dCache.org

gPlazma2: Plugins and Configuration

Karsten Schwank

Zeuthen, 17.4.2012

Fermilab NDGF Nordic DataGrid Facility EUROPEAN MIDDLEWARE INITIATIVE DESY HELMHOLTZ ASSOCIATION

Overview

- Basics
- Plugins
- Migrating from v1 to v2
- Introducing Argus
- Introducing Kerberos
- Examples
  - The WLCG Case
  - Using Kerberos and NIS
- Summary

Basics

Authorization with gPlazma2 is

- A 4 step process
  - Authenticate – "Who are we talking to?"
  - Map – "How does the authenticated user fit into our site?"
  - Account – "Is the account currently banned?"
  - Session – "What is the user allowed to access?"

Configuration of gPlazma2 is

- Done via the file `/etc/dcache/gplazma.conf`

Step 1: Authentication (auth)

Who are we talking to?

- Pin "Principals" to the subject
- Plugins:
  - KPWD – dCache's own file based mechanism
  - VOMS – Virtual Organization Membership Service
  - X509 – X.509 certificate extractor
  - JAAS – Java Authentication and Authorization Service
  - XACML – Use a XACML server (e.g., GUMS)
  - gPlazma1 – Use old gPlazma

auth:kpwd

- KPWD

gplazma.kpwd.file [/etc/dcache/dcache.kpwd]

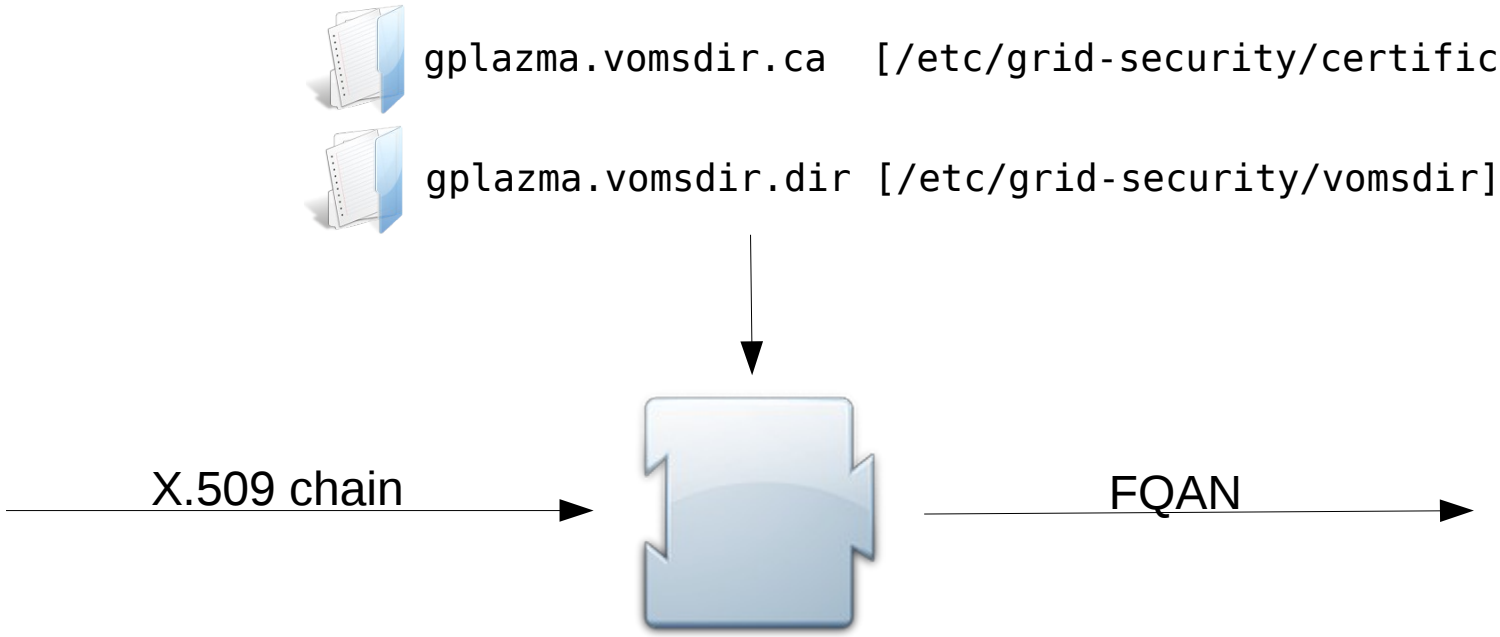
```
login behrmann read-write 1000 1000 /foo /bar /
  /O=Grid/O=NorduGrid/OU=ndgf.org/CN=Gerd Behrmann
  behrmann@ndgf.org

passwd behrmann aec59c36 read-write 1000 1000 / /
```

