dCache.org

Some gPlazma things

Paul Millar

Berlin, 2013.05.28

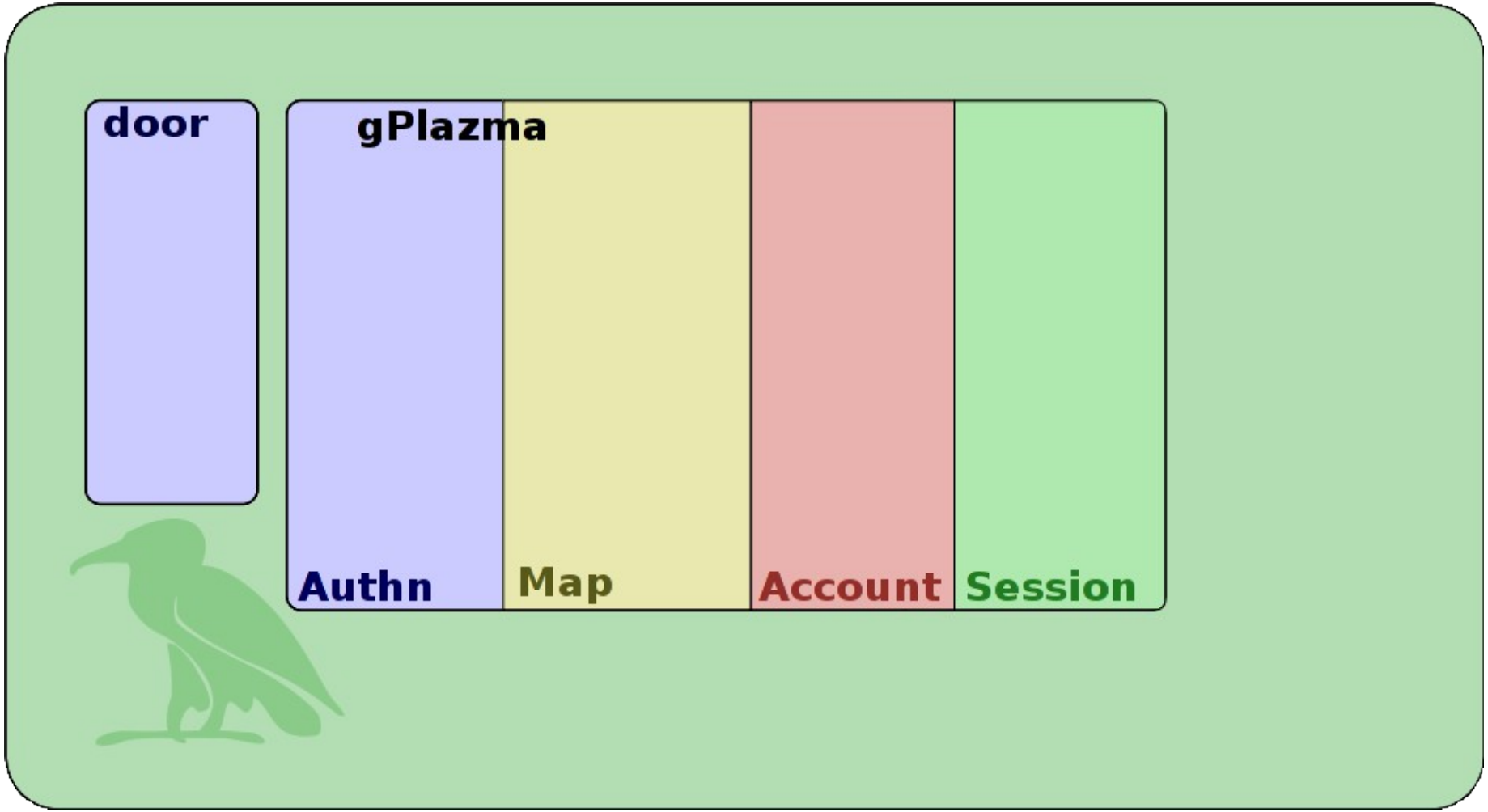
Fermilab NDGF Nordic DataGrid Facility EUROPEAN MIDDLEWARE INITIATIVE DESY HELMHOLTZ ASSOCIATION LSDMA

