dCache.org

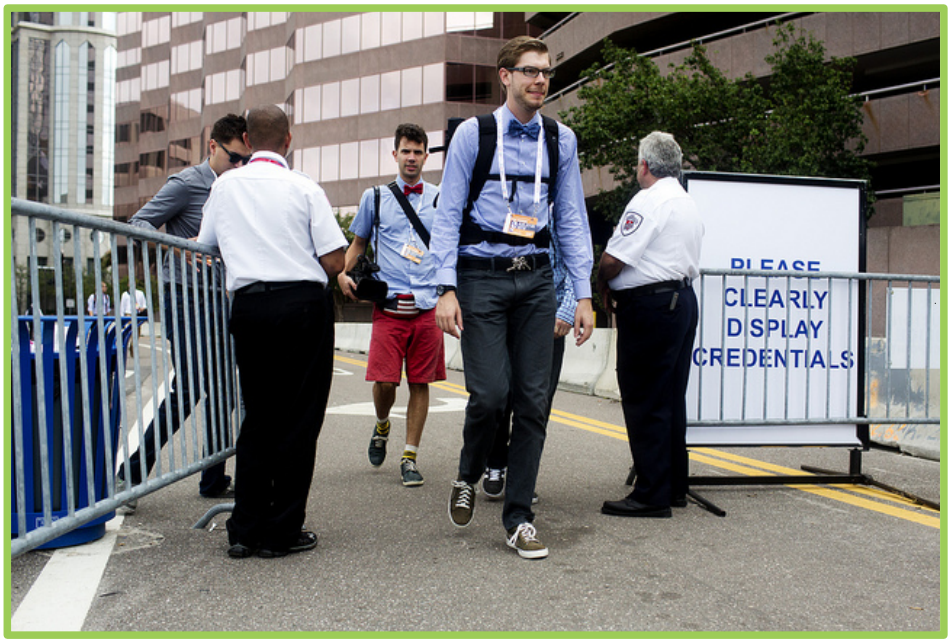
Identity: Theory, Practice and Future directions

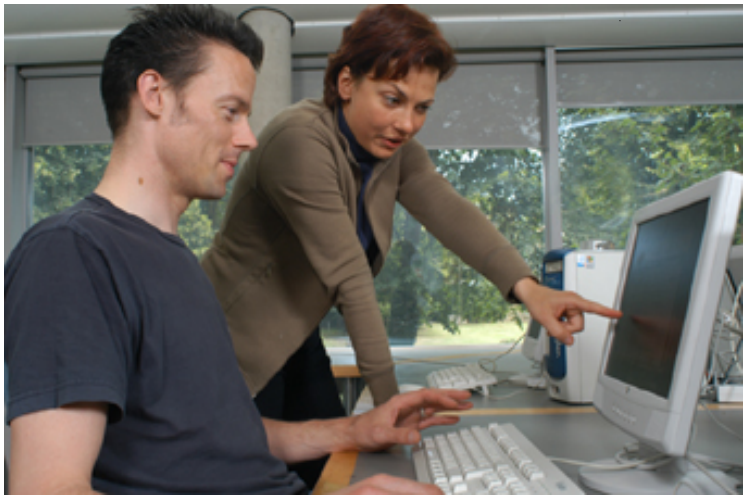
Paul Millar

Academia Sinica, 2014-03-24

Fermilab NDGF Nordic DataGrid Facility DESY HELMHOLTZ ASSOCIATION LSDMA

This talk: in a nutshell





Can read a file?

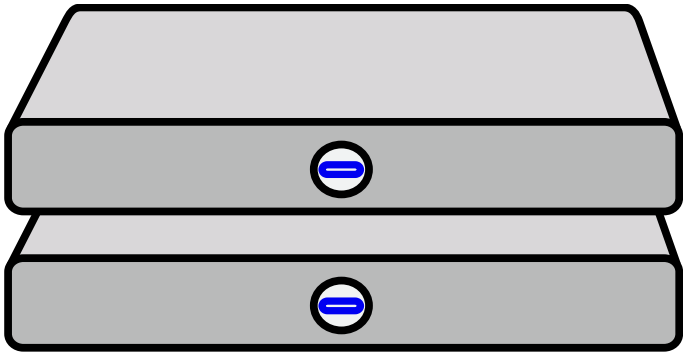
Can write a file?

Can delete a file?

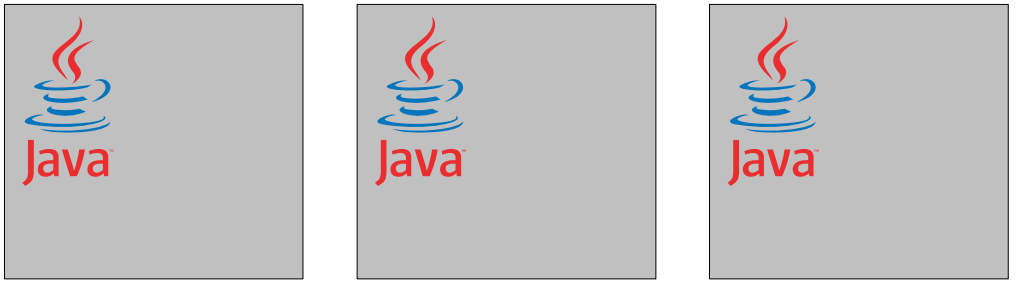
Stage file from tape?

Can create directory?