auth:x509

- X.509 certificate extractor

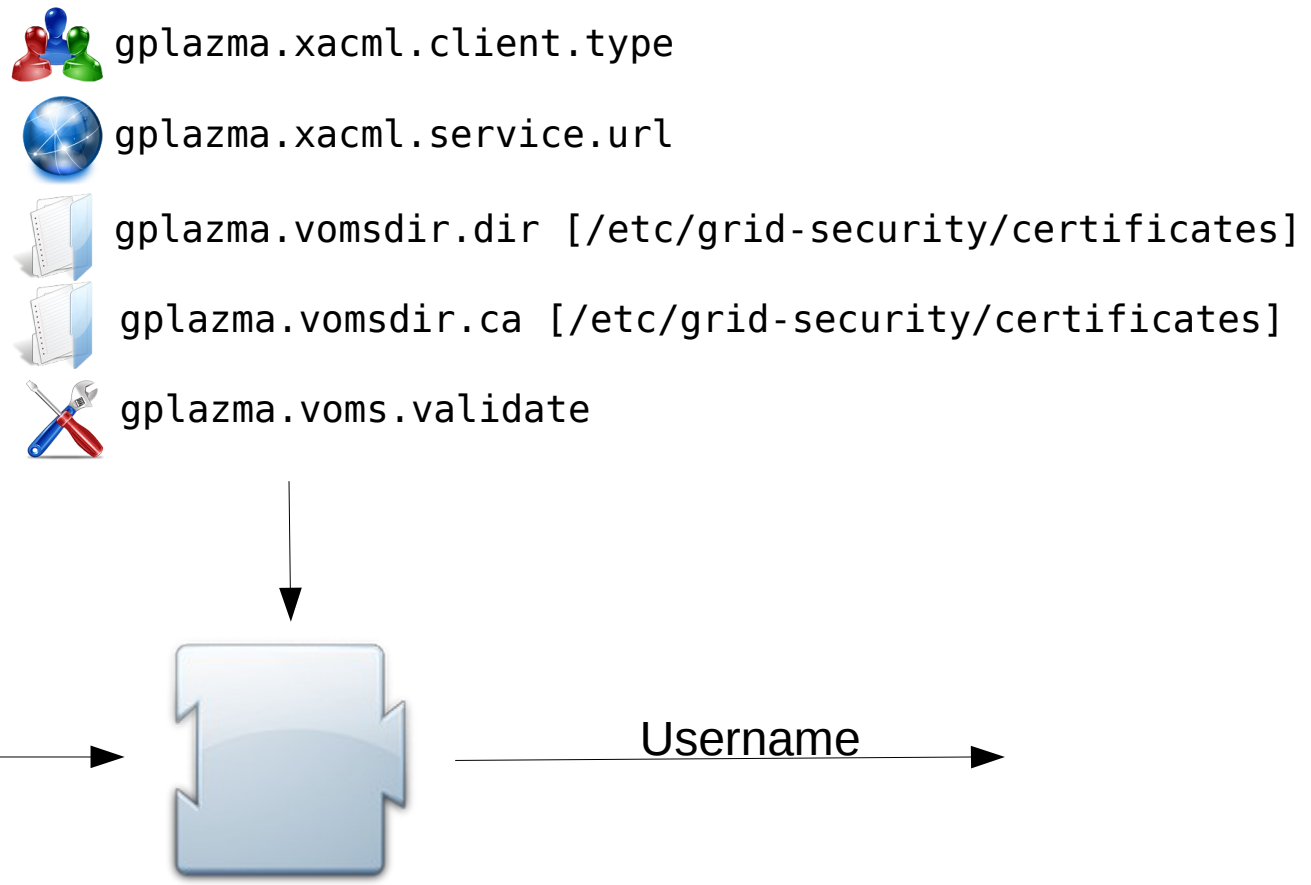
auth:voms

- Virtual Organization Membership Service



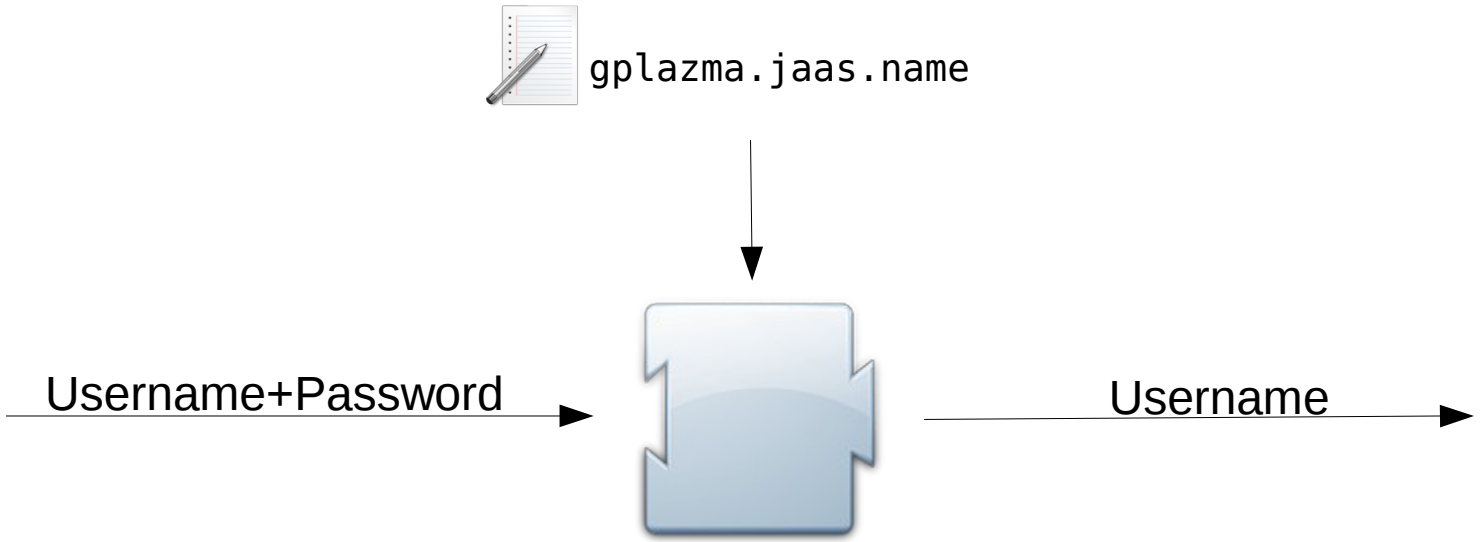
auth:xacml

- XACML



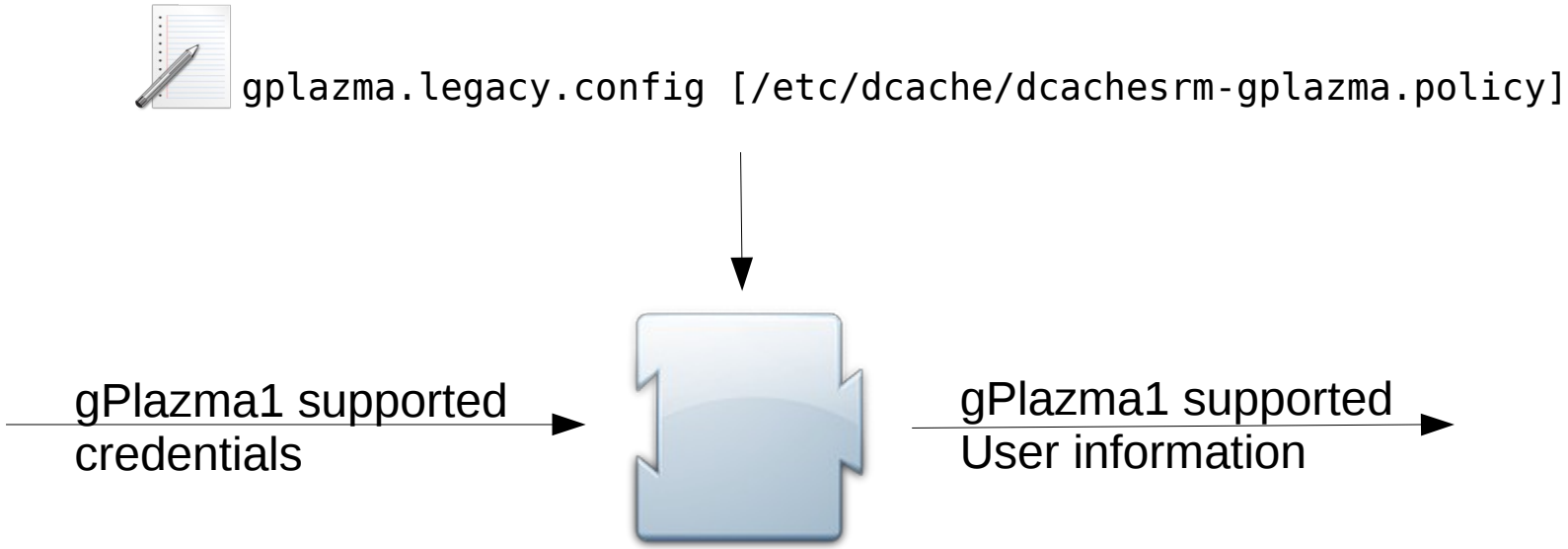
auth:jaas

- Java Authentication and Authorization Service



auth:gplazma1

- Use gPlazma1 as a plugin



Step 2: Mapping (map)

How does the authenticated user fit in our site?

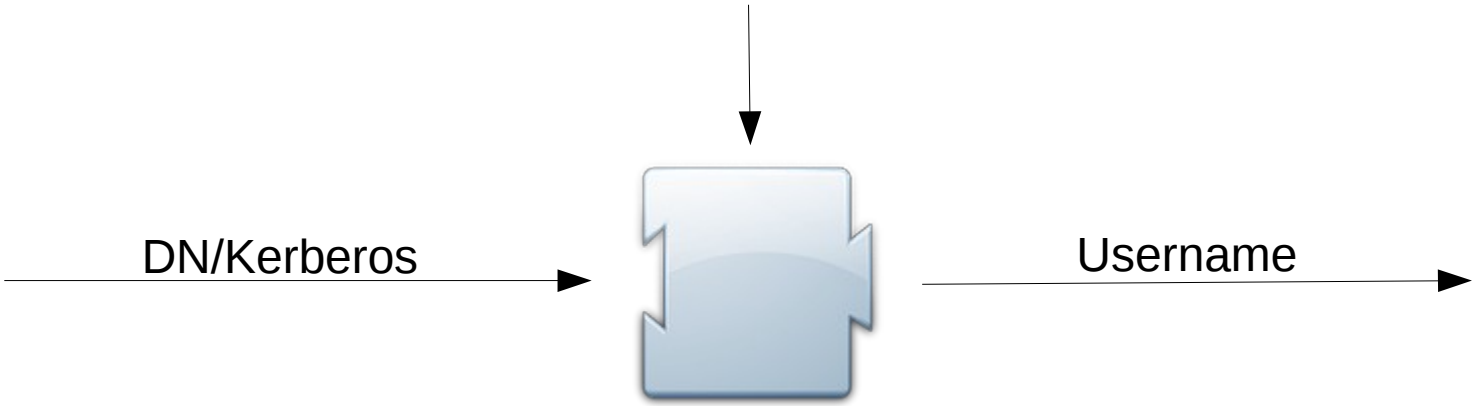
- Use the "principals" from auth step to assign a local name to the subject
- Plugins:
  - KPWD: dCache's file based solution
  - KRB5: Kerberos
  - NSSwitch: Username and Groupname
  - NIS: Network Information System
  - AuthzDB: Local file based solution
  - GridMap: Local file based solution
  - VoRoleMap: Local file based solution
  - gPlazma1

map:kpwd

- KPWD

gplazma.kpwd.file [/etc/dcache/dcache.kpwd]

```
mapping "/O=Grid/O=NorduGrid/OU=ndgf.org/CN=Gerd Behrmann" behrmann
```

