Aquilon Atomic and Test Deployments
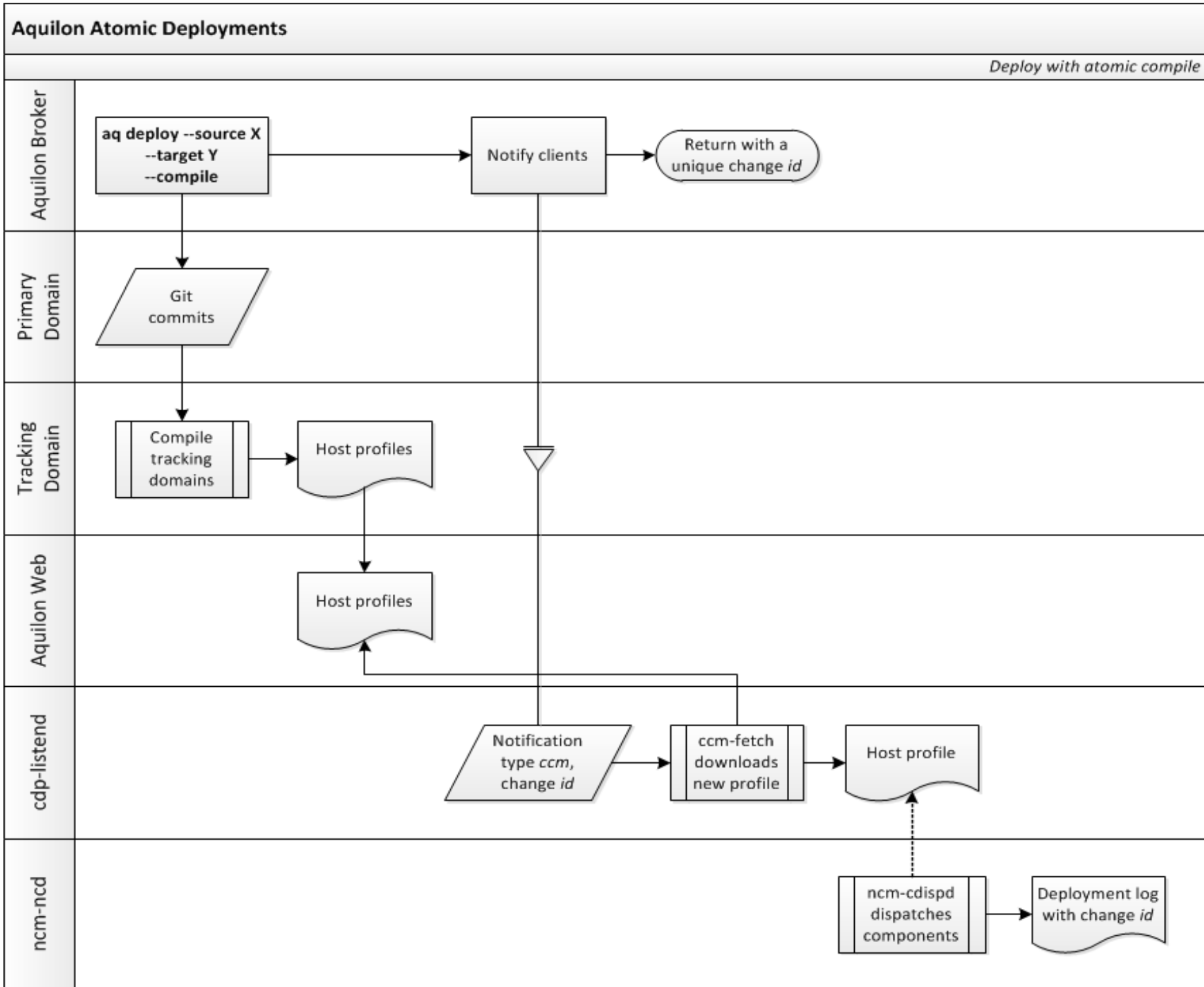
Mark R. Bannister