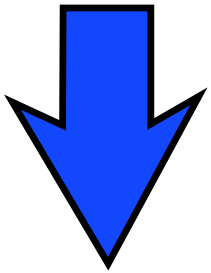
dCache deployment architecture



One or more **nodes** in dCache cluster

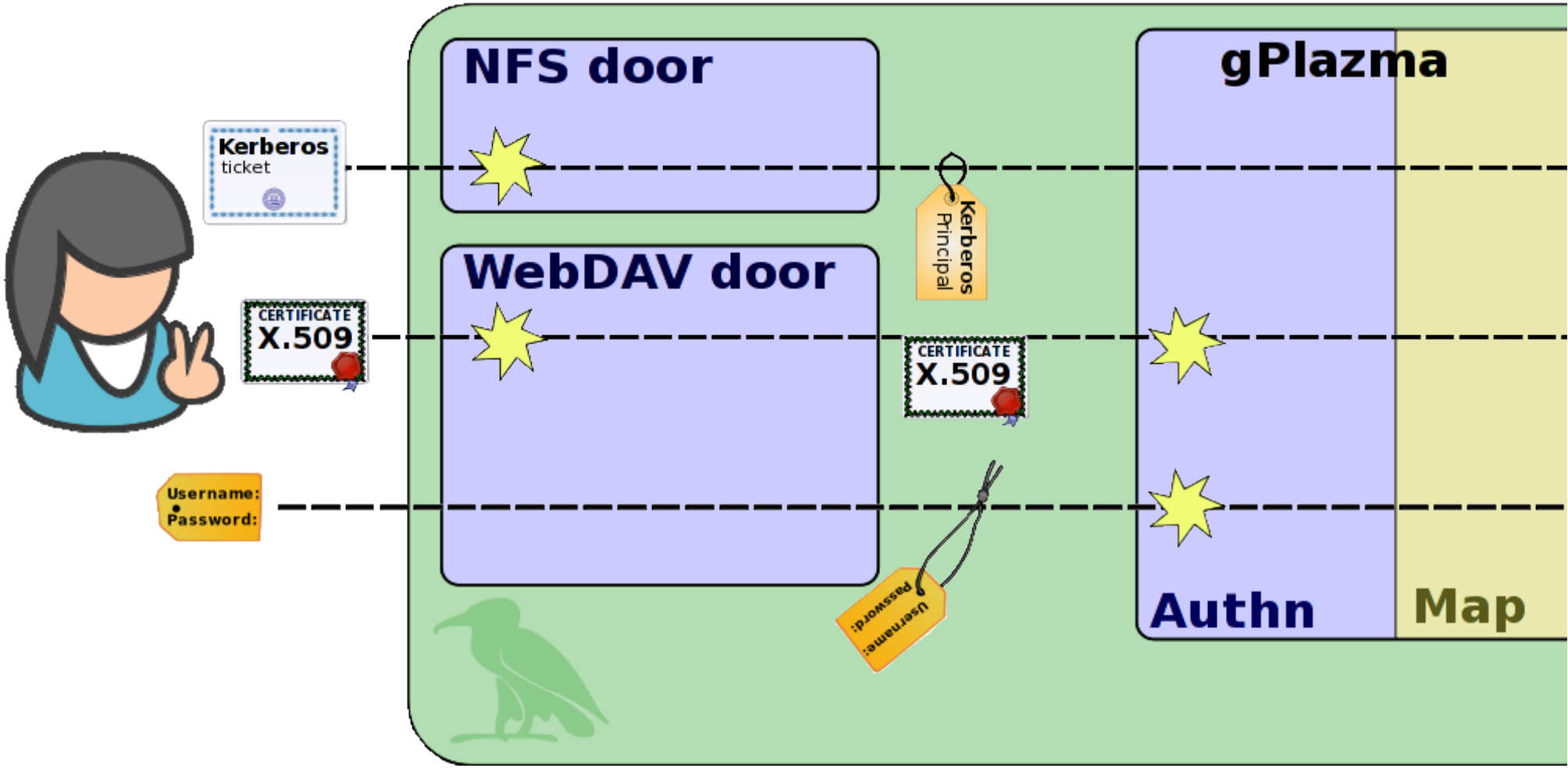


One or more **domains** on server



One or more **services** in domain

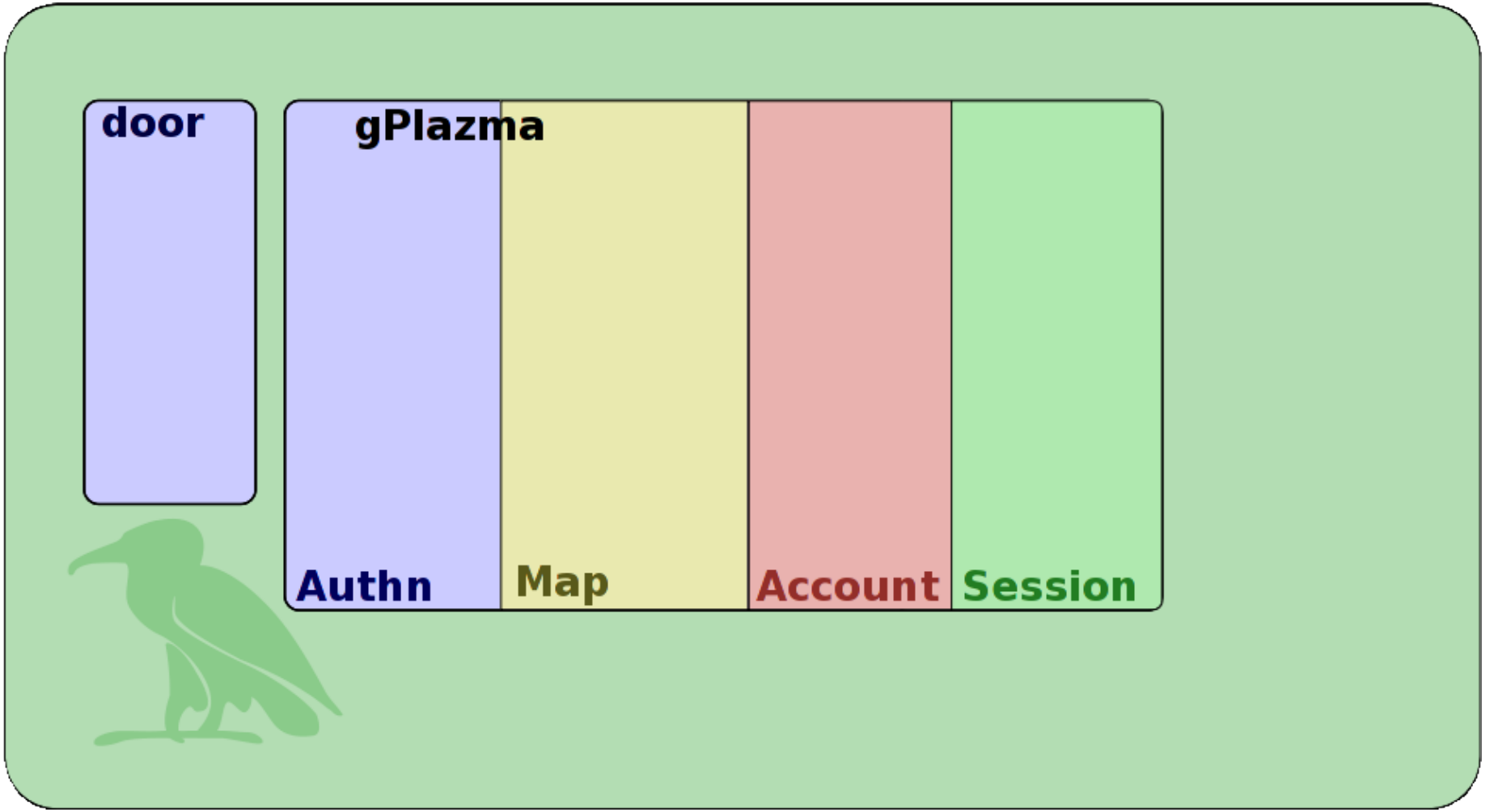
Authentication: door, both or gPlazma



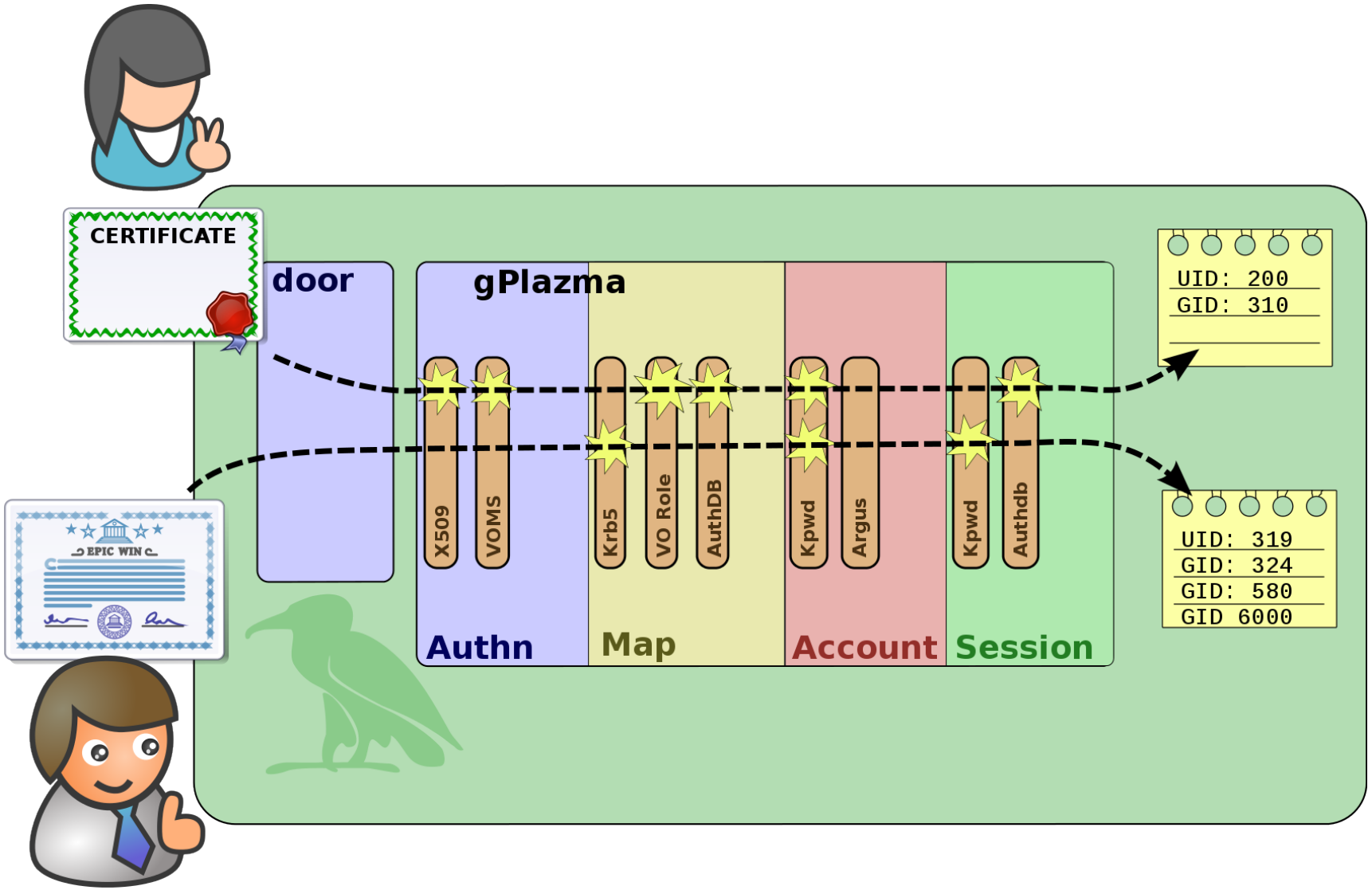
gPlazma as part of dCache

- Users use client software
- Client (software) connects to a door
- Client convinces door, or door+gPlazma, or gPlazma who the user is
- Communication between a door and gPlazma is a "login request" message, which gPlazma sends a "login response"
- gPlazma converts that to a uid, one or more gids, a root directory (might be /), a home directory (might be /) and an optional read-only flag.
  - Exceptions: anonymous access in HTTP/WebDAV (if enabled)

