The CDB evolution - part 4

10/07 - Universidad Autónoma de Madrid

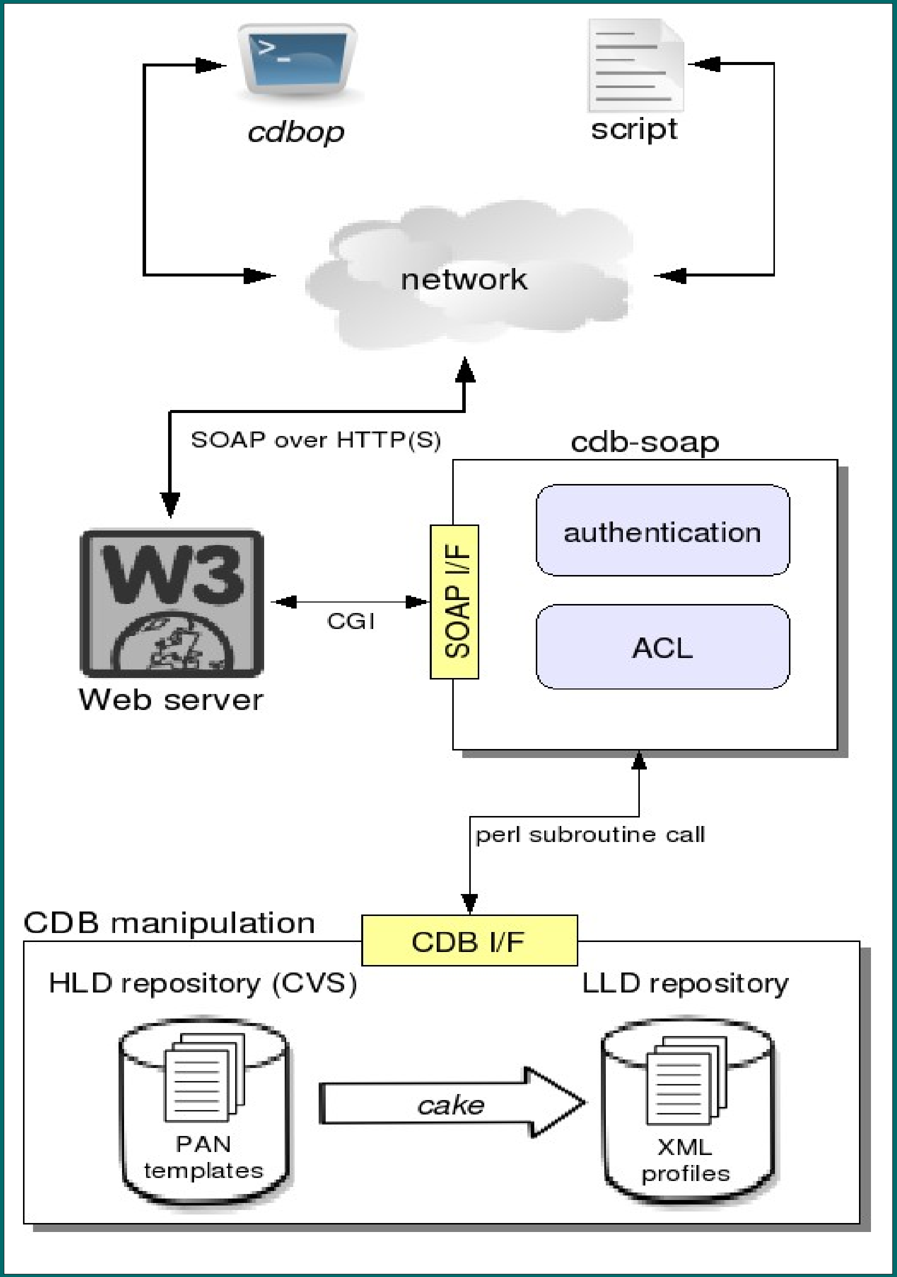
Marco Emilio Poleggi – *CERN-IT/INFN-CNAF*

