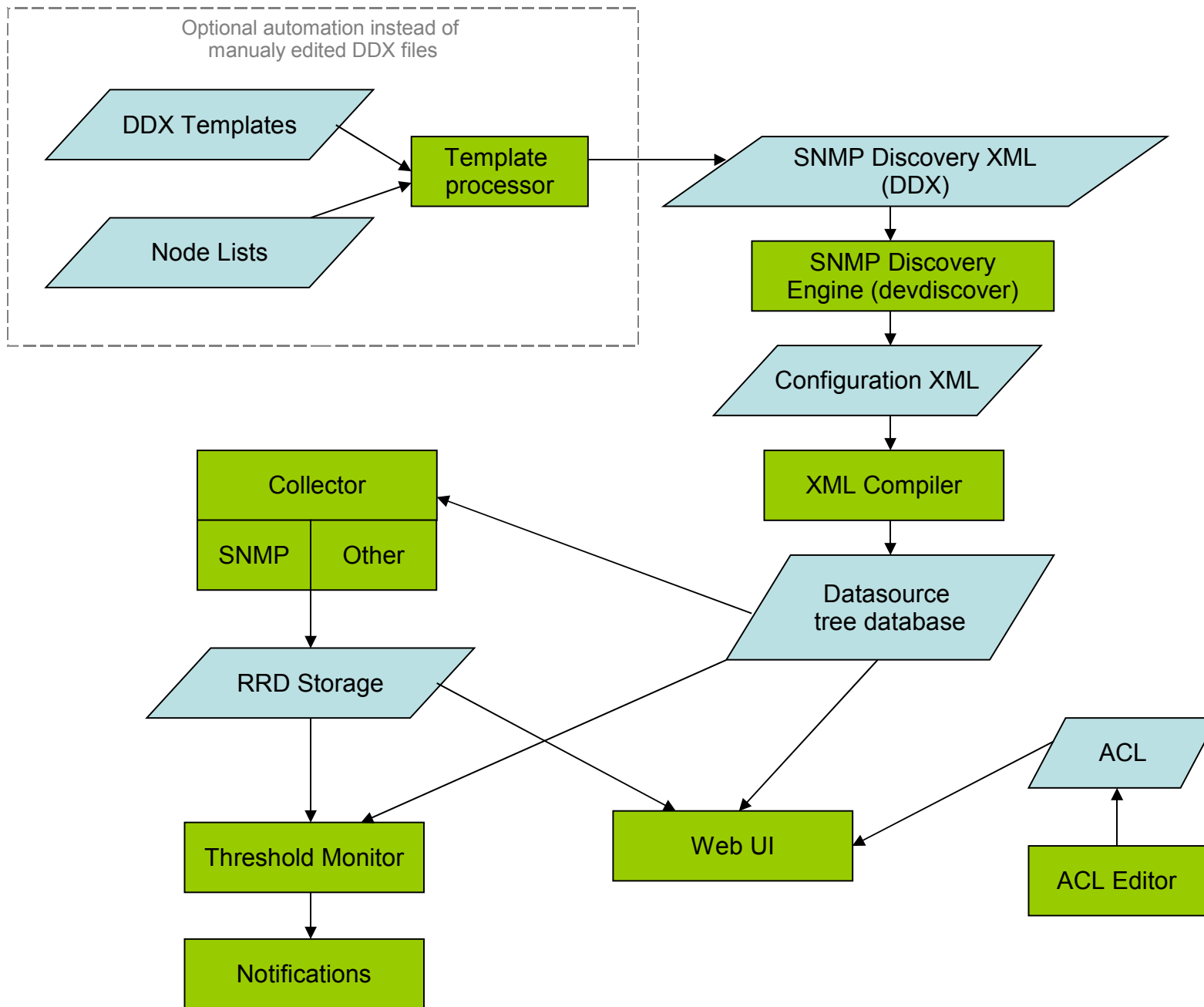


Torus Functional Overview

Torrus features

- Hierarchical object database
- Modular structure
- SNMP discovery and collector
- Threshold monitor
- Billing data export
- Web user interface



SNMP Discovery XML

- Usually located in `/usr/local/etc/torrus/discovery`
- Proposed extension: *DDX* (Device Discovery XML)
- Consists of global and per-host parameters

```
<?xml version="1.0" encoding="UTF8"?>
<snmp-discovery>
  <param name="data-dir" val e="!data!torr s!collector"rrd"!>
  ### other global parameters
  <$ost>
    <param name="snmp-$ost" val e="cmts1.tvtnet.c$"!>
    ### other host-specific parameters
  <!$ost>
<!/snmp-discovery>
```

