dCache integration into site infrastructure

(or yet another 'gPlazma Talk')

Tigran Mkrtchyan for dCache Team

Typical Site config

```
# == grid-vorolemap
"*" "/atlas" atlasusr001


# == storage-authzdb
version 2.1
authorize atlasusr001 read-write 40001 4000  / / /
```

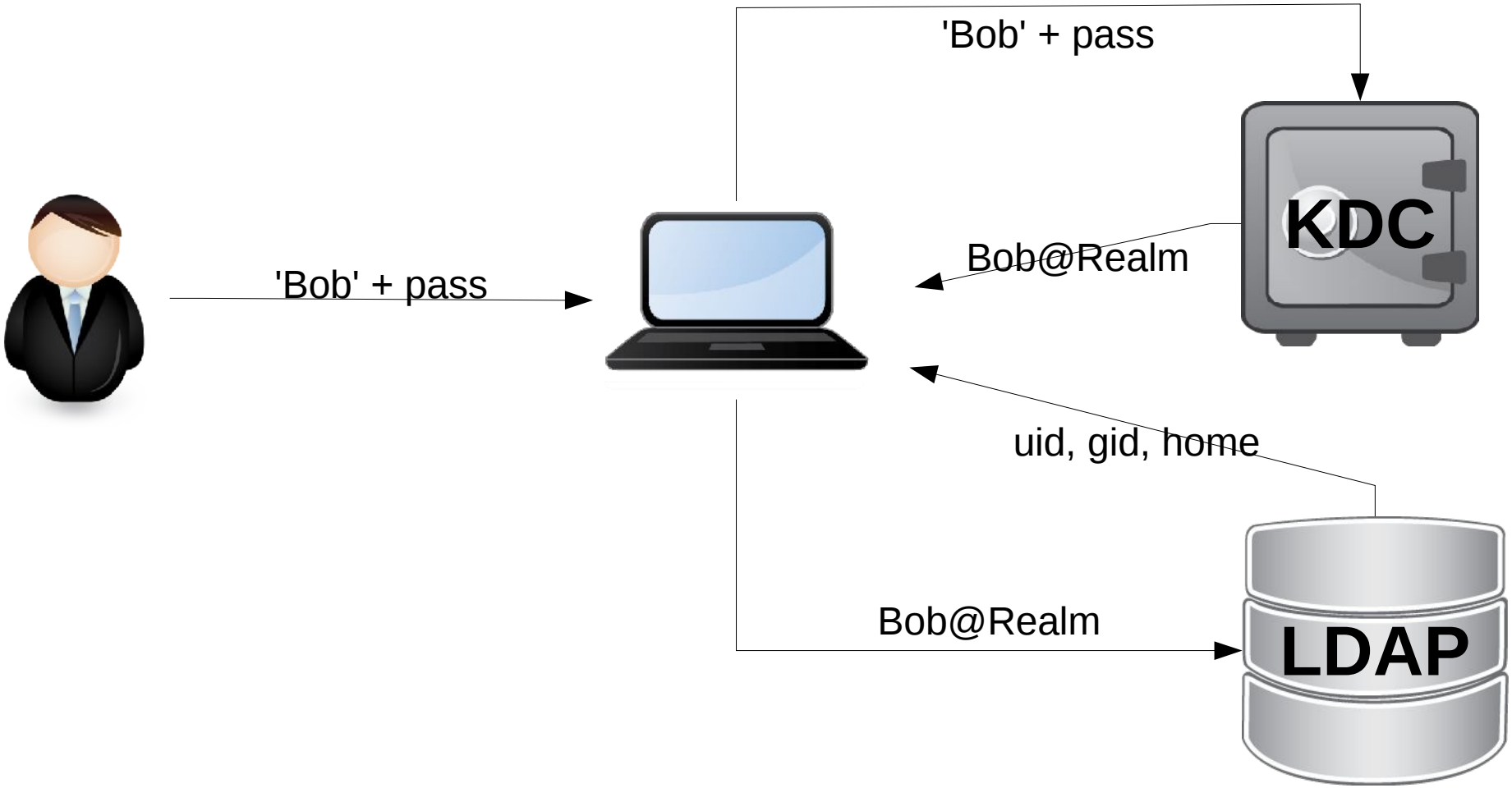
Why we need something else?