`Marco.Emilio.Poleggi@cern.ch`

Outline

- □ CDB overview
- □ Tasks completed since last workshop @ TCD
- □ Left behind since the last workshop @ TCD
- □ What's new
- □ The CDB Deployer
- □ Boiling in the pot

CDB overview



Three-tier architecture

- □ SOAP client
  - cdbop interactive/batch shell
  - scripts
- □ SOAP middle-ware
  - Apache + *cdb-soap* CGI
  - stateless: each connection conveys one command
- □ CDB back-end
  - it's a library, not a server
  - templates compiled via cake
  - templates stored in *CVS*
  - stateful: partially transactional semantic through ‘sessions”

Tasks since last workshop @ TCD

- Completed
  - Template area "deployer"
  - Other mostly CERN-CC-related activities on the template set framework (procedures for server upgrade might be useful elsewhere ;-)):
    - "De-optimization"of templates
    - First stage introduction
- Left behind
  - Fine-grained CDB locking with fair queuing. Long term.
  - A common authentication service. Long term.

What's new

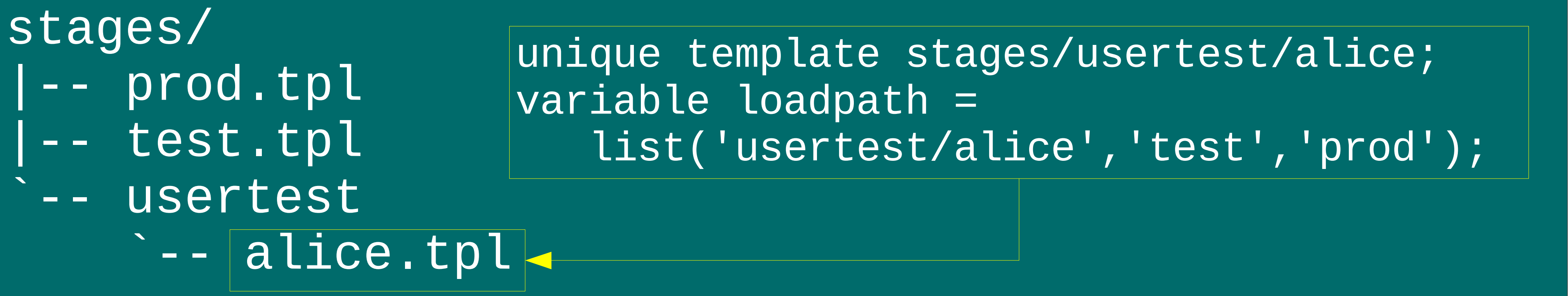
- **`cdb-cli (a.k.a. cdbop)`**
  - Name "canonicalization"for better namespaces' handling
  - Support for user-defined procedures
  - New `cp, mv` and `deploy` commands
    - First unit tests introduced :-)
  - Minimal support for disconnected operation
- **`cdb-soap`**
  - Mostly bug fixes
- **`cdb`**
  - Better log normalization
  - New tool `cdb-stat` for extracting performance indexes from logs
  - Template staging support

The CDB “Deployer”...

- Problem: Alice wants to test a (part of) configuration without affecting the production environment
  - She has a set of nodes dedicated to testing
  - Now she has to *manually*:
    - Replicate many `pro_foo_bar` templates, e.g. pro_type_`mycluster` into `new_type_mycluster`
    - Adapt `new_*` to the new setup
    - Test the new setup on a dedicated cluster, usually just a node in production
    - Move back the `new_*` stuff to `pro_*`
  - Pretty boring and error prone, isn't it?
    - Mess with templates' names :-(
    - Tests on production machines :-(
    - Not “humanly”scalable!