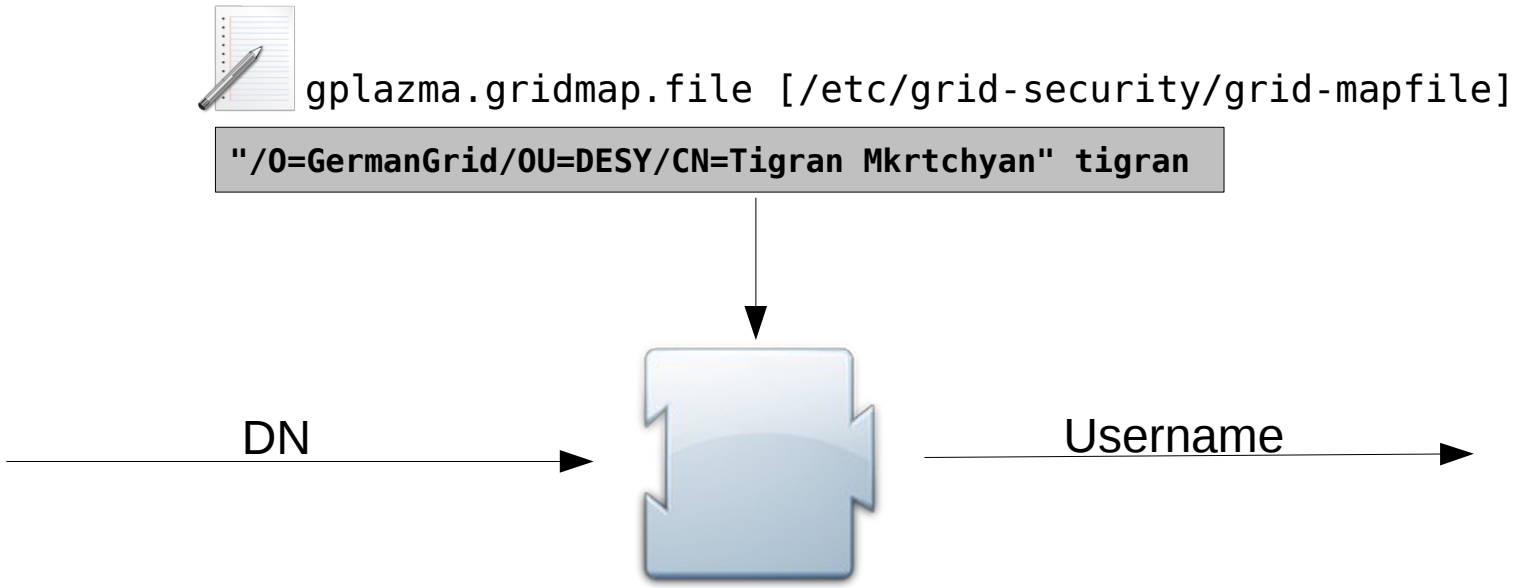


map:krb5

- Kerberos

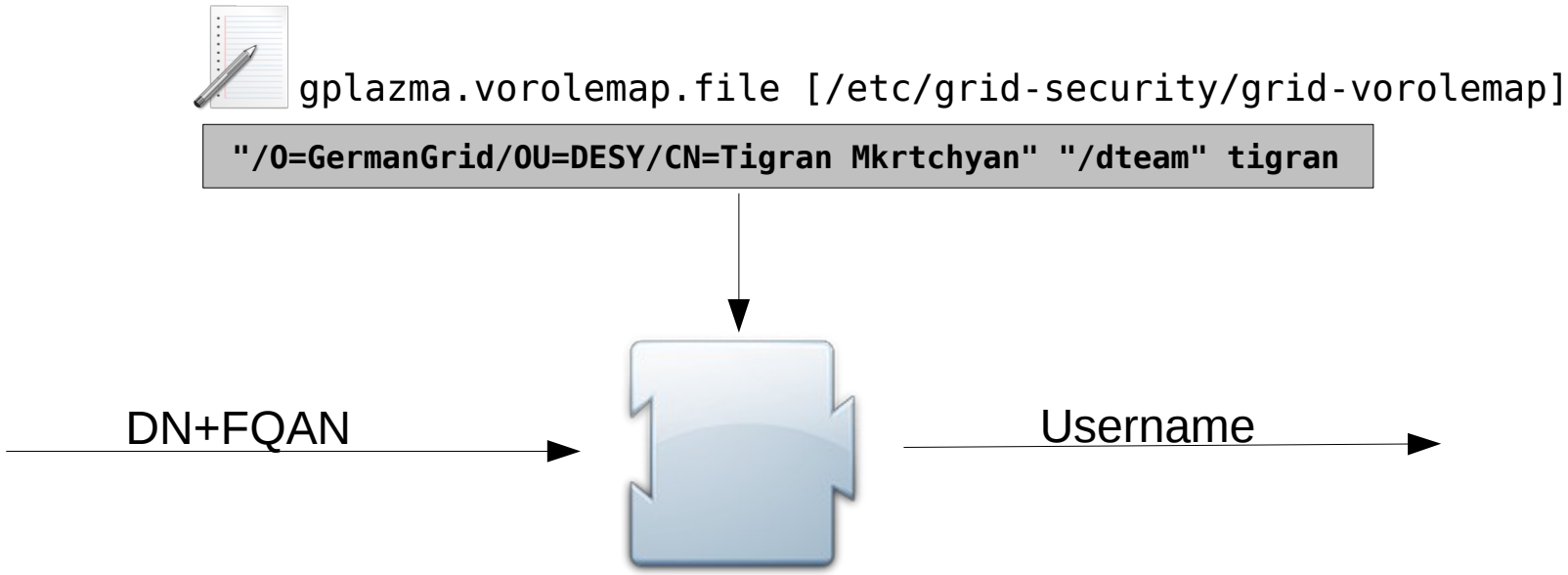
map:gridmap

- GridMap



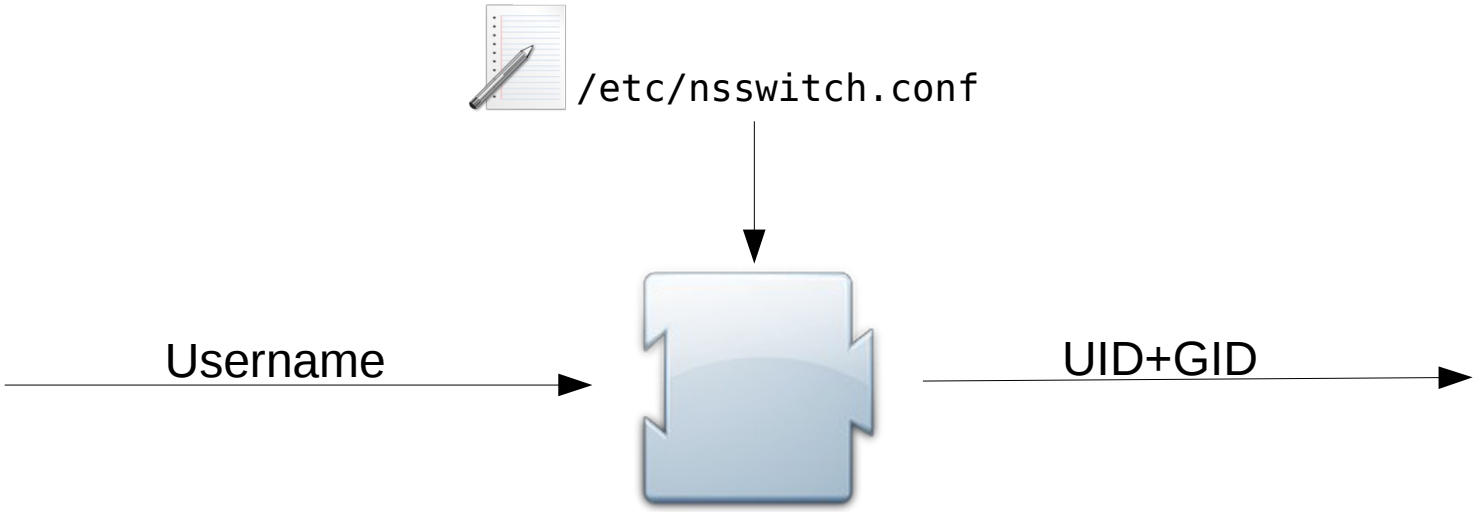
map:vorolemap

- VoRolemap



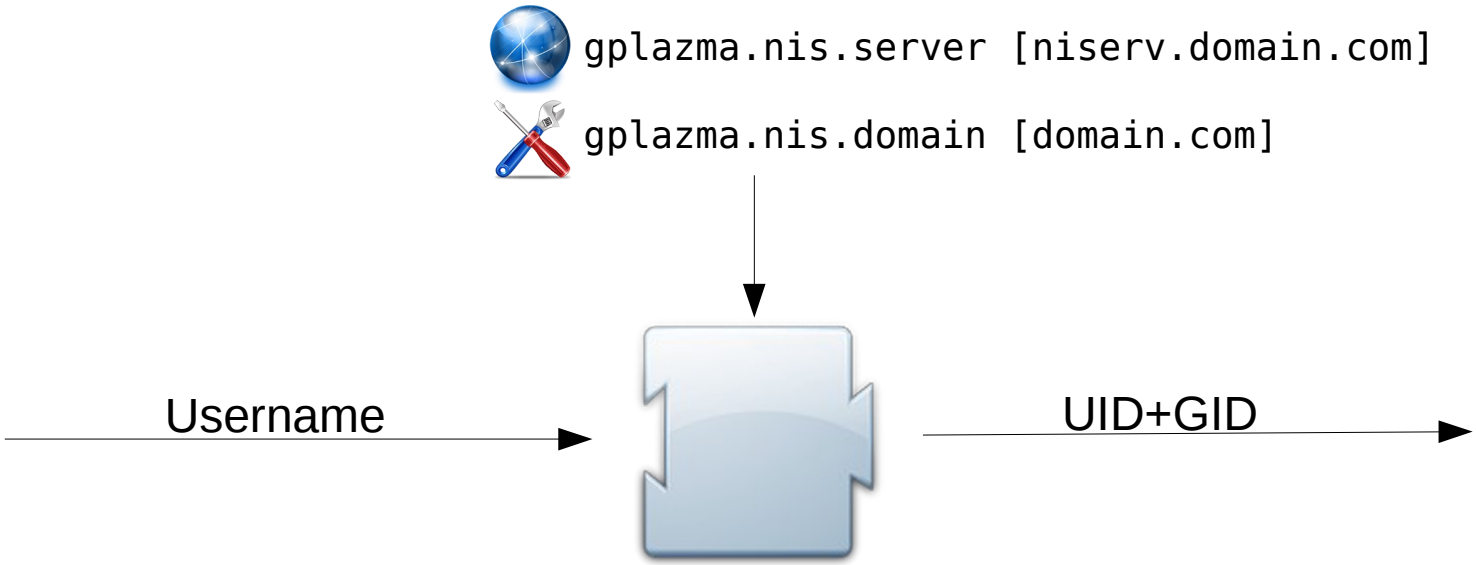
map:nsswitch

- NSSwitch



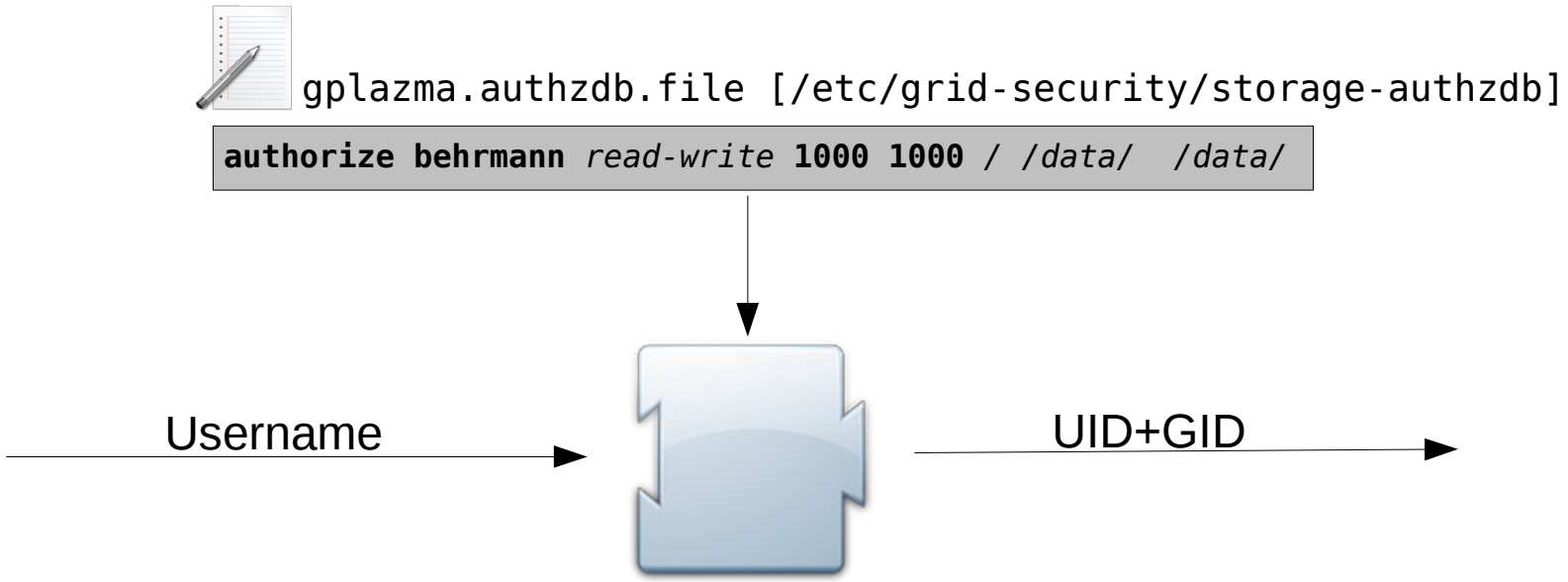
map:nis

- NIS


gplazma.nis.server [niserv.domain.com]
gplazma.nis.domain [domain.com]
Username
UID+GID

