DDS
Dynamic Deployment System

Andrey Lebedev
Anar Manafov
GSI
2015-11-25

The Dynamic Deployment System

is a tool-set that automates and significantly simplifies a deployment of user defined processes and their dependencies on any resource management system using a given topology

Basic concepts

DDS:

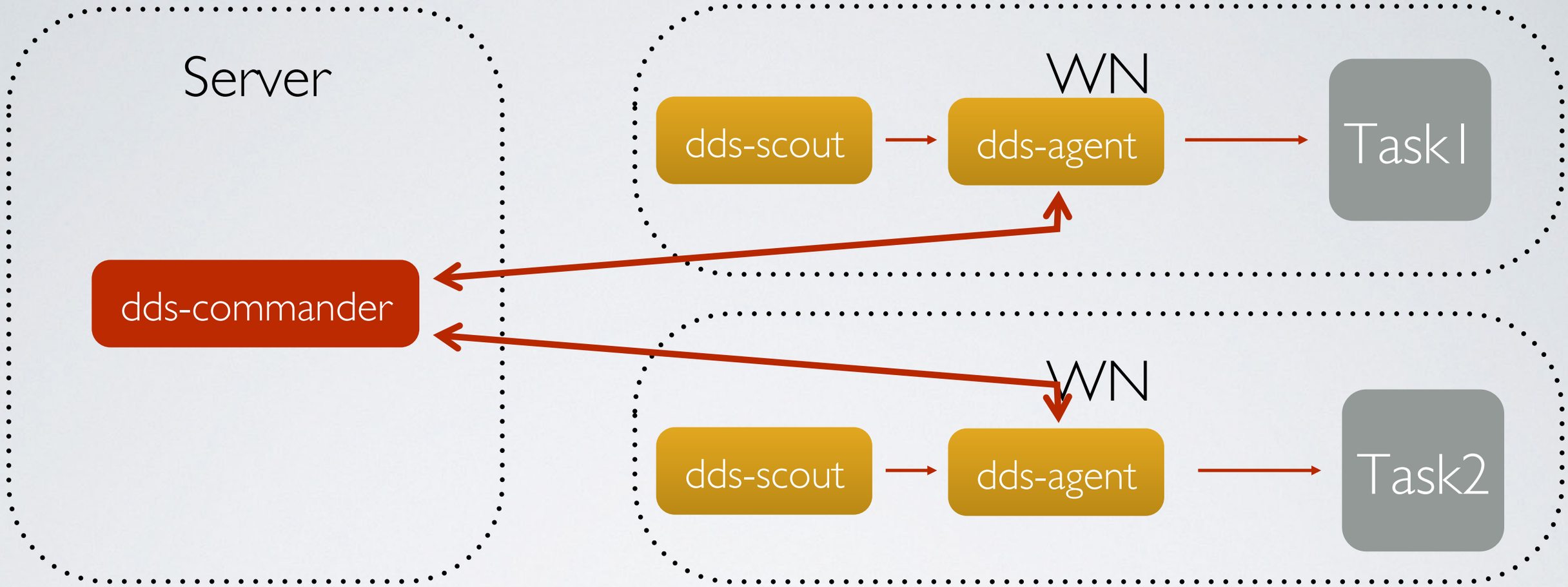
- implements a single-responsibility-principle command line tool-set and APIs,
- treats users' tasks as black boxes,
- doesn't depend on RMS (provides deployment via SSH, when no RMS is present),
- supports workers behind FireWalls (outgoing connection from WNs required),
- doesn't require pre-installation on WNs,
- deploys private facilities on demand with isolated sandboxes,
- provides a key-value properties propagation service for tasks,
- provides a rules based execution of tasks.

The contract

The system takes so called “topology file” as the input.
Users describe desired tasks and their dependencies using this file.

```
<topology id="myTopology">

  <decltask id="task1">
      <exe reachable="false">/Users/andrey/Test1.sh -l</exe>
  </decltask>

  <decltask id="task2">
      <exe>/Users/andrey/DDS/Test2.sh</exe>
  </decltask>

  <main id="main">
      <task>task1</task>
      <task>task2</task>
  </main>

</topology>
```