...The CDB “Deployer”: stages

- But, hold on, we have namespaces!
  - `pro_foo_bar` becomes `/pro/foo/bar`
  - The first namespace becomes an “area” or “stage”:
    - A place where to look for included templates
    - `/test/foo/bar, /preprod/foo/bar, /prod/foo/bar, ...`
  - “Staged” vs “non-staged” templates:
    - Non-staged declaration: `template test/foo/bar;`
    - Staged declaration: `template foo/bar;`
  - Meta-templates (non-staged) are used for recording where to look in:



```
stages/
|-- prod.tpl
|-- test.tpl
`-- usertest
    `-- alice.tpl
```

```
unique template stages/usertest/alice;
variable loadpath =
   list('usertest/alice','test','prod');
```

...The CDB “Deployer”: [re-]staging

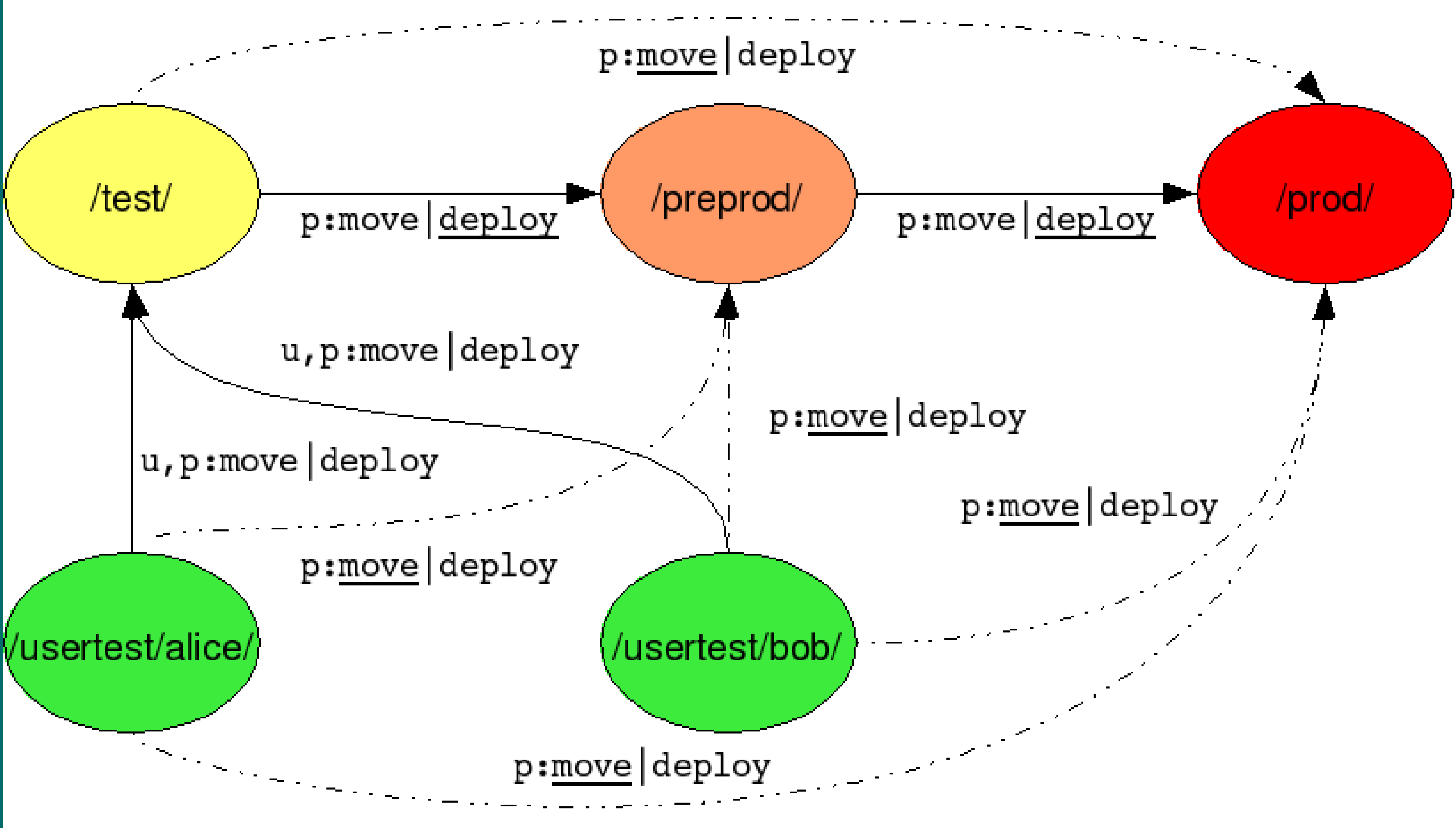
- How to “[re-]stage” now?
  - Profiles are *staged* by including a meta-template like the one above:

```
object template mynode;
include stages/test;
...
```