Quick intro/reminder* of gPlazma

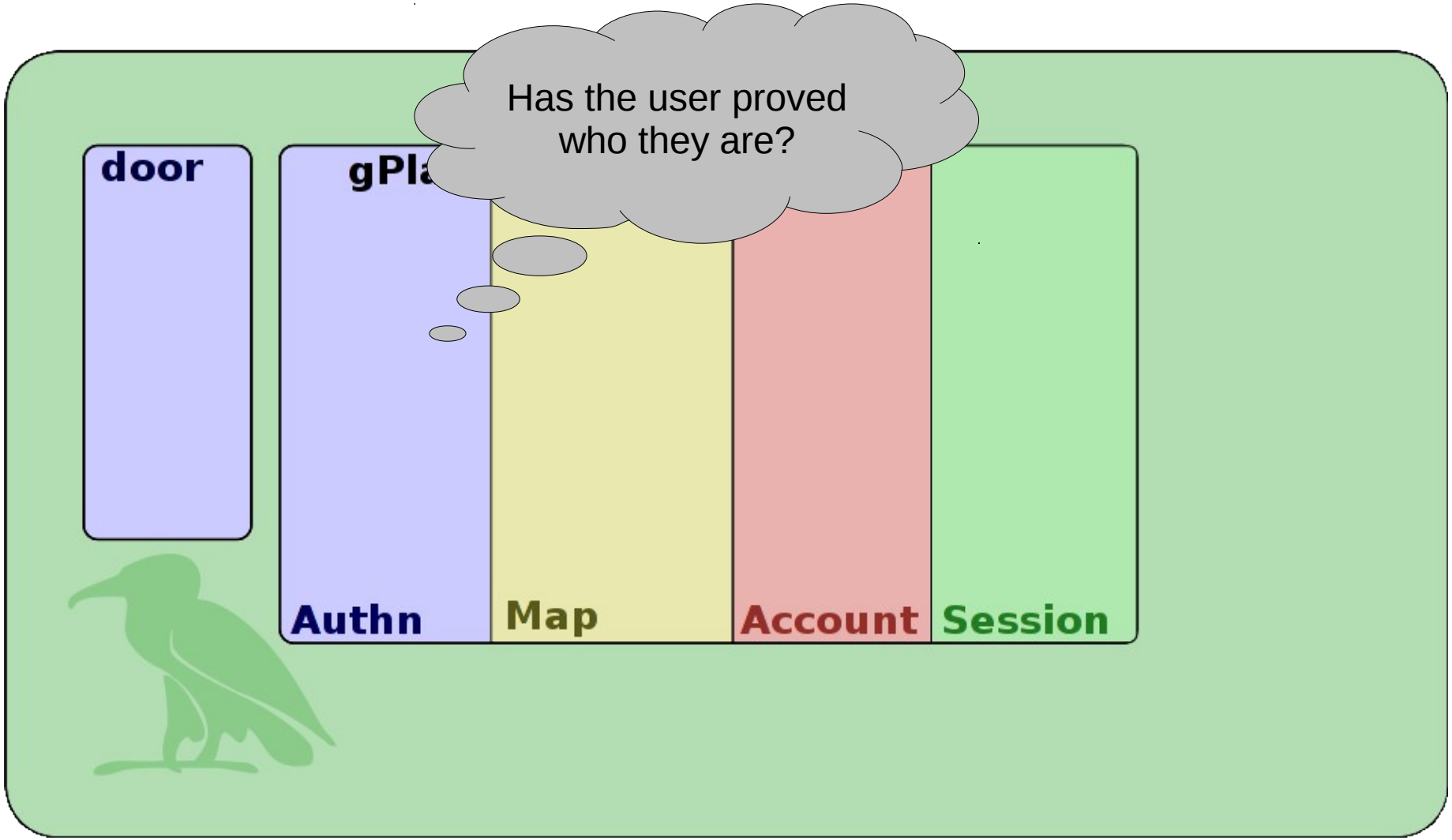
* delete as appropriate

Need to identify users

Logging in: the four phases



Logging in: first phase



Similar to entering a country

- Is it a valid ID card?
- Does the person look like the photo?
- Pull out information

```
Nationality: German
```