Logging in: the four phases



Logging in: four phases, using plugins



Example plugin #1: X.509 plugin

Example plugin #2: gridmap

Example plugin #3: ldap plugin

Worked example #1: Username + Password

Worked example #2: Kerberos

Worked example #3: X.509 and VOMS

Future directions: token service

Future directions: SAML for web-browser

Future directions: SAML for non-browser

The End

Backup Slides

Need to identify the users

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**

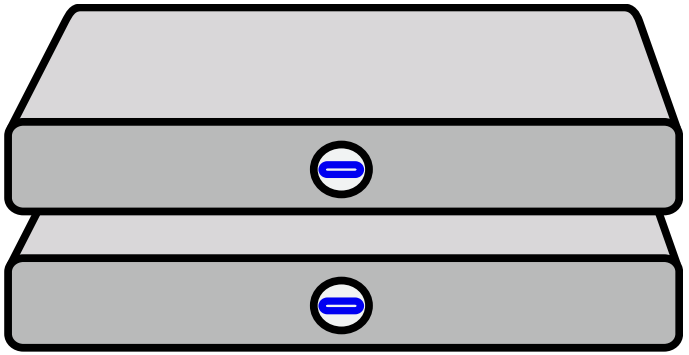
DoB: **1964-08-12**

Place of Birth: **Berlin**