DDS Workflow



dds-server *start*

dds-submit -r *ssh* --ssh-rms-cfg *ssh_hosts.cfg*

dds-topology --set *topology_test.xml*

dds-topology --*activate*

DDS SSH plugin cfg file

ssh_hosts.cfg

@bash_begin@
@bash_end@

flp, lxi0234.gsi.de, , /tmp/dds_wrk, 8
epn, lxi235.gsi.de, , /tmp/dds_wrk, 10

Highlights since last meeting

1. task requirements based on the worker node name in the SSH configuration,
2. internal statistics tracking and accumulation,
3. custom commands for user tasks and ext. utils,
4. new user APIs: env. properties, custom protocol commands,
5. dds-submit: learned a localhost RMS,
6. improved internal transport protocol,
7. improved test coverage,
8. tutorials for key-value propagation and custom commands

… many more other fixes and stability improvements

more details here: https://github.com/FairRootGroup/DDS/blob/master/ReleaseNotes.md

Requirements

- Two possibilities to specify a task and collection requirement based on:

1. host name of the computing node

```
<declrequirement id="reqHost">
    <hostPattern type="hostname" value="lxi(1|2|3|4).gsi.de"/>
</declrequirement>
```

2. worker node name in the SSH configuration

```
<declrequirement id="reqNodeNameSSH">
    <hostPattern type="wnname" value="flp*"/>
</declrequirement>
```

hosts.cfg

@bash_begin@
@bash_end@

flp, lxi0234.gsi.de, , /tmp/dds_wrk, 8
epn, lxi235.gsi.de, , /tmp/dds_wrk, 10

Internal statistics tracking

Message size, message queue size for read and write operations.

dds-stat *enable*

dds-stat *disable*

dds-stat *get*

Statistics is accumulated on the commander server for each channel separately.

Stat engine does not effect the overall performance.

```
[arybalch@cn48 DDS]$ dds-stat get
dds-stat:
Number of active channels: 3332
Read (message size)
cmd                          mean        max         sum          count
cmdHANDSHAKE                 12 B        12 B        39.05 KB     3332
cmdSIMPLE_MSG                47 B        48 B        155.12 KB    3328
cmdREPLY_HOST_INFO           94 B        94 B        304.89 KB    3328
cmdREPLY_ID                  16 B        16 B        52.00 KB     3328
cmdGET_AGENTS_INFO           8 B         8 B         8 B          1
cmdACTIVATE_AGENT            8 B         8 B         8 B          1
cmdUPDATE_KEY                85 B        86 B        267.34 KB    3328
cmdWATCHDOG_HEARTBEAT        8 B         8 B         831.98 KB    106494
cmdSET_TOPOLOGY              56 B        56 B        56 B         1
cmdENABLE_STAT               8 B         8 B         8 B          1
Total                        13 B        94          1.61 MB      123142

Write (message size)
cmd                          mean        max         sum          count
cmdSHUTDOWN                  8 B         8 B         16 B         2
cmdSIMPLE_MSG                67 B        82 B        416.39 KB    6660
cmdREPLY_HANDSHAKE_OK        8 B         8 B         26.03 KB     3332
cmdGET_HOST_INFO             8 B         8 B         26.00 KB     3328
cmdGET_ID                    8 B         8 B         26.00 KB     3328
cmdSET_ID                    16 B        16 B        52.00 KB     3328
cmdREPLY_AGENTS_INFO         670.52 KB   670.52 KB   670.52 KB    1
cmdASSIGN_USER_TASK          272 B       273 B       864.83 KB    3328
cmdACTIVATE_AGENT            8 B         8 B         26.00 KB     3328
cmdUPDATE_KEY                79 B        86 B        434.69 MB    5.54112e+06
cmdPROGRESS                  24 B        24 B        78.05 KB     3330
Total                        41.94 KB    686616      436.83 MB    5.57108e+06

Write (message queue size - bytes)
    mean        max         sum
    43.17 KB    670.52 KB   436.83 MB

Write (message queue size - messages)
        mean          max          sum
       16.96         1664      5571085
```

Custom commands (1)

Sending of custom commands from user tasks and ext. utilities.

Two use cases:

1. User task which connects to DDS agent
2. Ext. utility which connects to DDS commander

A custom command is a standard part of the DDS protocol.
From the user perspective a command can be any text, for example, JSON or XML.
A custom command recipient is defined by a condition.

Condition types:

1. Internal channel ID which is the same as sender ID.
2. Path in the topology: main/RecoGroup/TrackingTask.
3. Hash path in the topology: main/RecoGroup/TrackingTask_23.

Broadcast custom command
to all tasks with this path.

Task index.