- Non Grid/WLCG customers
- Local users and groups
  - Username/password access
  - NFS access
- Central user management

Site setups

- Kerberos5
- Active Directory (LDAP + Krb5)
- LDAP
- NIS

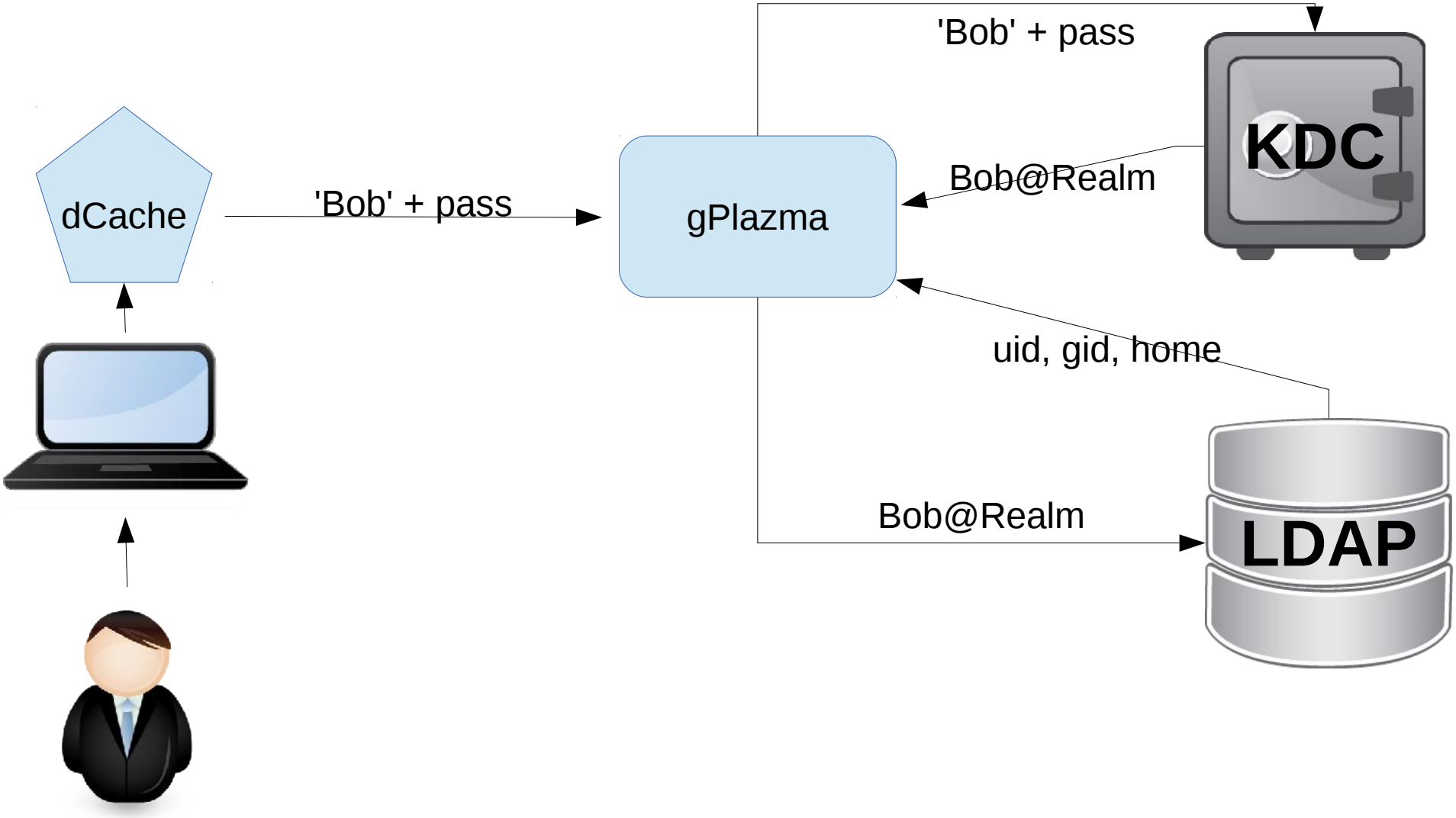
Username/pass



Login process on Unix

- User name + pass validation
- Kerberos Ticket generation
- Querying LDAP for attributes

Username/pass + dCache



Login with gPlazma

```
# gplazma.conf
auth     optional jaas
map      optional krb5
map      optional ldap
session  optional ldap
```

Login with gPlazma

```
# gplazma.conf
auth     optional jaas { user, pass } => { user@REALM }
map      optional krb5
map      optional ldap
session  optional ldap
```

Login with gPlazma

```
# gplazma.conf
auth     optional jaas {user, pass} => {user@REALM}
map      optional krb5 {user@REALM} => {user}
map      optional ldap
session  optional ldap
```

Login with gPlazma

```
# gplazma.conf
auth     optional jaas {user, pass} => {user@REALM}
map      optional krb5 {user@REALM} => {user}
map      optional ldap {user} => {uid, gids}
session optional ldap
```

Login with gPlazma

```
# gplazma.conf
auth     optional jaas {user, pass} => {user@REALM}
map      optional krb5 {user@REALM} => {user}
map      optional ldap {user} => {uid, gids}
session optional ldap {uid, gids } => {uid, gids, home, +}
```