Authenticate and extract principals

Name: **Erika Mustermann**

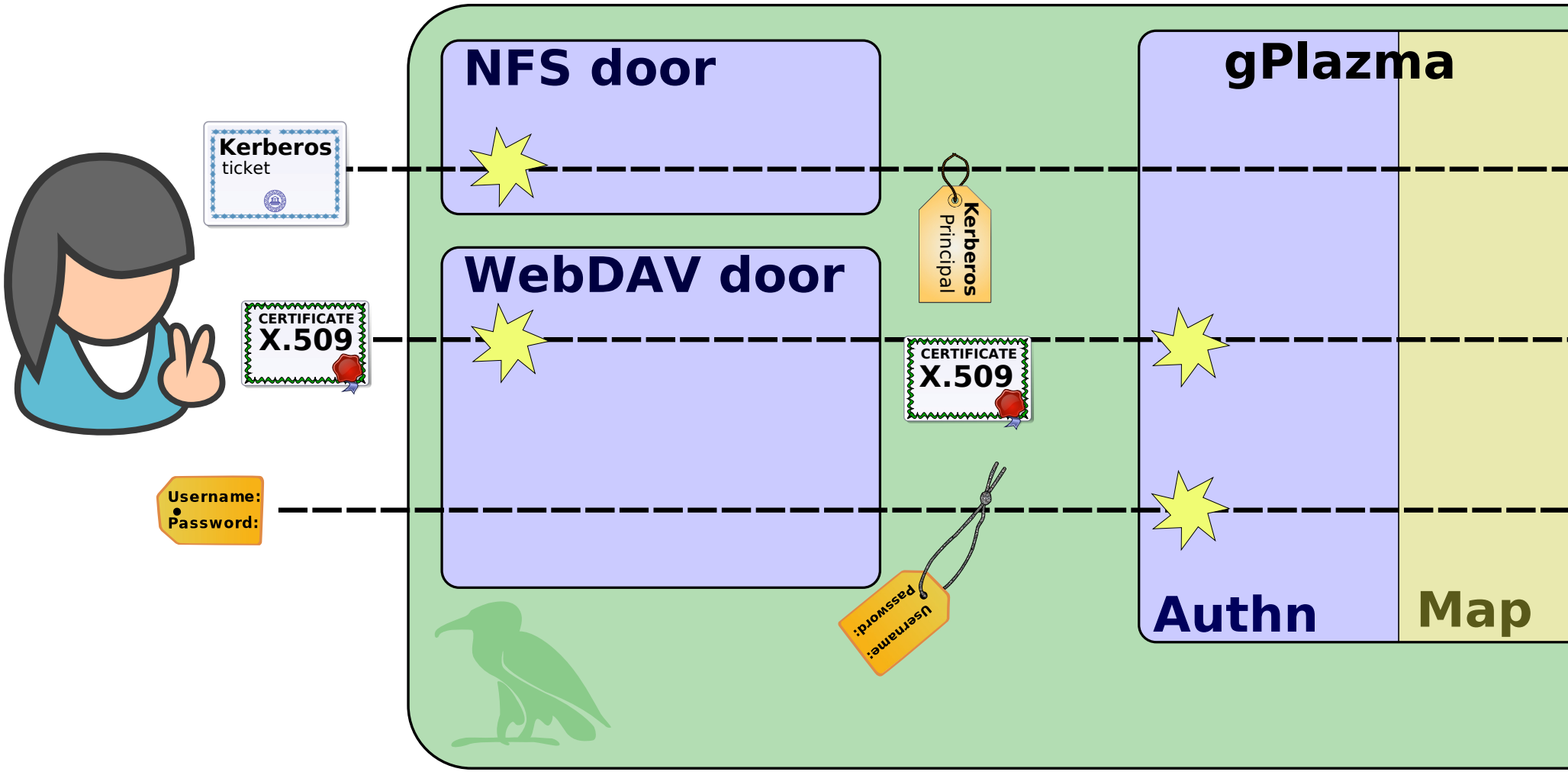
DoB: **1964-08-12**

Place of Birth: **Berlin**

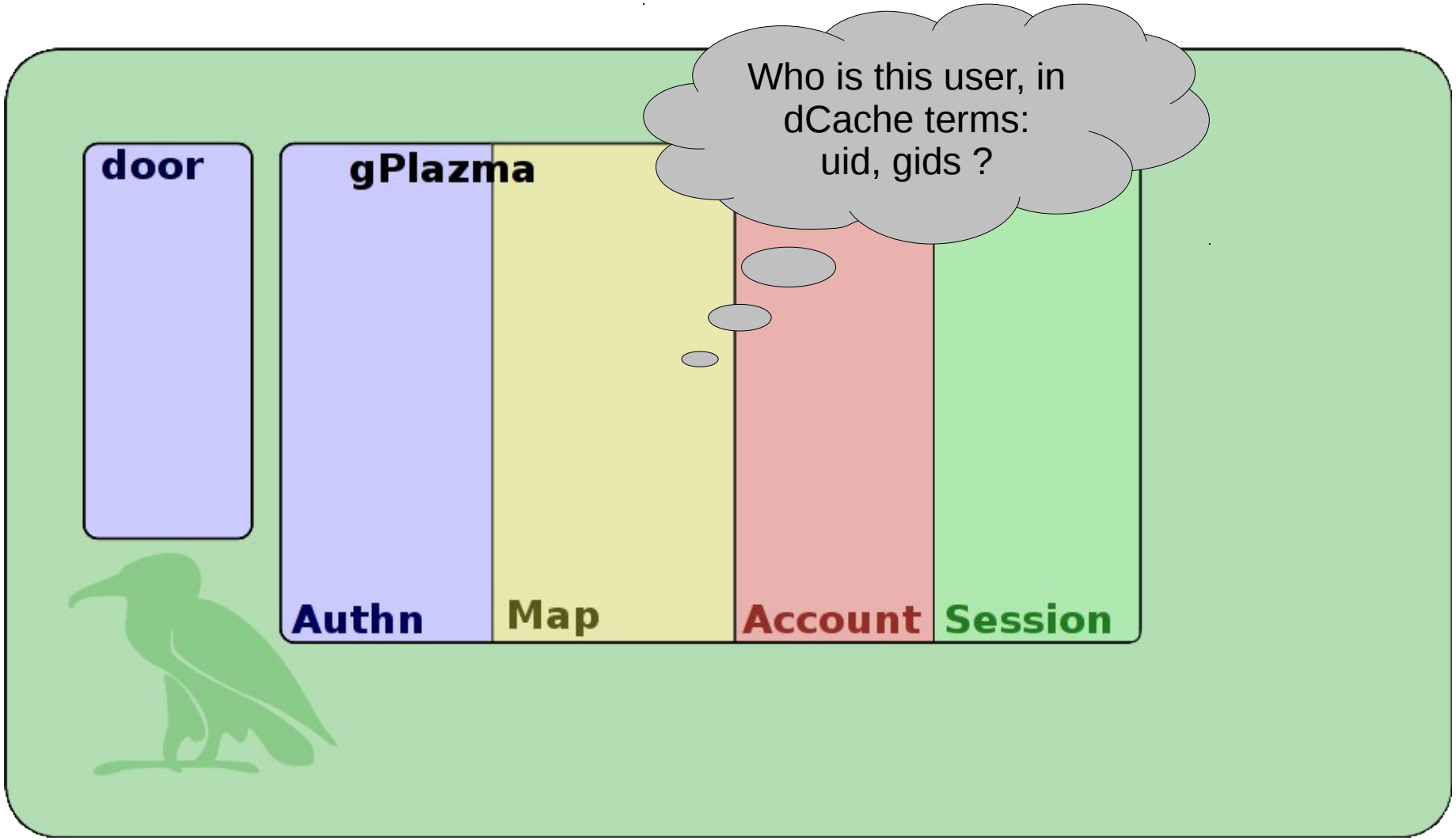
Credential

Principals

Authentication: door, both or gPlazma



Logging in: second phase



Mapping principals

DN:/C=UK/O=eScience/OU=Glasgow/L=Compserv/CN=paul millar

DN:/C=DE/O=GermanGrid/OU=DESY/CN=Paul Millar

Kerberos: paul@DESY.DE

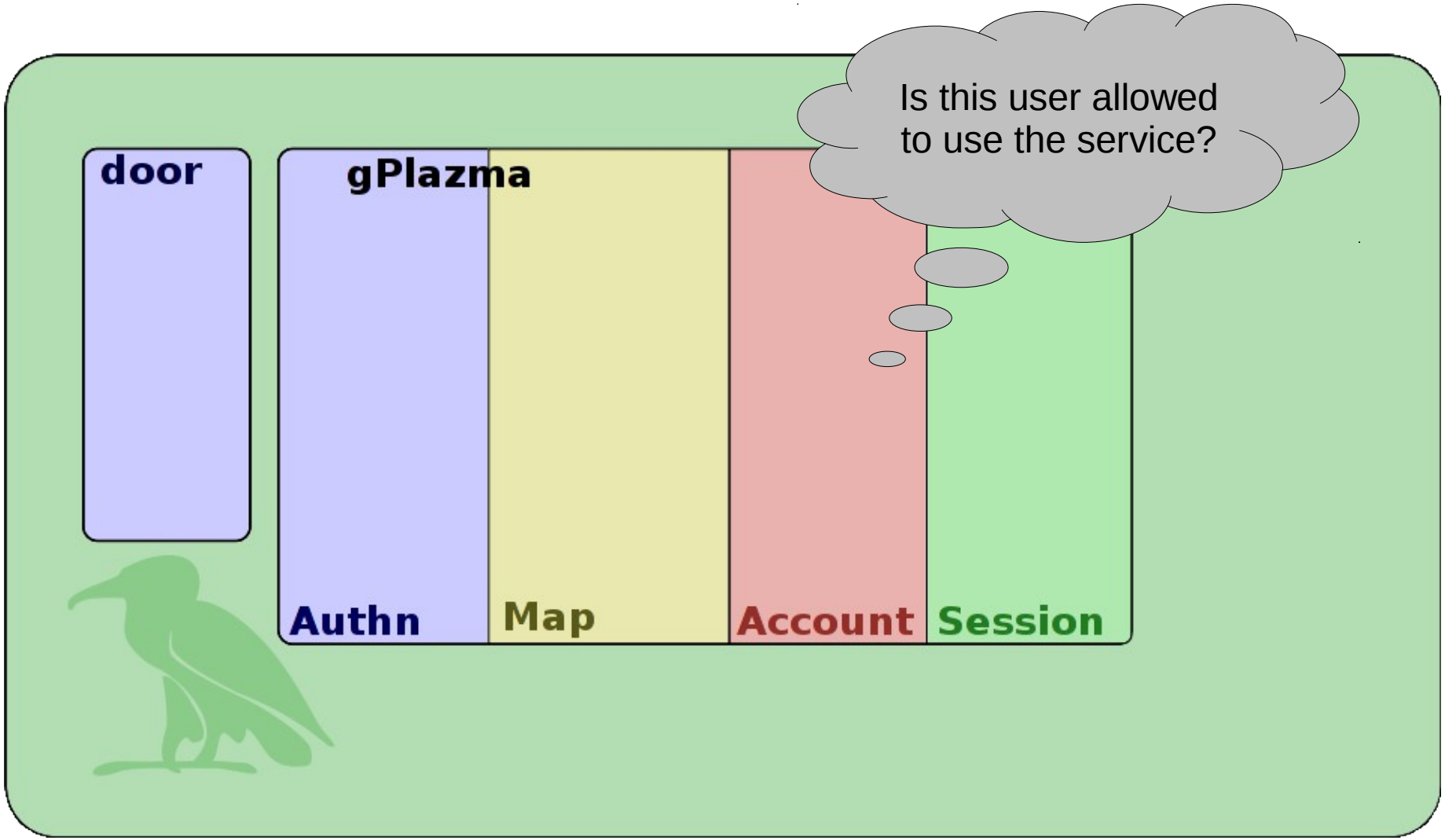
uid: 15691

FQAN: /atlas

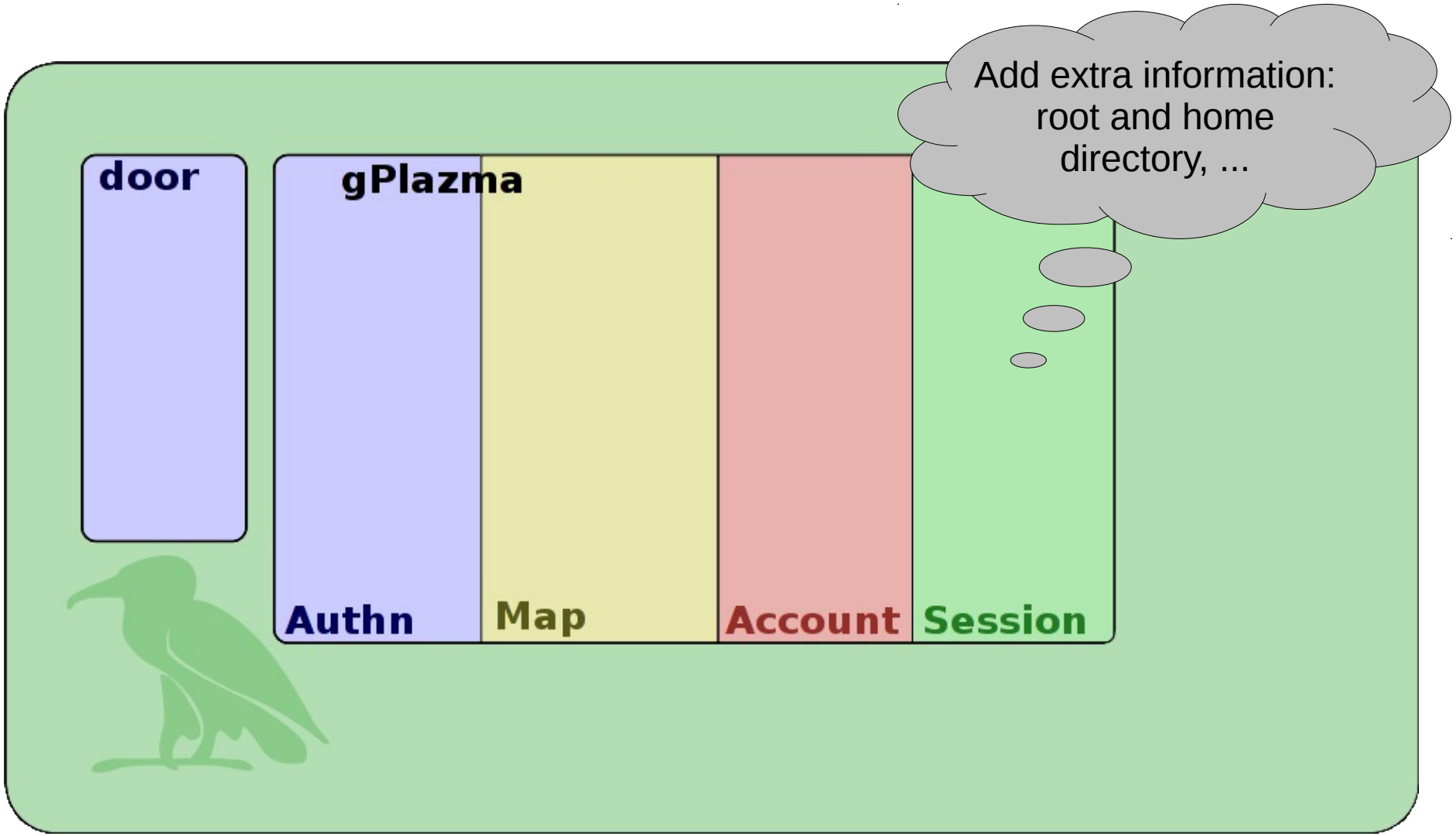
Kerberos: atlas@DESY.DE

gid: 1000

Logging in: third phase



Logging in: final phase



Each phase uses plugins

- Each plugin has a **small, concrete job**

  Each gPlazma-1 file format is supported by a gPlazma-2 plugin

  Plugins can support one or more phase

- Plugins **succeed or fail**

  Not all failures are "bad"

- You get to **configure**:

  which plugins run in each phase,

  in which the order plugins run within a phase,

  how gPlazma treats a plugin's success or failure.

Migrating from gPlazma-1 to gPlazma-2

You should act now

- In dCache v2.2
  - gPlazma-1 still exists, but only a gPlazma-2 **plugin**
  - Default gPlazma-2 configuration uses gPlazma-1 plugin
  - Same functionality now available via new plugins,
  - Can edit gPlazma-2 config file to use the new plugins.
- In dCache v2.6
  - **gPlazma 1 has gone** completely
  - Upgrading from 2.2 to 2.6 forces you to use new gPlazma-2 plugins,
  - Default configuration assumes a grid site (x509, voms, vorolemap, gridmap, authzdb)

The procedure for updating

- **Create** a new file: `gplazma-new.conf`

  details in a moment

- **Test** new configuration:

  if you don't have a test instance, start new domain running gPlazma (with distinct cell.name and using `gplazma-new.conf`) and a test door, configured to use this gPlazma.

