dCache.org

# Masterclass II: identity and gPlazma

**Paul Millar**

Taipei, 2013.03.17

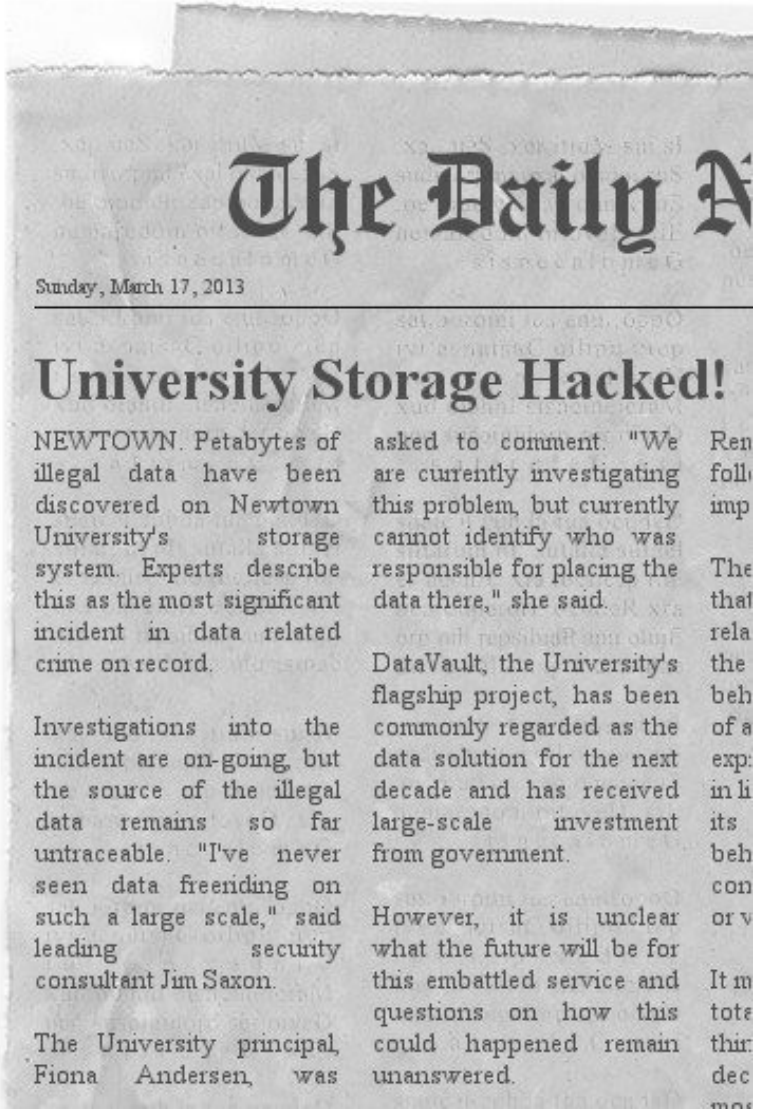
Fermilab NDGF Nordic DataGrid Facility EUROPEAN MIDDLEWARE INITIATIVE DESY HELMHOLTZ ASSOCIATION

# A storage system has valuable





# Your reputation has value



The Daily N

Sunday, March 17, 2013

## University Storage Hacked!

NEWTOWN. Petabytes of illegal data have been discovered on Newtown University's storage system. Experts describe this as the most significant incident in data related crime on record.

Investigations into the incident are on-going, but the source of the illegal data remains so far untraceable. "I've never seen data freeriding on such a large scale," said leading security consultant Jim Saxon.

The University principal, Fiona Andersen, was asked to comment. "We are currently investigating this problem, but currently cannot identify who was responsible for placing the data there," she said.

DataVault, the University's flagship project, has been commonly regarded as the data solution for the next decade and has received large-scale investment from government.

However, it is unclear what the future will be for this embattled service and questions on how this could happened remain unanswered.

# System needs to be dependable

Users doesn't want to learn that someone they've never heard of has deleted their data.

# Need to identify the users

# Many possibilities

Password:
- KDC
- NIS
- Site db
- Config. file

X.509

Kerberos Ticket

**Authentication**

- NIS
- LDAP
- Site db
- Config. file

**User details**

# The challenge

How does dCache support **all** of these methods

… even the ones we haven't heard of?

# Who are you?

# Who are you?

ZARA

**Erika Mustermann**
Cheif Design Manager

Zara
Tauentzienstraße 7a
(Charlottenburg)
10789 Berlin
Germany
+49 30 2101 6247
mustermann@zara.de

# What happens when entering country



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

  `Nationality: German`

- Make decision

# Credential vs Principal


Credentials

Name: **Wile E. Coyote**

ACME customer ID: **11493**

Passport number: **0008103314**

Bank account number: **001213921**
Banks with: **United ACME Bank**

Member-of: **Antagonists Anonymous**

Principals

# Authenticate and extract principals


Name: **Erika Mustermann**

DoB: **1964-08-12**

Place of Birth: **Berlin**

Credential

Principals

# Proving who you are: in computer world

- Based on one or more of:
  - knowing a secret
  - Holding something difficult to fake
  - Some biometric identifier
- Examples:
  - Username + password
  - X509 certificate (+ private key)
  - Kerberos
  - Federated identity:

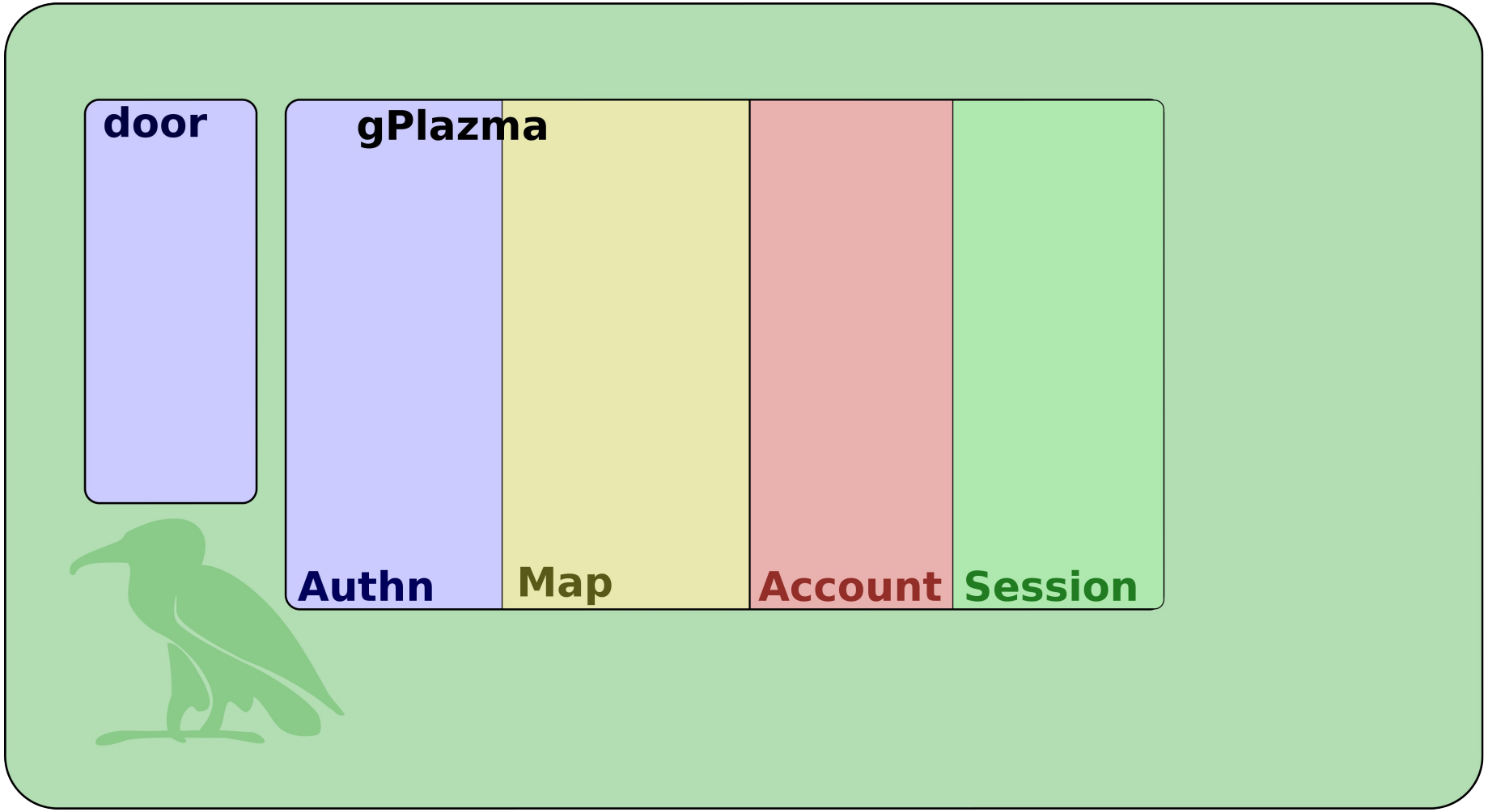


Shibboleth OpenID OAUTH BrowserID

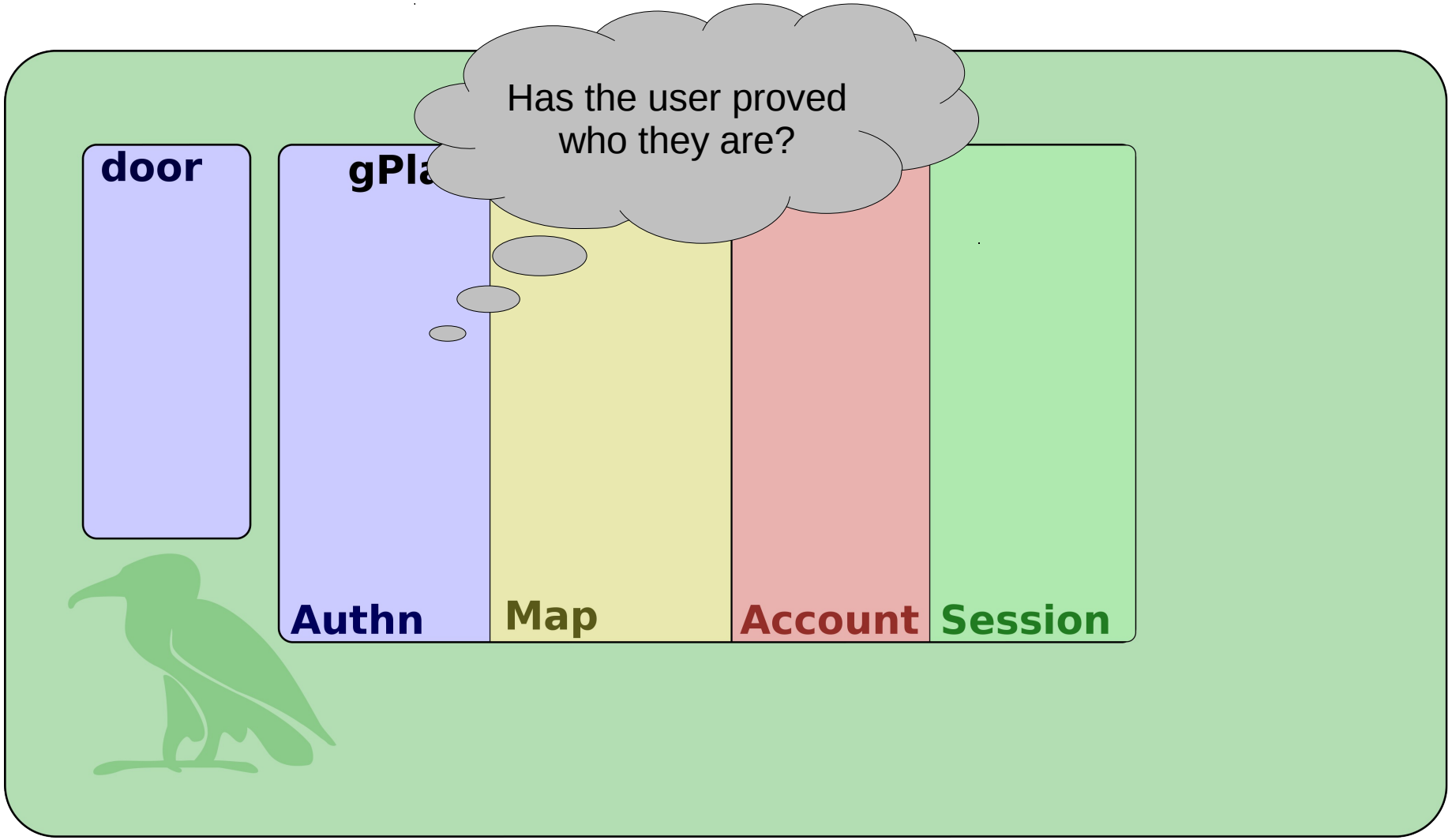
# gPlazma: concepts

*"The only way you can predict the future is to build it."* -Alan Kay

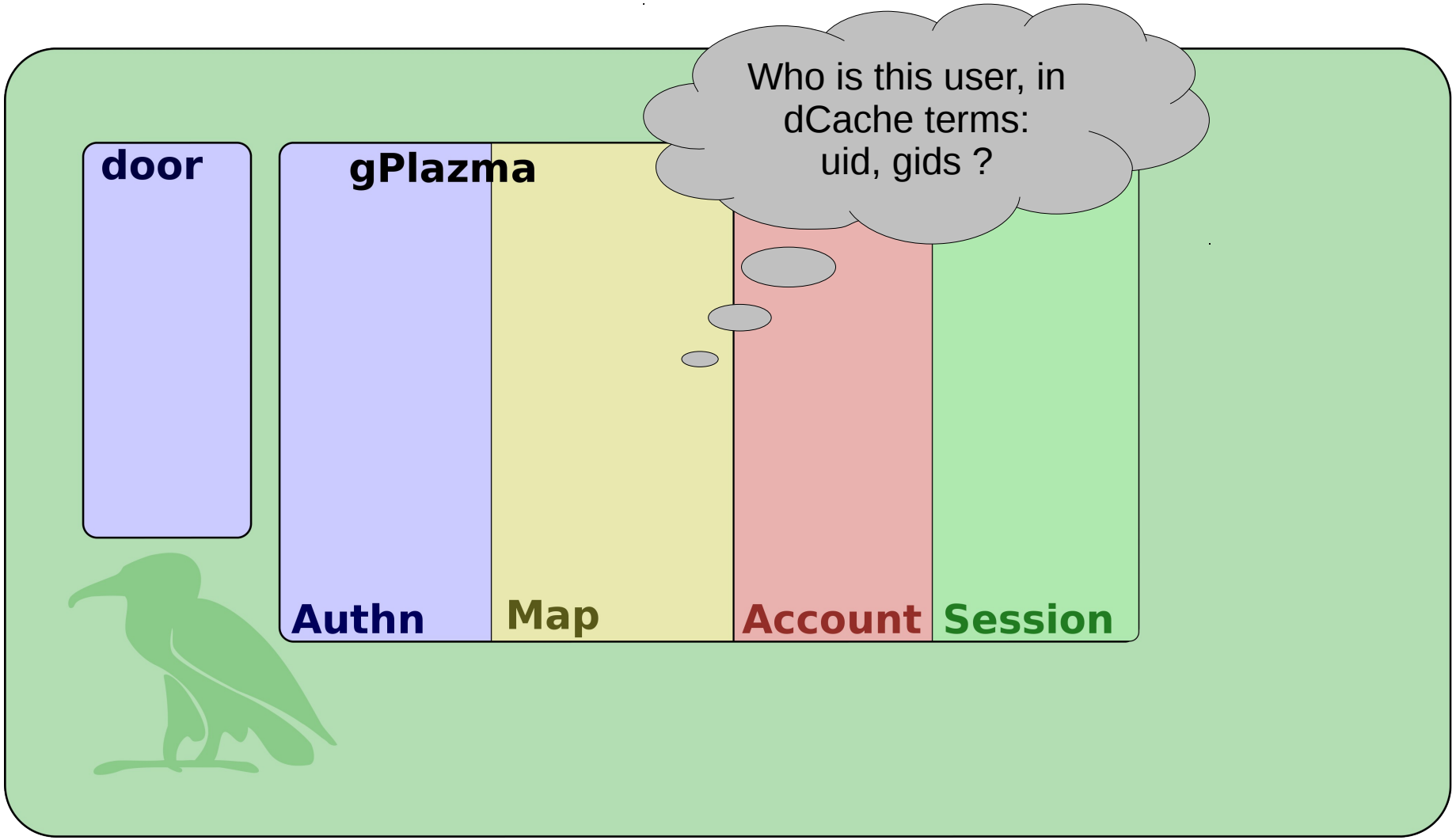
# Logging in: the four phases



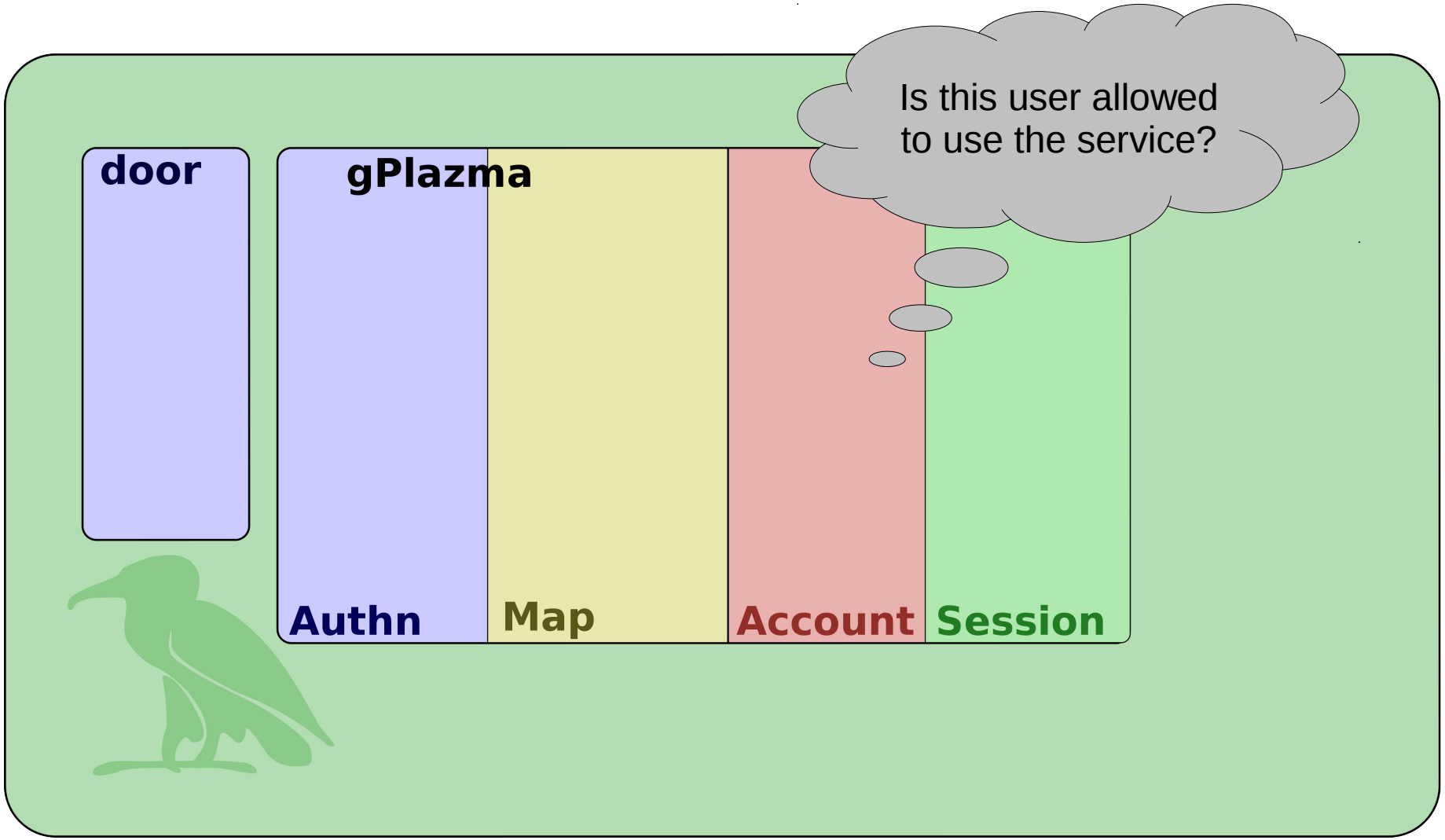
# Logging in: the four phases



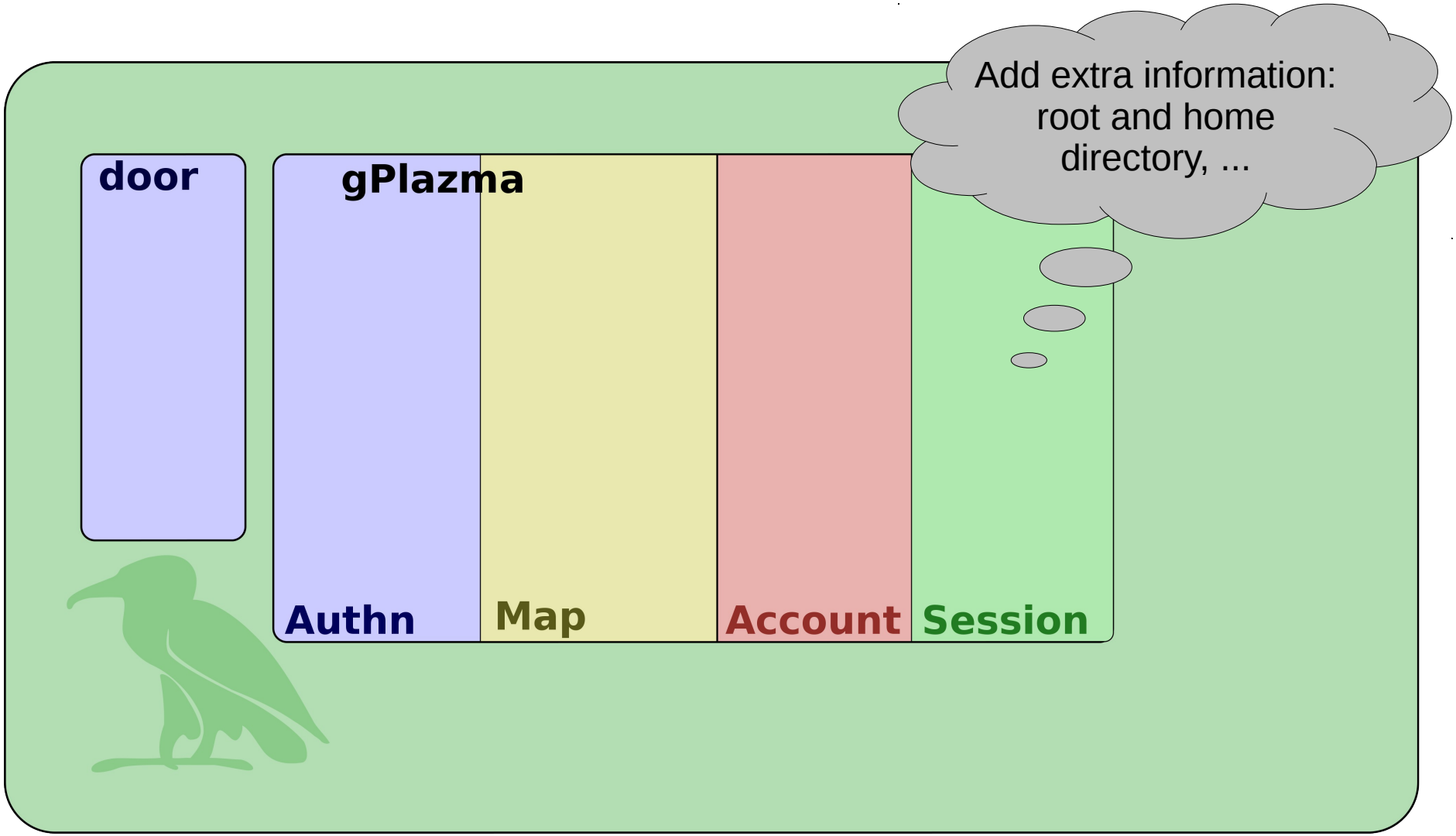
# Logging in: the four phases



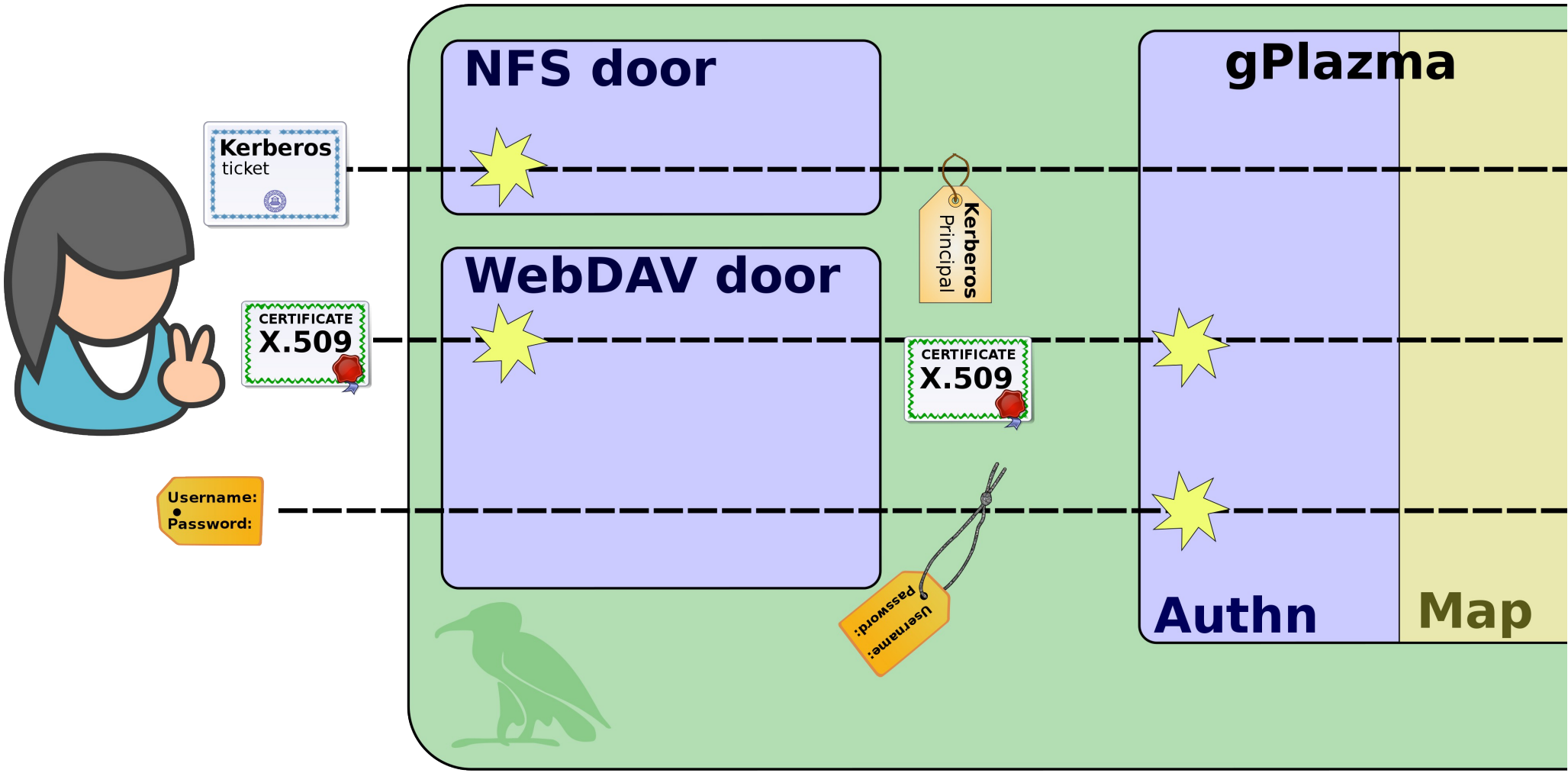
# Logging in: the four phases



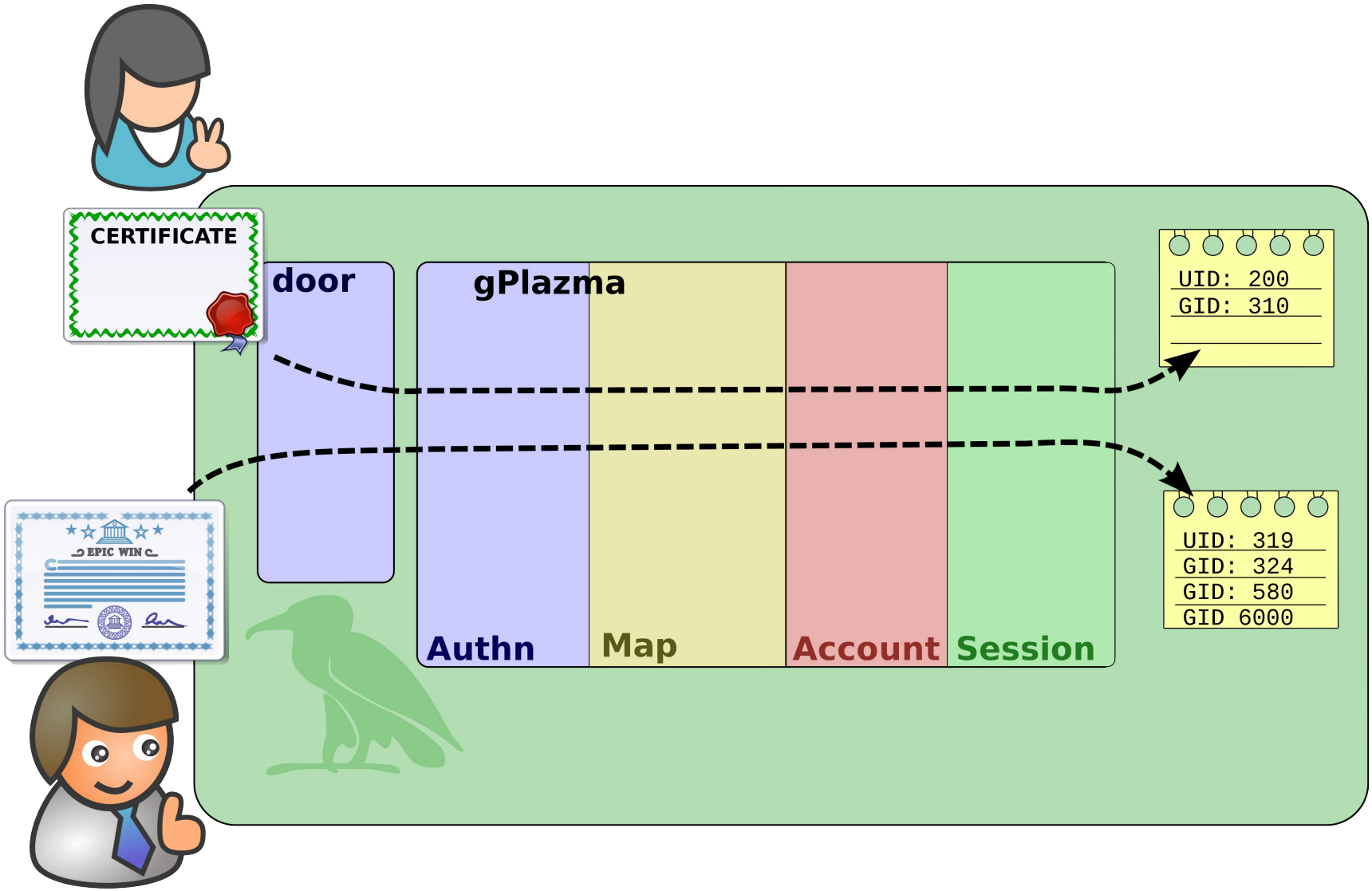
# Logging in: the four phases



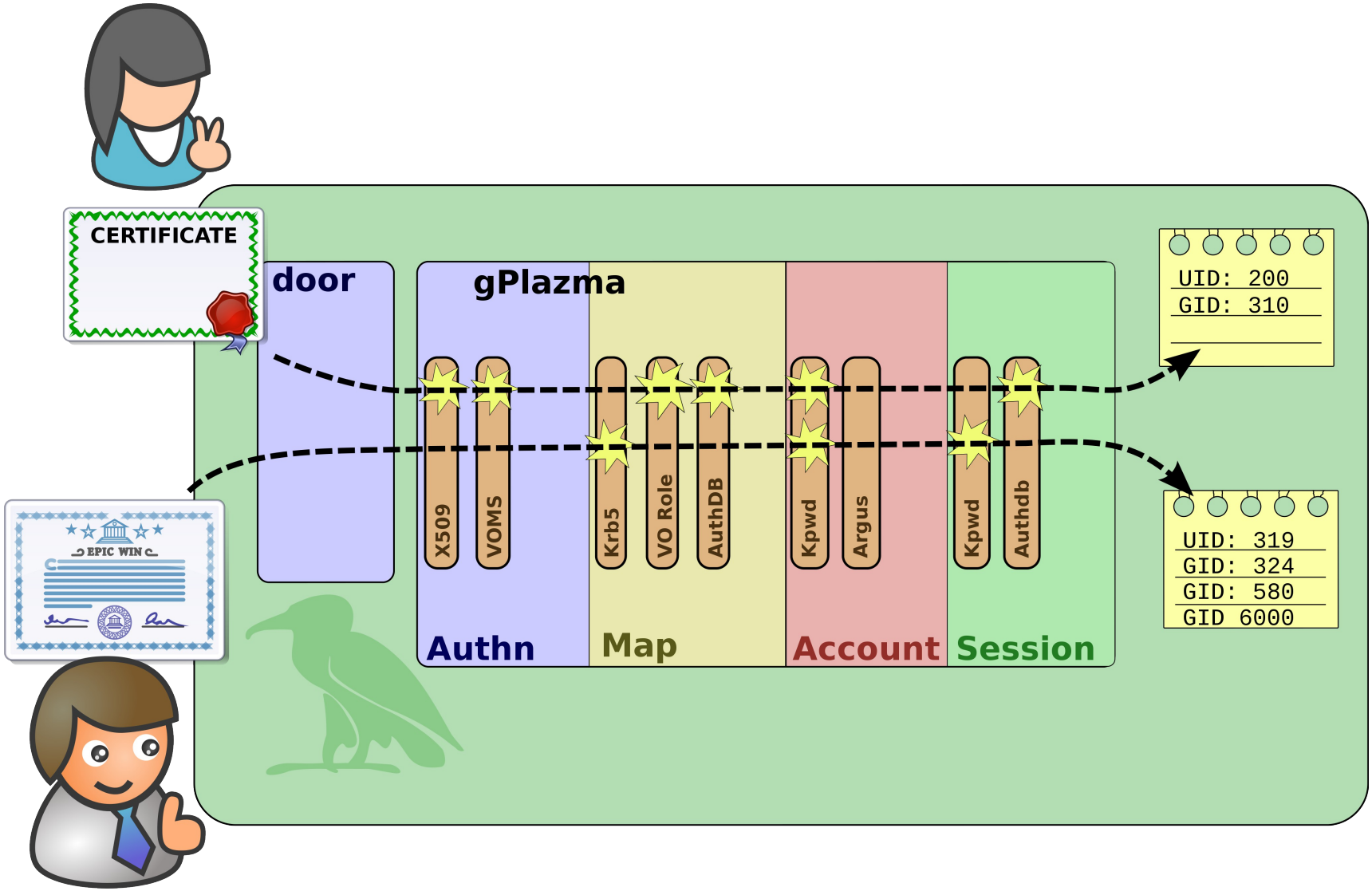
# Authentication: door, both or gPlazma



# Logging in: four phases



# Logging in: four phases, using plugins



# Example: username + password

/etc/dcache/gplazma.conf

```
# phase    wiring       plugin
```

# Example: username + password

- Use Kerberos server to check password.

/etc/dcache/gplazma.conf

```
# phase    wiring        plugin
auth       required      jaas
```

# Example: username + password

- Use Kerberos server to check password.
- Use NIS to map username to uid & gid

/etc/dcache/gplazma.conf

```
# phase    wiring        plugin
auth       required      jaas
map        required      nis
```

# Example: username + password

- Use Kerberos server to check password.
- Use NIS to map username to uid & gid.
- Use NIS again to discover user's home directory

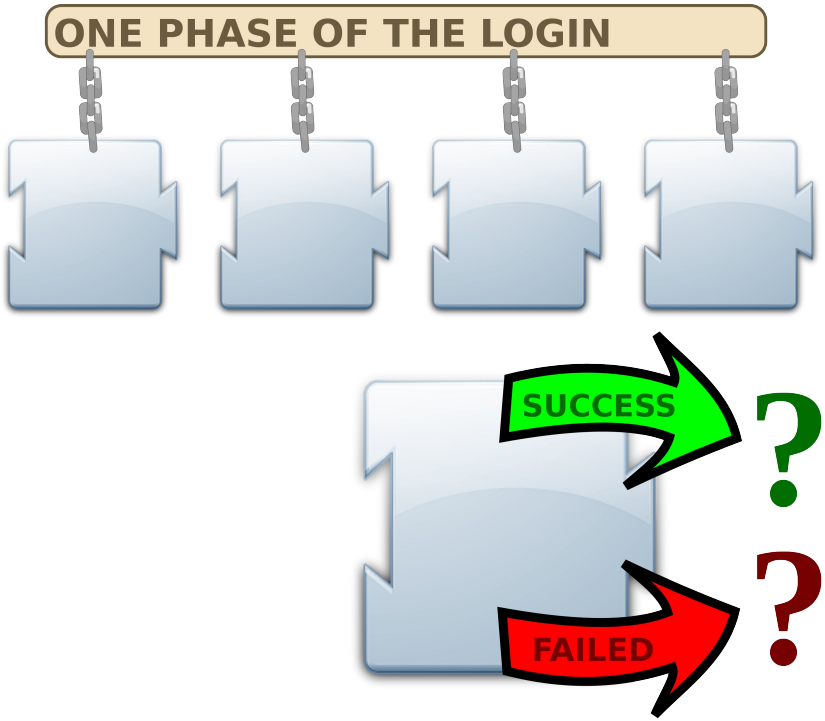
/etc/dcache/gplazma.conf

```
# phase    wiring        plugin
auth       required      jaas
map        required      nis
session    required      nis
```

# Wiring together a phase

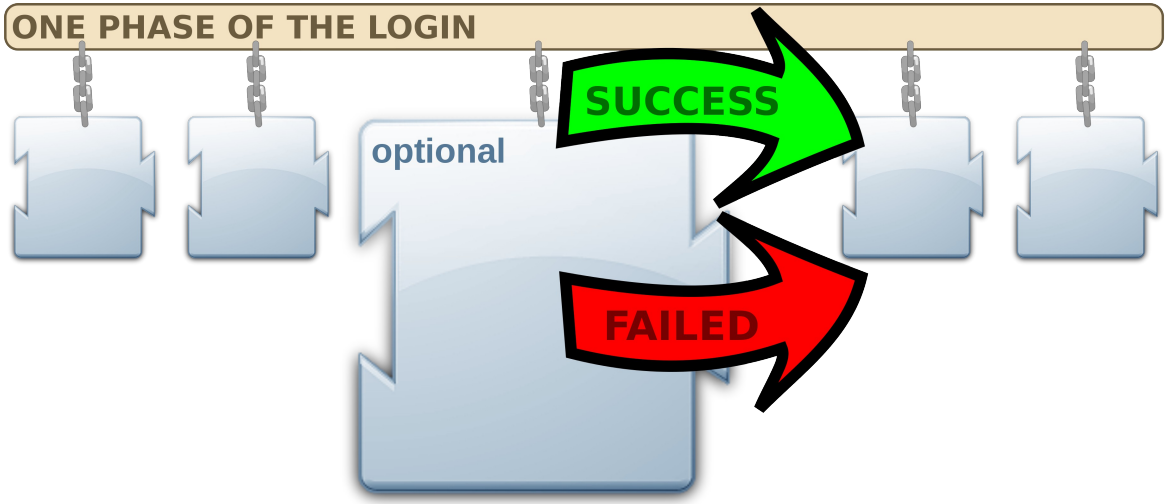
- Each plugin either **succeeds** or **fails**.