Important DDX parameters

- % `data-dir`: *where RRD files will be stored*
- % `domain-name`: *used for DNS-based snmp-host's*
- % `host-subtree`: *place in the tree hierarchy*
- % `snmp-community`, `snmp-version`, `snmp-timeout`,
`snmp-retries`, `snmp-port`
- % `snmp-host`: *hostname or IP address*
- % `output-file`: *where to save the discovery results*

DDX: output-bundle

- `<param name="output-bundle" value="NYC/devices1 !ml" />`
- Defines a configuration file that would include all files generated by this DDX
- Allows easy generation of one XML file per SNMP host

DDX: selectors

- Selectors are a way to add new actions to the discovered elements (interfaces, CPUs, power supplies, etc.).
- Most of customization you want to do on the discovery results can be done through selectors.
- Typical actions: adding threshold monitors and data export for billing.

DDX: selectors (cont.)

- This example adds the threshold monitor to all interfaces that have “DNS” in their descriptions:

```
<$ost>
  <param name="snmp-$ost" val e="1.'.(.)" !>
  <param name="sym*olic-name" val e="s+itc$1.example.com" !>
  <param name="o tp t-,ile" val e="-./012-"340!1.'.(.)xml" !>

  <param name="selectors" val e="dnstra,,ic" !>
  <param name="dnstra,,ic-selector-type" val e="5F3'86("7F"078" !>
  <param name="dnstra,,ic-selector-e!pr" val e="9i,3omment:" !>
  <param name="dnstra,,ic-ifComment" val e=";<=" !>
  <param name="dnstra,,ic-selector-actions"
    val e="7n8ytes0onitor> 4 t8ytes0onitor" !>
  <param name="dnstra,,ic-7n8ytes0onitor-ar#" val e="dnstra,,ic" !>
  <param name="dnstra,,ic-4 t8ytes0onitor-ar#" val e="dnstra,,ic" !>
< !$ost>
```


DDX: static tokensets

- Tokenset is a set of graphs that are viewable on a single web page.

```
<param name="define-to$ensets">
  pstream-peering& Upstream peering?
  large-c stomers& 2arge c stomers
<!param>

<$ost>
  <param name="snmp-$ost" val e="10.0.0.1"!>
  <param name="sym*olic-name" val e="core0'.example.net"!>
  <param name="o tp t-,ile" val e="example"net!core0'.example.net.xml"!>

  <param name="%&C'()*+,&+-,.,::to$enset-members">
    pstream-peering& 14=@"0?
    large-c stomers& Aiga*it-t$ernet0"'> =erial("0> Fast-t$ernet)"0
  <!param>
<!$ost>
```

Datasource trees

- Each tree can run multiple *collector* processes and one *monitor* process.
- Web interface access control lists set the user permissions per tree. Currently it is not possible to set different permissions inside the tree.
- Trees are defined in `torrus-siteconf i# pl`, usually located in `/usr/local/etc/torrus/conf`.

Datasource configuration XML

- Multiple XML files are compiled into one datasource tree.
- Usually they consist of discovery results and templates from Torrus distribution.
- In rare occasions, manual editing is required.
- Files are usually located in
`/usr/local/etc/torrus/!mlconfi#.`
- `% site-#lobal !ml` is usually included in all trees.

Basic commands

- Run SNMP discovery:
`torrus dd /in=012-340 dd! 5/verbose6`
- Compile XML configuration:
`torrus compile /tree=012-340 5/verbose6`
- Collector, monitor and Apache will reload the configuration automatically.

Startup and shutdown

- Torrus daemons are usually started via `/etc/init.d/torus start`
- **NEVER** stop the torrus daemons with "kill -9"
- If the daemons have stopped abnormally (e.g. because of server crash), stop all daemons and the Apache server and perform the database recovery.

Recommended setup

- One tree to run the collector – includes all devices.
- Multiple view-only trees for user web access and threshold monitoring
- Fine tuning of collector schedules is required for installations with more than 20-30 thousand SNMP objects (see User guide and Scalability guide).

Future developments

- Web UI with granular access control, custom properties, favorites, etc.
- Distributed and redundant architecture.
- Management console with web interface.