Custom commands (2)

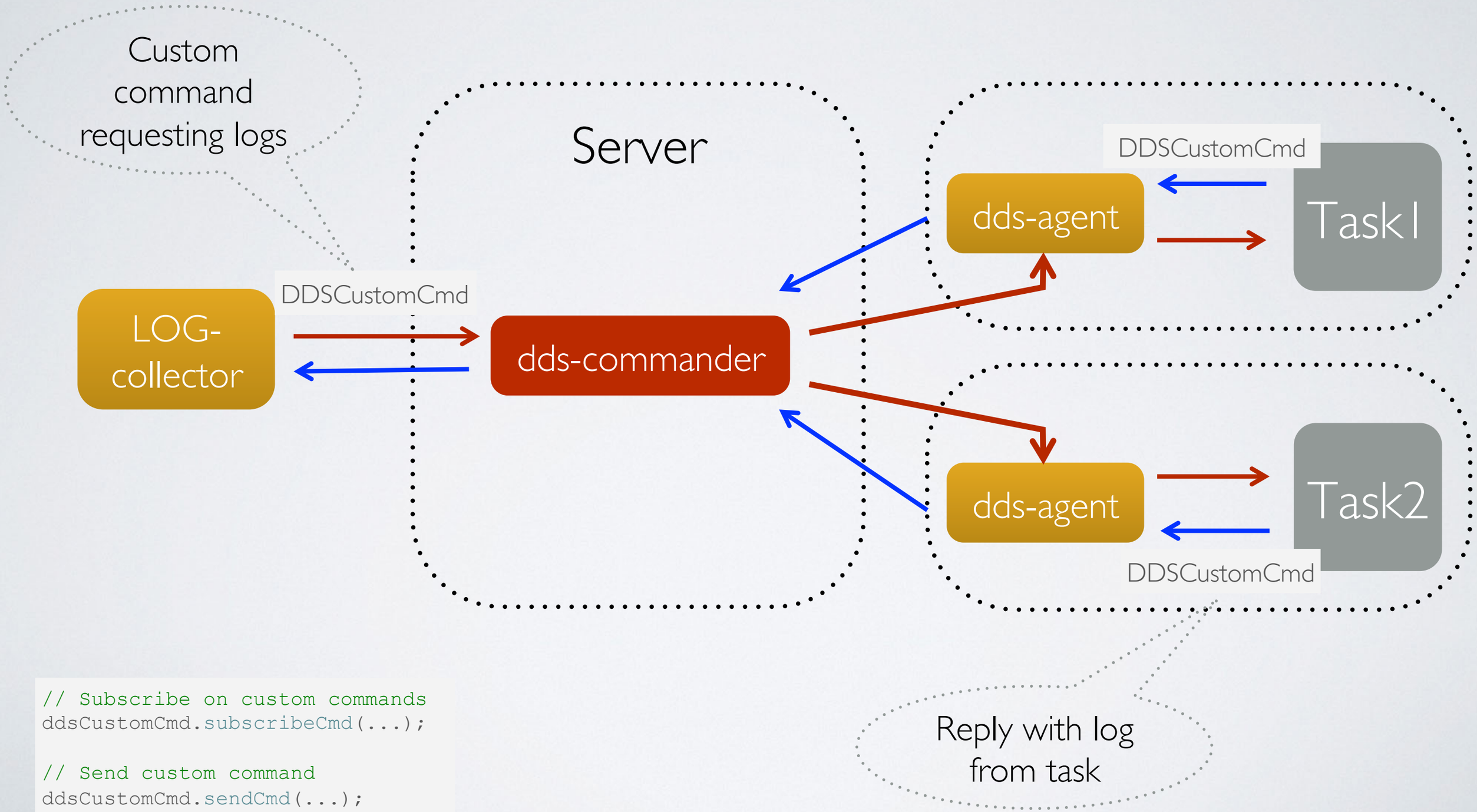
New library **dds-custom-cmd-lib** and header file “CustomCmd.h” with user API

```
#include “CustomCmd.h”

CCustomCmd ddsCustomCmd;

// Subscribe on custom commands
ddsCustomCmd.subscribeCmd(
    [](const string& _command, const string& _condition, uint64_t _senderId)
{
    cout << ”Command: " << _command << " condition: " << _condition
         << " senderId: " << _senderId << endl;

    // Send message back to sender
    if (_command == "please-reply”)
        ddsCustomCmd.sendCmd("reply“, to_string(_senderId));
});

// Subscribe on reply from DDS commander server
ddsCustomCmd.subscribeReply([](const string& _msg)
{
    cout << ”Message: " << _msg << endl;
});
```

For more information refer to Tutorial2 of DDS.