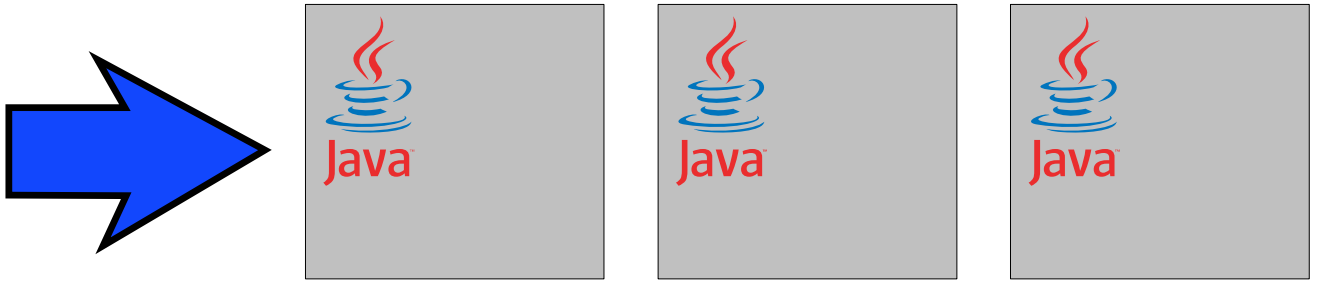
Credential

Principals

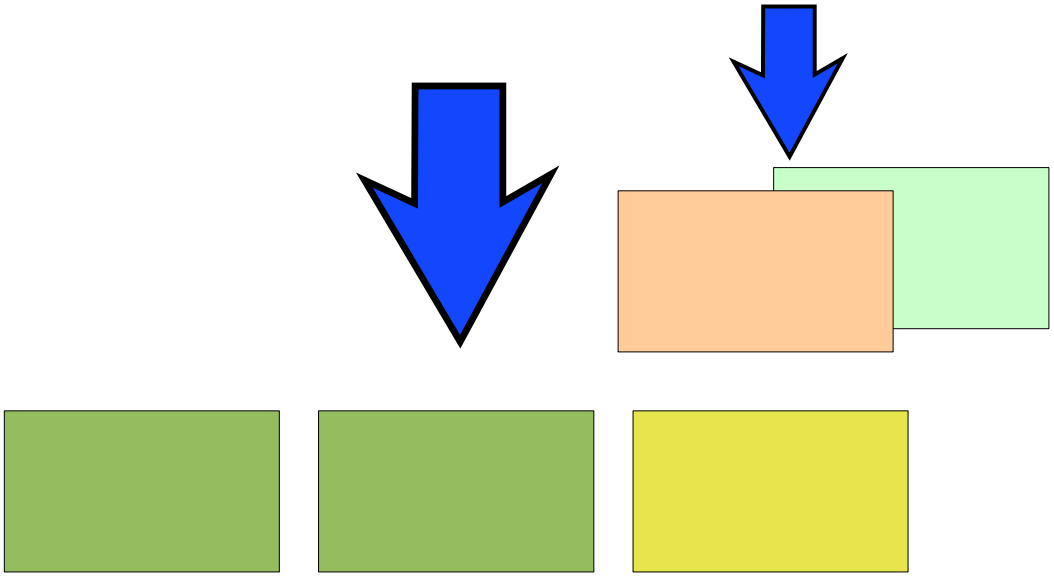
dCache deployment architecture



One or more **nodes** in dCache cluster

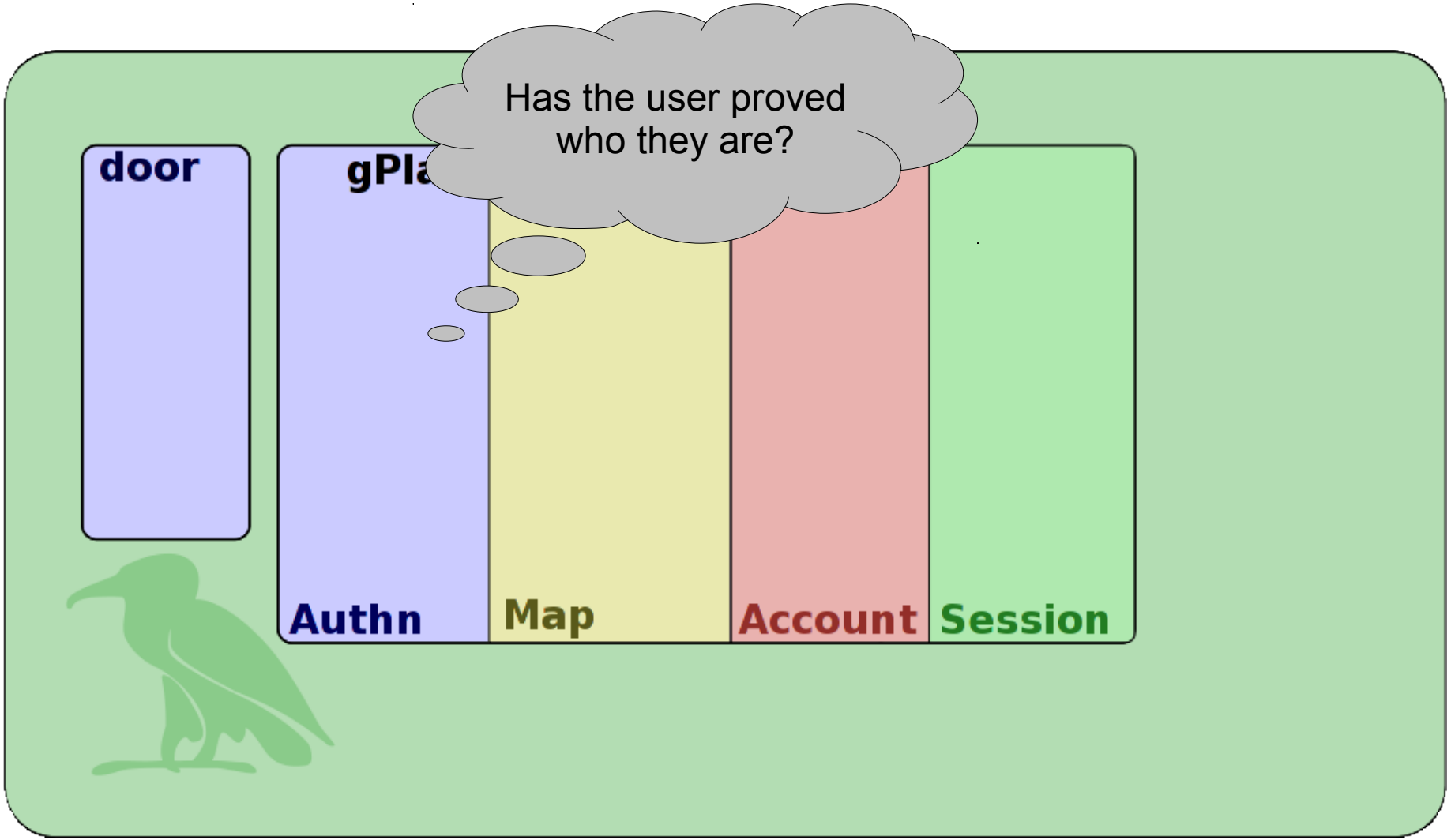


One or more **domains** on server

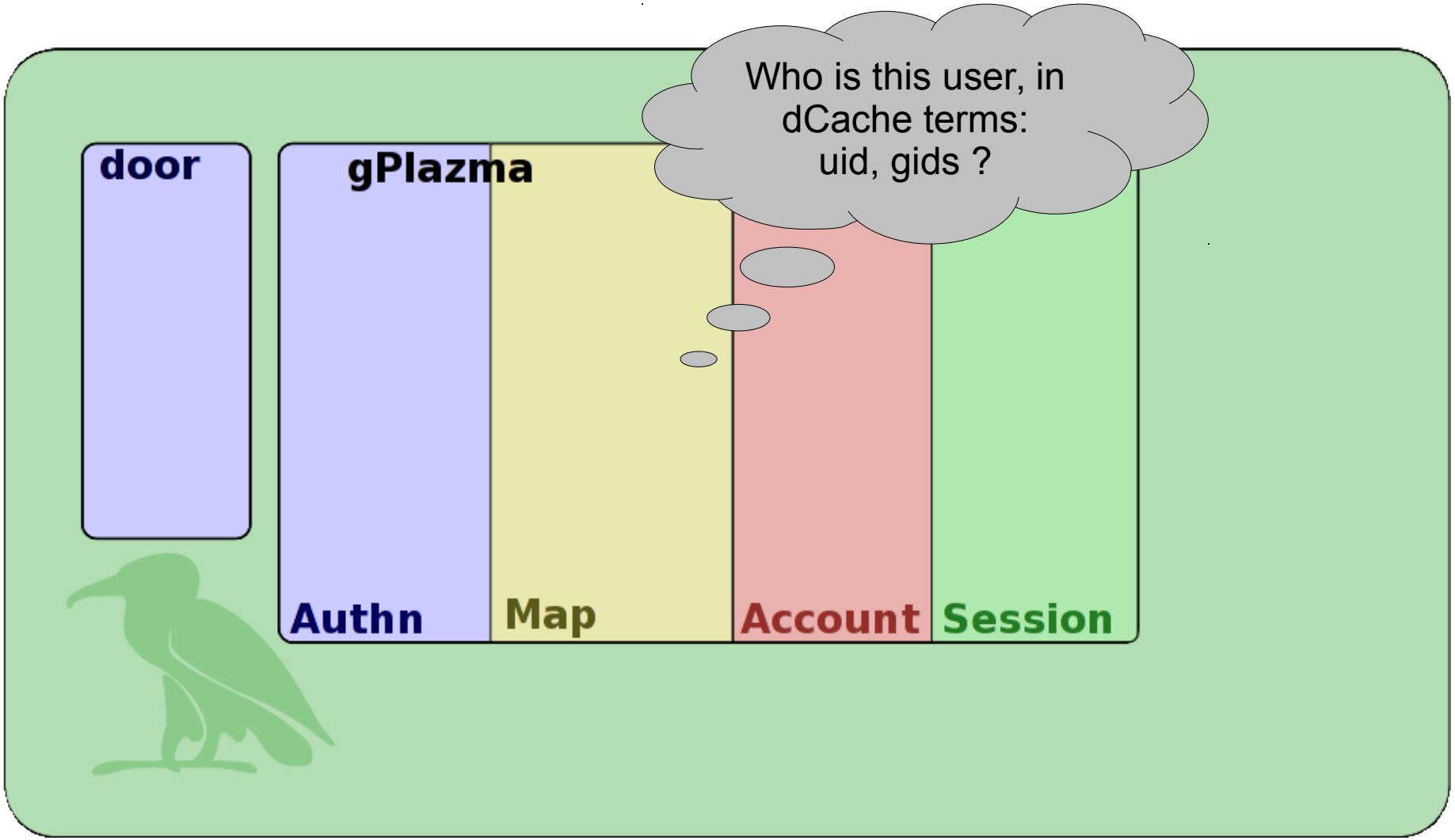


One or more **services** in domain

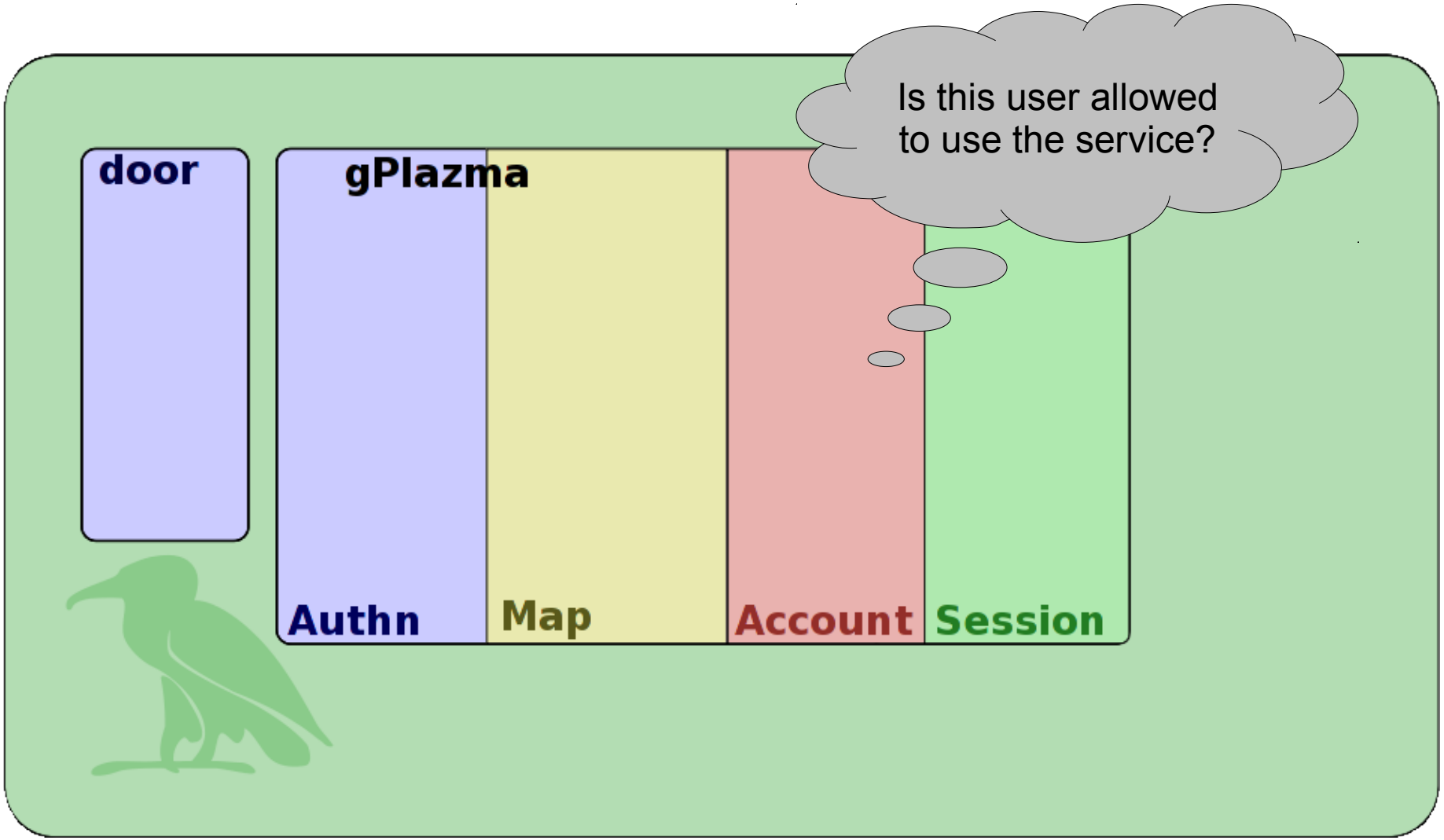
Logging in: the four phases



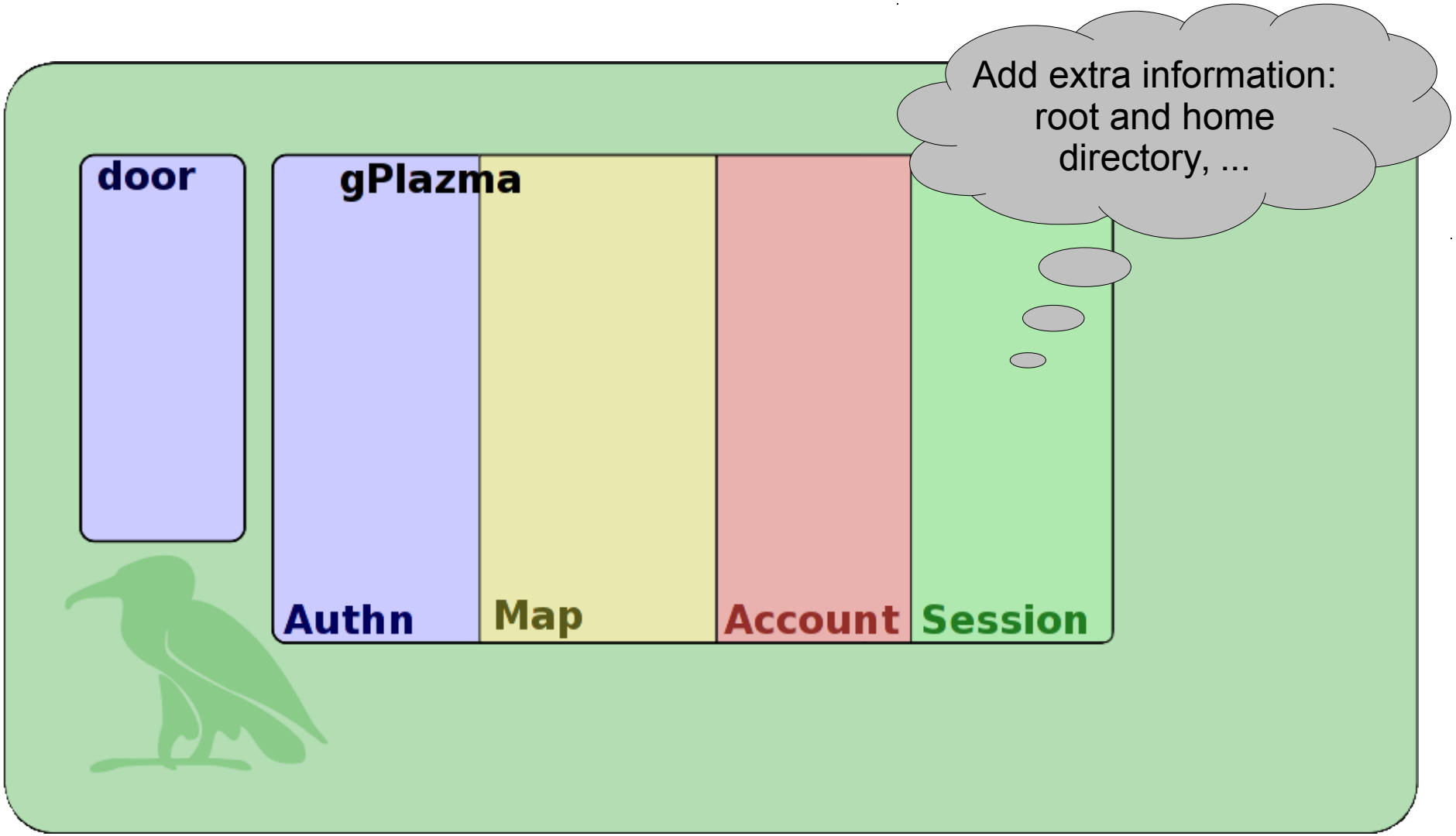
Logging in: the four phases



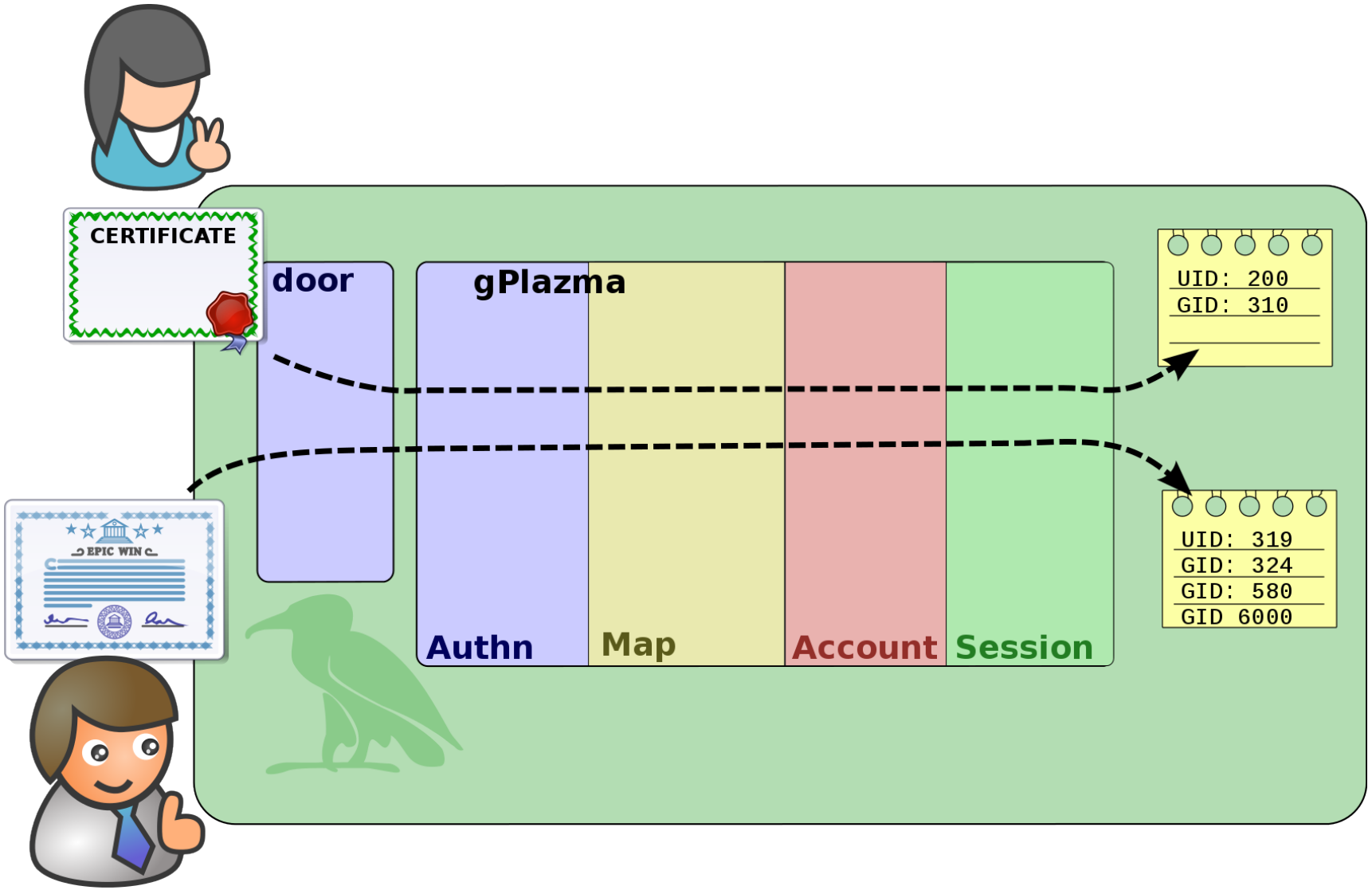
Logging in: the four phases