  The wiring describe what happens next:

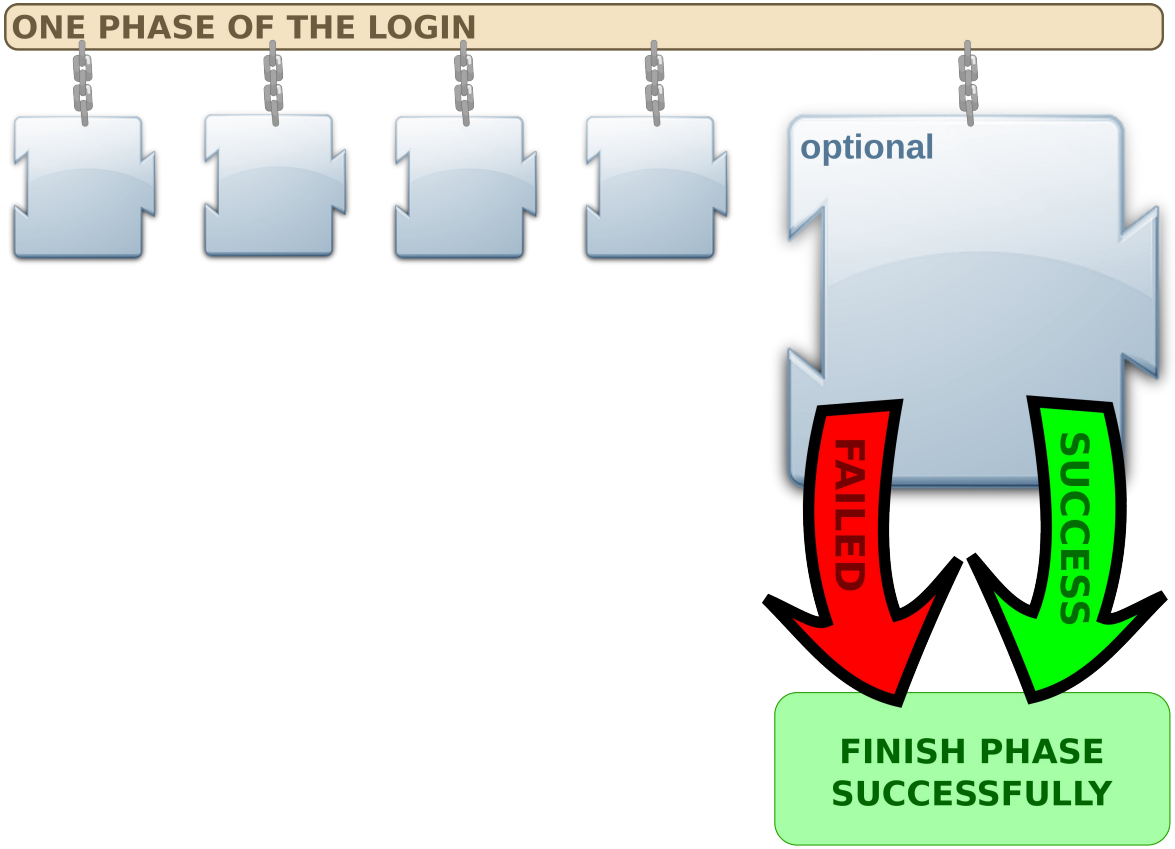



| Wiring | Description |
|---|---|
| `optional` | Always move onto next plugin in the phase |
| `sufficient` | Successful plugin finishes the phase with success |
| `requisite` | Failing plugin finishes the phase with failure |
| `required` | Failing plugin fails the phase but remaining plugins are still run |

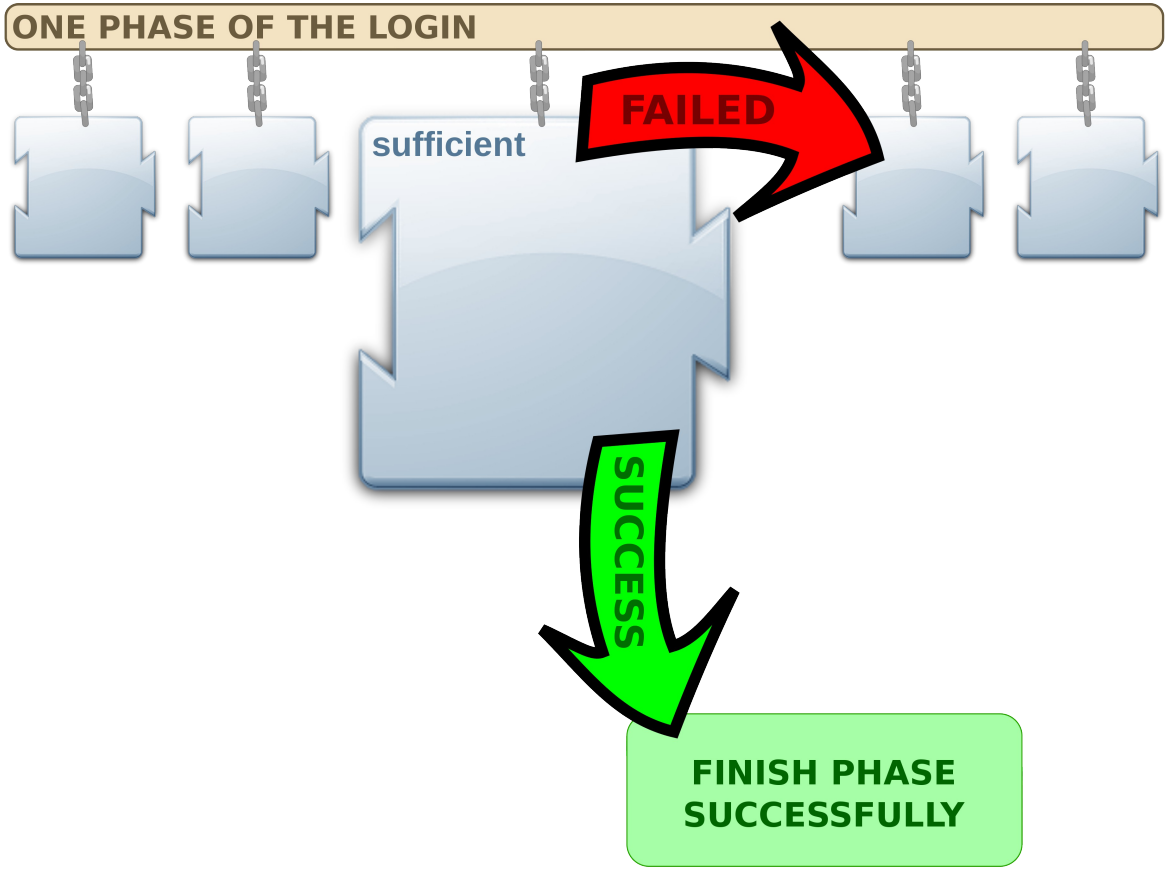
# optional: always move on



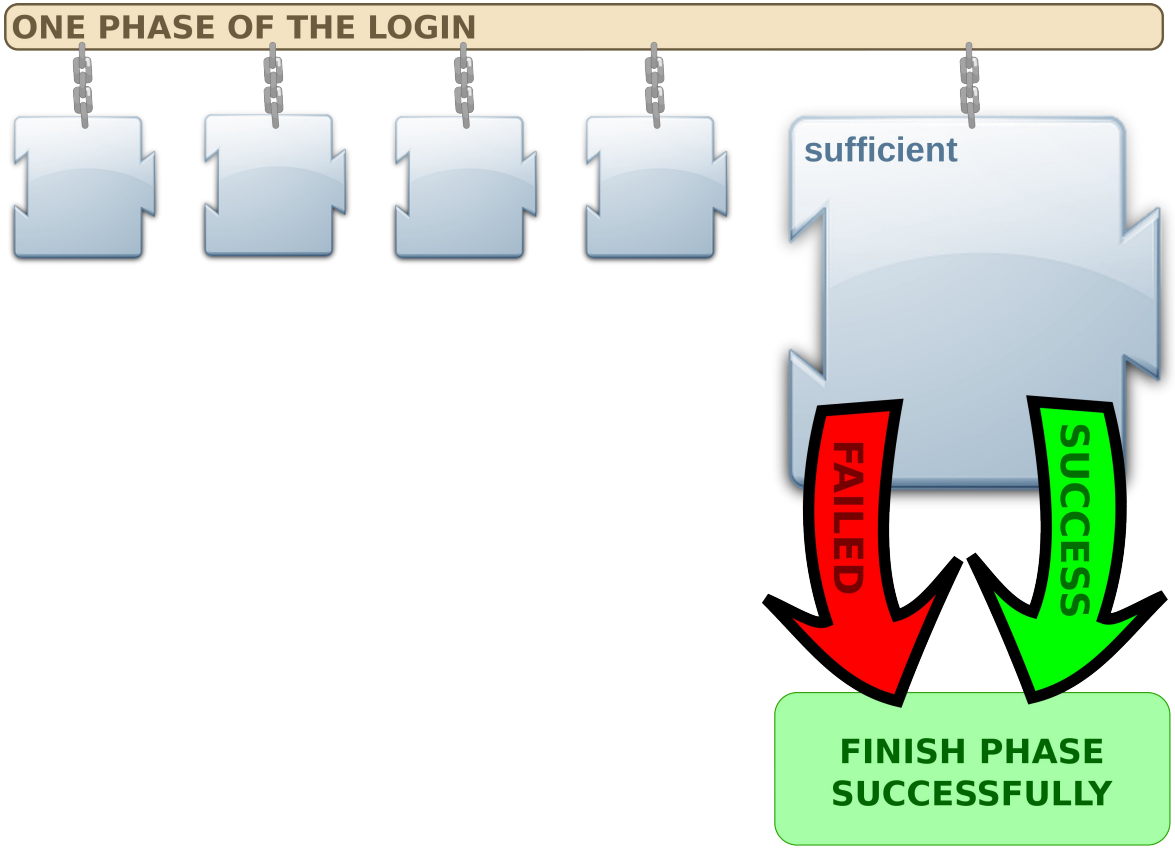
# optional: ends phase successfully



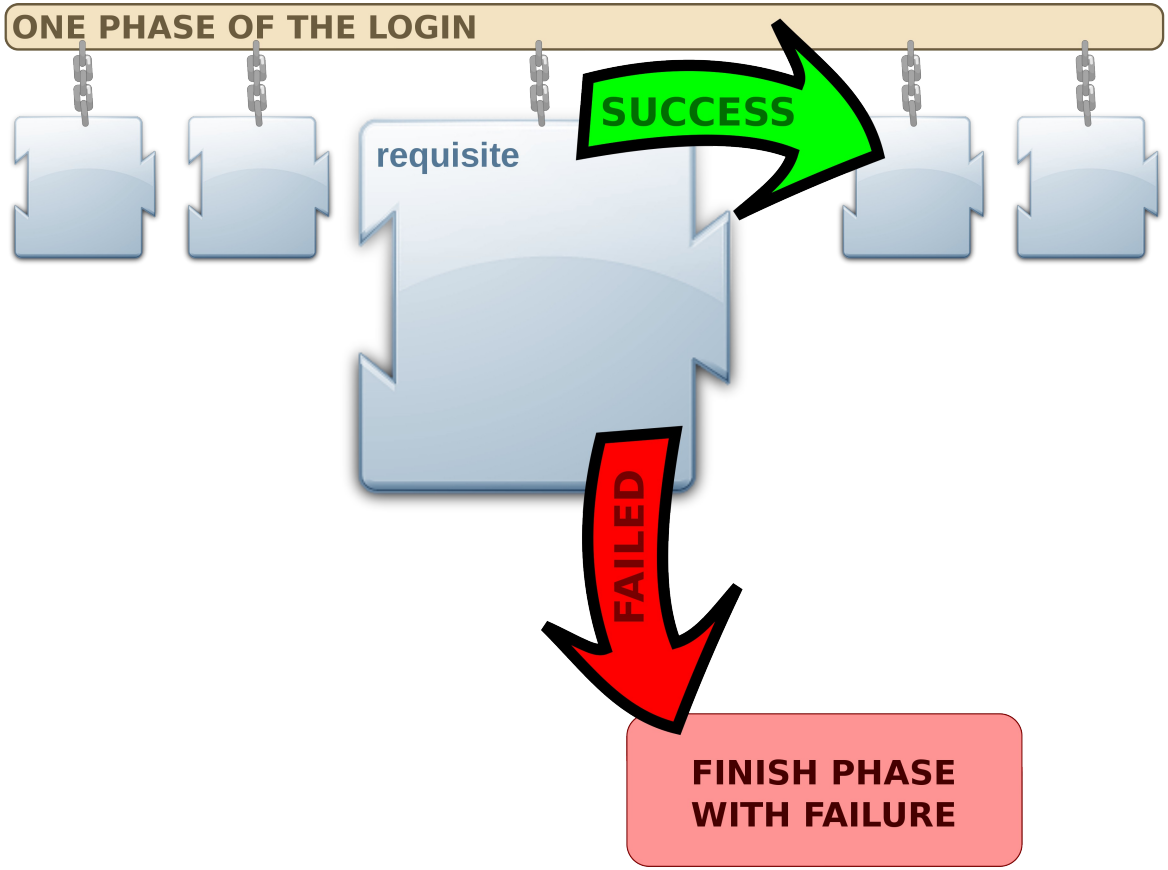
# sufficient: success finishes the phase