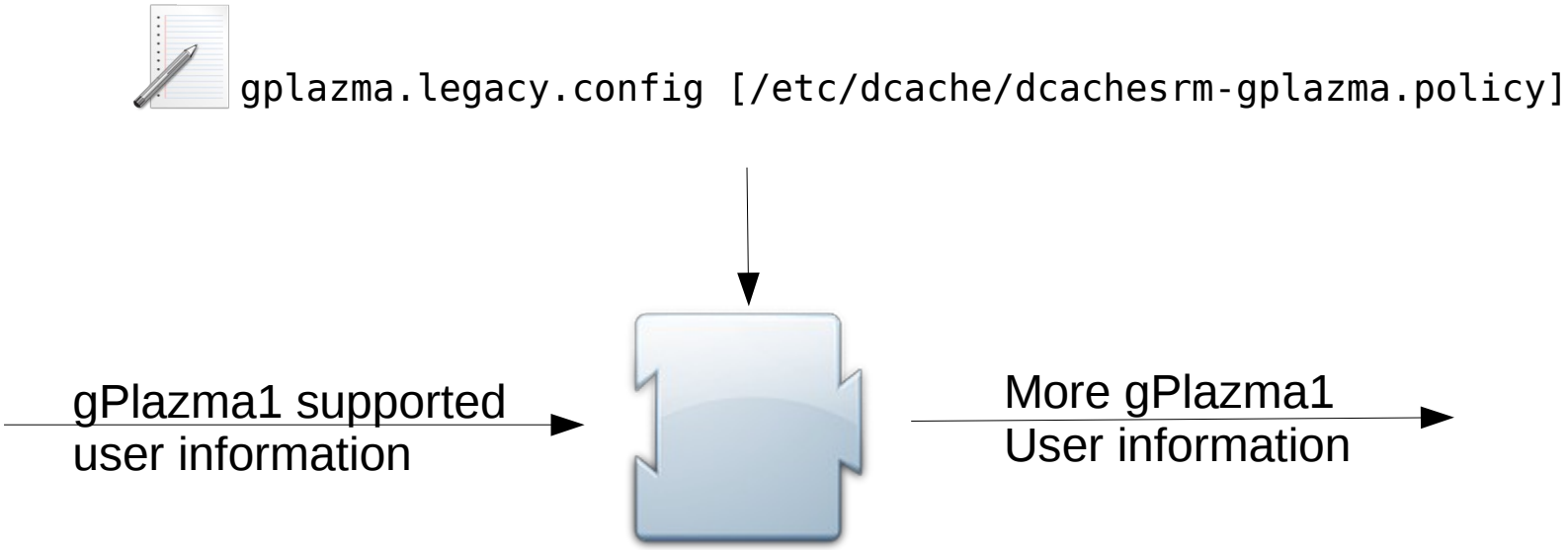
map:authzdb

- AuthzDB



map:gplazma1

- gPlazma1



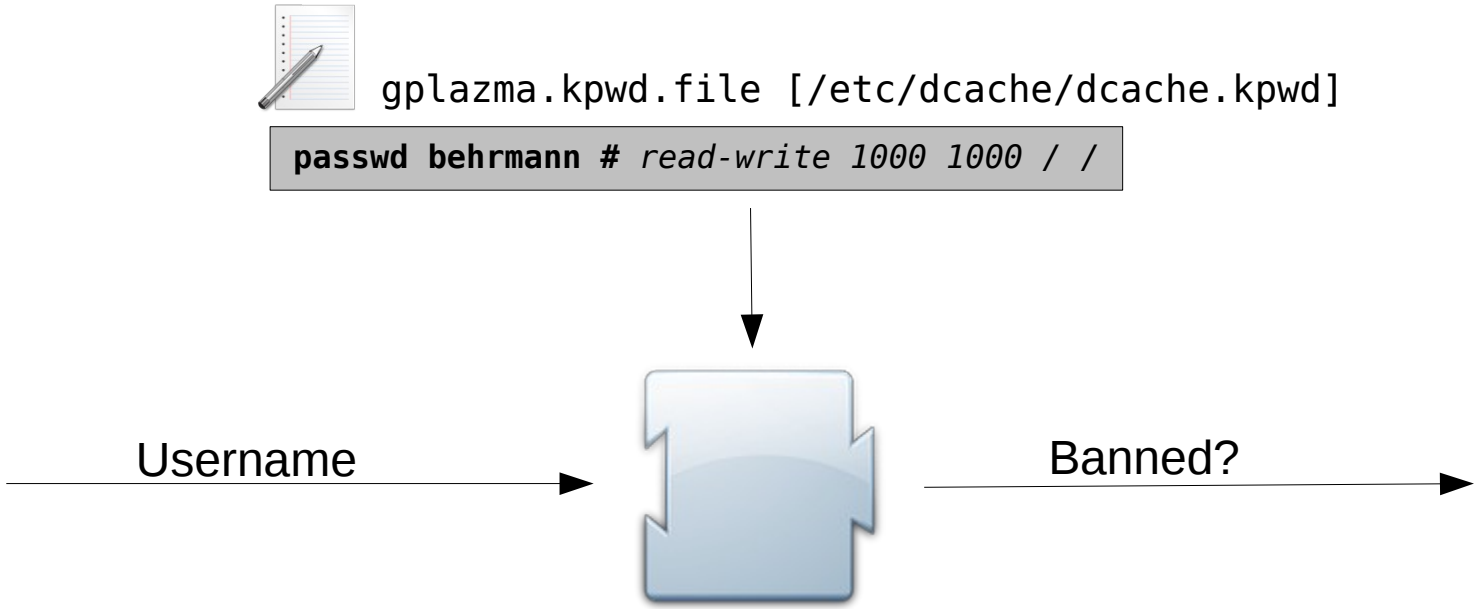
Step 3: Account

Is the account currently banned?

- Check if we have any reason not to allow the user to access our system
- Plugins:
  - KPWD: dCache's file based solution
  - Argus: a hierarchical centralized authentication and authorization service

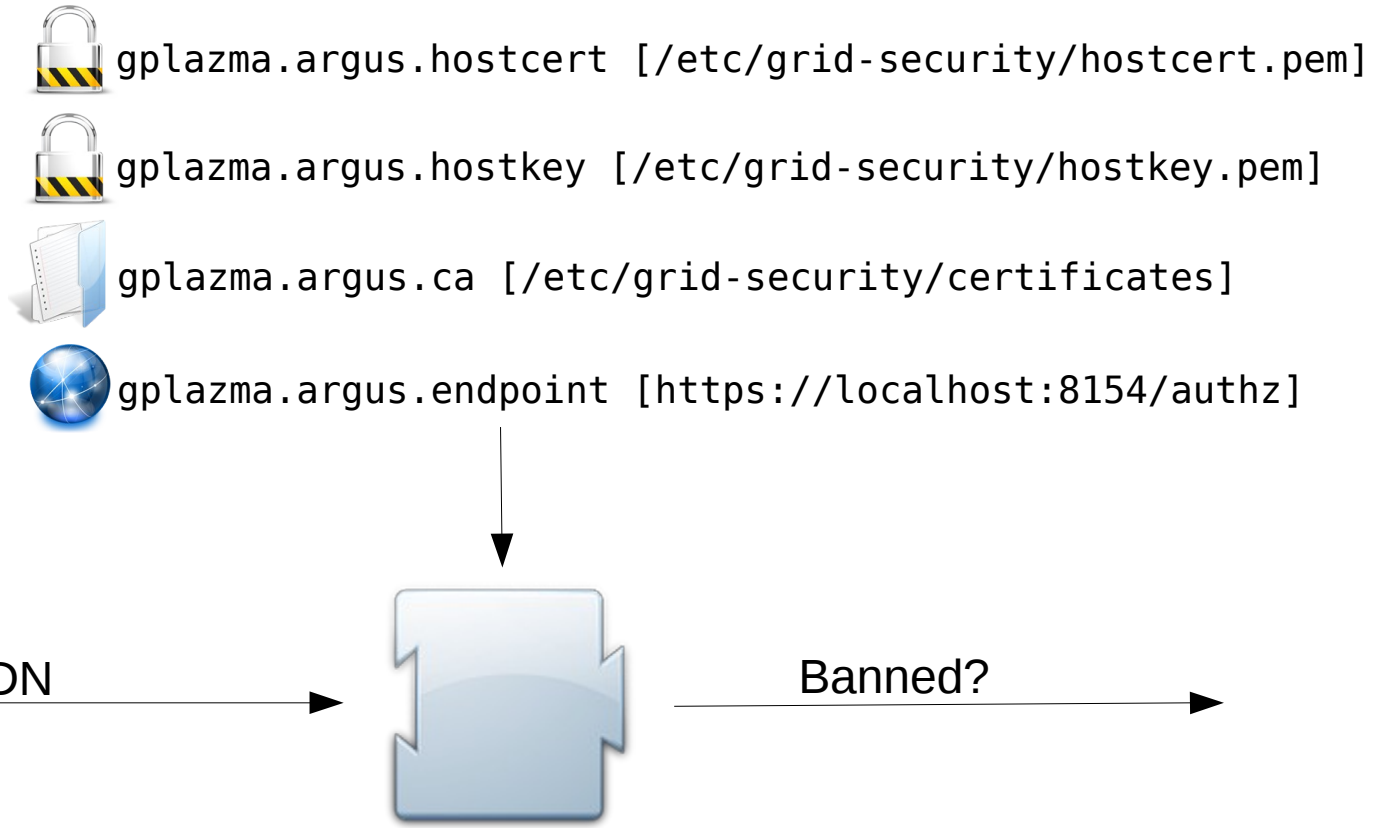
account:kpwd

- KPWD



account:argus

- Argus



Step 4: Session

What is the user allowed to access?