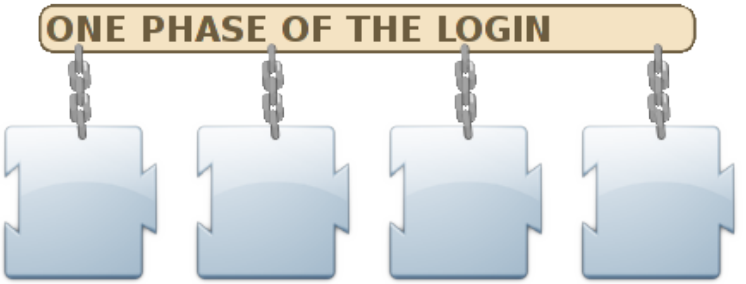
Logging in: the four phases



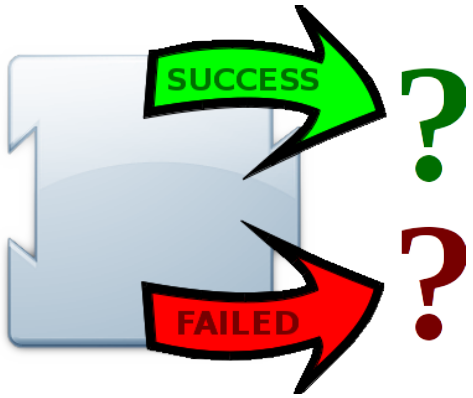
Logging in: four phases



Wiring plugins together

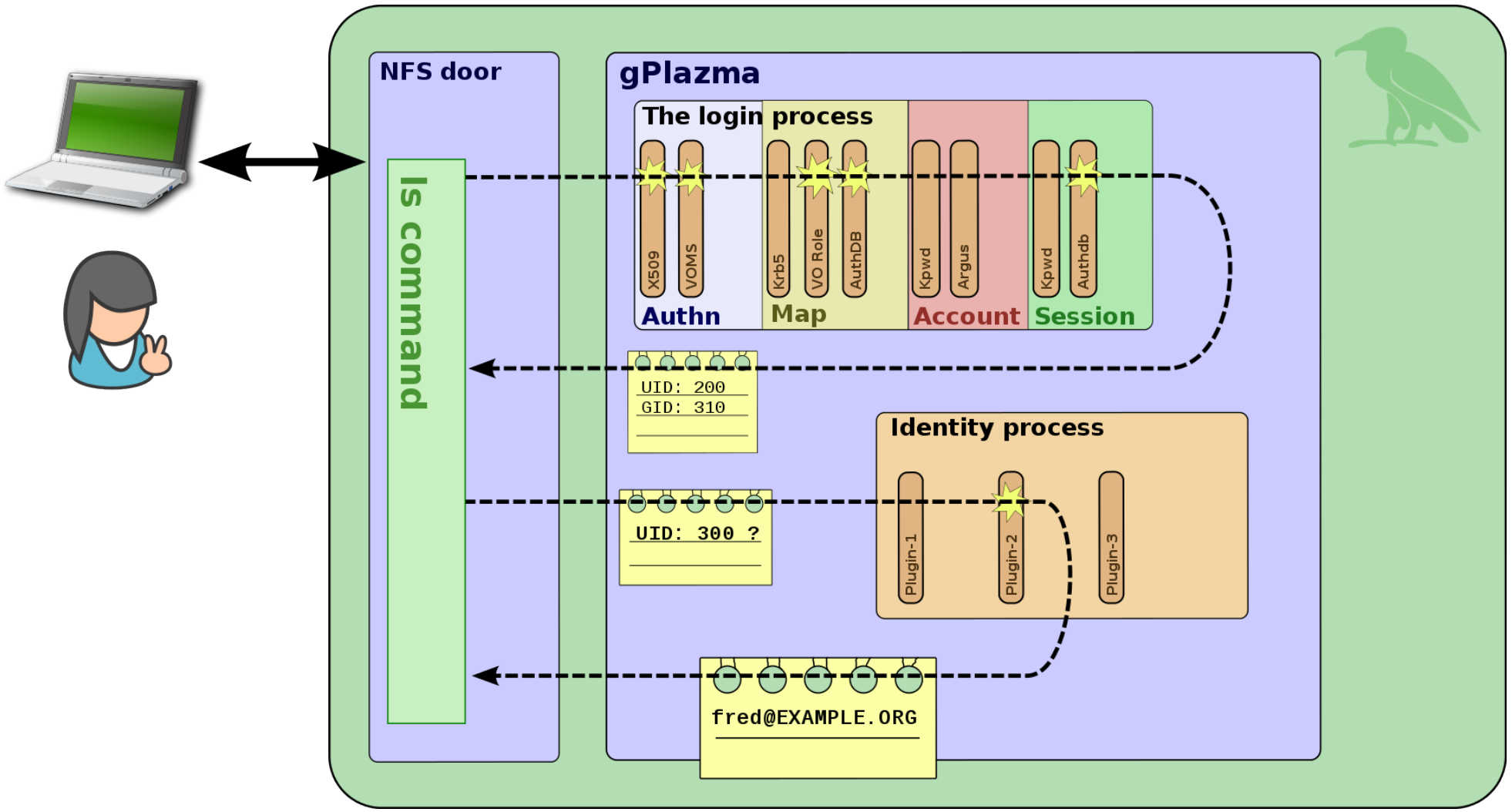


- Each phase is comprised of plugins
  - Result of a phase depends on the result of running these plugins
- When a plugin runs:
  - Running a plugin is either **success** or **failure**.
  - Plugins that fail sometimes is expected
- Four options describe what to do next:



Name	Description
`optional`	Success or failure of the plugin doesn't matter; always move onto next one 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

Something extra: identity mapping



Available plugins: auth phase

- x509
- voms
- kpwd
- htpasswd
- xacml
- jaas

Available plugins: map phase

- gridmap
- vorolemap
- krb5
- mutator
- nsswitch
- authzdb
- nis
- kpwd

Available plugins: account phase

- argus
- kpwd
- banfile

Available plugins: session phase

- authzdb
- nis
- kpwd

Available plugins: identity

- nis
- nsswitch

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

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
```

Example with simple X509