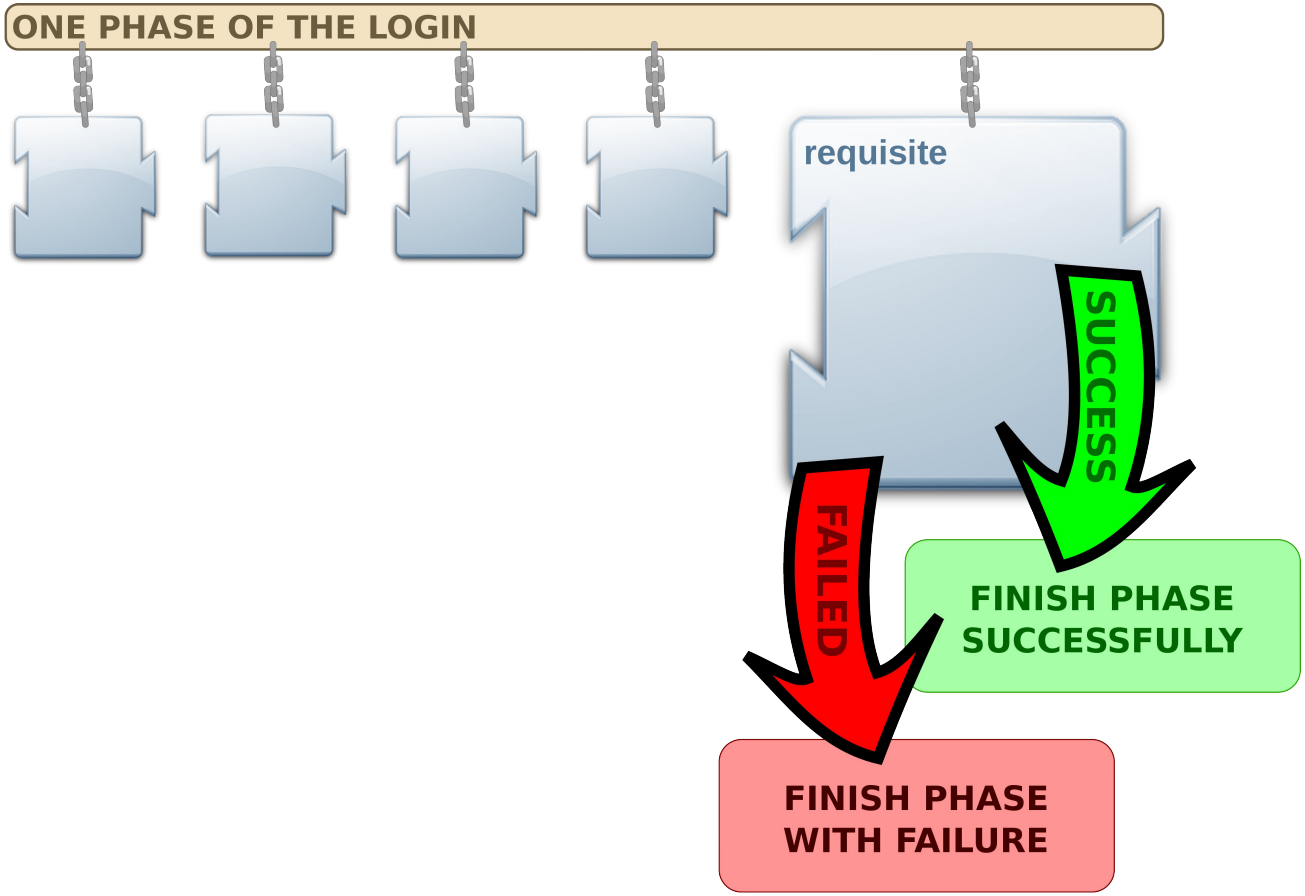
# sufficient: phase ends successfully



# requisite: failure will fail the phase

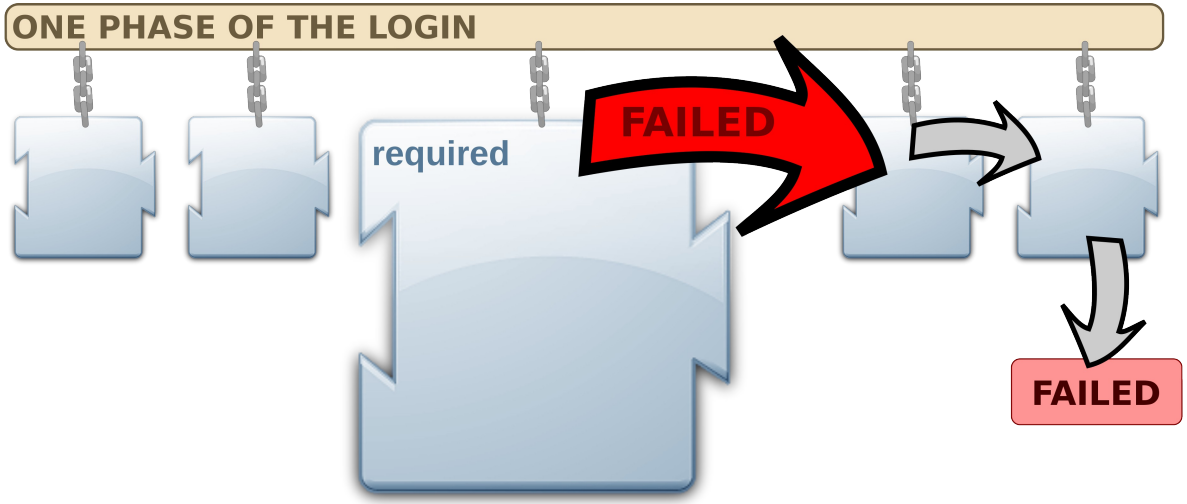


# requisite: .. also if last plugin

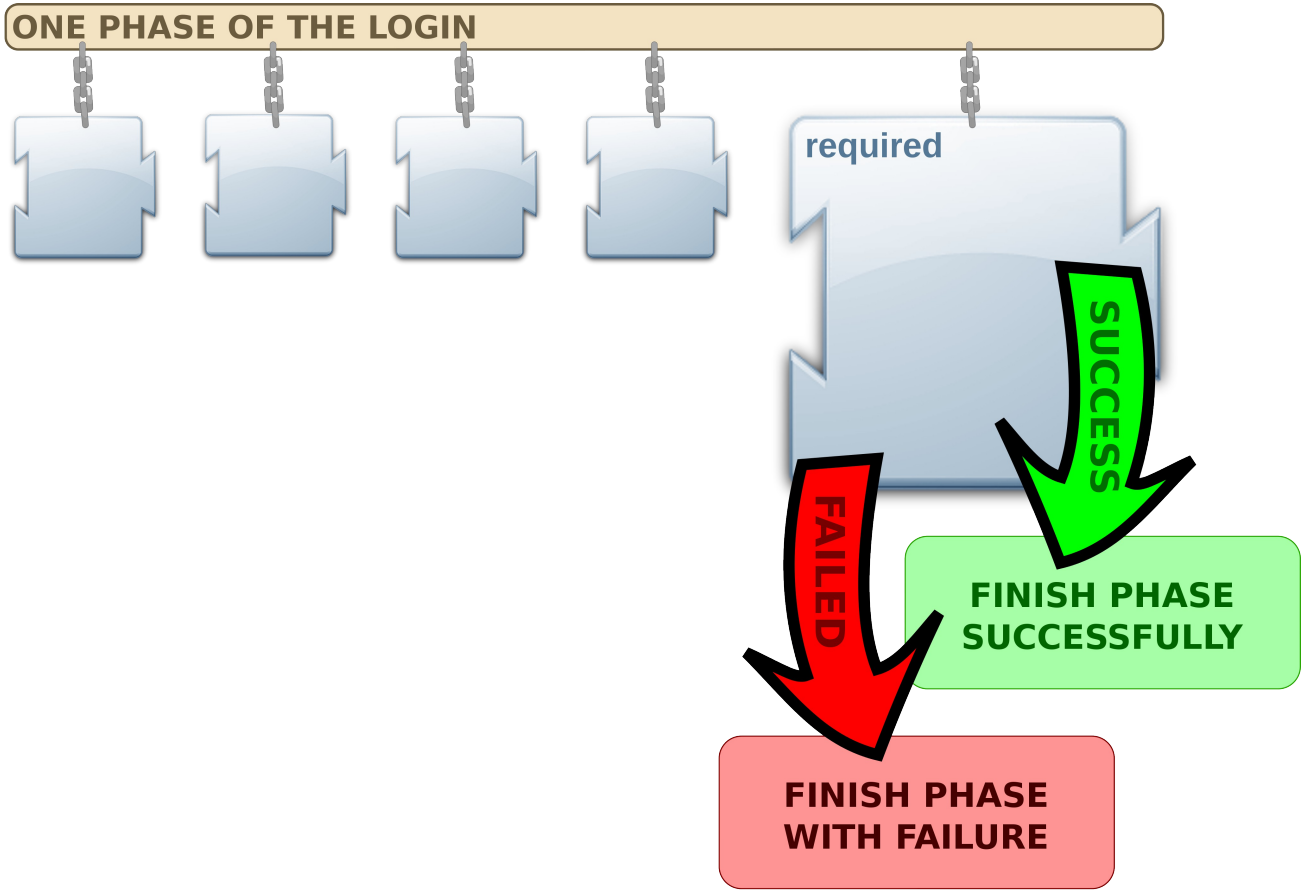


# required: success move onto next plugin

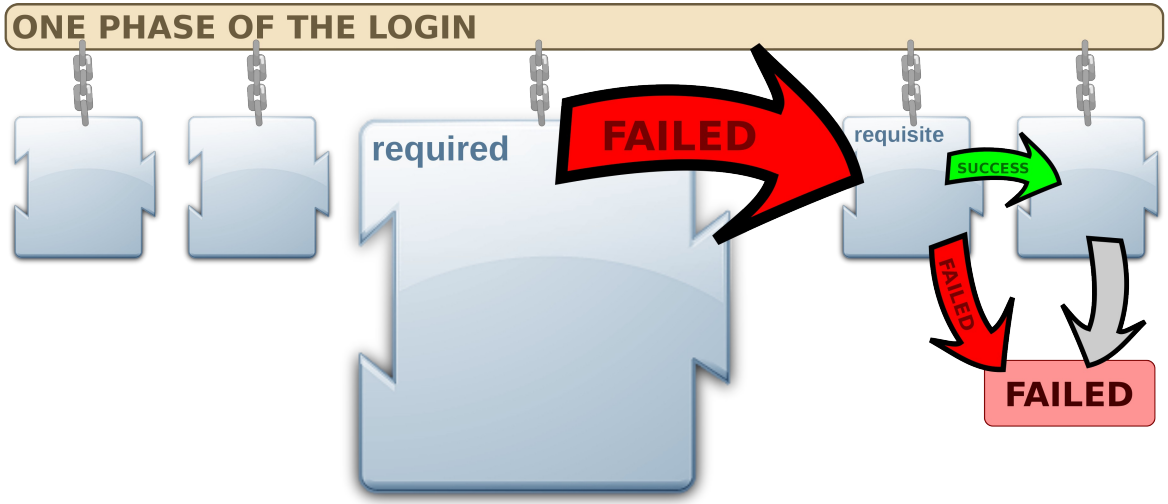
# required: failure continue, but fail phase



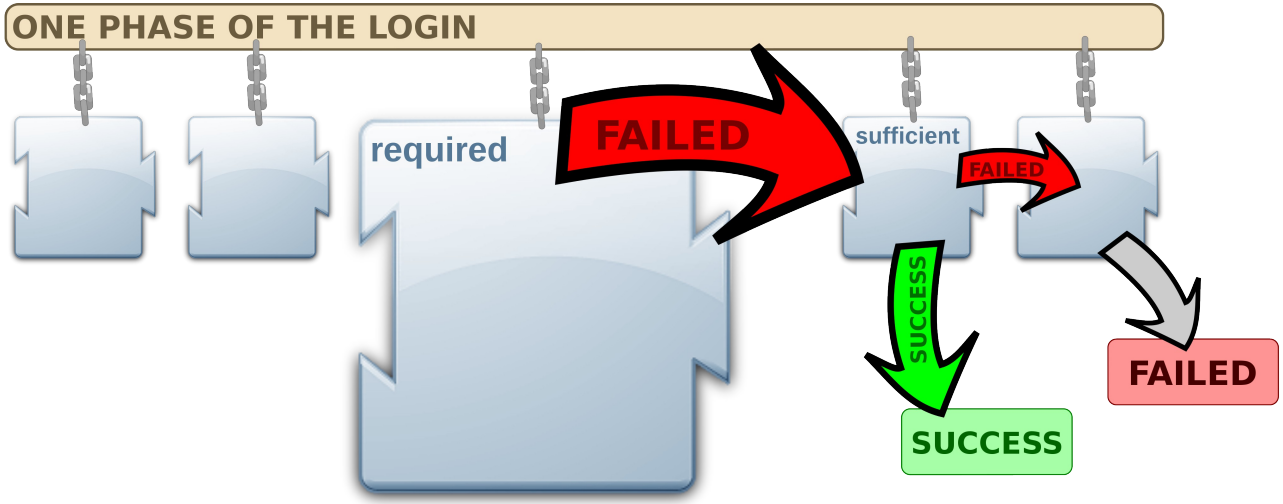
# required: similar to requisite



# required: complication #1



# required: complication #2



# gPlazma configuration file

```
auth    optional    x509
auth    optional    voms
auth    optional    kpwd        gplazma.kpwd.file=/etc/dcache.kpwd

map     optional    krb5
map     optional    vorolemap
map     sufficient  authzdb
map     requisite   kpwd

account requisite   argus

session optional    authzdb
session optional    kpwd
```

# gPlazma configuration file

**First column is required and identifies in which phase the plugin is being configured**

```
auth     optional    x509
auth     optional    voms
auth     optional    kpwd        gplazma.kpwd.file=/etc/dcache.kpwd

map      optional    krb5
map      optional    vorolemap
map      sufficient  authzdb
map      requisite   kpwd

account  requisite   argus

session  optional    authzdb
session  optional    kpwd
```

# gPlazma configuration file

**Second column is required and identifies the wiring: what to do next**

```
auth    optional   x509
auth    optional   voms
auth    optional   kpwd        gplazma.kpwd.file=/etc/dcache.kpwd

map     optional   krb5
map     optional   vorolemap
map     sufficient authzdb
map     requisite  kpwd

account requisite  argus

session optional   authzdb
session optional   kpwd
```

# gPlazma configuration file

**Third column is required and identifies which plugin is being configured**

```
auth     optional   x509
auth     optional   voms
auth     optional   kpwd       gplazma.kpwd.file=/etc/dcache.kpwd

map      optional   krb5
map      optional   vorolemap
map      sufficient authzdb
map      requisite  kpwd

account  requisite  argus

session  optional   authzdb
session  optional   kpwd
```

# gPlazma configuration file

**Final column is optional and provide local configuration. Configuration is normally achieved in dcache.conf and layout file**

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

map     optional    krb5
map     optional    vorolemap
map     sufficient  authzdb
map     requisite   kpwd

account requisite   argus

session optional    authzdb
session optional    kpwd
```

```
gplazma.kpwd.file=/etc/dcache.kpwd
```

# gPlazma configuration file

**The plugin configuration order is the order in which they are run**

**x509, voms, kpwd**
**krb5, vorolemap, authzdb, kpwd**
**argus**
**authzdb, kpwd**

```
auth     optional    x509\
auth     optional    voms
auth     optional    kpwd        gplazma.kpwd.file=/etc/dcache.kpwd

map      optional    krb5
map      optional    vorolemap
map      sufficient  authzdb
map      requisite   kpwd

account  requisite   argus

session  optional    authzdb
session  optional    kpwd
```

# gPlazma configuration file

**The same plugin may appear multiple times, in different phases**

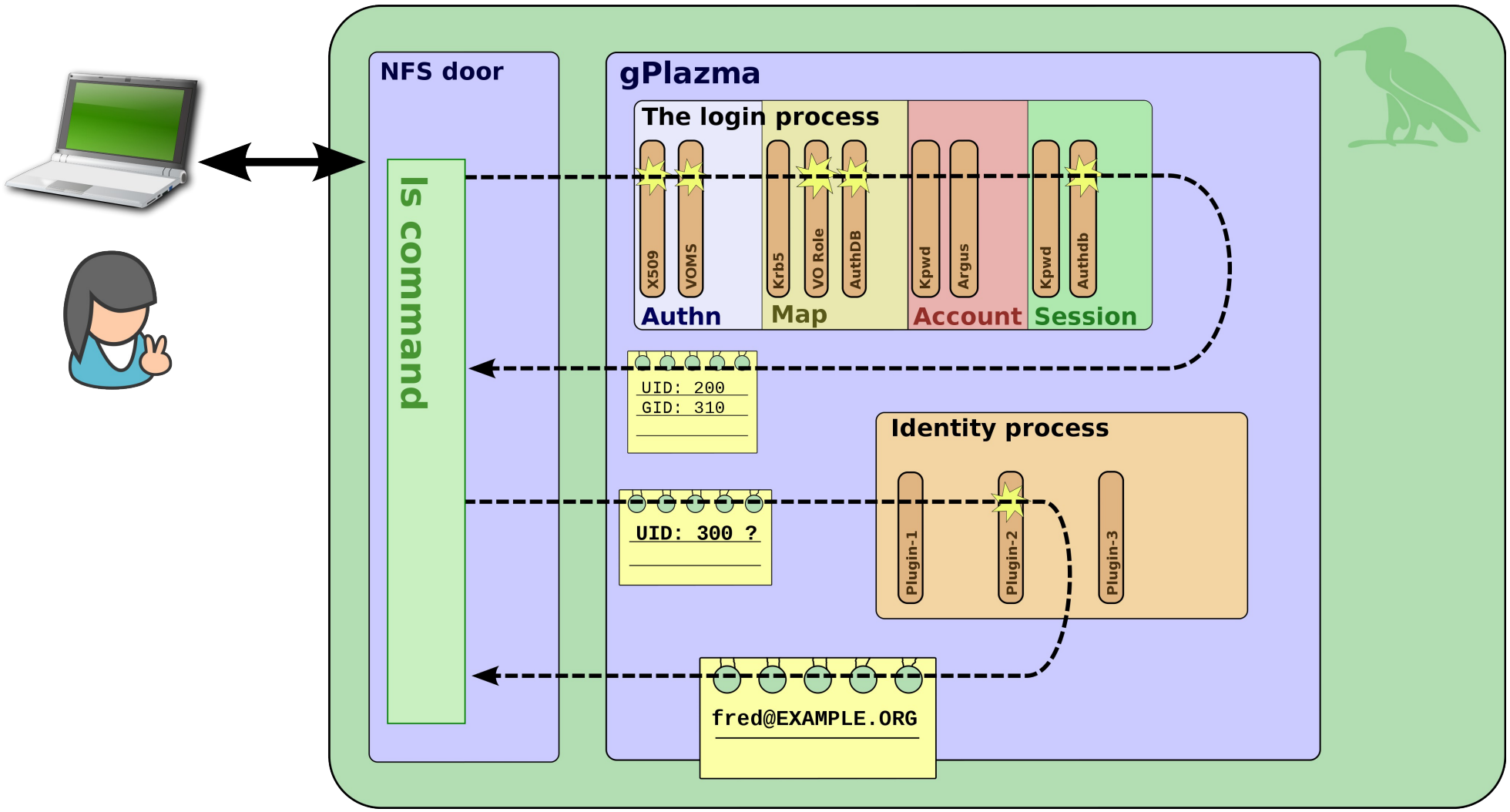
```
auth    optional    x509
auth    optional    voms
auth    optional    kpwd          gplazma.kpwd.file=/etc/dcache.kpwd

map     optional    krb5
map     optional    vorolemap
map     sufficient  authzdb
map     requisite   kpwd