  - Alice needs just to replicate templates in her “sandbox” `usertest/alice`, or in another stage:
    - Use `cp`, `mv` and preferentially `deploy`
    - Templates *move* between stages (usually *forward*)
    - Profiles are expected to *persist* in their stages, though service managers can manually re-stage them at their convenience
  - Rollback: for now successful deploy operation must rolled back manually
    - Next to come: `drollback` command acting on CVS tags.

…The CDB ‘Deployer’: policies

- 'u' = *normal* user, 'p' = *privileged/power* user
- Full lines = *regular* transitions, dashed lines = *exceptional* transitions
- Underlined actions = *preferred*



…The CDB ‘Deployer’: example

- Starting setup:
  - Two templates + two profiles
  - No stages

```
/components/grub/config
/hardware/disk/STD_80
/prod_node
/test_node
```

- Final setup:
  - Two stages `test/` and `prod/`
  - One replicates template `components/grub/config`

```
/prod/components/grub/config
/prod/hardware/disk/STD_80
/profiles/prod_node
/profiles/test_node
/stages/prod
/stages/test
/test/components/grub/config
```

…The CDB ‘Deployer’: example

- Preparing the stage framework
  - Prepare two meta-templates
  - Add the needed namespaces and meta-templates
  - Move the profiles in the new location

edit

```
unique template stages/test;
variable loadpath = list('test', 'prod');
```

edit

```
unique template stages/prod;
variable loadpath = list('prod');
```

cdbop

```
> add_ns profiles
> add_ns prod
> add_ns test
> add stages/test.tpl stages/prod.tpl
> mv -f -a prod_node test_node profiles/
```

…The CDB 'Deployer': example…

- Staging profiles first time
  - All configuration is staged in `prod/` because the other area is empty

edit

```
object template prod_node;
include stages/prod;
include components/grub/config;
include hardware/disk/STD_80;
```

edit

```
object template test_node;
include stages/test;
include components/grub/config;
include hardware/disk/STD_80;
```

cdbop

```
> update profiles/prod_node.tpl profiles/test_node.tpl
> deploy -a components/ prod/components/
> deploy -a hardware/ prod/hardware/
> commit
> rm_ns -r /components/grub/
> rm_ns -r /hardware/disk/
> commit
```

…The CDB ‘Deployer”: example…

- Preparing a test stage
  - Use `deploy` with “skip removal”option
    - Ignore warnings about missing `foo/*` files: deploy looks for templates in each subnamespace
  - Now `test_node` gets `components/grub/config` from stage `test/`

cdbop

```
> deploy -a -s /prod/components/ /test/components/
> commit
```

- Once done with testing, deploy again into `prod/`
  - Standard deploy moves templates
    - Option `-a` stands for “all or nothing”

cdbop

```
> deploy -a /test/ /prod/
> commit
```

Boiling in the pot

Ideas

- *Bug #24156*: Visualization/navigation tool in progress
  - Pan parser - `cdb-tpl-view` - available and partially integrated with `pangraph`
  - Other option: waiting for panc-8
- SOAP eats too much memory (again)!
  - Make use of file attachment features

Issues

- Manpower :-(
- Reverting committed changes
  - Increasing CDB users --> higher mess up probability
  - Using CVS tags would easily solve the problem, but needs “harnessing”users' freedom

Am I forgetting something?
Any RFE, wish, remark?