```
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
 |
```

Example with voms proxy

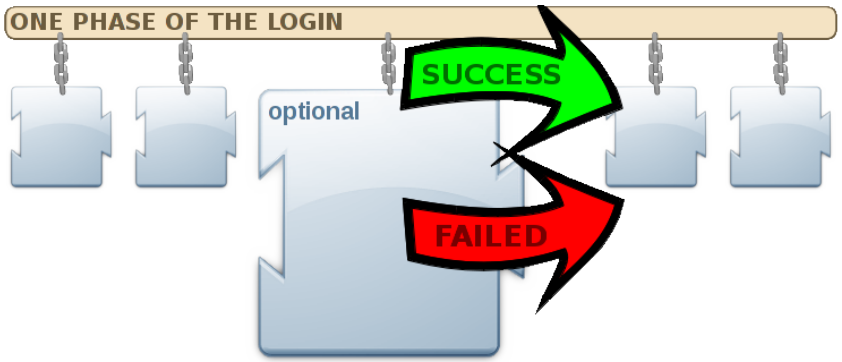
```
LOGIN FAIL
 |    in: S:/131.169.137.140
 |        /C=DE/ST=Hamburg/O=dCache.ORG/CN=Kermit the frog
 |        X509 Certificate chain:
 |          |
 |          +--CN=proxy,CN=Kermit the frog,O=dCache.ORG,ST=Hamburg,C=DE [11549466642107437257]
 |          |    |
 |          |    +--Issuer: CN=Kermit 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
 |
```

Thank you and good night!

For readers at home, there is more information in following slides ...

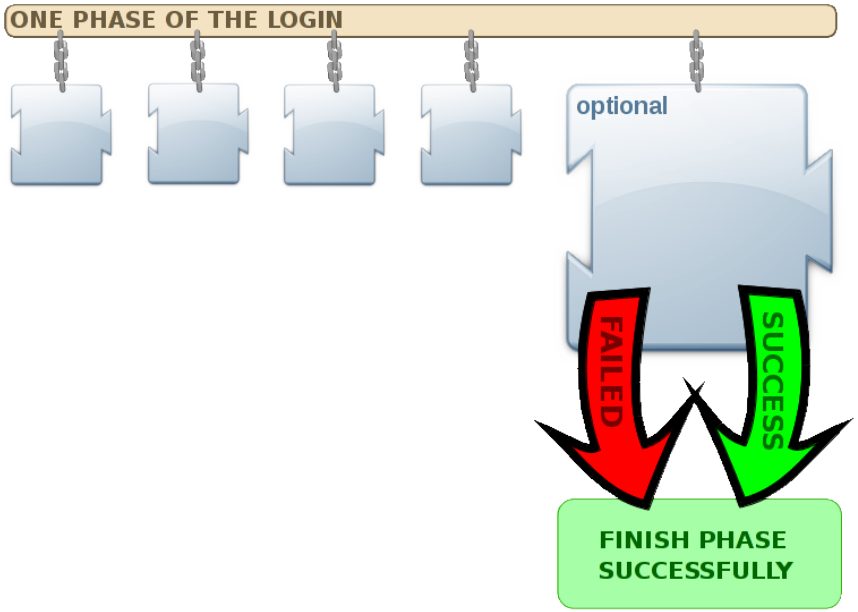
gPlazma wiring in detail

Wiring plugins together: optional



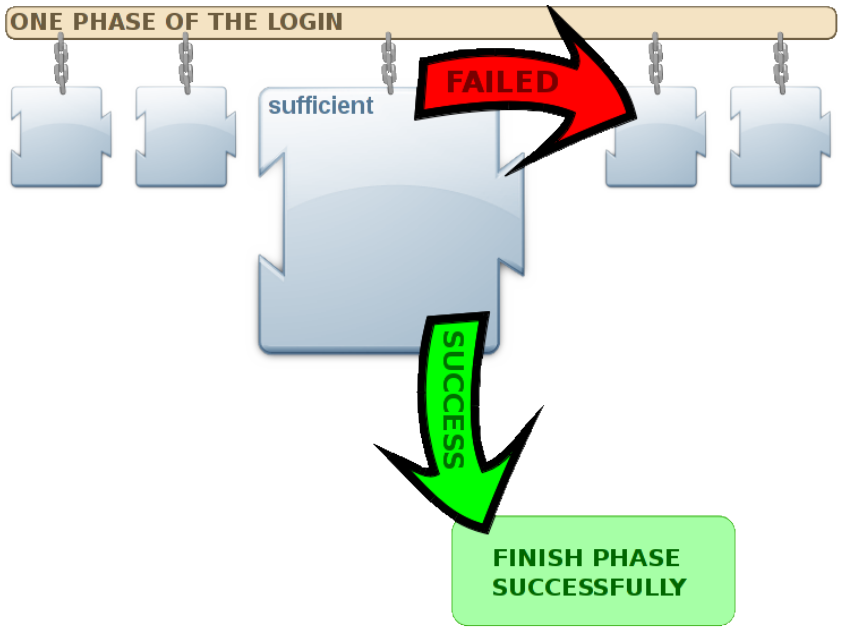
- If a plugin is `optional` then always move on to the next plugin

Wiring plugins together: optional



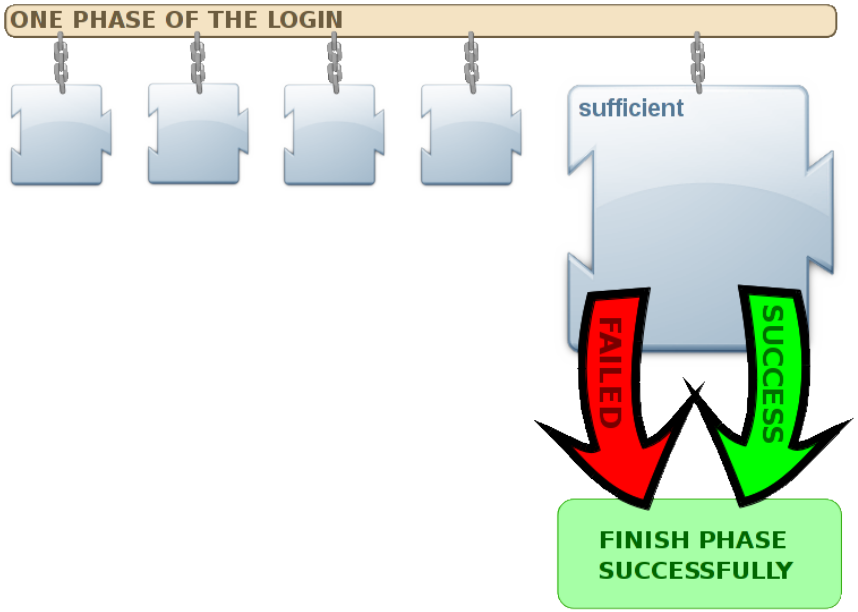
- If the last plugin is `optional`, the phase always succeeds

Wiring plugins together: sufficient



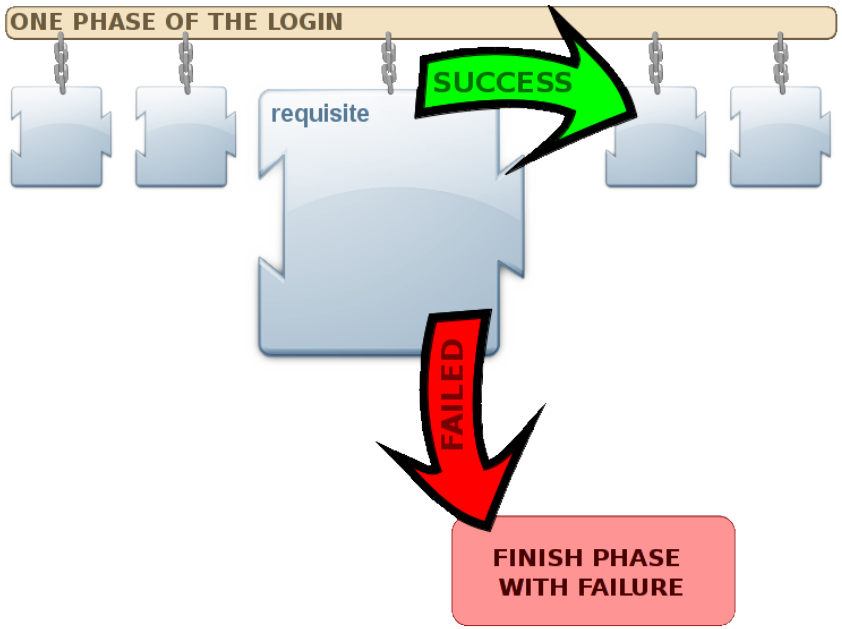
- If a `sufficient` plugin is successful, then end the phase immediately
- If a `sufficient` plugin fails, move on to the next plugin

Wiring plugins together: sufficient



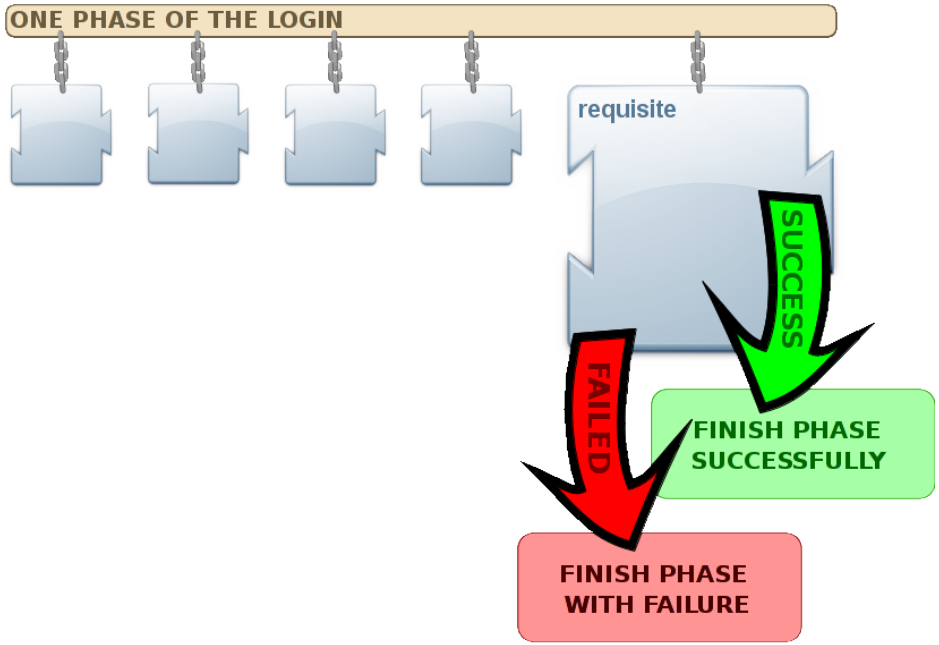
- If the last plugin in the phase is `sufficient` then the phase will always be successful.

Wiring plugins together: requisite



- If a `requisite` plugin fails then fail the phase immediately.
- If a `requisite` plugin is successful, continue to the next plugin

Wiring plugins together: requisite