account requisite   argus

session optional    authzdb
session optional    kpwd
```

# Something extra: identity mapping



# Plugins that come with dCache

# Anyone can write plugins

- Plugins can be written by anyone

  Sites can interface with bespoke infrastructure

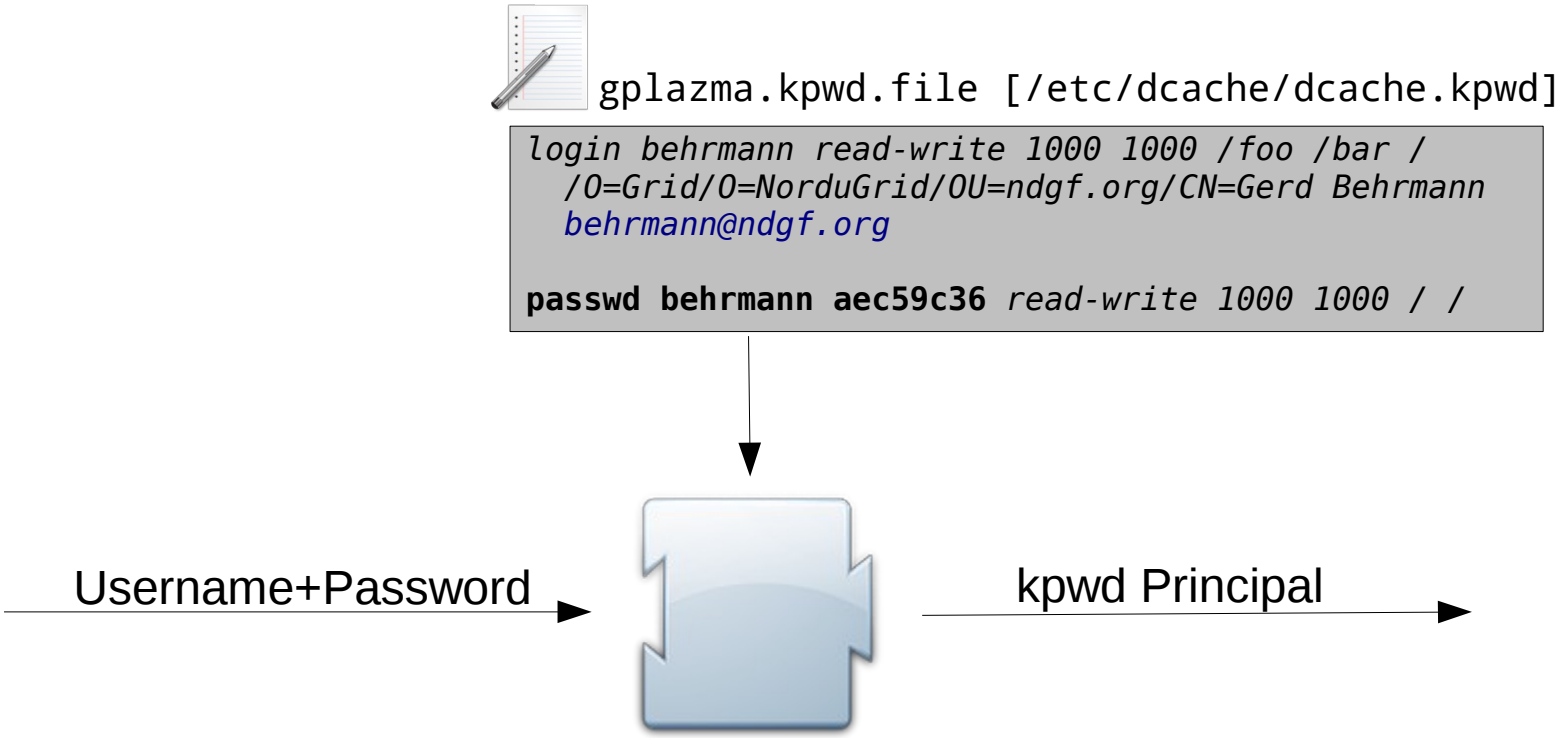
- Don't worry, you don't need to write code!
- dCache comes with **13 plugins**:

  enough to cover most situations,

  more can be added; e.g., contribution from external developers.

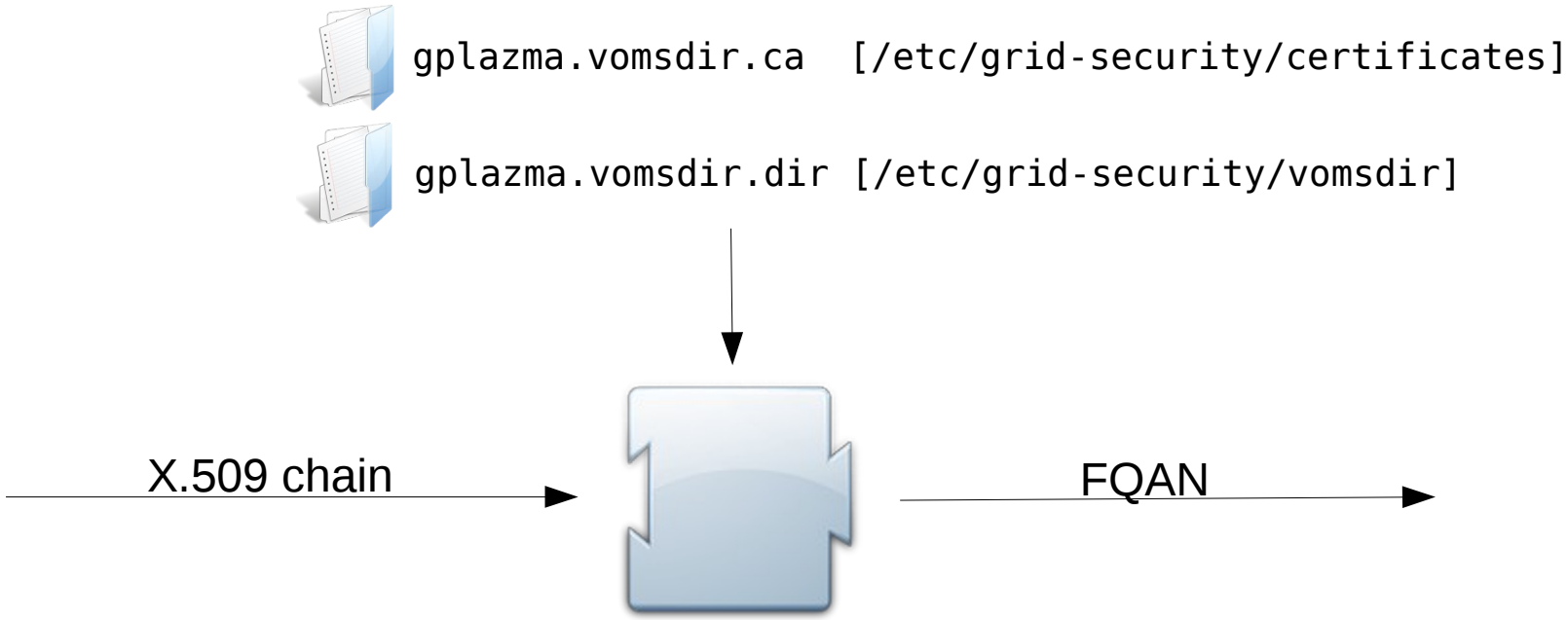
# The five auth plugins

# kpwd: username+password

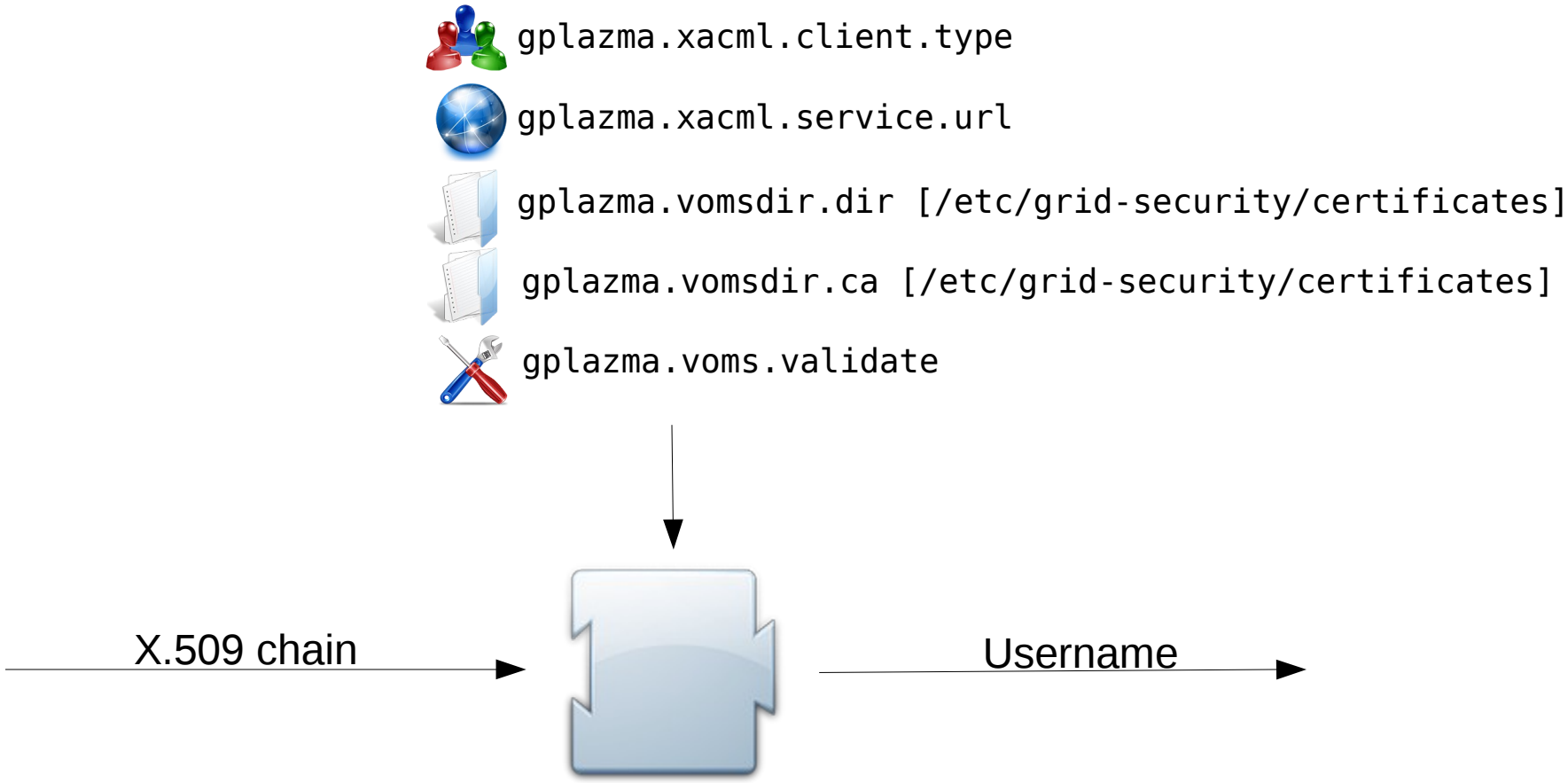


# x509: extract DN

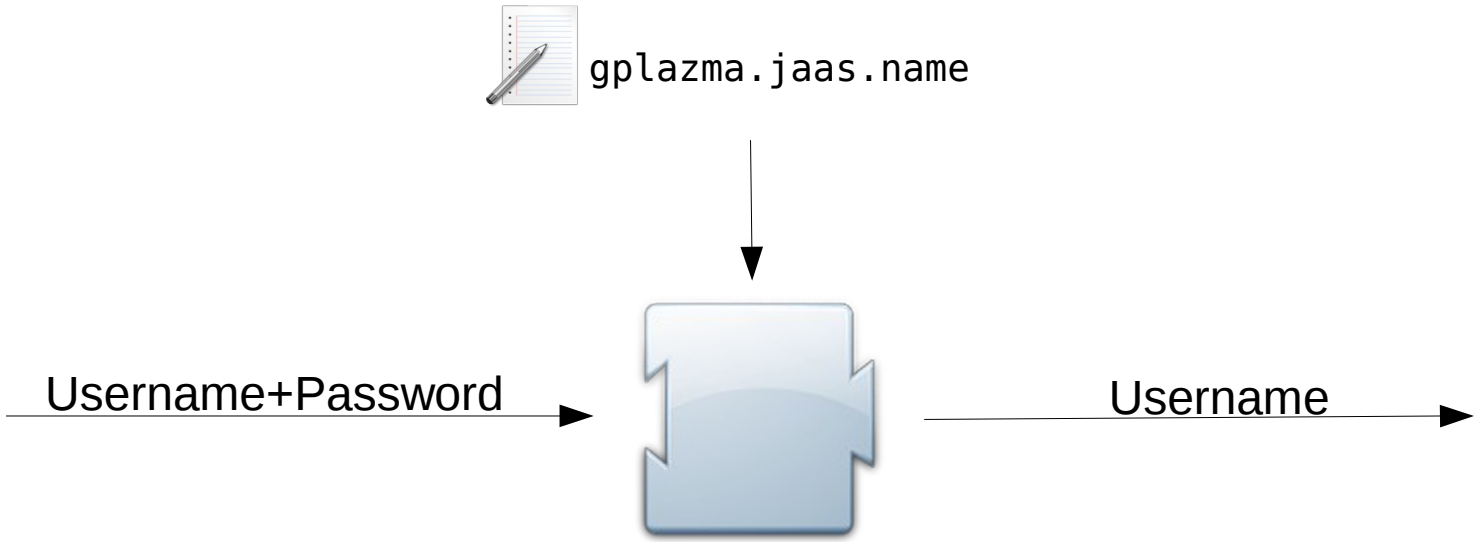
# voms: extract FQANs



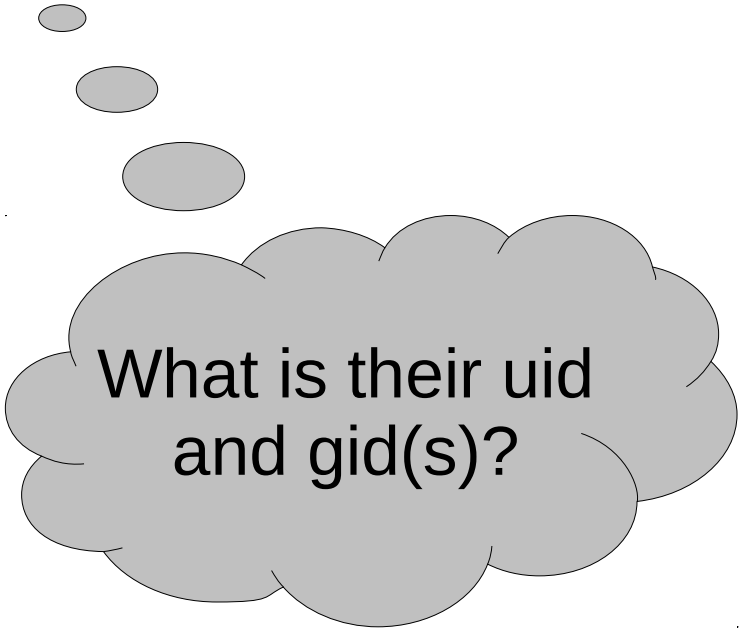
# xacml: call out to external server



# jaas: try to 'login' elsewhere



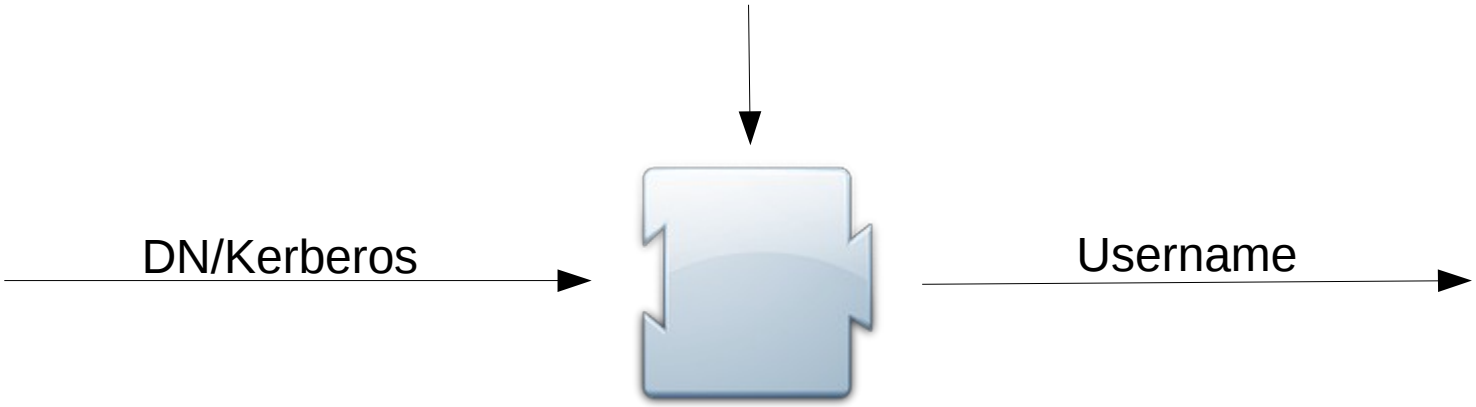
# The eight map plugins



# kpwd: use local file

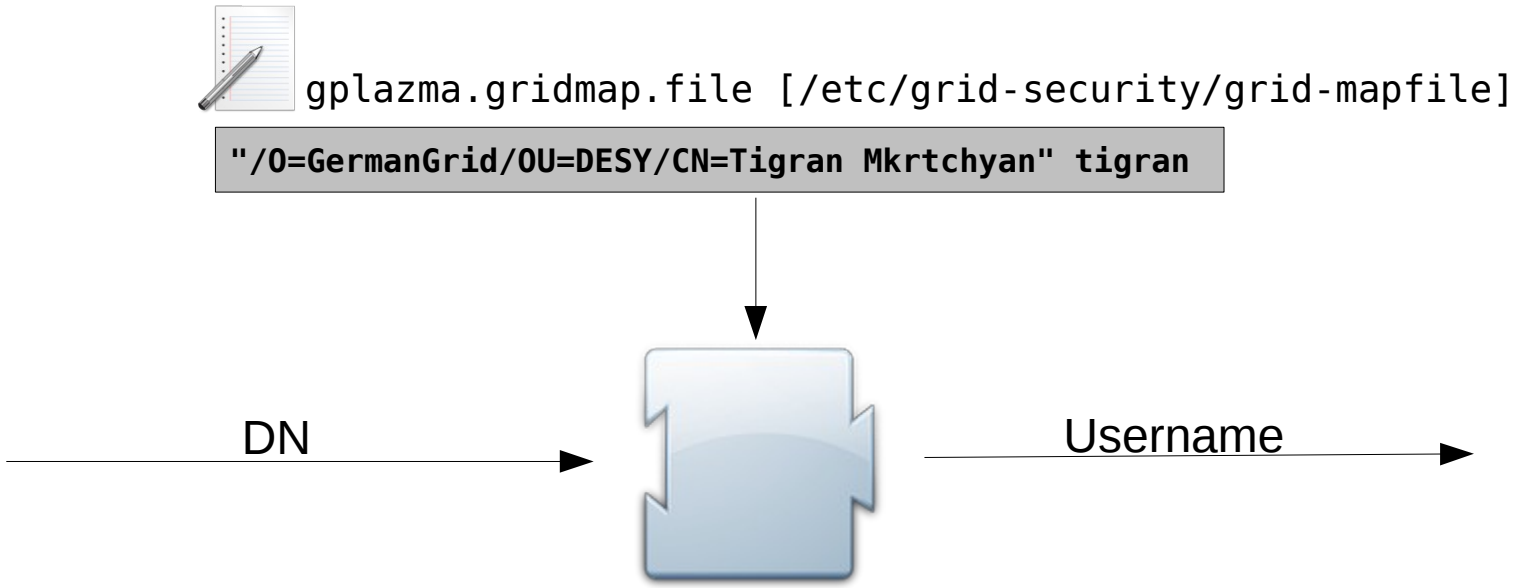