Quattor Workshop, October 2016

Objectives

- Rock solid test deployments!
- Rock solid back-outs!
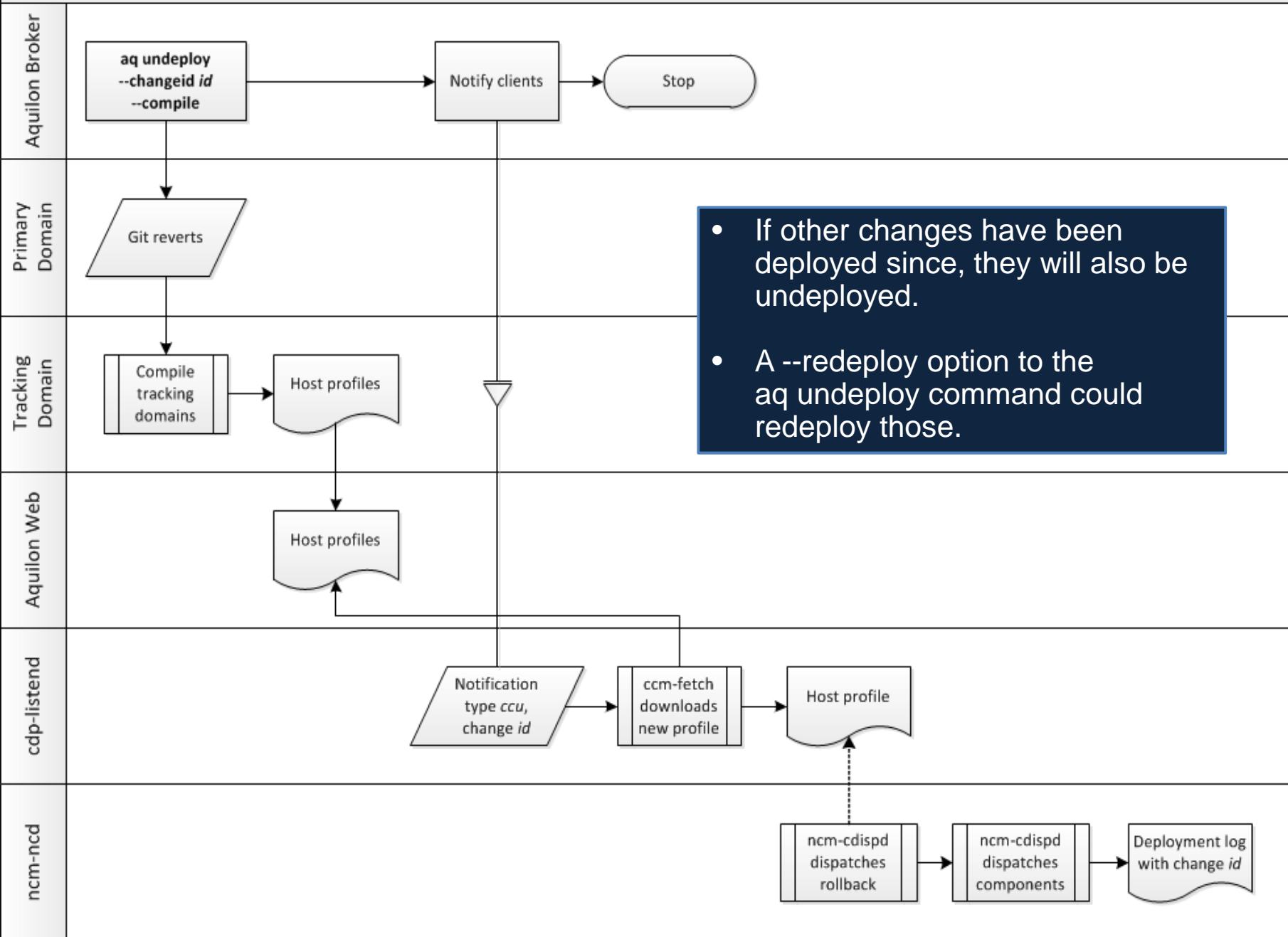- But what does that mean?