- Use the local name to assign home and root directory.
- Plugins:
  - KPWD: dCache's file based solution
  - NIS: Network Information System
  - NSSwitch: Name Service Switch
  - AuthzDB: Local file based solution
  - gPlazma1: Use old gPlazma as plugin

session:kpwd

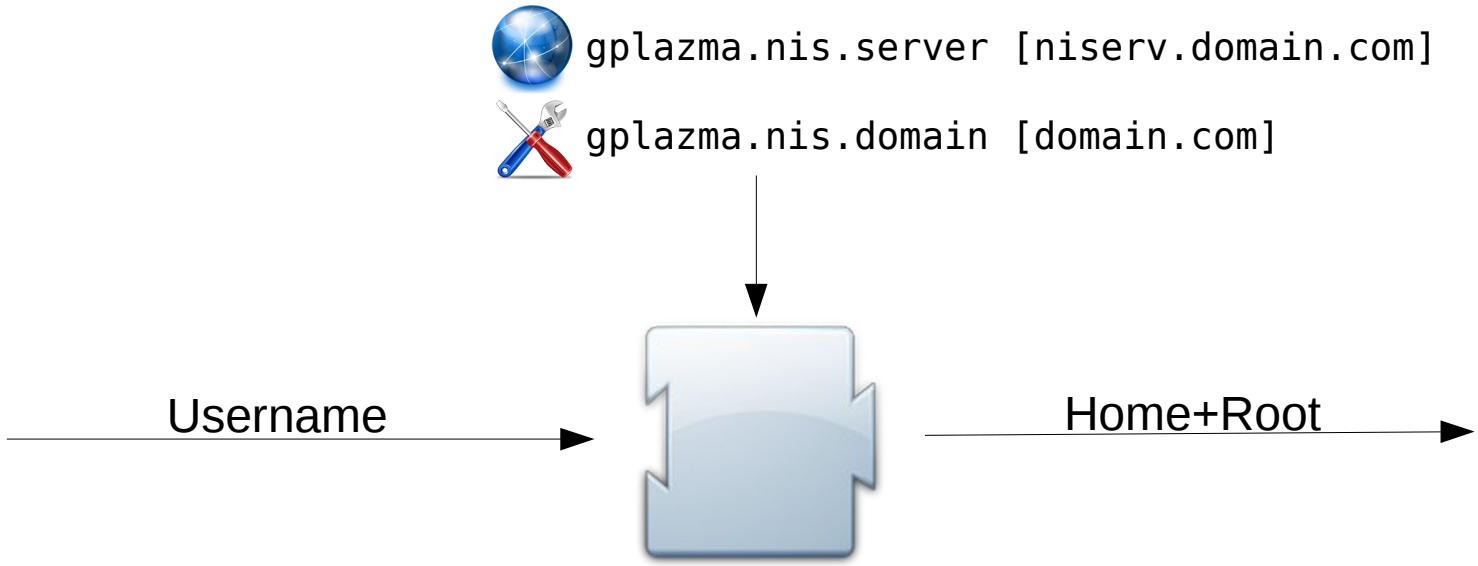
- KPWD

gplazma.kpwd.file [/etc/dcache/dcache.kpwd]

```
login behrmann read-write 1000 1000 /home /root /
  /O=Grid/O=NorduGrid/OU=ndgf.org/CN=Gerd Behrmann
  behrmann@ndgf.org

passwd behrmann aec59c36 read-write 1000 1000 / /
```