gplazma.kpwd.file [/etc/dcache/dcache.kpwd]

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

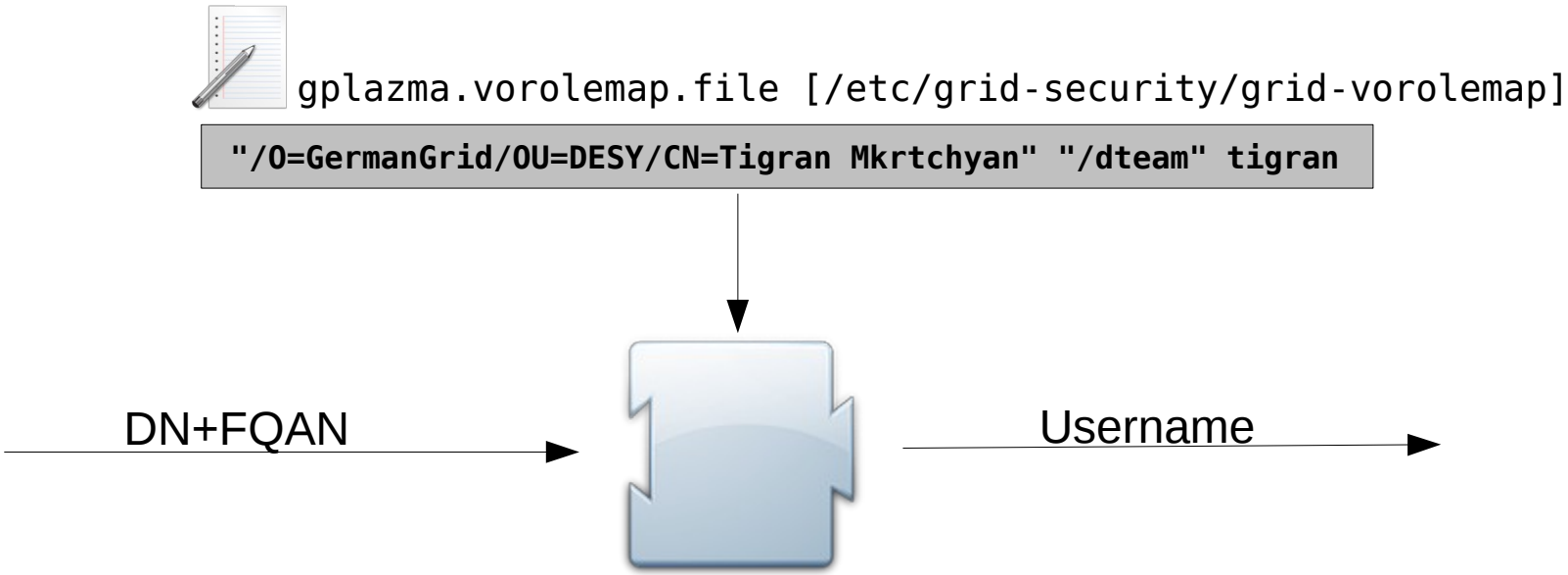


# krb5: Kerberos principal to username

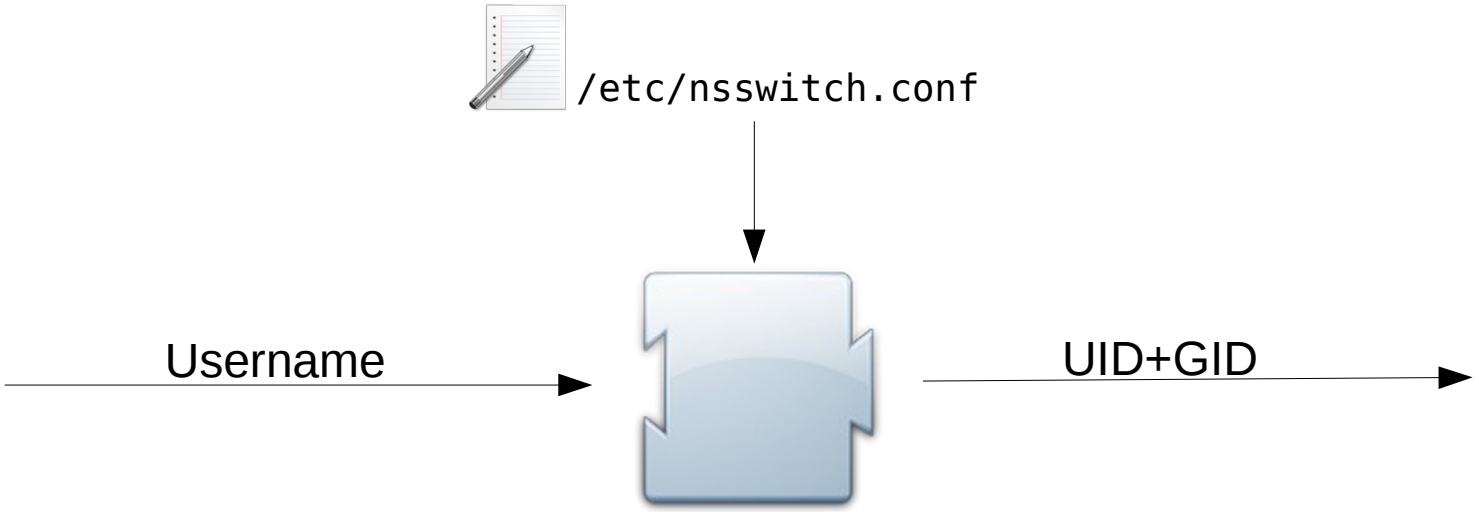
# gridmap: use local file



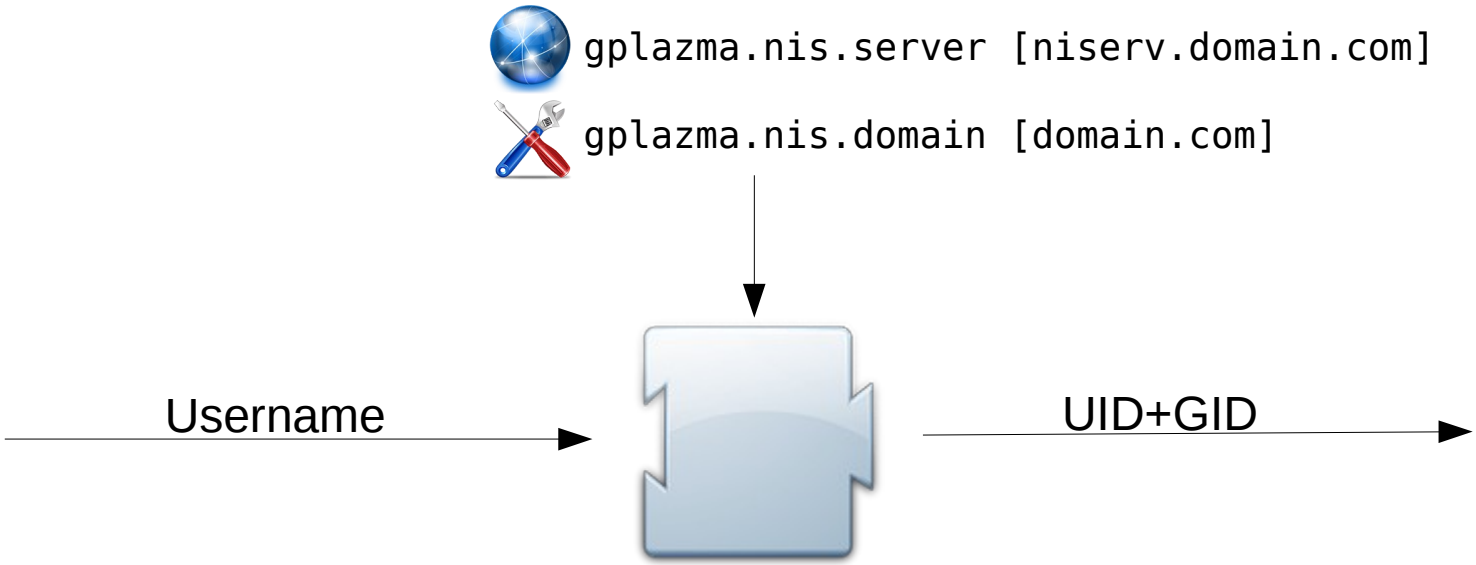
# vorolemap: use local file



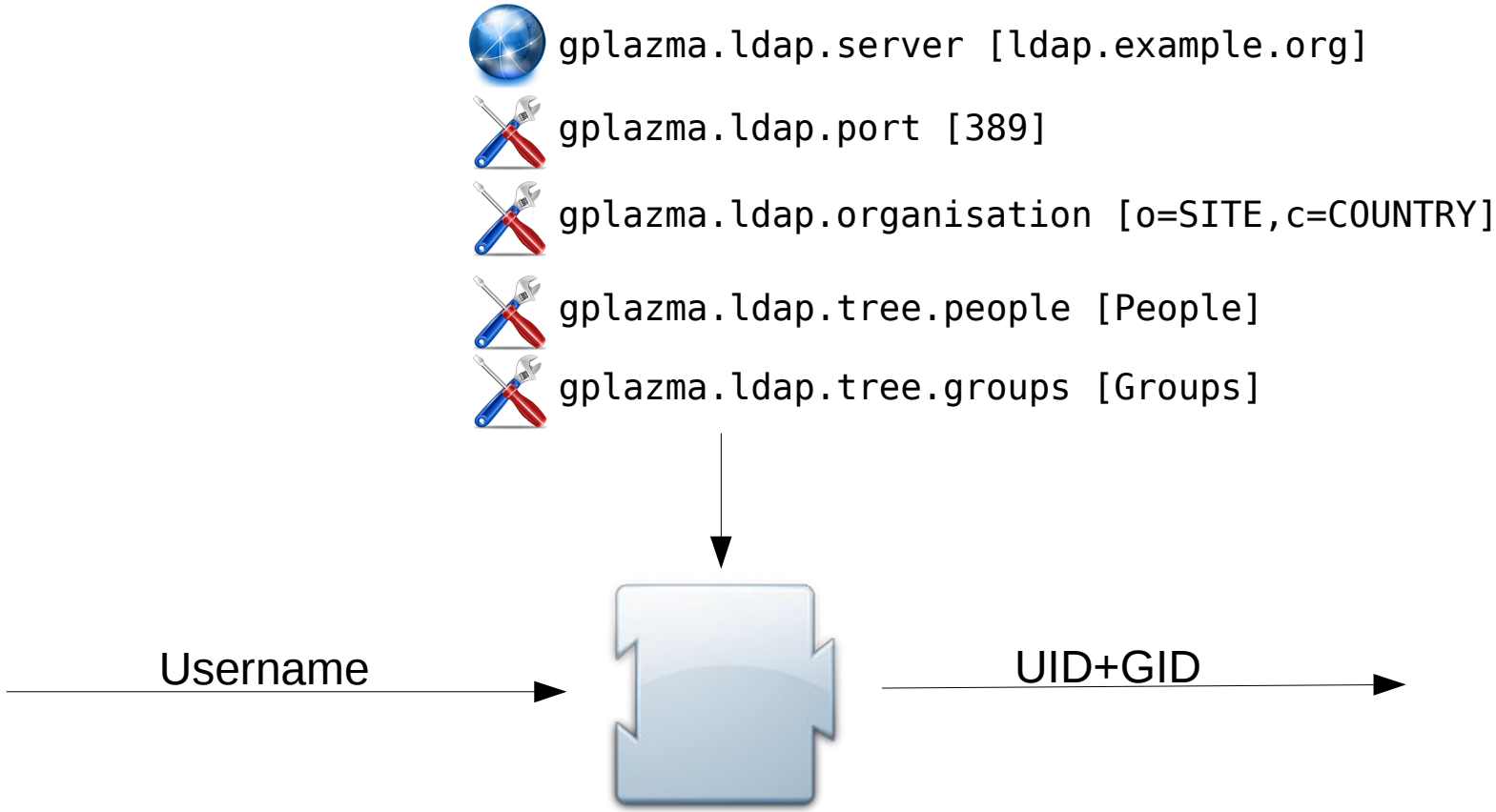
# nsswitch: use local OS mapping



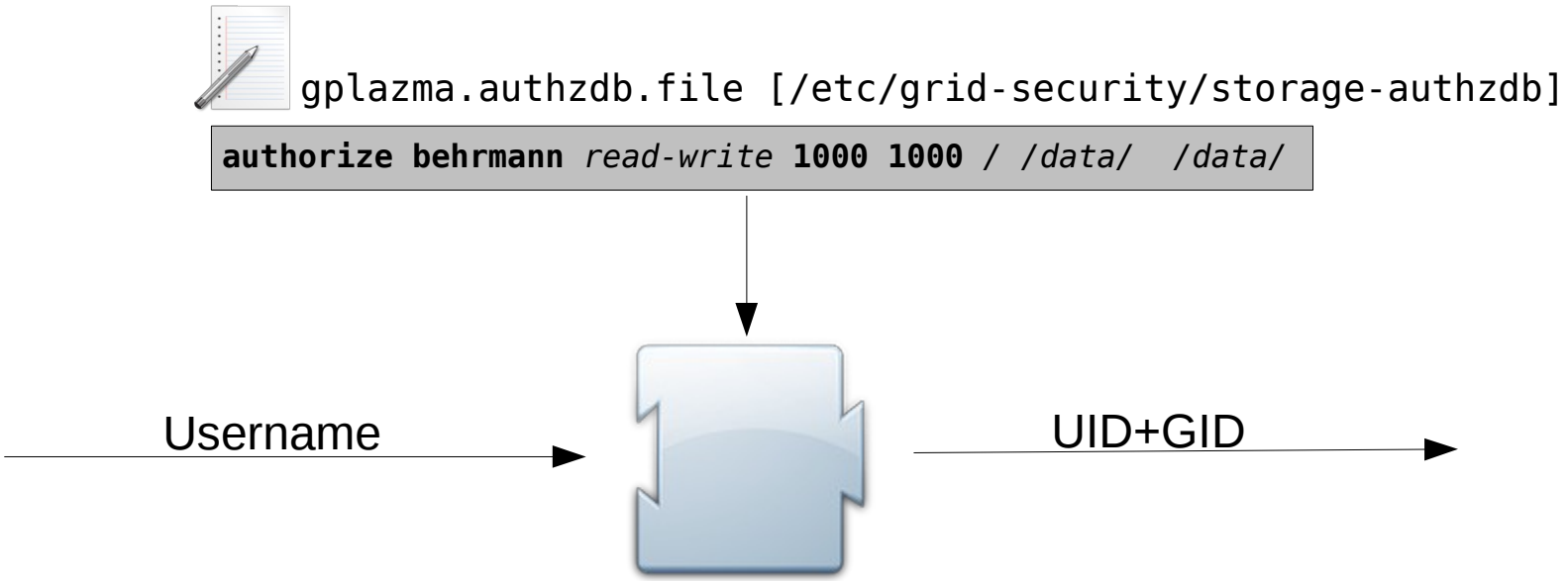
# nis: lookup in site infrastructure



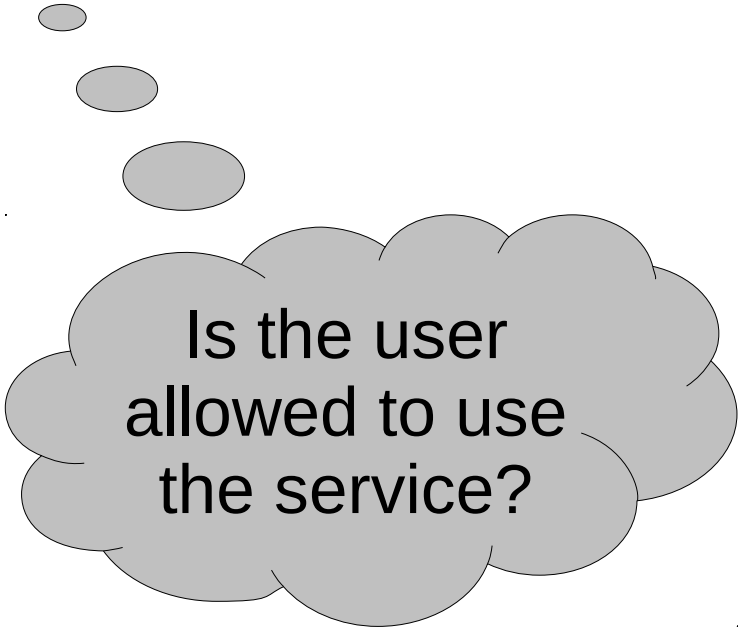
# ldap: lookup in site infrastructure



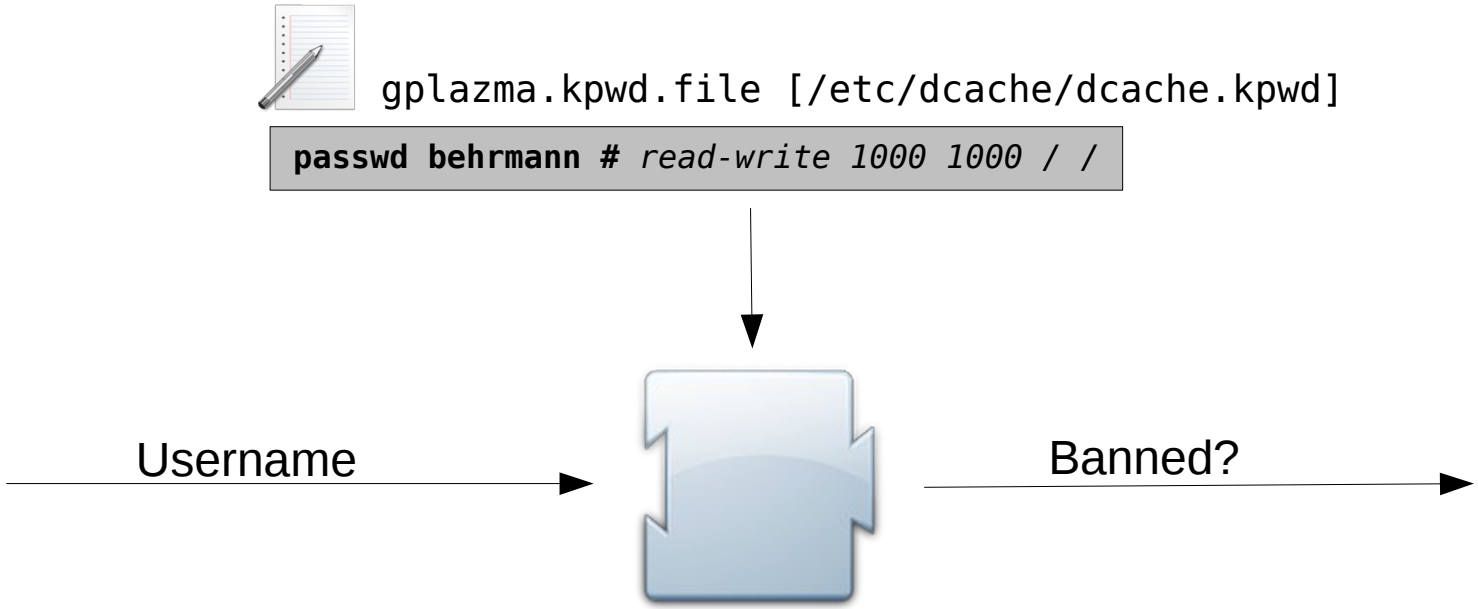
# authzdb: use local file



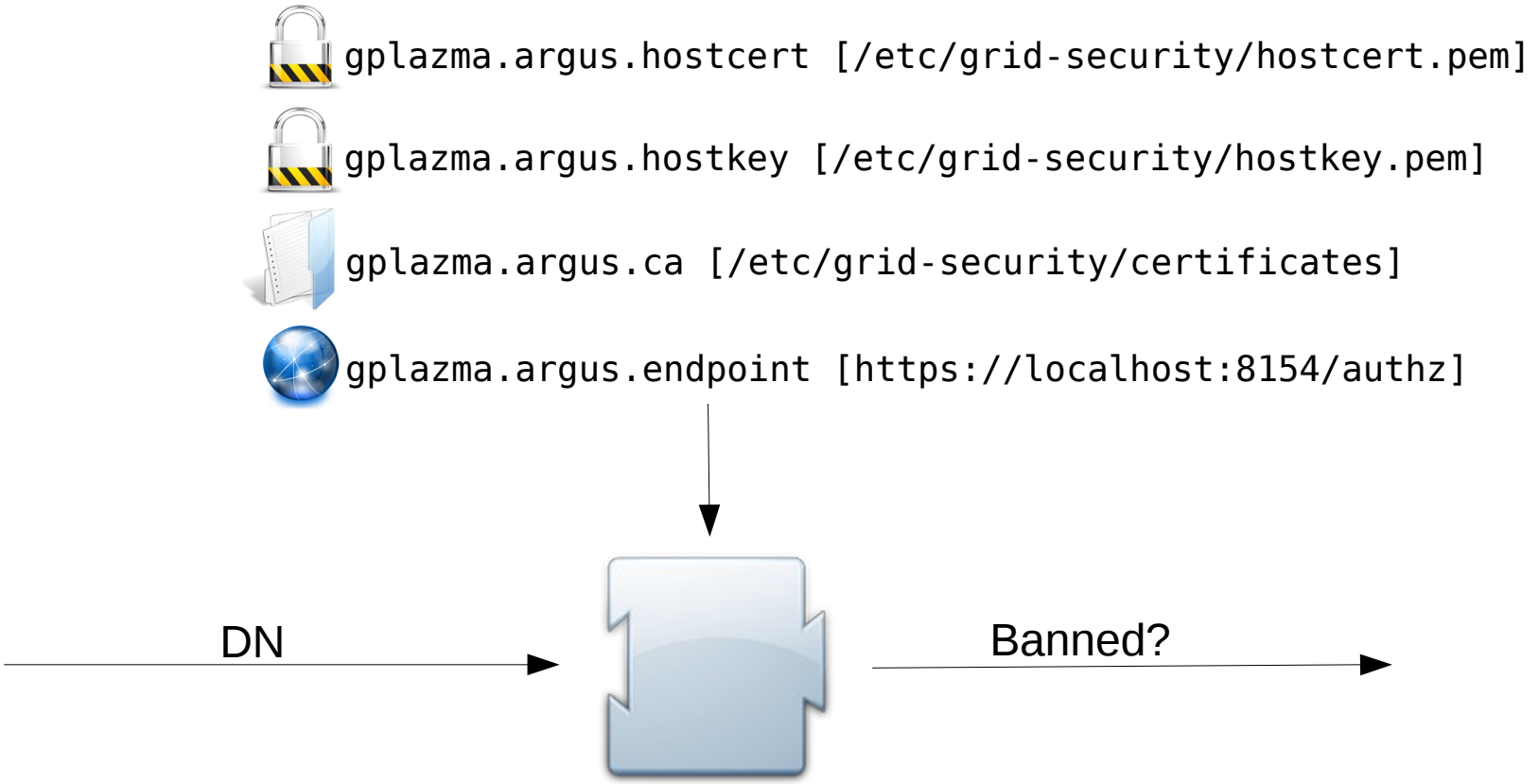
# The two account plugins



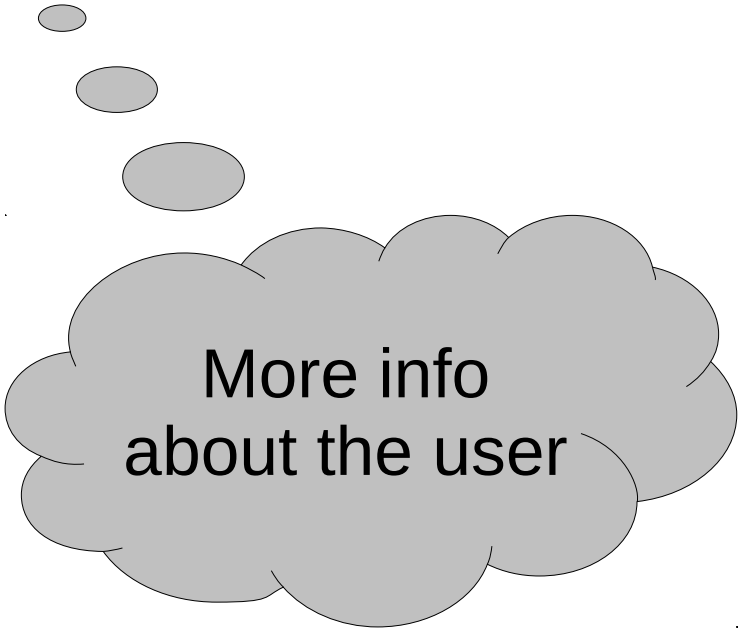
# kpwd: use local file



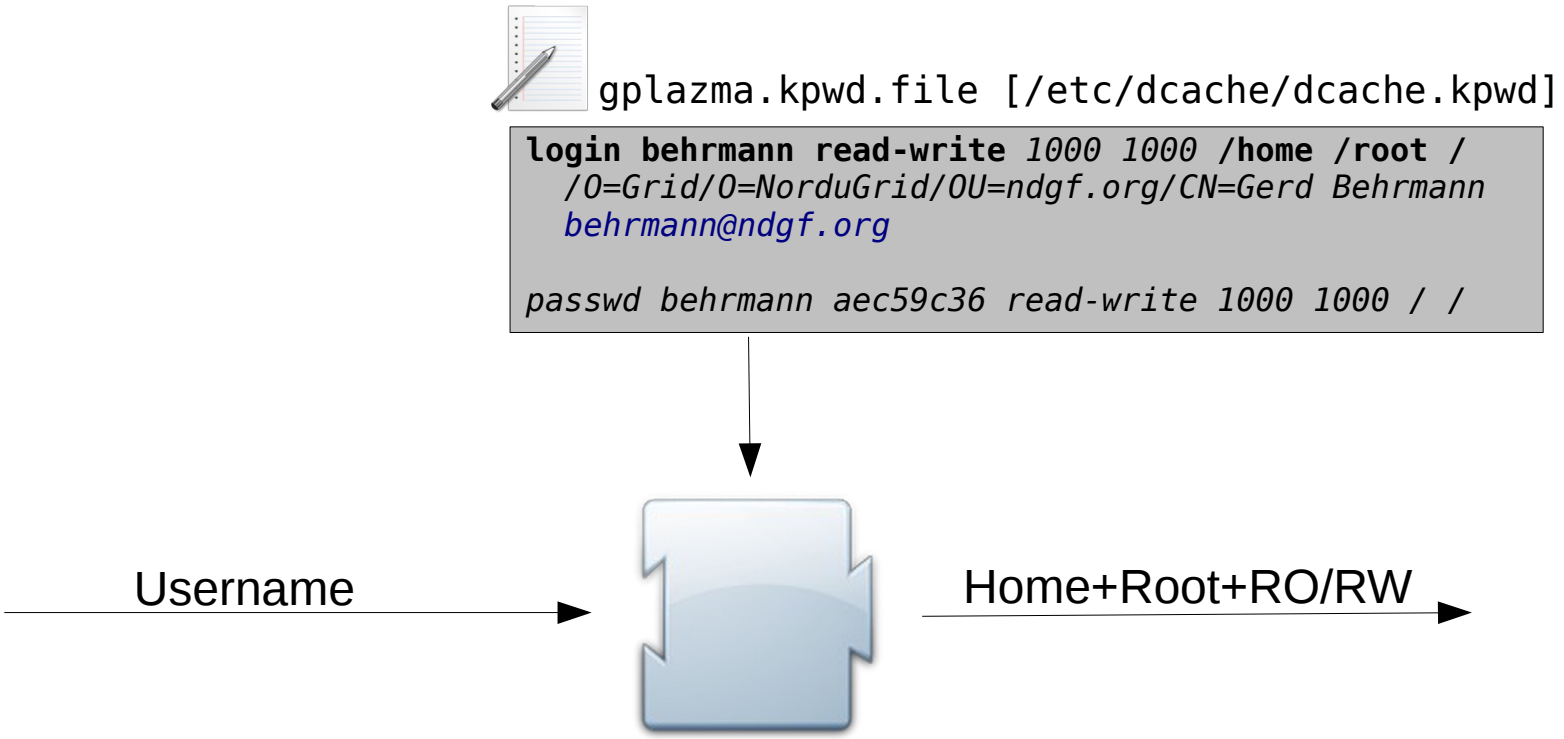
# argus: call out to external service



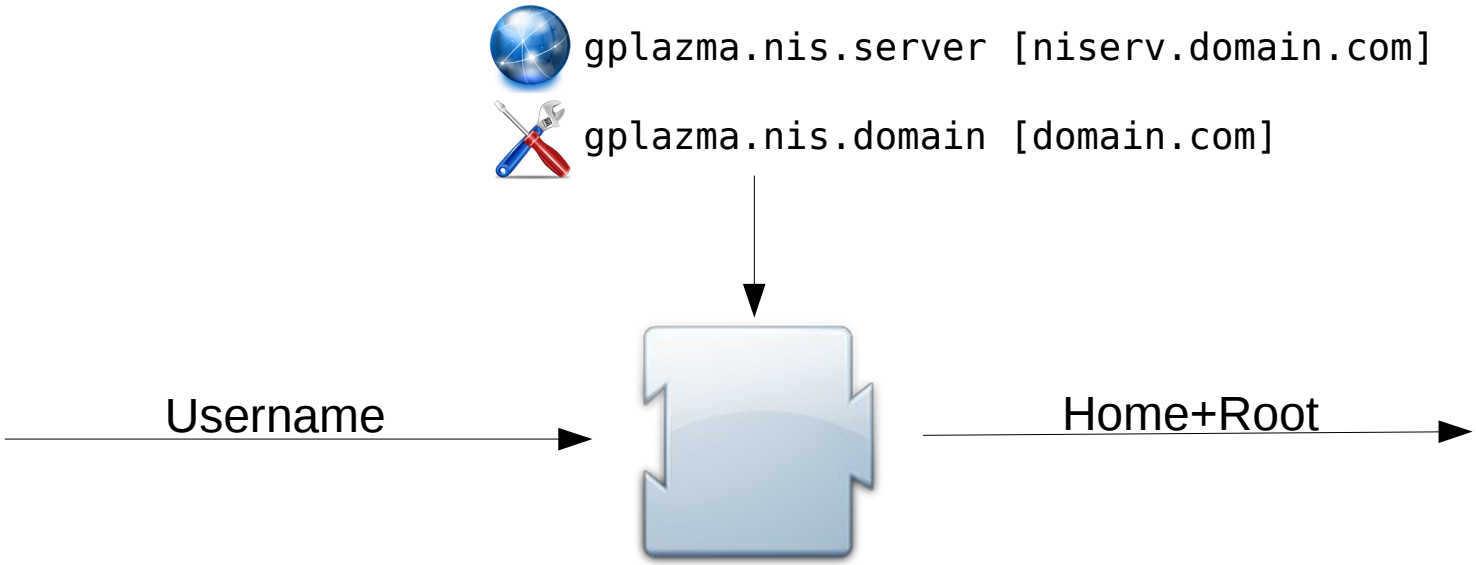
# The five session plugins



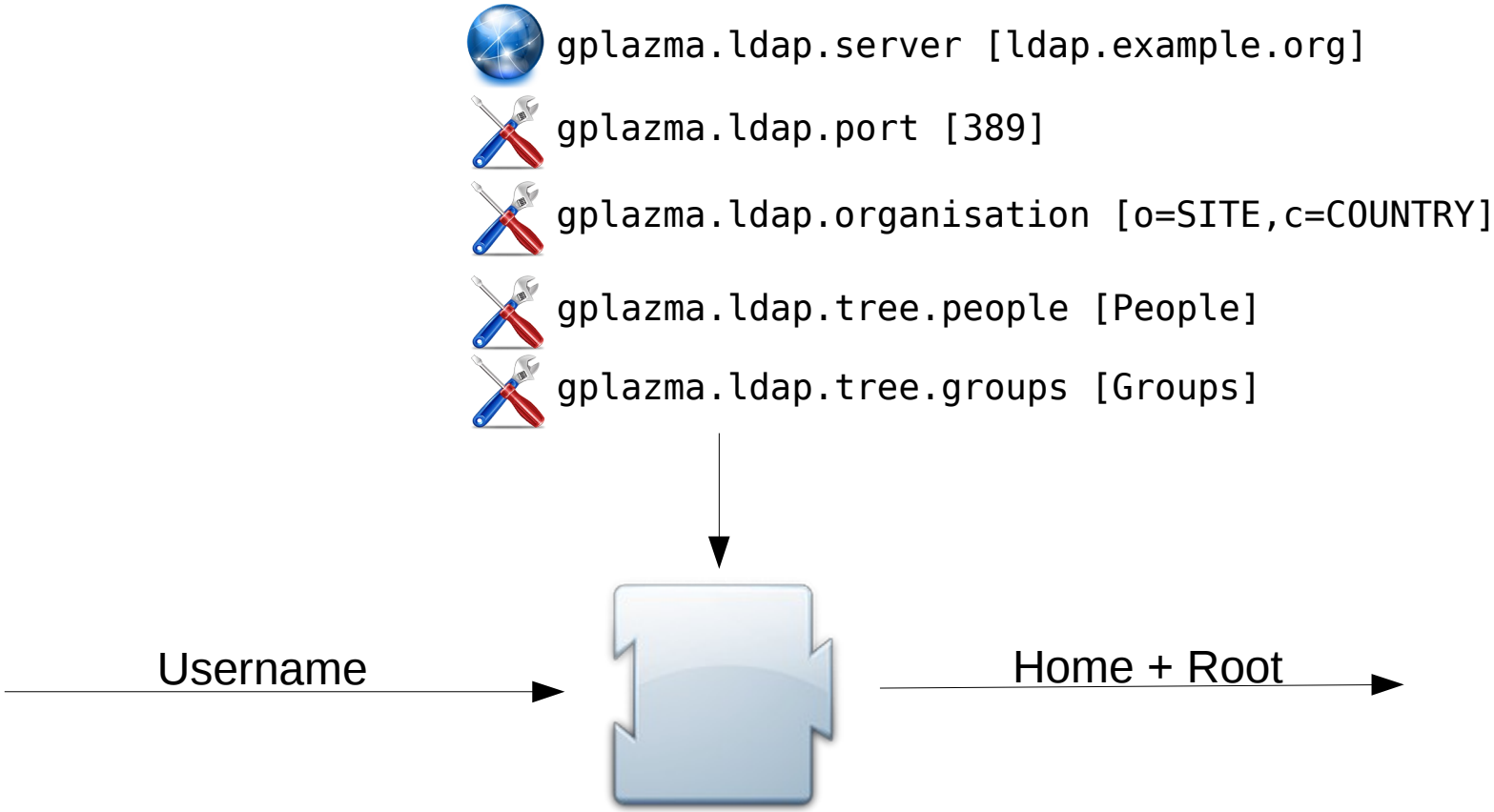