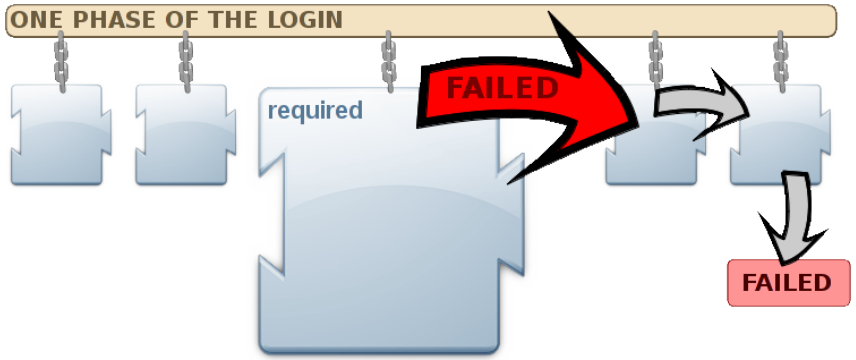


- If the last plugin is `requisite` then the success of the phase depends on the success of the last plugin

Wiring plugins together: required

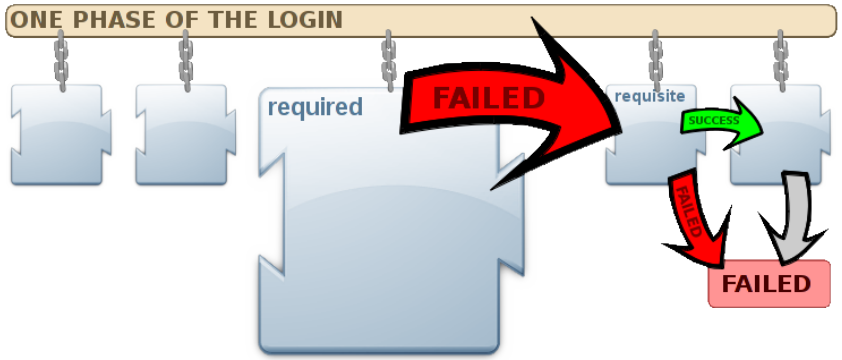
- If a `required` plugin succeeds then move onto the next plugin

Wiring plugins together: required



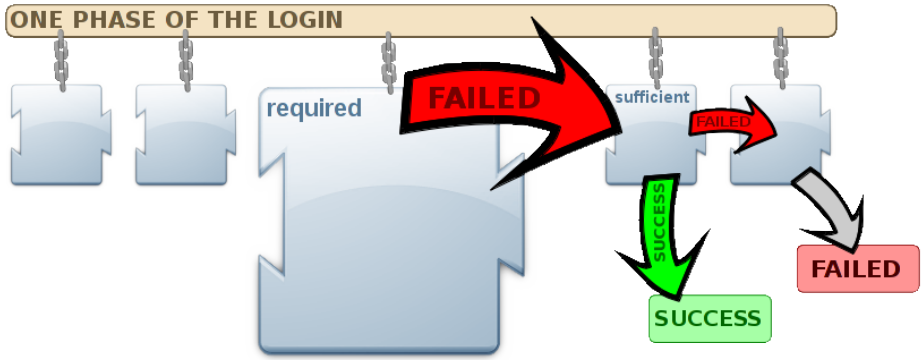
- If a `required` plugin fails then continue onto the next plugin, but ultimately fail with an error from this plugin
- unless ...

Wiring plugins together: required



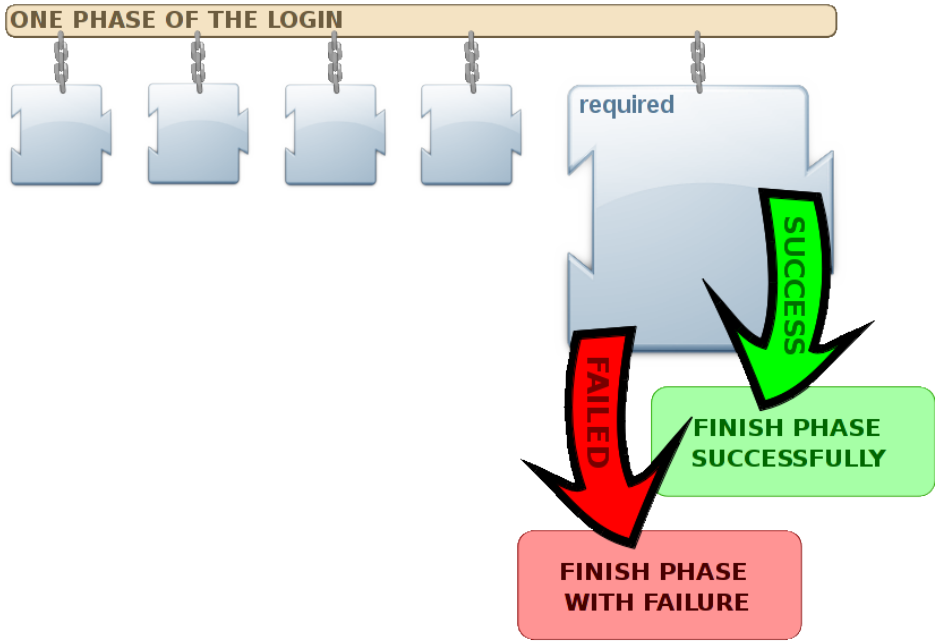
- If one of the plugins after the failing `required` plugin is `requisite` and also fails then immediately fail the phase with the error from the failing `required` plugin.

Wiring plugins together: required



- If one of the plugins after the failing `required` plugin is `sufficient` and succeeds then end the phase successfully.

Wiring plugins together: required



- If the last plugin is `required` then the success of the phase depends on the success of the final plugin

gPlazma configuration

```
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

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

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

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

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

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

The same plugin may appear multiple times, usually 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
```