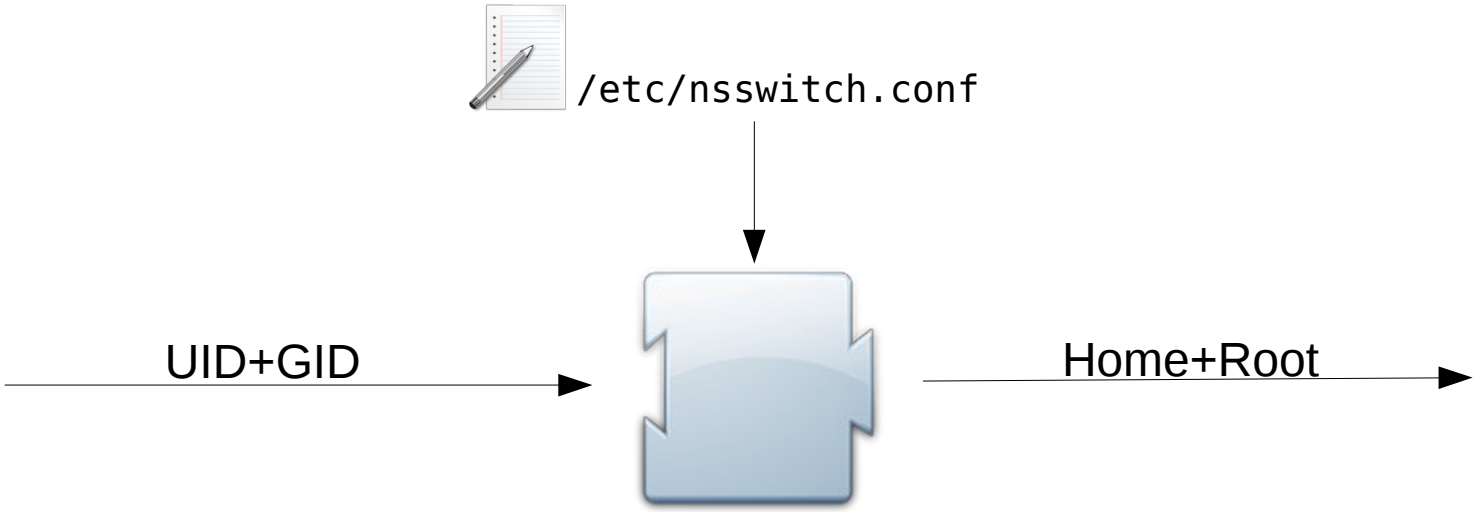
session:nis

- NIS



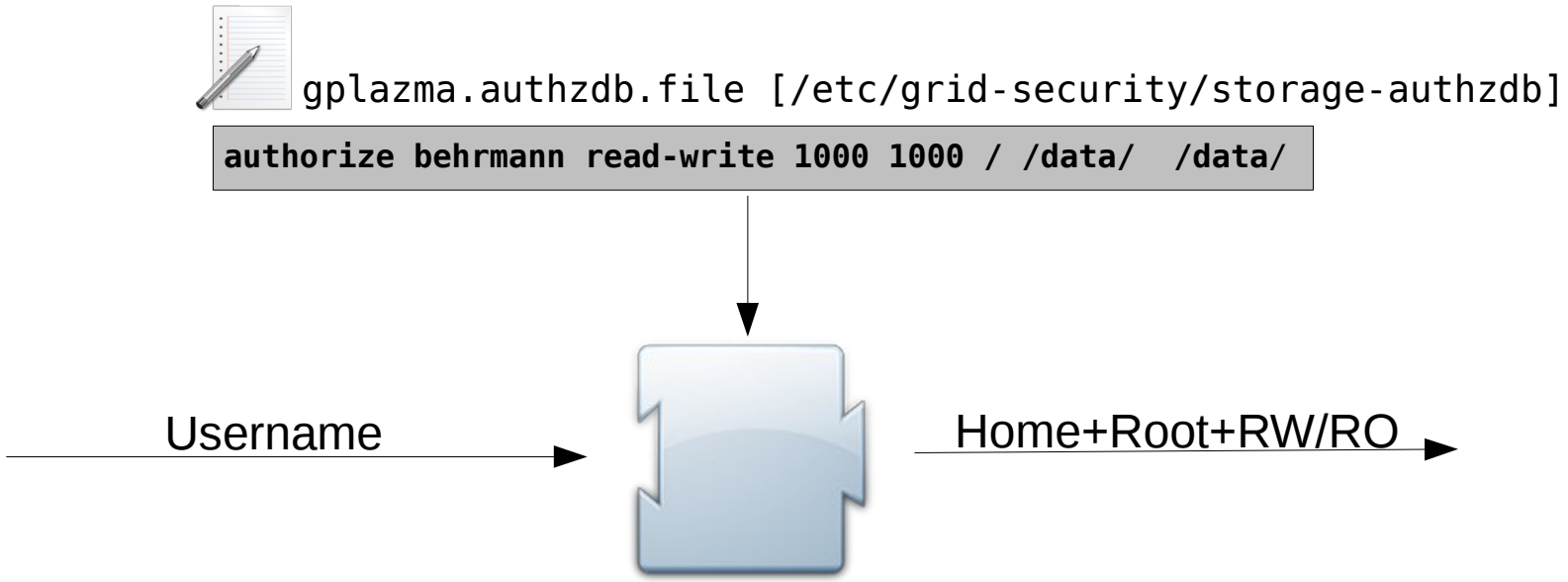
session:nsswitch

- NSSwitch



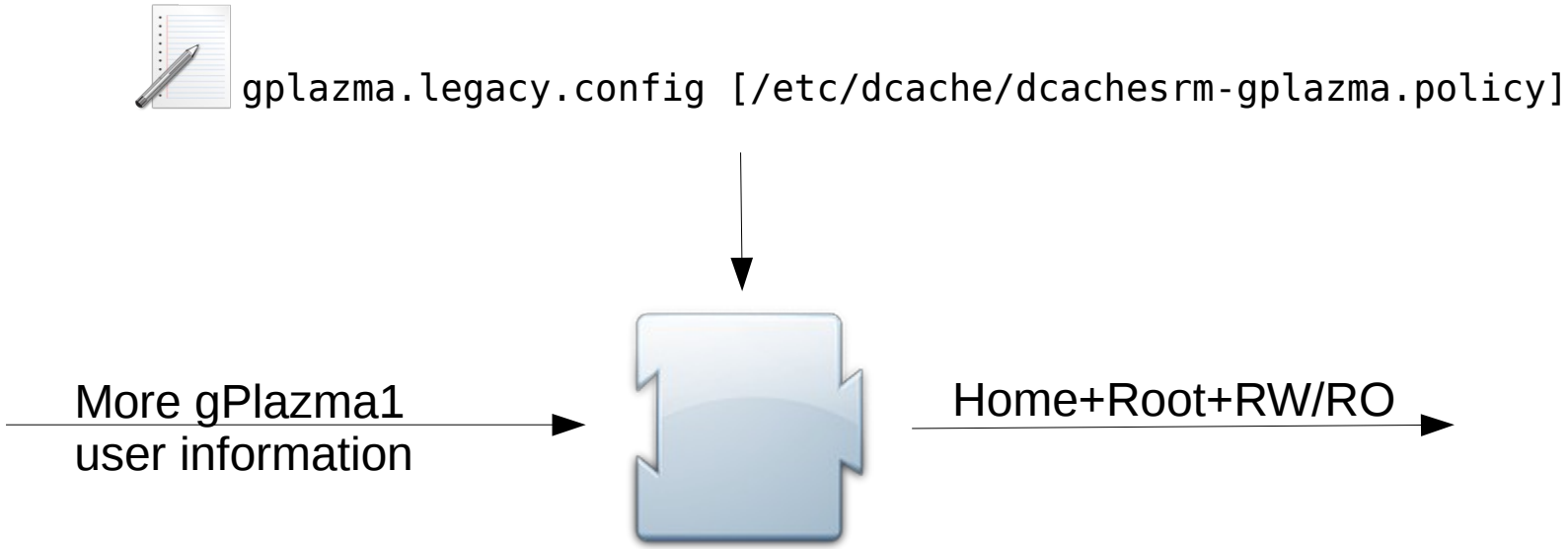
session:authzdb

- AuthzDB



session:gplazma1

- gPlazma1



Moving from v1 to v2

v1 → v2 plugins

gPlazma v1 plugin	gPlazma v2 plugins, for each phases			
	Auth	Map	Account	Session
`kpwd`	`opt: x509,` `opt: kpwd`	`suf: kpwd`	`req: kpwd`	`suf: kpwd`
`grid-mapfile`	`opt: x509`	`opt: gridmap,` `suf: authzdb`	`req: gridmap`	`suf: authzdb`
`gplazmalite-vorole-mapping`	`opt: x509,` `opt: voms`	`opt: vorolemap,` `suf: authzdb`	`req: vorolemap`	`suf: authzdb`
`xacml-vo-mapping`	`opt: xacml`	`suf: authzdb`	`req: authzdb`	`suf: authzdb`

Key: opt = optional, suf = sufficient, req = requisite

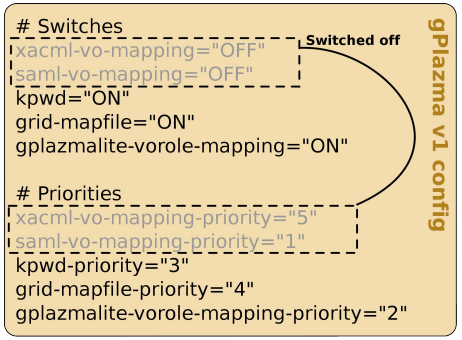
v1 → v2: example

gPlazma v1 config

```
# Switches
xacml-vo-mapping="OFF"
saml-vo-mapping="OFF"
kpwd="ON"
grid-mapfile="ON"
gplazmalite-vorole-mapping="ON"

# Priorities
xacml-vo-mapping-priority="5"
saml-vo-mapping-priority="1"
kpwd-priority="3"
grid-mapfile-priority="4"
gplazmalite-vorole-mapping-priority="2"
```

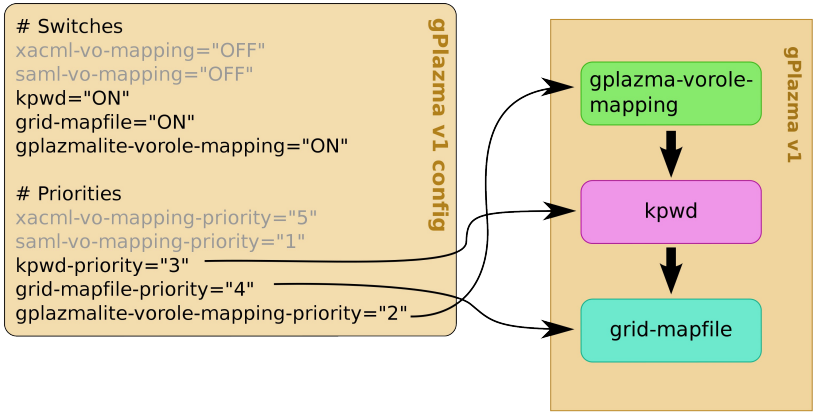
- Top part of gPlazma v1 config file

v1 → v2: example

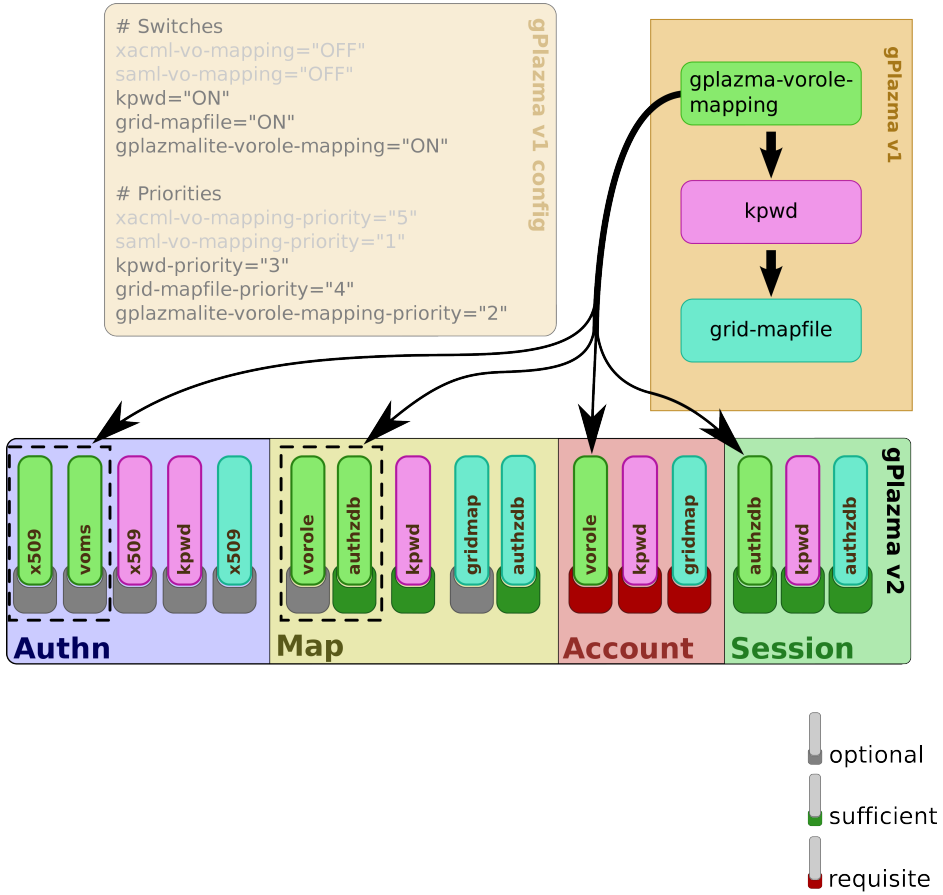


- Ignore plugins that are switched off

v1 → v2: example



- Consider the remaining plugins in their execution order

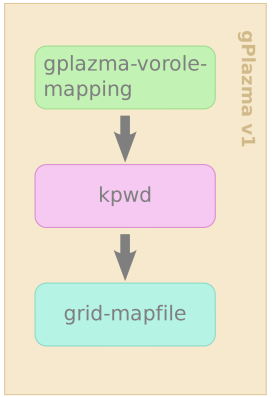


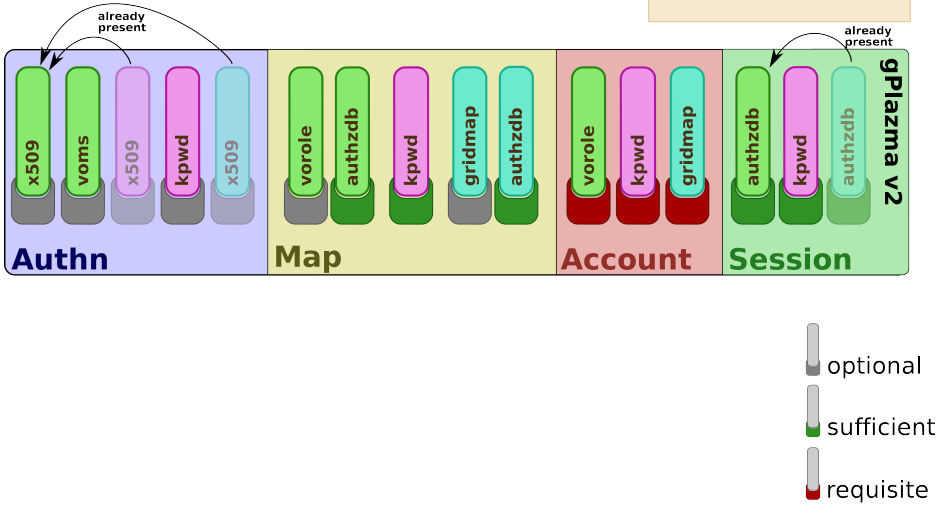
- Use table to build initial gPlazma2 configuration

v1 → v2: example

```
# Switches
xacml-vo-mapping="OFF"
saml-vo-mapping="OFF"
kpwd="ON"
grid-mapfile="ON"
gplazmalite-vorole-mapping="ON"

# Priorities
xacml-vo-mapping-priority="5"
saml-vo-mapping-priority="1"
kpwd-priority="3"
grid-mapfile-priority="4"
gplazmalite-vorole-mapping-priority="2"
```