- **Deploy** new configuration:
  - Copy `gplazma.conf` file as `gplazma-old.conf`
  - Register site "at risk" in GOC-DB, if you're paranoid
  - Copy `gplazma-new.conf` as `gplazma.conf`.
- Note: there's no need to restart dCache.

gPlazma-2 plugins for gPlazma1 “plugin”s

Phase	gPlazma-1 “plugin”s			
	kpwd	**grid-mapfile**	**Gplazmalite-vorole-mapping**	**xacml-vo-mapping**
auth	`opt: x509,` `opt: kpwd`	`opt: x509`	`opt: x509,` `opt: voms`	`opt: xacml`
map	`suf: kpwd`	`opt: gridmap,` `suf: authzdb`	`opt: vorolemap,` `suf: authzdb`	`suf: authzdb`
account	`req: kpwd`	`req: gridmap`	`req: vorolemap`	`req: authzdb`
session	`suf: kpwd`	`suf: authzdb`	`suf: authzdb`	`suf: authzdb`

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

Creating gplazma-new.conf file

- For each enabled gPlazma-1 "plugin" there is a set of gPlazma-2 plugins you need to configure
- Write down the gPlazma-2 plugins corresponding to each gPlazma-1 "plugin" in decreasing order of priority:

  For the auth phase:
  - Write down the auth phase gPlazma-2 plugins for the highest priority gPlazma-1 "plugin"
  - Write down the auth phase gPlazma-2 plugins for the next highest priority gPlazma-1 "plugin"

    ...etc...

  and so on for map, account and session phases.
- If the same plugin appears multiple times in **auth** or **session** phase then you can keep the first plugin and remove the remainder.
- Do **NOT** remove duplicate plugins in **map** phase

Migration worked example

Worked example

This is an example `dcachesrm-gplazma.policy` file.

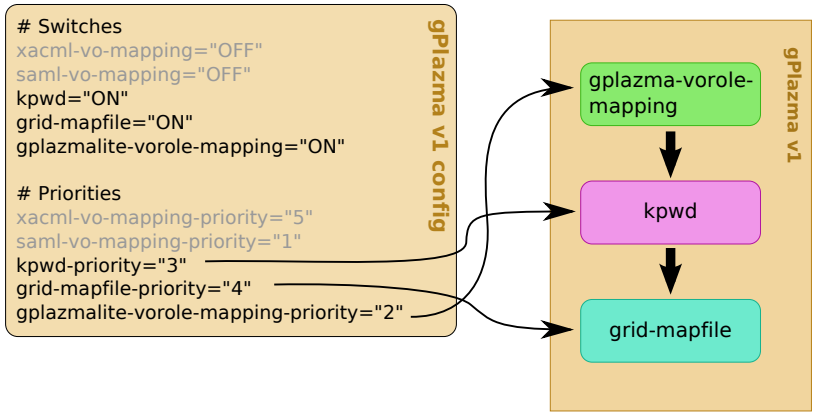
```
# Switches
xacml-vo-mapping="OFF"
saml-vo-mapping="OFF"
kpwd="ON"
grid-mapfile="ON"
gplazmalite-vorole-mapping="ON"

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

# ... rest of configuration file would go here ...
```

Worked example

Ignore “plugin”s that are switched off.

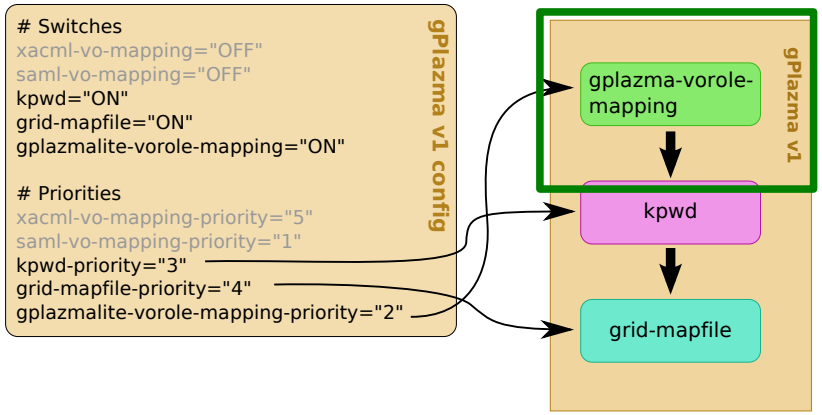
```
# Switches
xacml-vo-mapping="OFF"
saml-vo-mapping="OFF"
kpwd="ON"
grid-mapfile="ON"
gplazmalite-vorole-mapping="ON"

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

# ... rest of configuration file would go here ...
```