# kpwd: use local file



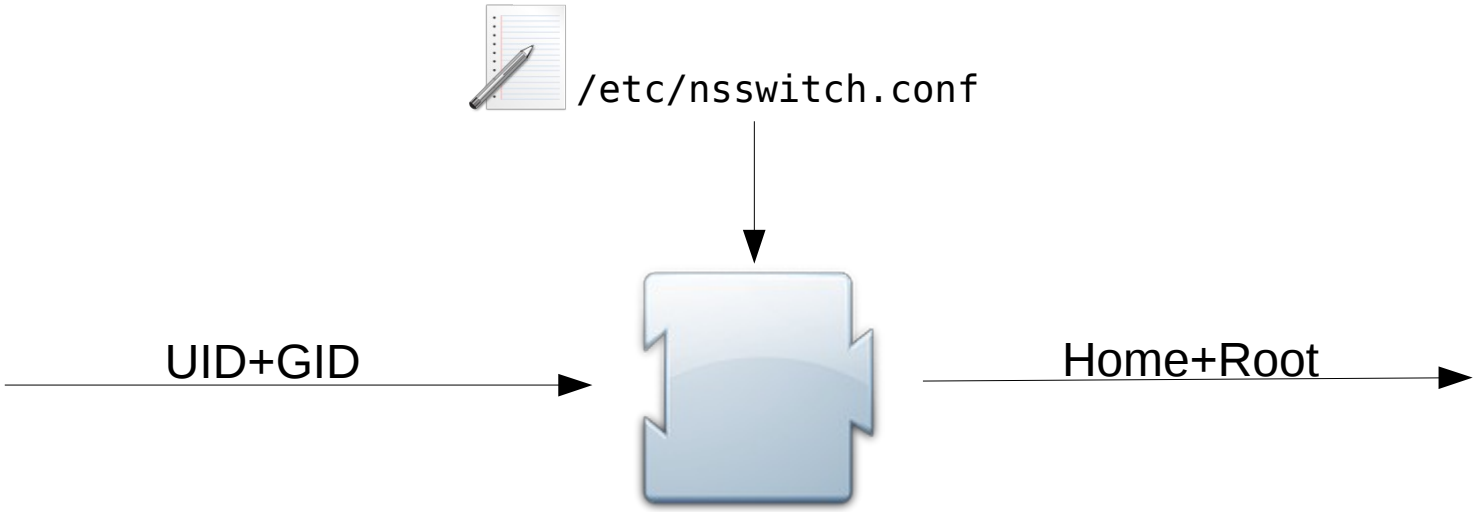
# nis: lookup in site infrastructure



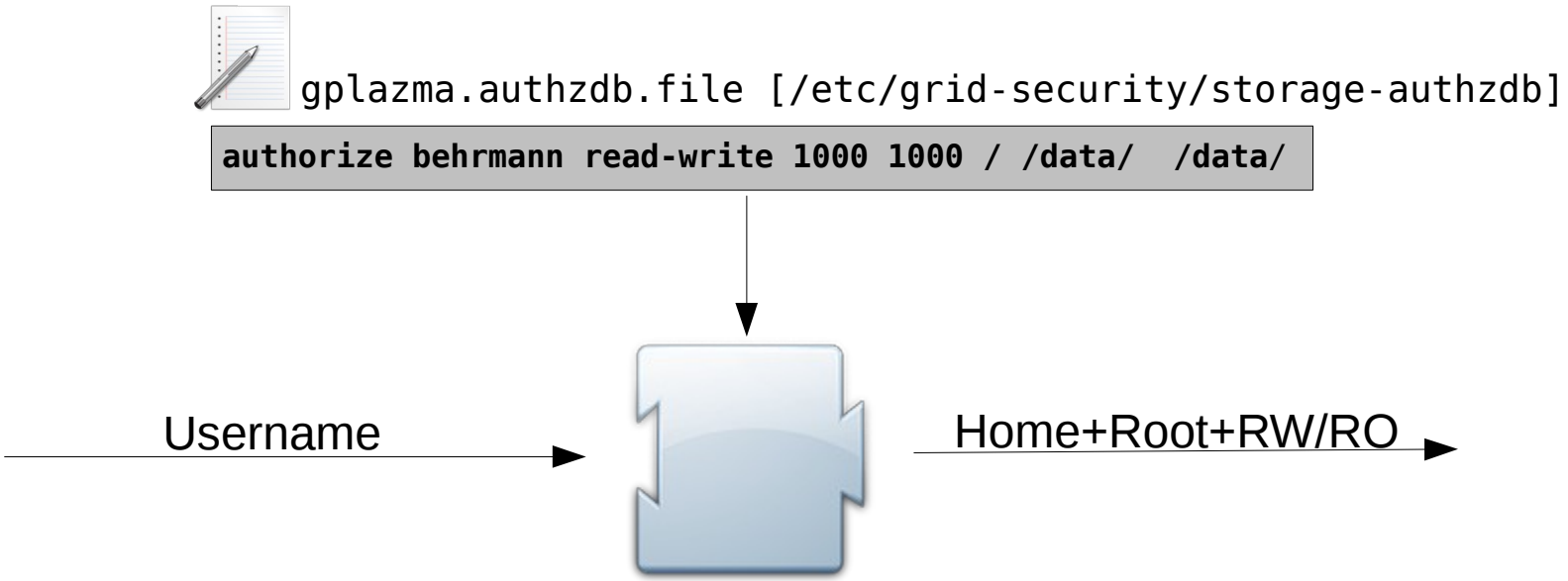
# ldap: lookup in site infrastructure



# nsswitch: use local OS mapping

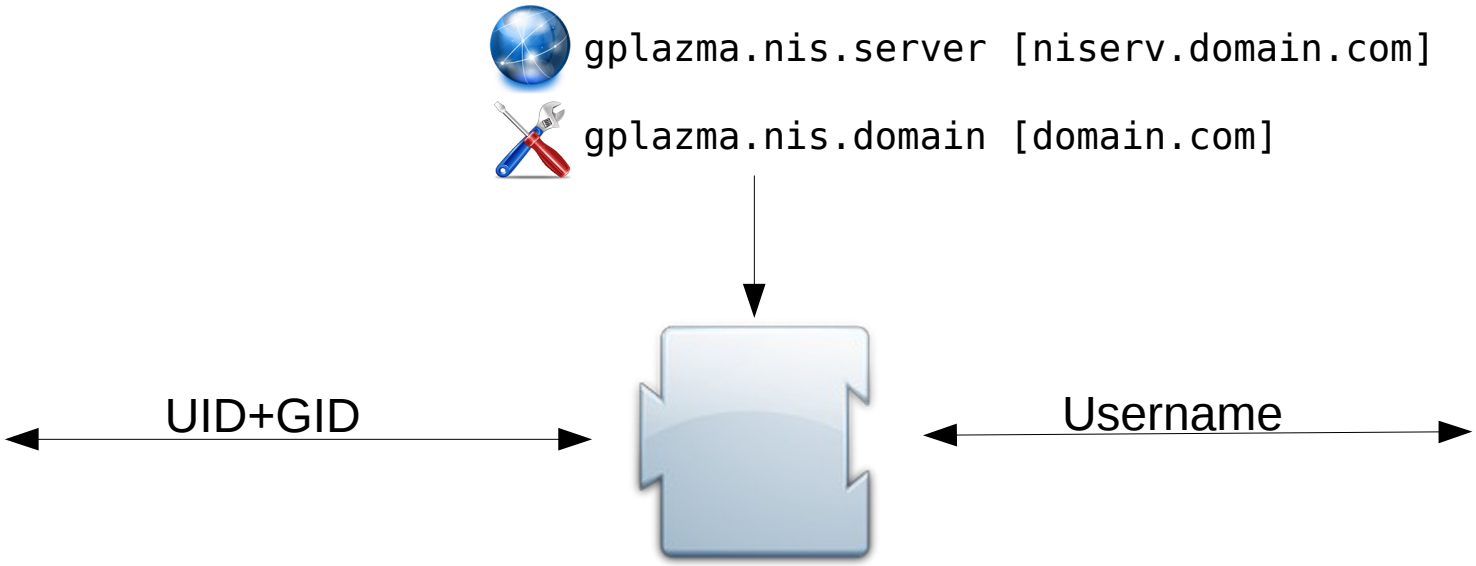


# authzdb: use local file

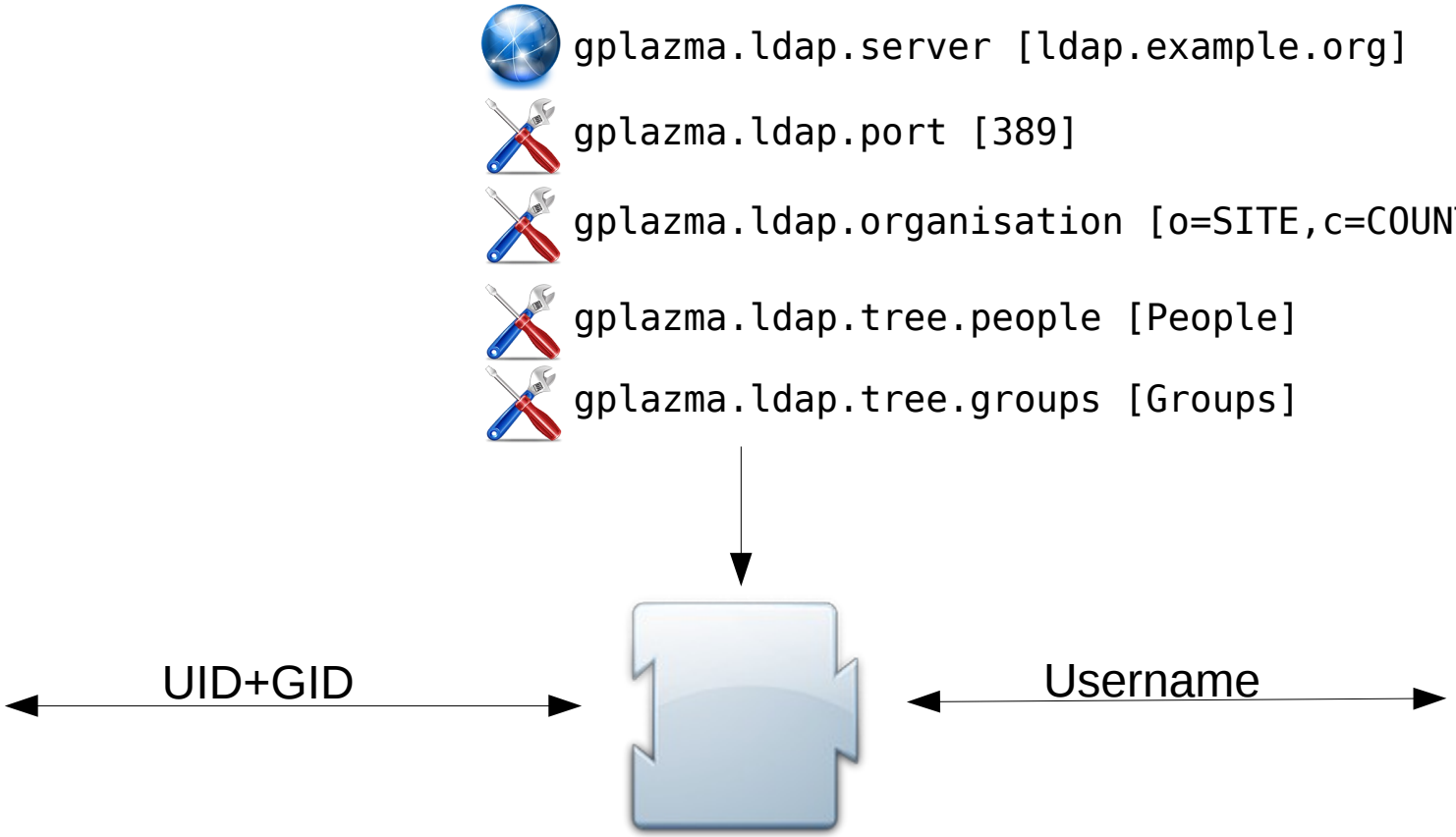


# The three identity plugins

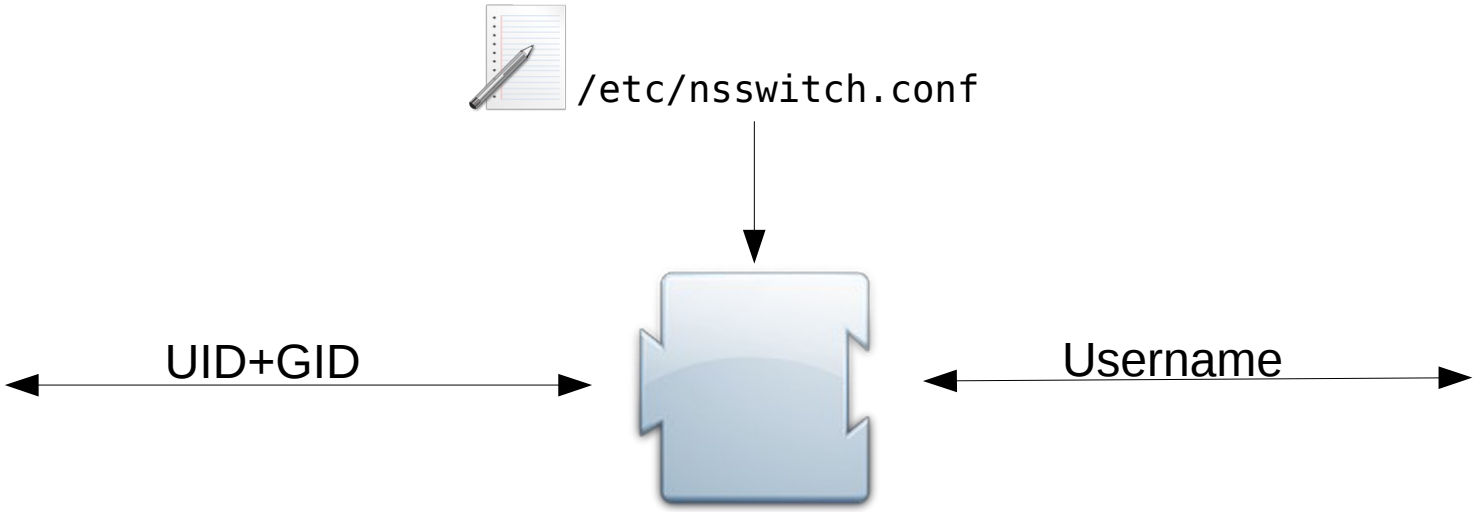
# nis: use site infrastructure



# ldap: lookup in site infrastructure



# nsswitch: use local OS mapping



# One final thing..

# Life's not always that easy

- When user fails to login in they are **not** told why

  (for good security reasons)

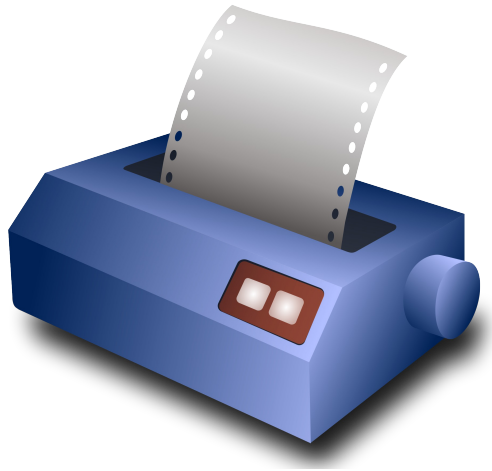
  … but the admin may need to investigate further

- Simply logging the failing plugin might not be enough:

  e.g., cooperating plugins and logins
  that fail in the validation step

- Need an overview of the
  login process

# Introducing gPlazma result printer



# Simple example