gPlazma v1 config



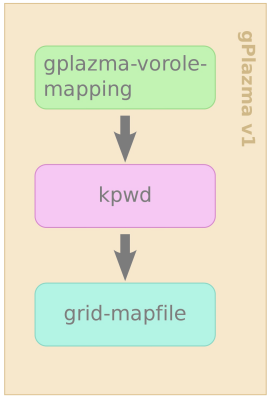


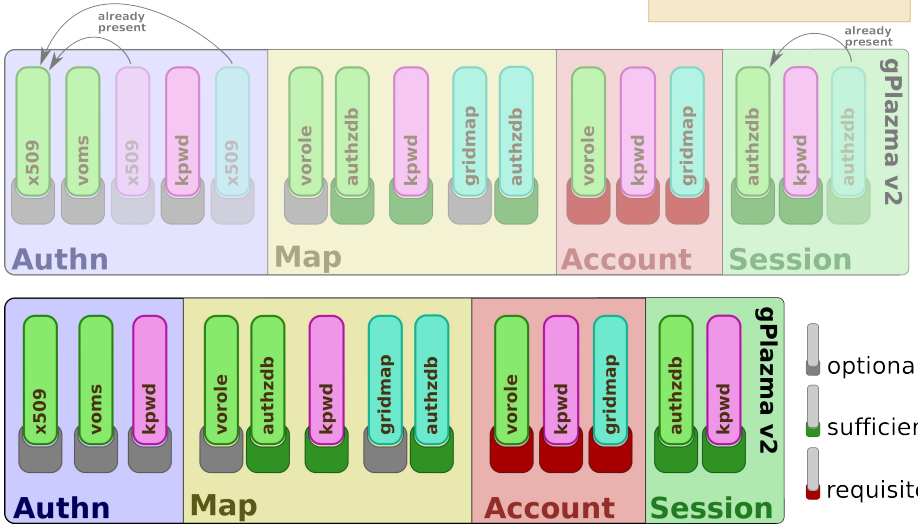
- Notice that there are some duplicates

v1 → v2: example

```
# Switches
xacml-vo-mapping="OFF"
saml-vo-mapping="OFF"
kpwd="ON"
grid-mapfile="ON"
gplazmalite-vorole-mapping="ON"

# Priorities
xacml-vo-mapping-priority="5"
saml-vo-mapping-priority="1"
kpwd-priority="3"
grid-mapfile-priority="4"
gplazmalite-vorole-mapping-priority="2"
```

gPlazma v1 config





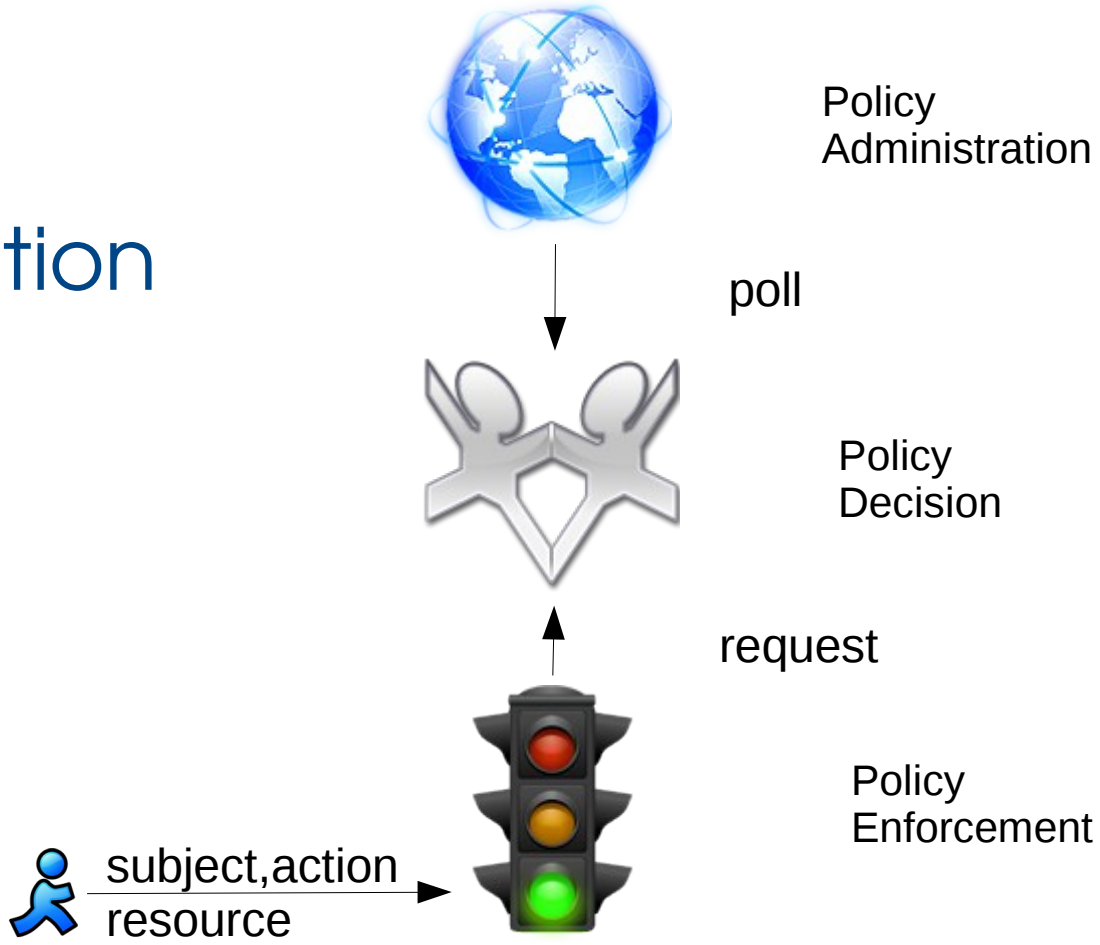
- Adjust configuration to remove duplication

Commercials

Argus

Introducing Argus

- Centralized Policies
- Hierarchical Distribution
- Authentication
- Authorization



Commercials End

See now: The standard case feat. Argus

Example: WLCG

/etc/dcache/gplazma.conf

```
# step    modifier     plugin    params k=v
```

Example: WLCG

- Users are authenticated by X.509 certificates with voms

/etc/dcache/gplazma.conf

```
# step    modifier     plugin    params k=v
auth      optional     x509
auth      optional     voms
```

Example: WLCG

- Users are authenticated by X.509 certificates with voms
- Mapping by VoRoleMap and AuthzDB

/etc/dcache/gplazma.conf

```
# step    modifier     plugin    params k=v
auth      optional     x509
auth      optional     voms
map       optional     vorolemap
map       optional     authzdb
```

Example: WLCG

- Users are authenticated by X.509 certificates with voms
- Mapping by VoRoleMap and AuthzDB
- Banning by Argus

/etc/dcache/gplazma.conf

```
# step    modifier     plugin    params k=v
auth      optional     x509
auth      optional     voms
map       optional     vorolemap
map       optional     authzdb
account   requisite    argus
```

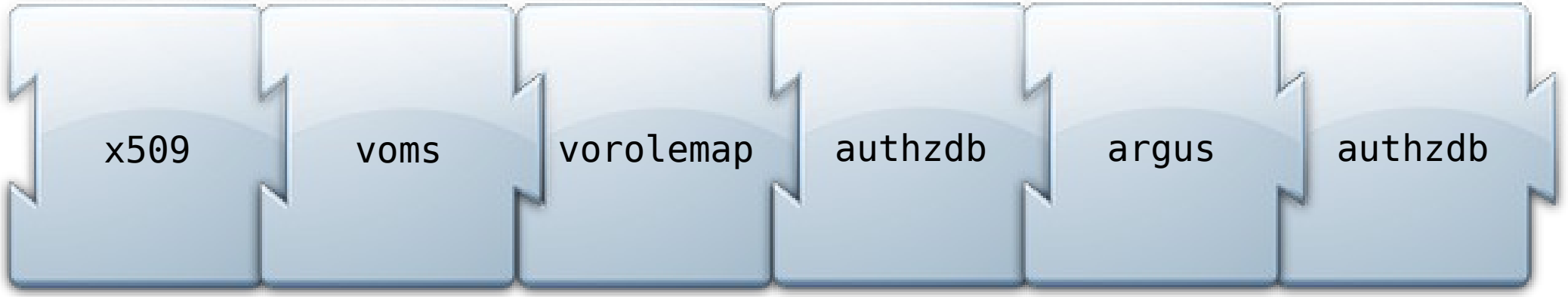
Example: WLCG

- Users are authenticated by X.509 certificates with voms
- Mapping by VoRoleMap and AuthzDB
- Banning by Argus
- Session parameters by AuthzDB

/etc/dcache/gplazma.conf

```
# step    modifier     plugin    params k=v
auth      optional     x509
auth      optional     voms
map       optional     vorolemap
map       optional     authzdb
account   requisite    argus
session   optional     authzdb
```

Example: WLCG



X.509 Chain + DN	X.509 Chain +FQAN	DN + FQAN + Username	Username + UID + GID	DN + banned?	UID+GID + home folder + root folder