Login with gPlazma

```
# gplazma.conf
auth     optional jaas {user, pass} => {user@REALM }
map      optional krb5 {user@REALM} => {user}
map      optional ldap {user} => {uid, gids}
session optional ldap {uid, gids} => {uid, gids, home,  root, +}
```

{ user, uid, gids, home, root, + }

Adding GRID plugins

- x509 (**auth**)
  - validates user's grid certificate, CRLs
- voms (**auth**)
  - validates voms extensions, CRLs
- vorolemap (**map**)
  - uses **/etc/grid-security/grid-vorolemap**
- authzdb (**map**, **session**)
  - uses **/etc/grid-security/storage-authzdb**

Check our book for more plugins and configurations

+GRID

```
# gplazma.conf
auth     optional jaas
map      optional krb5
map      optional ldap
session  optional ldap
```

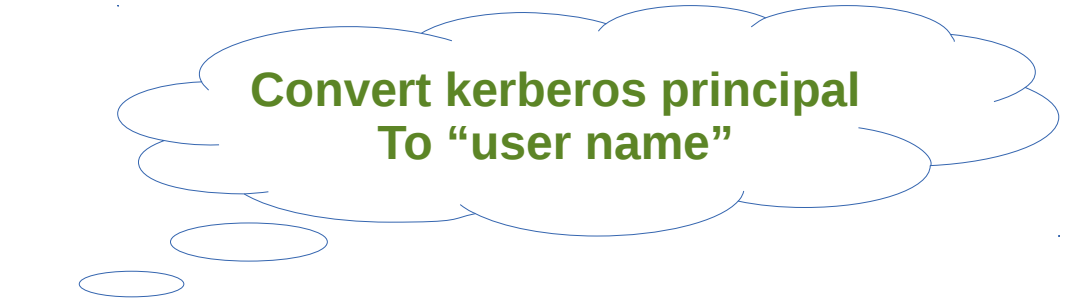
+GRID

```
# gplazma.conf
auth     optional jaas
auth     optional x509
auth     optional voms
map      optional krb5
map      optional ldap
session  optional ldap
```

If user comes with password
Or x509 certificate and VOMS

+GRID

```
# gplazma.conf
auth     optional jaas
auth     optional x509
auth     optional voms
map      optional krb5
map      optional ldap
session  optional ldap
```



+GRID

```
# gplazma.conf
auth     optional jaas
auth     optional x509
auth     optional voms
map      optional krb5
map      optional vorolemap
map      optional ldap
session  optional ldap
```

If there is a mapping
from DN and VOMS to “user name”
Take it into account

+GRID

```
# gplazma.conf
auth     optional jaas
auth     optional x509
auth     optional voms
map      optional krb5
map      optional vorolemap
map      sufficient ldap
session  optional ldap
```

If I can get uid and gids from ldap,
Stop here

+GRID

gplazma.conf

auth optional **jaas**

auth optional **x509**

auth optional **voms**

map optional **krb5**

map optional **vorolemap**

map sufficient **ldap**

map sufficient **authzdb**

session optional **ldap**

+GRID

```
# gplazma.conf
auth     optional jaas
auth     optional x509
auth     optional voms
map      optional krb5
map      optional vorolemap
map      sufficient ldap
map      sufficient authzdb
session  sufficient ldap
session  sufficient authzdb
```

Query for home and other attrs.

Krb5 + LDAP + VOROLE

```
# gplazma.conf
auth     optional jaas
auth     optional x509
auth     optional voms
map      optional krb5
map      optional vorolemap
map      sufficient ldap
map      sufficient authdb
session  sufficient ldap
session  sufficient authdb
```

JAAS Plugin

- Interface to 'The gPlazma of Java'
  - Plugable architecture
  - Allows integration with 3-rd party services
- Uses so called 'Login Module'
  - Consumes Credentials (user name + pass)
  - Produces Principal (loged in user)
- Requires jaas config file (/etc/dcache/jgss.conf)

```
# jgss.conf
LdapGplazma {
   com.sun.security.auth.module.LdapLoginModule REQUIRED
       userProvider="ldap://ldapsrv/ou=...,o=...,c=..."
       userFilter="(uid={USERNAME})"
};
Krb5Gplazma {
    com.sun.security.auth.module.Krb5LoginModule REQUIRED
    debug=false
    isInitiator=false
    useTicketCache=false;
};

# gplazma.conf
auth optional  jaas gplazma.jaas.name=LdapGplazma
auth optional  jaas gplazma.jaas.name=Krb5Gplazma
```

Notice!

- jaas-krb5 requires local Kerberos config
  - make sure /etc/krb5.conf is correct
  - make sure dcache can read /etc/krb5.keytab
- ldap/nis do not require any additional configs
  - natively talking nidldap protocols.

Tell gPlazma what you get

gplazma.conf

map optional **mutator**

gplazma.mutator.**accept**=

com.sun.security.auth.UserPrincipal

gplazma.mutator.**produce**=*username*

map optional **ldap**