Worked example



Consider the remaining plugins in their execution order

Worked example



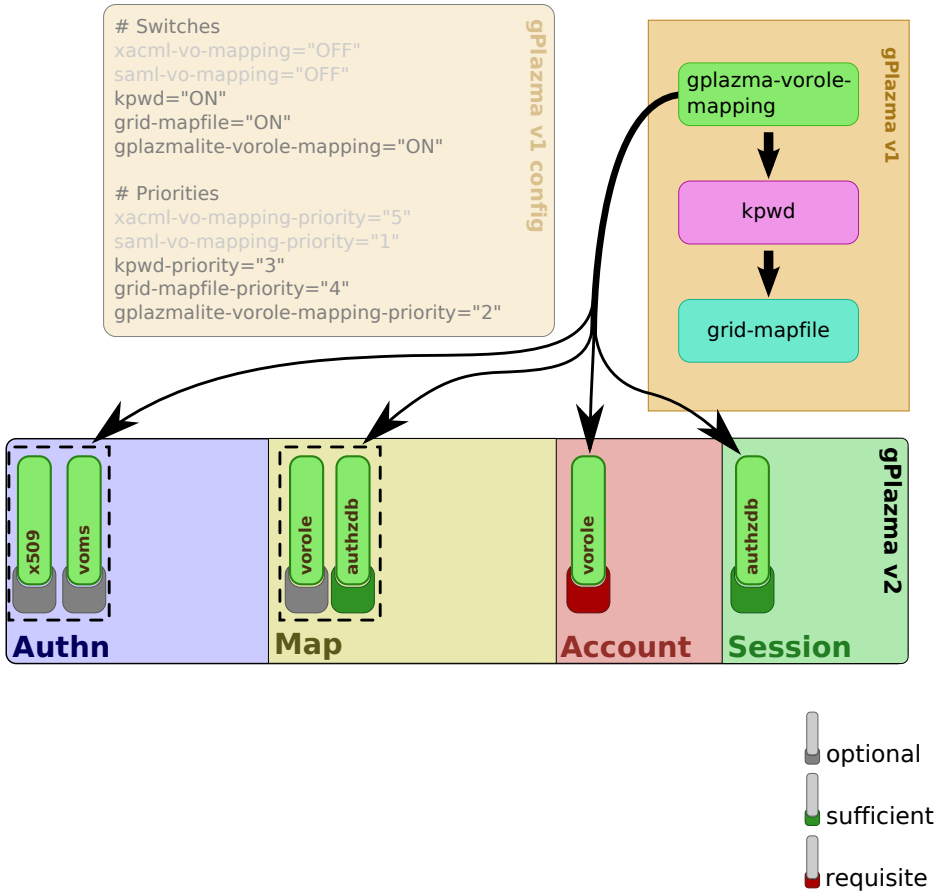
First, gplazma-vorole-map plugin

gPlazma-2 plugins for gplazmalite-vorol...

Phase	gPlazma 1 "plugin"s			
	kpwd	grid-mapfile	**Gplazmalite-vorole-mapping**	xacml-vo-mapping
auth	opt: x509, opt: kpwd	opt: x509	**opt: x509, opt: voms**	opt: xacml
map	suf: kpwd	opt: gridmap, suf: authzdb	**opt: vorolemap, suf: authzdb**	suf: authzdb
account	req: kpwd	req: gridmap	**req: vorolemap**	req: authzdb
session	suf: kpwd	suf: authzdb	**suf: authzdb**	suf: authzdb

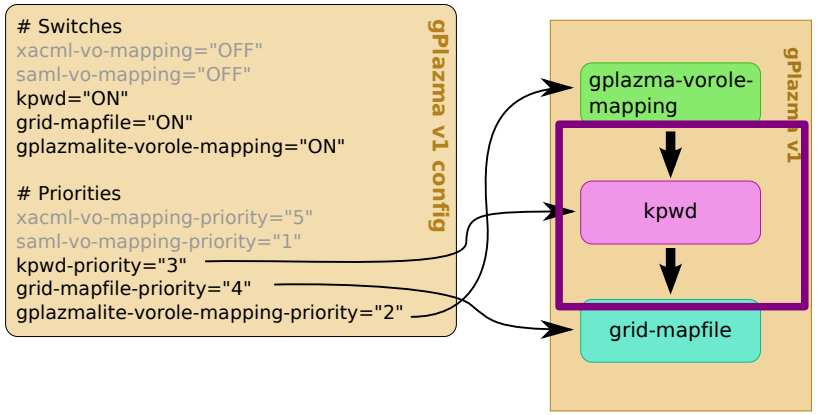
Key: opt = optional, suf = sufficient, req = requisite

Worked example




```
auth optional x509
auth optional voms

map optional vorole
map sufficient authdb

account requisite vorole

session sufficient authdb
```

Worked example



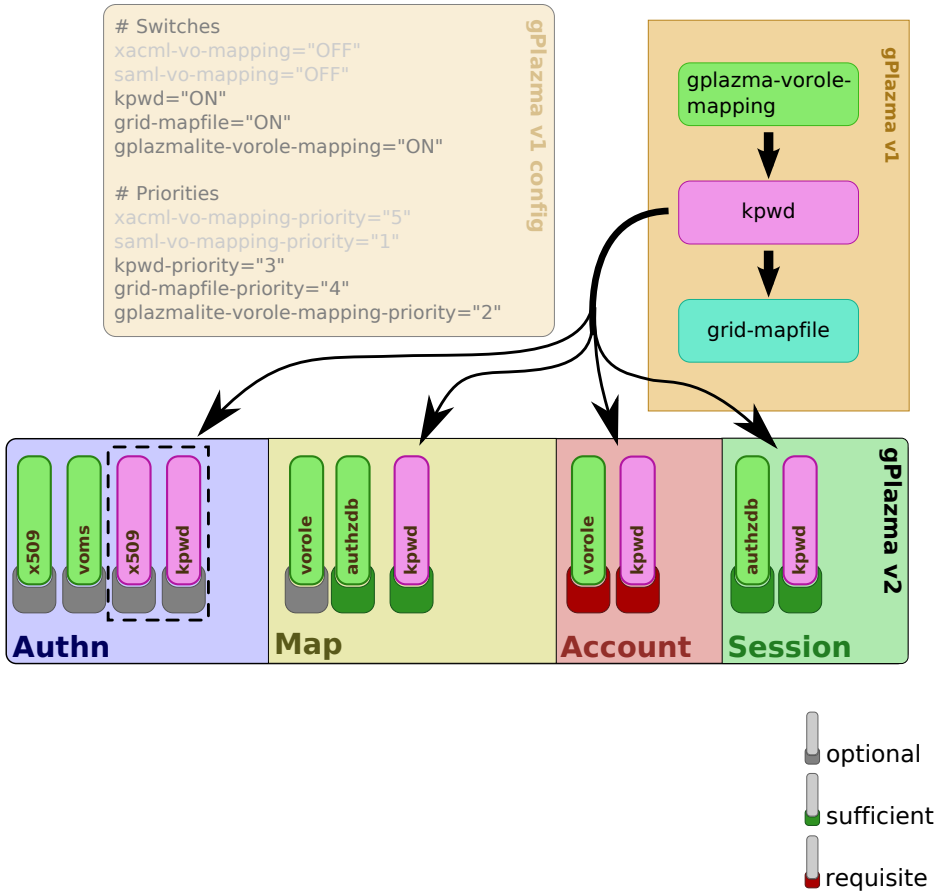
Next, the kpwd plugin

gPlazma-2 plugins for kpwd

Phase	gPlazma-1 "plugin"s			
	kpwd	**grid-mapfile**	**Gplazmalite-vorole-mapping**	**xacml-vo-mapping**
auth	**opt: x509,** **opt: kpwd**	opt: x509	opt: x509, opt: voms	opt: xacml
map	**suf: kpwd**	opt: gridmap, suf: authzdb	opt: vorolemap, suf: authzdb	suf: authzdb
account	**req: kpwd**	req: gridmap	req: vorolemap	req: authzdb
session	**suf: kpwd**	suf: authzdb	suf: authzdb	suf: authzdb