More commercials

Identity mapping and Kerberos

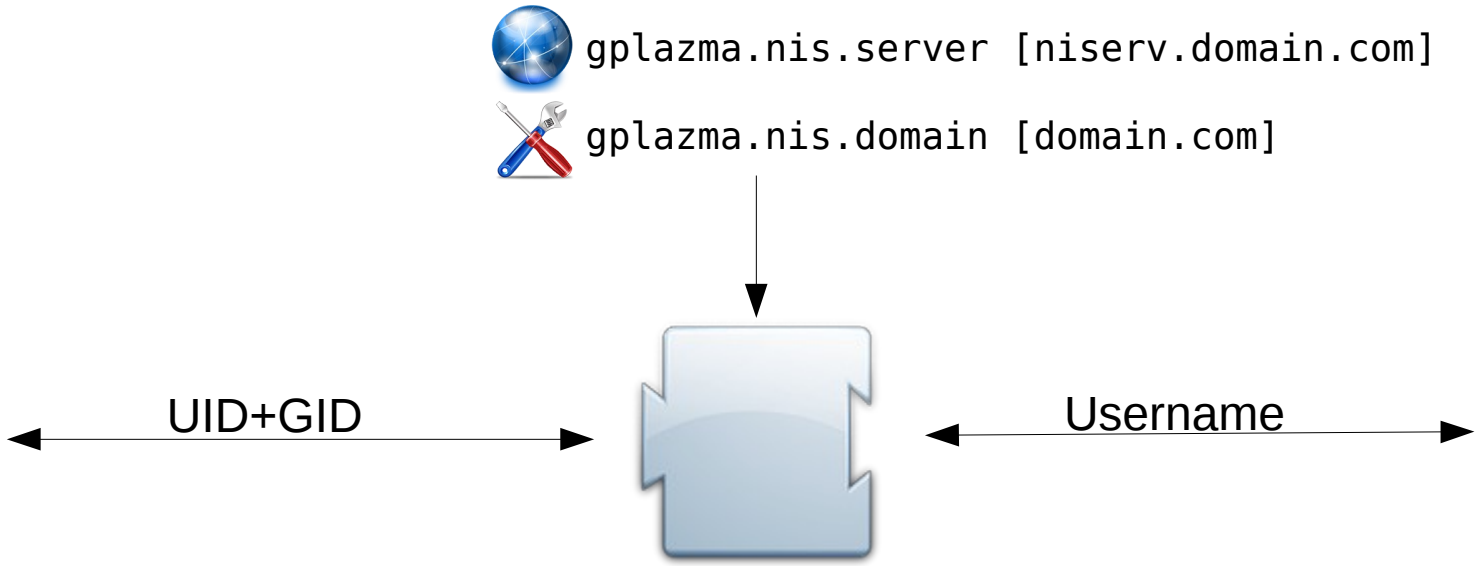
Identity Service

What's your name again?

- Map Username to UID **and reverse**
- Is **not** part of the login process
- Used by NFS 4.1 server
- Plugins:
  - NIS
  - NSSwitch

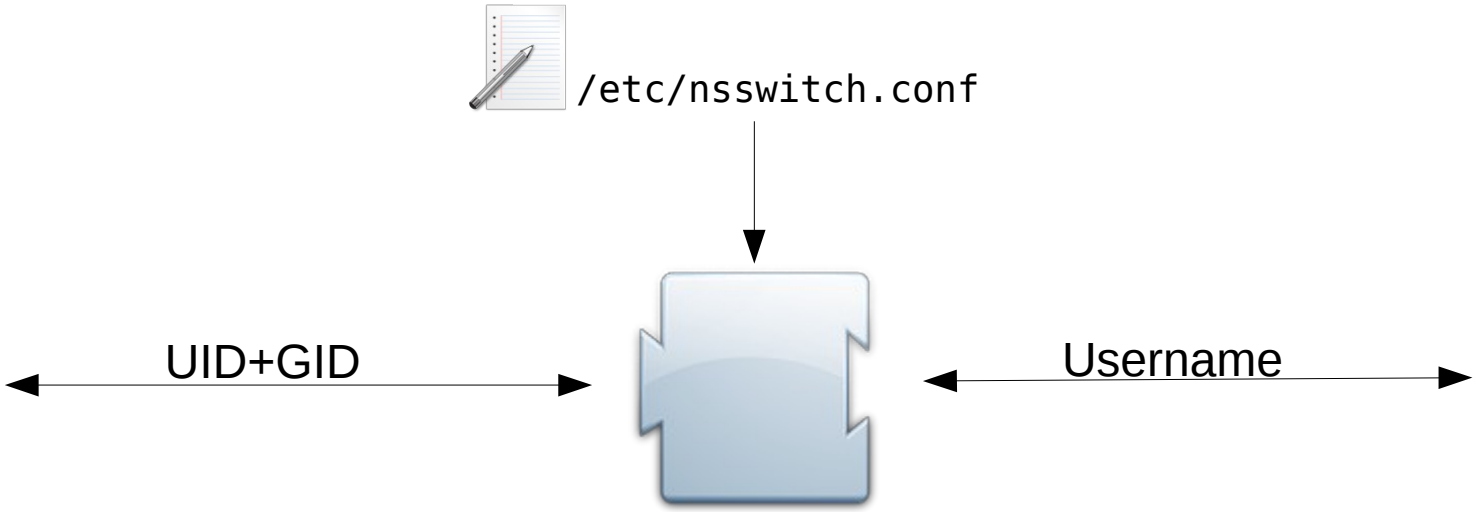
identity:nis

- NIS



identity:nss

- NSSwitch



Another example

Identity mapping and Kerberos in action

Example: Kerberos + NIS

/etc/dcache/gplazma.conf

```
# step    modifier     plugin    params k=v
```

Example: Kerberos + NIS

- Authentication is done by dCache "door".

/etc/dcache/gplazma.conf

```
# step    modifier     plugin    params k=v
```

Example: Kerberos + NIS

- Authentication is done by dCache "door"
- Mapping to Username is done by krb5 plugin

/etc/dcache/gplazma.conf

```
# step    modifier     plugin    params k=v
map       optional     krb5
```

Example: Kerberos + NIS

- Authentication is done by dCache "door"
- Mapping to Username is done by krb5 plugin
- Mapping to UID+GID is done by NIS plugin

/etc/dcache/gplazma.conf

```
# step    modifier     plugin    params k=v
map       optional     krb5
map       optional     nis
```

Example: Kerberos + NIS

- Authentication is done by dCache "door"
- Mapping to Username is done by krb5 plugin
- Mapping to UID+GID is done by NIS plugin
- Session attributes are added by NIS plugin

/etc/dcache/gplazma.conf

```
# step    modifier     plugin    params k=v
map       optional     krb5
map       optional     nis
session   optional     nis
```

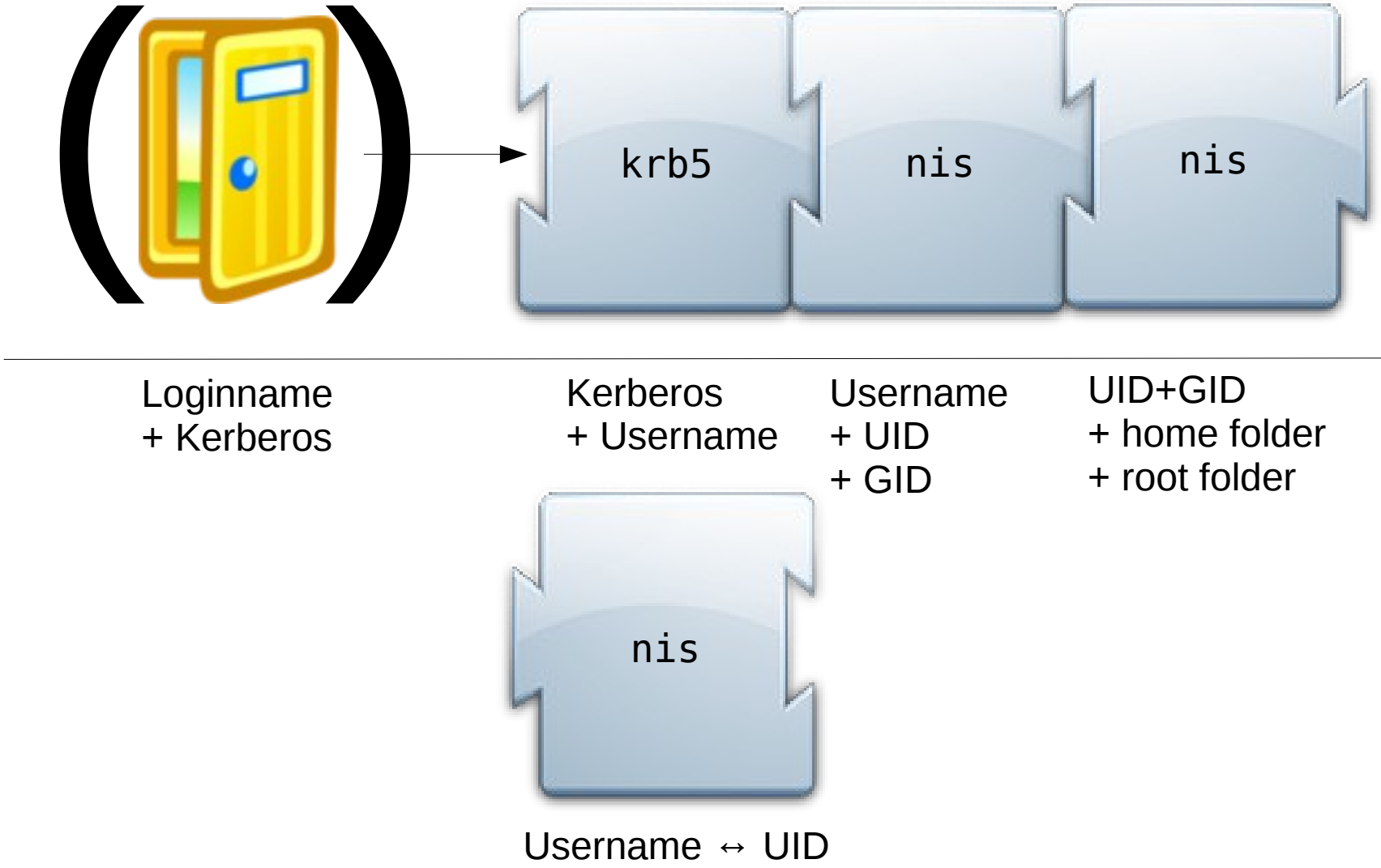
Example: Kerberos + NIS

- Authentication is done by dCache "door"
- Mapping to Username is done by krb5 plugin
- Mapping to UID+GID is done by NIS plugin
- Session attributes are added by NIS plugin
- Identity mapping by NIS plugin

/etc/dcache/gplazma.conf

```
# step     modifier     plugin     params k=v
map        optional     krb5
map        optional     nis
session    optional     nis
identity   optional     nis
```

Example: Kerberos + NIS



Summary

Use gPlazma2.