```
LOGIN FAIL
 |    in: host/zitpcx6184.desy.de@DESY.DE
 |
 +--AUTH OK
 |   |
 |   +--x509 OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--voms OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--kpwd OPTIONAL:FAIL (no username and password) => OK
 |
 +--MAP FAIL
 |   |  removed: host/zitpcx6184.desy.de@DESY.DE
 |   |
 |   +--krb5 OPTIONAL:OK => OK
 |   |      added: UserNamePrincipal[host/zitpcx6184.desy.de]
 |   |
 |   +--vorolemap OPTIONAL:FAIL (no record) => OK
 |   |
 |   +--authzdb SUFFICIENT:FAIL (no mapping exists for host/zitpcx6184.desy.de) => OK
 |   |
 |   +--kpwd REQUISITE:FAIL (no login name) => FAIL (ends the phase)
 |
 +--(ACCOUNT) skipped
 |
 +--(SESSION) skipped
 |
 +--(VALIDATION) skipped
```

# Simple example

**The information supplied by the door**

```
LOGIN FAIL
 |   in: host/zitpcx6184.desy.de@DESY.DE
 |
 +--AUTH OK
 |   |
 |   +--x509 OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--voms OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--kpwd OPTIONAL:FAIL (no username and password) => OK
 |
 +--MAP FAIL
 |   |  removed: host/zitpcx6184.desy.de@DESY.DE
 |   |
 |   +--krb5 OPTIONAL:OK => OK
 |   |      added: UserNamePrincipal[host/zitpcx6184.desy.de]
 |   |
 |   +--vorolemap OPTIONAL:FAIL (no record) => OK
 |   |
 |   +--authzdb SUFFICIENT:FAIL (no mapping exists for host/zitpcx6184.desy.de) => OK
 |   |
 |   +--kpwd REQUISITE:FAIL (no login name) => FAIL (ends the phase)
 |
 +--(ACCOUNT) skipped
 |
 +--(SESSION) skipped
 |
 +--(VALIDATION) skipped
```

# Simple example

**The different phases and whether they where successful, failed or weren't even run**

```
LOGIN FAIL
 |    in: host/zitpcx6184.desy.de@DESY.DE
 |
 +--AUTH OK
 |  |
 |  +--x509 OPTIONAL:FAIL (no X509 certificate chain) => OK
 |  |
 |  +--voms OPTIONAL:FAIL (no X509 certificate chain) => OK
 |  |
 |  +--kpwd OPTIONAL:FAIL (no username and password) => OK
 |
 +--MAP FAIL
 |  |  removed: host/zitpcx6184.desy.de@DESY.DE
 |  |
 |  +--krb5 OPTIONAL:OK => OK
 |  |     added: UserNamePrincipal[host/zitpcx6184.desy.de]
 |  |
 |  +--vorolemap OPTIONAL:FAIL (no record) => OK
 |  |
 |  +--authzdb SUFFICIENT:FAIL (no mapping exists for host/zitpcx6184.desy.de) => OK
 |  |
 |  +--kpwd REQUISITE:FAIL (no login name) => FAIL (ends the phase)
 |
 +--(ACCOUNT) skipped
 |
 +--(SESSION) skipped
 |
 +--(VALIDATION) skipped
```

# Simple example

**Summary of principals added or removed by a phase**

```
LOGIN FAIL
 |    in: host/zitpcx6184.desy.de@DESY.DE
 |
 +--AUTH OK
 |   |
 |   +--x509 OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--voms OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--kpwd OPTIONAL:FAIL (no username and password) => OK
 |
 +--MAP FAIL
 |   |  removed: host/zitpcx6184.desy.de@DESY.DE
 |   |
 |   +--krb5 OPTIONAL:OK => OK
 |   |      added: UserNamePrincipal[host/zitpcx6184.desy.de]
 |   |
 |   +--vorolemap OPTIONAL:FAIL (no record) => OK
 |   |
 |   +--authzdb SUFFICIENT:FAIL (no mapping exists for host/zitpcx6184.desy.de) => OK
 |   |
 |   +--kpwd REQUISITE:FAIL (no login name) => FAIL (ends the phase)
 |
 +--(ACCOUNT) skipped
 |
 +--(SESSION) skipped
 |
 +--(VALIDATION) skipped
```

# Simple example

Which plugins were run

```
LOGIN FAIL
 |    in: host/zitpcx6184.desy.de@DESY.DE
 |
 +--AUTH OK
 |   |
 |   +--x509 OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--voms OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--kpwd OPTIONAL:FAIL (no username and password) => OK
 |
 +--MAP FAIL
 |    |  removed: host/zitpcx6184.desy.de@DESY.DE
 |   |
 |   +--krb5 OPTIONAL:OK => OK
 |   ||      added: UserNamePrincipal[host/zitpcx6184.desy.de]
 |    |
 |   +--vorolemap OPTIONAL:FAIL (no record) => OK
 |    |
 |   +--authzdb SUFFICIENT:FAIL (no mapping exists for host/zitpcx6184.desy.de) => OK
 |    |
 |   +--kpwd REQUISITE:FAIL (no login name) => FAIL (ends the phase)
 |
 +--(ACCOUNT) skipped
 |
 +--(SESSION) skipped
 |
 +--(VALIDATION) skipped
```

# Simple example

**Principals added by a plugin**

```
LOGIN FAIL
 |    in: host/zitpcx6184.desy.de@DESY.DE
 |
 +--AUTH OK
 |    |
 |    +--x509 OPTIONAL:FAIL (no X509 certificate chain) => OK
 |    |
 |    +--voms OPTIONAL:FAIL (no X509 certificate chain) => OK
 |    |
 |    +--kpwd OPTIONAL:FAIL (no username and password) => OK
 |
 +--MAP FAIL
 |    |  removed: host/zitpcx6184.desy.de@DESY.DE
 |    |
 |    +--krb5 OPTIONAL:OK => OK
 |    |       added: UserNamePrincipal[host/zitpcx6184.desy.de]
 |    |
 |    +--vorolemap OPTIONAL:FAIL (no record) => OK
 |    |
 |    +--authzdb SUFFICIENT:FAIL (no mapping exists for host/zitpcx6184.desy.de) => OK
 |    |
 |    +--kpwd REQUISITE:FAIL (no login name) => FAIL (ends the phase)
 |
 +--(ACCOUNT) skipped
 |
 +--(SESSION) skipped
 |
 +--(VALIDATION) skipped
```

# Simple example

```
LOGIN FAIL
 |    in: host/zitpcx6184.desy.de@DESY.DE
 |
 +--AUTH OK
 |   |
 |   +--x509 OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--voms OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--kpwd OPTIONAL:FAIL (no username and password) => OK
 |
 +--MAP FAIL
 |   |  removed: host/zitpcx6184.desy.de@DESY.DE
 |   |
 |   +--krb5 OPTIONAL:OK => OK
 |   |      added: UserNamePrincipal[host/zitpcx6184.desy.de]
 |   |
 |   +--vorolemap OPTIONAL:FAIL (no record) => OK
 |   |
 |   +--authzdb SUFFICIENT:FAIL (no mapping exists for host/zitpcx6184.desy.de) => OK
 |   |
 |   +--kpwd REQUISITE:FAIL (no login name) => FAIL (ends the phase)
 |
 +--(ACCOUNT) skipped
 |
 +--(SESSION) skipped
 |
 +--(VALIDATION) skipped
```

**The wiring and whether the plugin was successful.**

# Simple example

```
LOGIN FAIL
 |    in: host/zitpcx6184.desy.de@DESY.DE
 |
 +--AUTH OK
 |   |
 |   +--x509 OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--voms OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--kpwd OPTIONAL:FAIL (no username and password) => OK
 |
 +--MAP FAIL
 |   |  removed: host/zitpcx6184.desy.de@DESY.DE
 |   |
 |   +--krb5 OPTIONAL:OK => OK
 |   |      added: UserNamePrincipal[host/zitpcx6184.desy.de]
 |   |
 |   +--vorolemap OPTIONAL:FAIL (no record) => OK
 |   |
 |   +--authzdb SUFFICIENT:FAIL (no mapping exists for host/zitpcx6184.desy.de) => OK
 |   |
 |   +--kpwd REQUISITE:FAIL (no login name) => FAIL (ends the phase)
 |
 +--(ACCOUNT) skipped
 |
 +--(SESSION) skipped
 |
 +--(VALIDATION) skipped
```

The error message if plugin fails

No message if plugin succeeds

# Simple example

**Wiring and plugin result combined**

```
LOGIN FAIL
 |    in: host/zitpcx6184.desy.de@DESY.DE
 |
 +--AUTH OK
 |   |
 |   +--x509 OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--voms OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--kpwd OPTIONAL:FAIL (no username and password) => OK
 |
 +--MAP FAIL
 |   |  removed: host/zitpcx6184.desy.de@DESY.DE
 |   |
 |   +--krb5 OPTIONAL:OK => OK
 |   |      added: UserNamePrincipal[host/zitpcx6184.desy.de]
 |   |
 |   +--vorolemap OPTIONAL:FAIL (no record) => OK
 |   |
 |   +--authzdb SUFFICIENT:FAIL (no mapping exists for host/zitpcx6184.desy.de) => OK
 |   |
 |   +--kpwd REQUISITE:FAIL (no login name) => FAIL (ends the phase)
 |
 +--(ACCOUNT) skipped
 |
 +--(SESSION) skipped
 |
 +--(VALIDATION) skipped
```

# Simple example

Easy to spot why login failed

```
LOGIN FAIL
 |    in: host/zitpcx6184.desy.de@DESY.DE
 |
 +--AUTH OK
 |   |
 |   +--x509 OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--voms OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--kpwd OPTIONAL:FAIL (no username and password) => OK
 |
 +--MAP FAIL
 |   | removed: host/zitpcx6184.desy.de@DESY.DE
 |   |
 |   +--krb5 OPTIONAL:OK => OK
 |   |      added: UserNamePrincipal[host/zitpcx6184.desy.de]
 |   |
 |   +--vorolemap OPTIONAL:FAIL (no record) => OK
 |   |
 |   +--authzdb SUFFICIENT:FAIL (no mapping exists for host/zitpcx6184.desy.de) => OK
 |   |
 |   +--kpwd REQUISITE:FAIL (no login name) => FAIL(ends the phase)
 |
 +--(ACCOUNT) skipped
 |
 +--(SESSION) skipped
 |
 +--(VALIDATION) skipped
```

# Simple example

```
LOGIN FAIL
 |    in: host/zitpcx6184.desy.de@DESY.DE
 |
 +--AUTH OK
 |   |
 |   +--x509 OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--voms OPTIONAL:FAIL (no X509 certificate chain) => OK
 |   |
 |   +--kpwd OPTIONAL:FAIL (no username and password) => OK
 |
 +--MAP FAIL
 |   |  removed: host/zitpcx6184.desy.de@DESY.DE
 |   |
 |   +--krb5 OPTIONAL:OK => OK
 |   |      added: UserNamePrincipal[host/zitpcx6184.desy.de]
 |   |
 |   +--vorolemap OPTIONAL:FAIL (no record) => OK
 |   |
 |   +--authzdb SUFFICIENT:FAIL (no mapping exists for host/zitpcx6184.desy.de) => OK
 |   |
 |   +--kpwd REQUISITE:FAIL (no login name) => FAIL (ends the phase)
 |
 +--(ACCOUNT) skipped
 |
 +--(SESSION) skipped
 |
 +--(VALIDATION) skipped
```

# Example with X.509

**Door supplies X509 certificate chain with single certificate**

```
LOGIN FAIL
 |    in: S:/131.169.252.82
 |        X509 Certificate chain:
 |          |
 |          +--CN=Alexander Paul Millar,OU=DESY,O=GermanGrid,C=DE [16724]
 |               |
 |               +--Issuer: CN=GridKa-CA,O=GermanGrid,C=DE
 |               +--Validity: OK for 366 days, 20 hours, 30 minutes and 13.0 seconds
 |               +--Algorithm: SHA-1 with RSA
 |               +--Subject alternative names: paul.millar@desy.de
 |               +--Key usage: digital signature, key encipherment, data encipherment
 |
 |   out: GidPrincipal[1000,primary]
 |        UidPrincipal[1000]
 |        UserNamePrincipal[paul]
 |        /C=DE/O=GermanGrid/OU=DESY/CN=Alexander Paul Millar
 |        KpwdPrincipal[paul]
 |
 +--AUTH OK
 |    |    added: /C=DE/O=GermanGrid/OU=DESY/CN=Alexander Paul Millar
 |    |
 |    +--x509 OPTIONAL:OK => OK
 |    |      added: /C=DE/O=GermanGrid/OU=DESY/CN=Alexander Paul Millar
 |    |
 |    +--voms OPTIONAL:FAIL (no FQANs) => OK
 |    |
 |    +--kpwd OPTIONAL:FAIL (no username and password) => OK
 |
```

**x509 plugin extracts DN**

# Example with VOMS proxy certificate

Embedded VOMS certificate

```
LOGIN FAIL
 |    in: S:/131.169.137.140
 |        /C=DE/ST=Hamburg/O=dCache.ORG/CN=Kermit the frog
 |        X509 Certificate chain:
 |          |
 |          +--CN=proxy,CN=Kermit the frog,O=dCache.ORG,ST=Hamburg,C=DE [11549466642107437257]
 |          |    |
 |          |    +--Issuer: CN=Kermit the frog,O=dCache.ORG,ST=Hamburg,C=DE
 |          |    +--Validity: OK for 11 hours, 59 minutes and 42.0 seconds
 |          |    +--Algorithm: SHA-1 with RSA
 |          |    +--Attribute certificates:
 |          |    |    |
 |          |    |    +--C=DE,O=GermanGrid,OU=DESY,CN=host/grid-voms.desy.de
 |          |    |         +--Validity: OK for 11 hours, 59 minutes and 42.0 seconds
 |          |    |         +--Algorithm: SHA-1 with RSA
 |          |    |         +--FQANs: /desy, /desy/workshop
 |          |    +--Key usage: digital signature, key encipherment, data encipherment, key agreement
 |          |
 |          +--CN=Kermit the frog,O=dCache.ORG,ST=Hamburg,C=DE [11549466642107437257]
 |          |    |
 |          |    +--Issuer: CN=dCache.ORG CA,O=dCache.ORG,ST=Hamburg,C=DE
 |          |    +--Validity: OK for 357 days, 10 hours, 23 minutes and 52.0 seconds
 |          |    +--Algorithm: SHA-1 with RSA
 |          |    +--Subject alternative names: kermit.the.frog@dcache.org
 |          |    +--Key usage: digital signature, key encipherment, data encipherment, key agreement
 |          |
 |          +--CN=dCache.ORG CA,O=dCache.ORG,ST=Hamburg,C=DE [11549466642107437183] (self-signed)
 |               |
 |               +--Validity: OK for 866 days, 12 hours, 13 minutes and 49.0 seconds
 |               +--Algorithm: SHA-1 with RSA
 |               +--Key usage: key certificate signing, CRL signing
 |
```

# Future directions

To infinity ... and beyond!

# Future direction

- **Autogenerate uid/gid** for auto-registration:

  previously unknown principals are assigned a uid or gid:

  X509 DNs → uid, FQAN → gid.

- **Discover FQANs** if user didn't provide them

  Not always possible for user to provide FQANs

  Likely only be used for reading.

- Add support for **federated identity** systems:

  Initially SAML, but also looking at OpenID

# Summary

dCache has gPlazma:

a ***powerful*** and ***flexible***

identity system