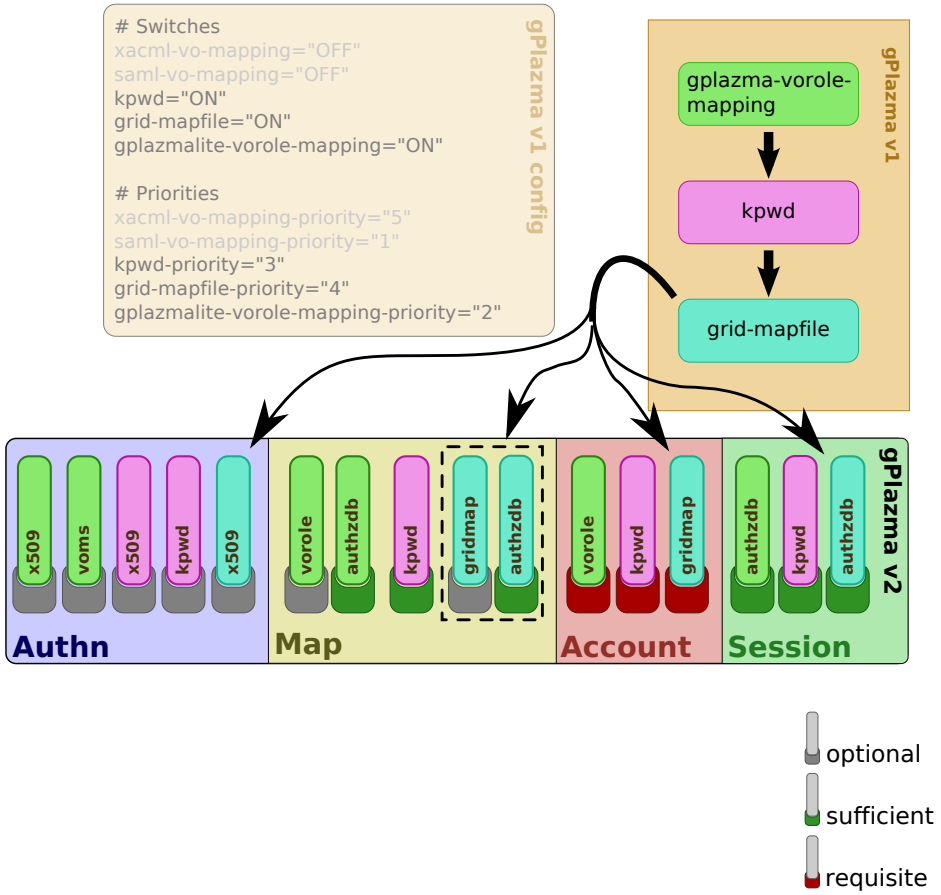
Key: opt = optional, suf = sufficient, req = requisite

Worked example




```
auth optional x509
auth optional voms
auth optional x509
auth optional kpwd

map optional vorole
map sufficient authdb
map sufficient kpwd

account requisite vorole
account requisite kpwd

session sufficient authdb
session sufficient kpwd
```

Worked example

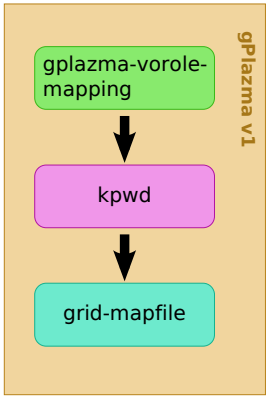


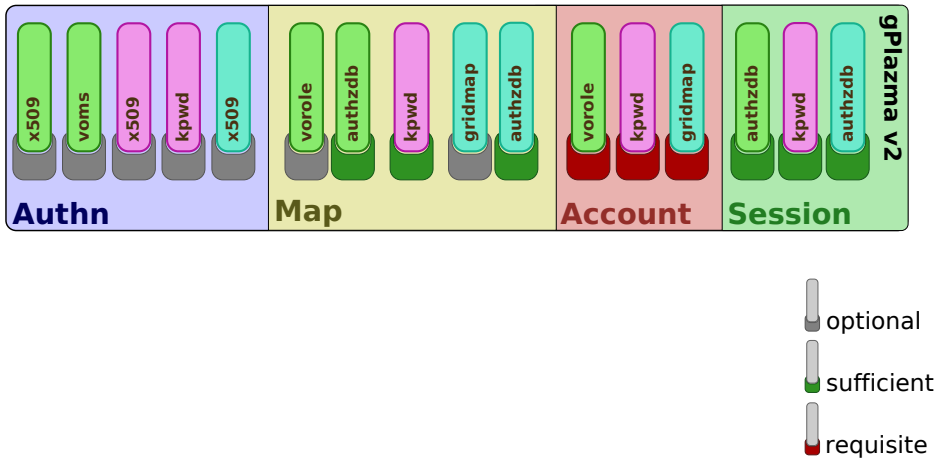

```
auth optional x509
auth optional voms
auth optional x509
auth optional kpwd
auth optional x509

map optional vorole
map sufficient authdb
map sufficient kpwd
map optional gridmap
map sufficient authdb

account requisite vorole
account requisite kpwd
account requisite gridmap

session sufficient authdb
session sufficient kpwd
session sufficient authdb
```