Objectives

- Be confident in the knowledge of the effects of deploying a change
- To be able to quickly and 100% reliably remove a change
- Be able to identify a list of changes that have recently been made on any single host
- To identify which change caused a particular artefact
- Be able to list all manual steps required to back out a change on an individual host (emergency recovery)

Aquilon Atomic Deployments

Aquilon Broker

aq deploy --source X
--target Y
--compile

Notify clients

Return with a unique change *id*

Primary Domain

Git commits

Tracking Domain

Compile tracking domains

Host profiles

Aquilon Web

Host profiles

cdp-listend

Notification type *ccm*, change *id*

ccm-fetch downloads new profile

Host profile

ncm-ncd

ncm-cdispd dispatches components

Deployment log with change *id*

4

Aquilon Atomic Deployments

Aquilon Broker

aq undeploy --changeid *id* --compile → Notify clients → Stop

Primary Domain

Git reverts

Tracking Domain

Compile tracking domains → Host profiles
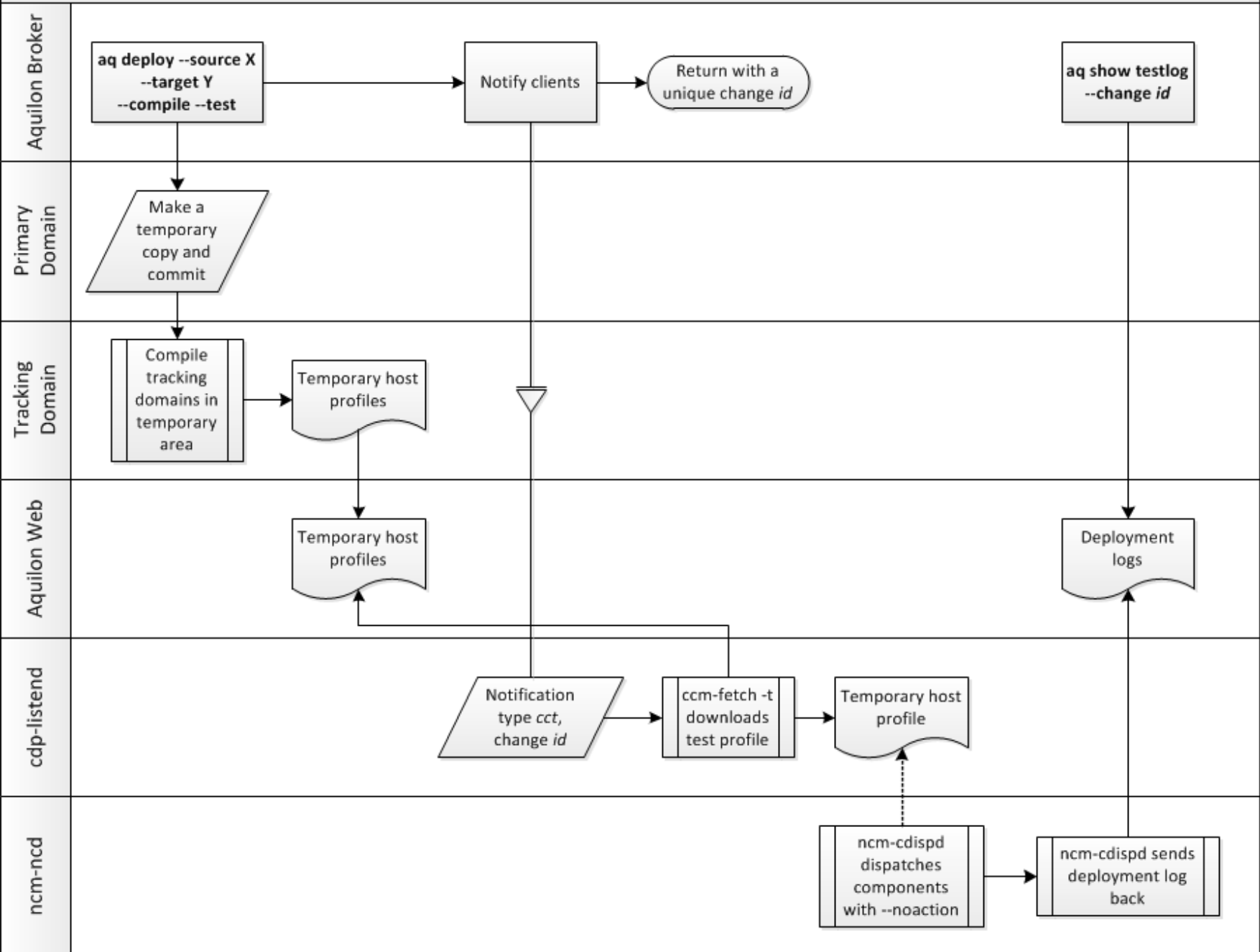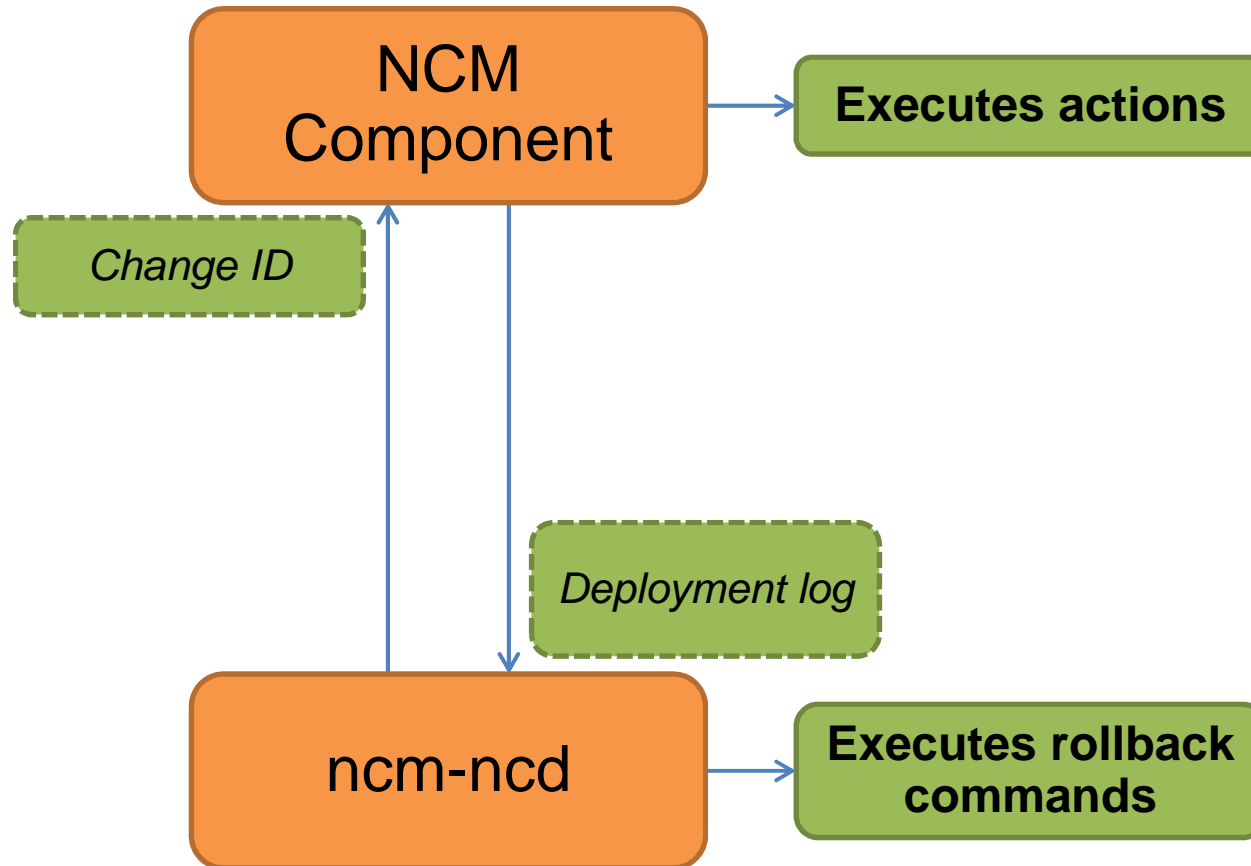
- If other changes have been deployed since, they will also be undeployed.

