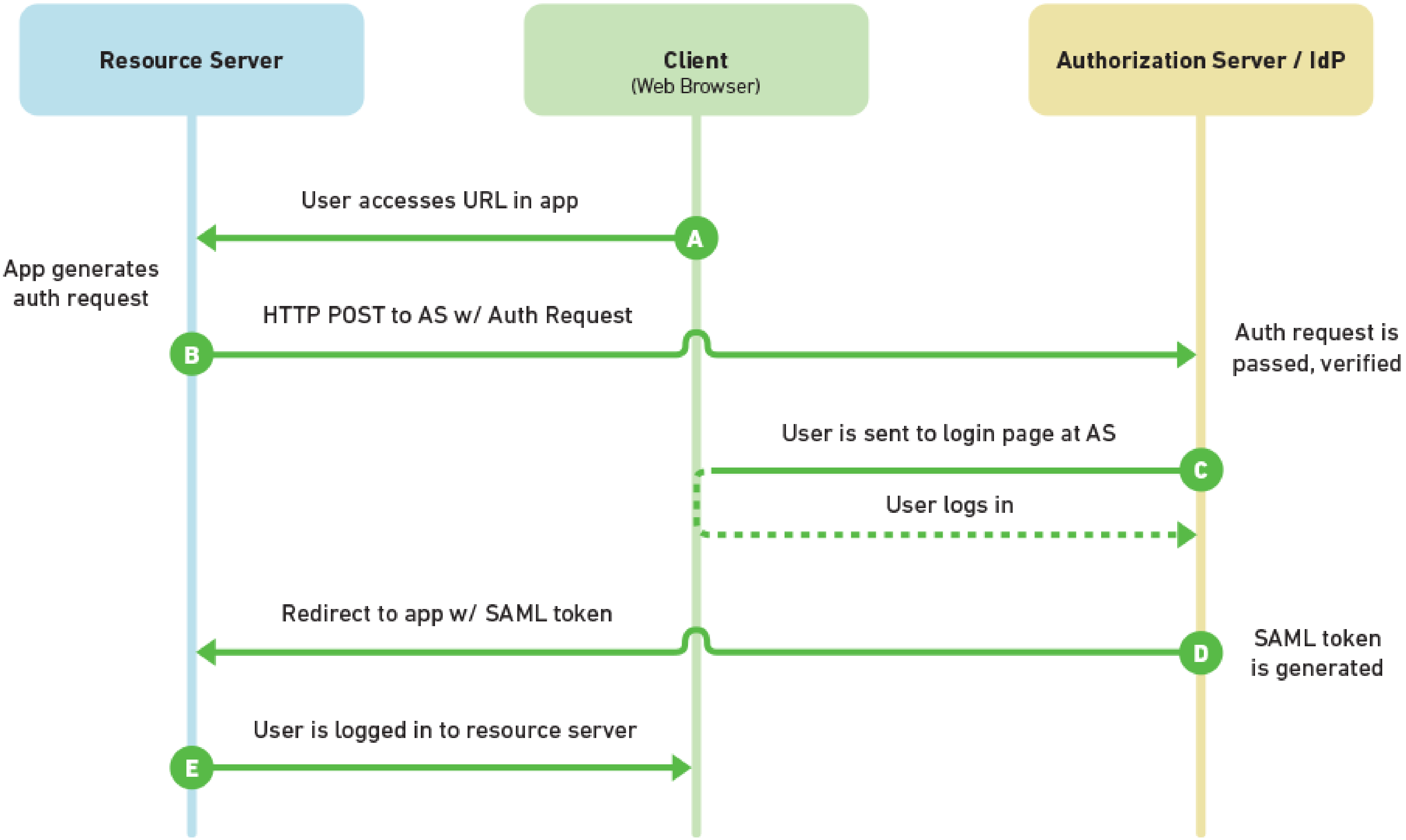


LDAP Schema should contain:
* Username
* Per-database username

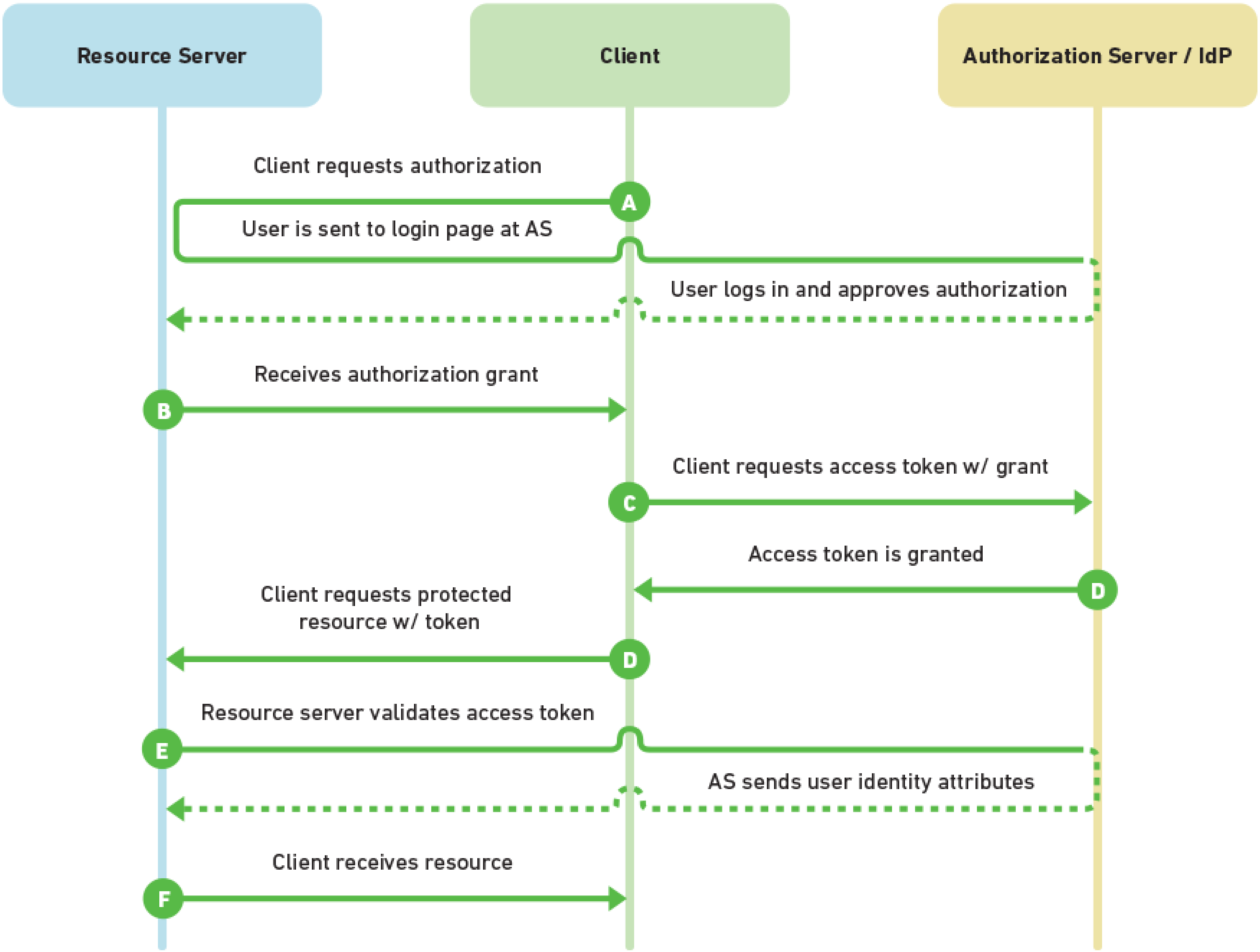
# SAML? OAuth? CAS? OpenID? AD? …?

- SSO vs “WebSSO”

# SAML 2.0 Flow


Resource Server
Client
(Web Browser)
Authorization Server / IdP
User accesses URL in app
A
App generates auth request
HTTP POST to AS w/ Auth Request
B
Auth request is passed, verified
User is sent to login page at AS
C
User logs in
Redirect to app w/ SAML token
D
SAML token is generated
User is logged in to resource server
E

# OAuth 2.0 Flow


Resource Server
Client
Authorization Server / IdP
Client requests authorization
A
User is sent to login page at AS
User logs in and approves authorization
Receives authorization grant
B
Client requests access token w/ grant
C
Access token is granted
D
Client requests protected resource w/ token
D
Resource server validates access token
E
AS sends user identity attributes
Client receives resource
F

# PicketLink Overview

- PicketLink is an umbrella project for security and identity management for Java Applications.
  - Java EE Application Security
  - Identity Management
  - Federation (SAML, OAuth, OpenID, ...)
  - Social Login (Facebook, Twiter, Google)
  - Mobile Applications Security
  - REST Applications Security

- Quickstart examples! =)

# Identity Provider

- Create security-domain
- Create new web-app
  - pom.xml - manifest deps: org.picketlink
  - web.xml
    - Configure container managed authentication
    - IDPHttpSessionListener
    - IDPFilter
  - **picketlink.xml**
    - SAML specific configuration
      - idp url, trusted domains, ...

# Service Provider

- Create security-domain
  - **SAML2LoginModule**
- picketlink.xml
  - IDP URL, SP URL
  - Keystore parameters

?