Worked 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



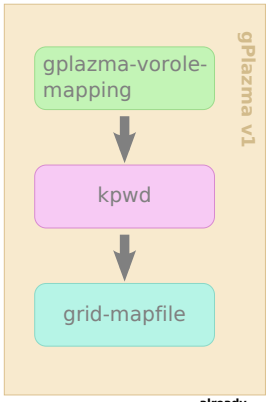


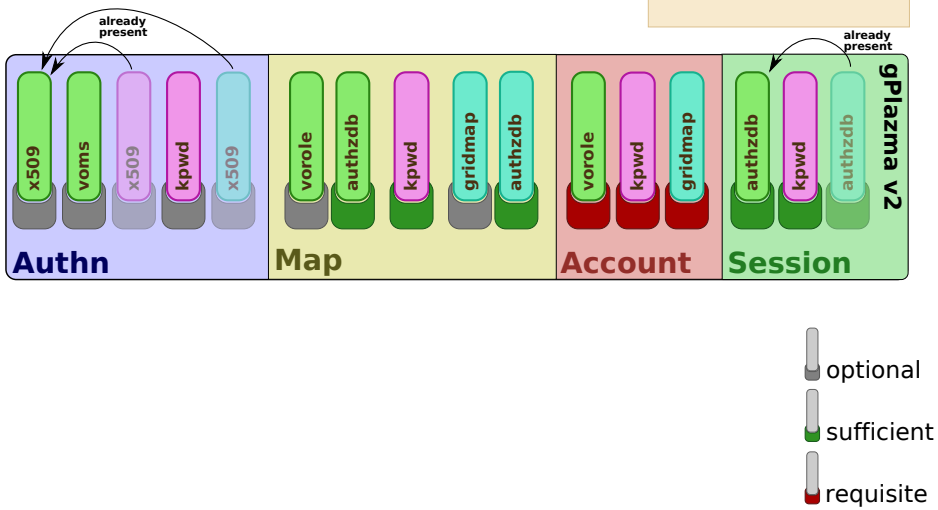

```
auth optional x509
auth optional voms
auth optional x509
auth optional kpwd
auth optional x509

map optional vorole
map sufficient authdb
map sufficient kpwd
map optional gridmap
map sufficient authdb

account requisite vorole
account requisite kpwd
account requisite gridmap

session sufficient authdb
session sufficient kpwd
session sufficient authdb
```

Worked 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



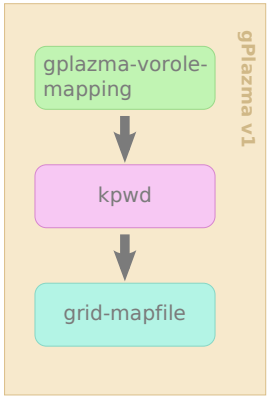


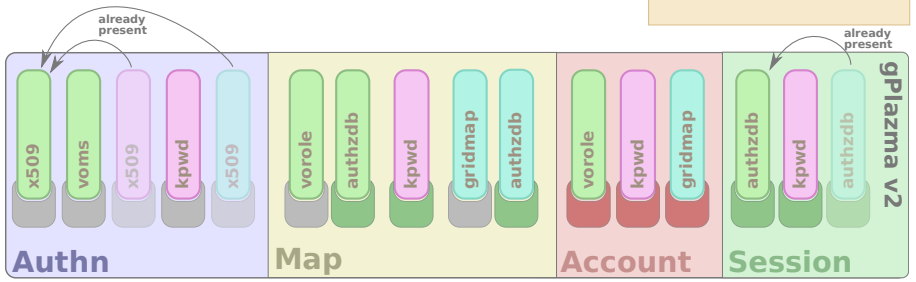

```
auth optional x509
auth optional voms
auth optional x509
auth optional kpwd
auth optional x509

map optional vorole
map sufficient authdb
map sufficient kpwd
map optional gridmap
map sufficient authdb

account requisite vorole
account requisite kpwd
account requisite gridmap

session sufficient authdb
session sufficient kpwd
session sufficient authdb
```

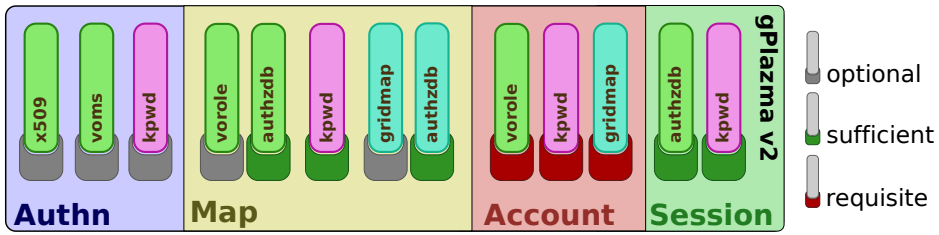
Worked example

```
# 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







Final result

```
auth optional x509
auth optional voms
auth optional kpwd

map optional vorole
map sufficient authdb
map sufficient kpwd
map optional gridmap
map sufficient authdb

account requisite vorole
account requisite kpwd
account requisite gridmap

session sufficient authdb
session sufficient kpwd
```

Customising plugin behaviour

- Some aspects of gPlazma-1 "plugins" may be customised in the `dcachesrm-gplazma.policy` file.

  (location of a file, GUMS server to contact, ...)

- With gPlazma-2, this is achieved with the normal dCache configuration
  - Properties for plugin foo start `gplazma.foo`

    e.g., gplazma.vorolemap.file, gplazma.authzdb.uid
  - Properties are documented in:

    `/usr/share/dcache/default/gplazma.properties`
  - Configure properties in `dcache.conf`, the layout file or in gplazma.conf

    if not sure, use `dcache.conf`

Example configuration

- In dcache.conf

```
gplazma.vorolemap.file=/etc/dcache/vorole.conf
```

- In layout file

```
[testDomain/gPlazma]
gplazma.vorolemap.file=/etc/dcache/vorole.conf
```

- In gplazma.con

```
map optional vorolemap gplazma.vorolemap.file=/etc/d[..]
```

Adding HTTP/WebDAV support

Just start WebDAV door

- Configure 'https-jglobus' for the webdavProtocol property

  (needed to support proxies)
- Start WebDAV door
- X.509 authenticated users should just work

Adding support for username+password

- Configure door to accept basic authn

  ```
  WebdavBasicAuthenication = enabled
  ```

  Make sure only https-jglobus is enabled.

- Add username+password supprt in gPlazma
  - Either use kpwd, jaas (with Kerberos) or jaas (external plugin activated)
  - See Ron's talk for details

Adding NFS mounted users

The simple approach: no Kerberos

- Start NFS v4.1 server
- Mount server on client machine
- uid/gid(s) on client machine are uid/gid(s) in dCache
  - If you have a mismatch then it'll be painful
- "Trusted host" security.

Integrating with Kerberos

- Configure NFS door to use Kerberos
- Add the krb5 plugin to map phase

  maps 'paul@DESY.DE' to 'paul'
- Ensure user is mapped to a uid

  Either use static mapping (e.g., kpwd) or call out to local infrastructure (e.g., nis, ldap)
- On client machine, mount with Kerberos security and with rpc.idmapd, rpc.gssd running

Summary

- Sites should be upgrading to gPlazma-2 now(-ish)
- Upgrading is (relatively) easy, may be tested and executed without down-time
- gPlazma-2 allows NFS mounting using Kerberos security