- A --redeploy option to the aq undeploy command could redeploy those.

Aquilon Web

Host profiles

cdp-listend

Notification type *ccu*, change *id* → ccm-fetch downloads new profile → Host profile

ncm-ncd

ncm-cdispd dispatches rollback → ncm-cdispd dispatches components → Deployment log with change *id*

5

Aquilon Test Deployments

Aquilon Broker

**aq deploy --source X
--target Y
--compile --test**

Notify clients

Return with a
unique change *id*

**aq show testlog
--change *id***

Primary Domain

Make a
temporary
copy and
commit

Tracking Domain

Compile
tracking
domains in
temporary
area

Temporary host
profiles

Aquilon Web

Temporary host
profiles

Deployment
logs

cdp-listend

Notification
type *cct*,
change *id*

ccm-fetch -t
downloads
test profile

Temporary host
profile

ncm-ncd

ncm-cdispd
dispatches
components
with --noaction

ncm-cdispd sends
deployment log
back

6

Execution Flow

NCM Component

Executes actions

Change ID

Deployment log

ncm-ncd

Executes rollback commands

Example CAF::Executor Object

```
CAF::Executor
-       Change ID #123
-       Task #1 => {
-               execute => {
-                       object => CAF::FileWriter(<config_file>, <backup_file>)
-                       desc   => "Add <new_item> to <config_file>"
-               }
-               undo => {
-                       object => same CAF::FileWriter (restore <backup_file>)
-                       desc   => "Restore <backup_file> to <config_file>"
-               }
-       }
-       Task #2 => {
-               execute => {
-                       object => CAF::Process(["pkill", "-HUP", "<process>"])
-                       desc   => "Send HUP signal to <process>"
-               }
-               undo => {
-                       # same as execute
-               }
-
-       }
-       Task #3 => {
-               execute => {
-                       object => CAF::Evaluate(sub {<arbitrary code>})
-                       desc   => "Some custom description"
-               }
-               undo => {
-                       object => CAF::Evaluate(sub {<arbitrary code>})
-                       desc   => "Some custom reverse description"
-                       reverse => 1   # When undoing, tasks marked with this flag are executed
-                                      # in reverse order
-               }
-       }
```

Example CAF::Executor Object

```
CAF::Executor
-   Change ID #123
-   Task #1 => {
-      execute => {
-         object => CAF::FileWriter(<config_file>, <backup_file>)
-         desc   => "Add <new_item> to <config_file>"
-      }
-      undo => {
-         object => same CAF::FileWriter (restore <backup_file>)
-         desc   => "Restore <backup_file> to <config_file>"
-      }
-   }
```

Example CAF::Executor Object

```
CAF::Executor
-    Change ID #123
-    Task #2 => {
-       execute => {
-          object => CAF::Process(["pkill", "-HUP",
"<process>"])
-          desc    => "Send HUP signal to <process>"
-       }
-       undo => {
-          # same as execute
-       }
-    }
```

Example CAF::Executor Object

```
CAF::Executor
-   Change ID #123
-   Task #3 => {
-      execute => {
-        object => CAF::Evaluate(sub {<arbitrary code>})
-        desc   => "Some custom description"
-      }
-      undo => {
-        object => CAF::Evaluate(sub {<arbitrary code>})
-        desc   => "Some custom reverse description"
-        reverse => 1   # When undoing, tasks marked with this
-                       # flag are executed in reverse order
-      }
-   }
```