Custom commands (3)

A possible use case: collect log from the user tasks




```
// Subscribe on custom commands
ddsCustomCmd.subscribeCmd(...);

// Send custom command
ddsCustomCmd.sendCmd(...);
```

dds-submit learned a localhost RMS

dds-submit -r *ssh* --ssh-rms-cfg *ssh_hosts.cfg*

if you want to run on the localhost

dds-submit -r *localhost*

dds-submit -r *localhost* –n *10*

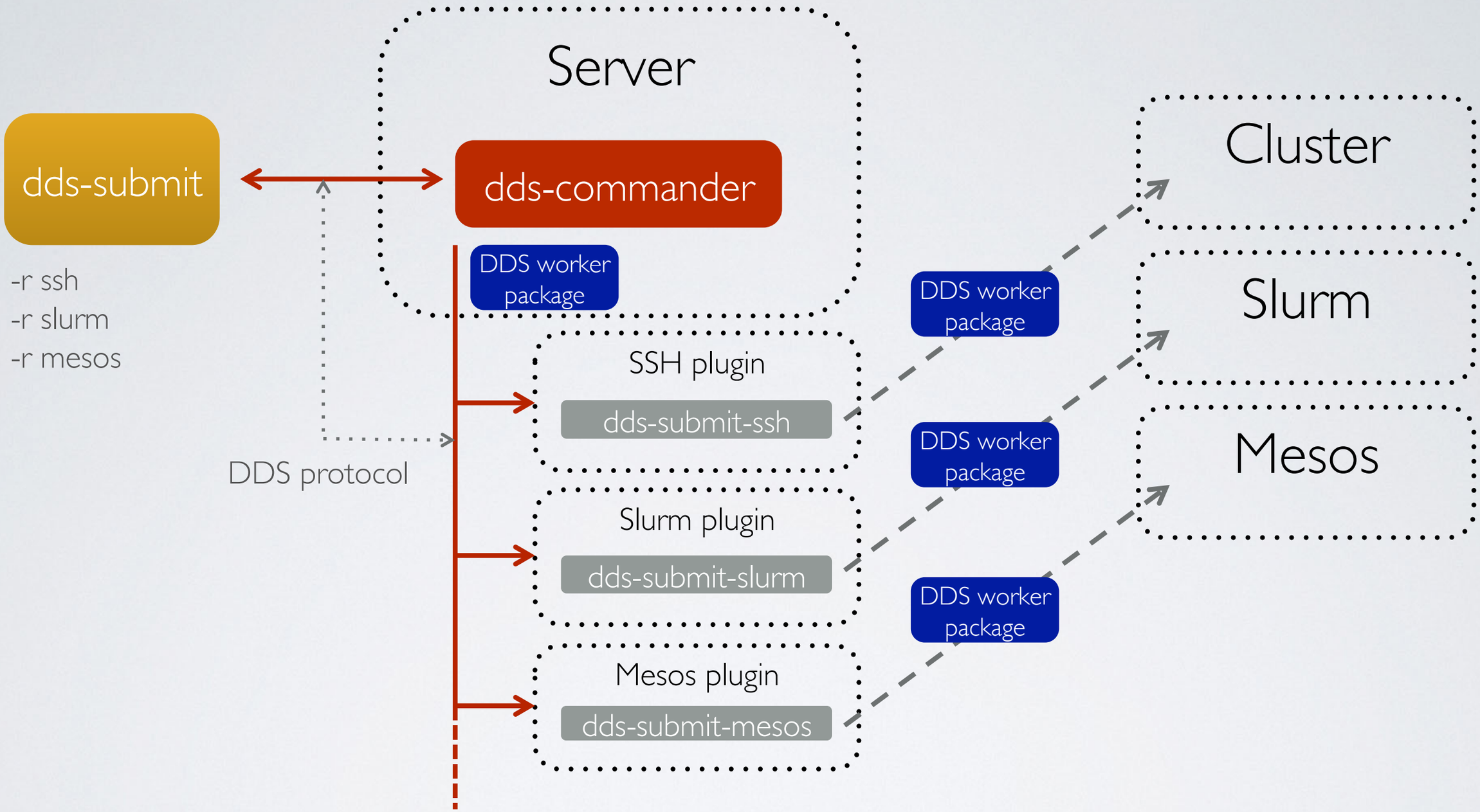
If -n is omitted, than the number of deployed agents is equal to the number of logical cores.

Documentation and tutorials

- User manual
- API documentation
- Tutorial1: key-value propagation
- Tutorial2: custom commands

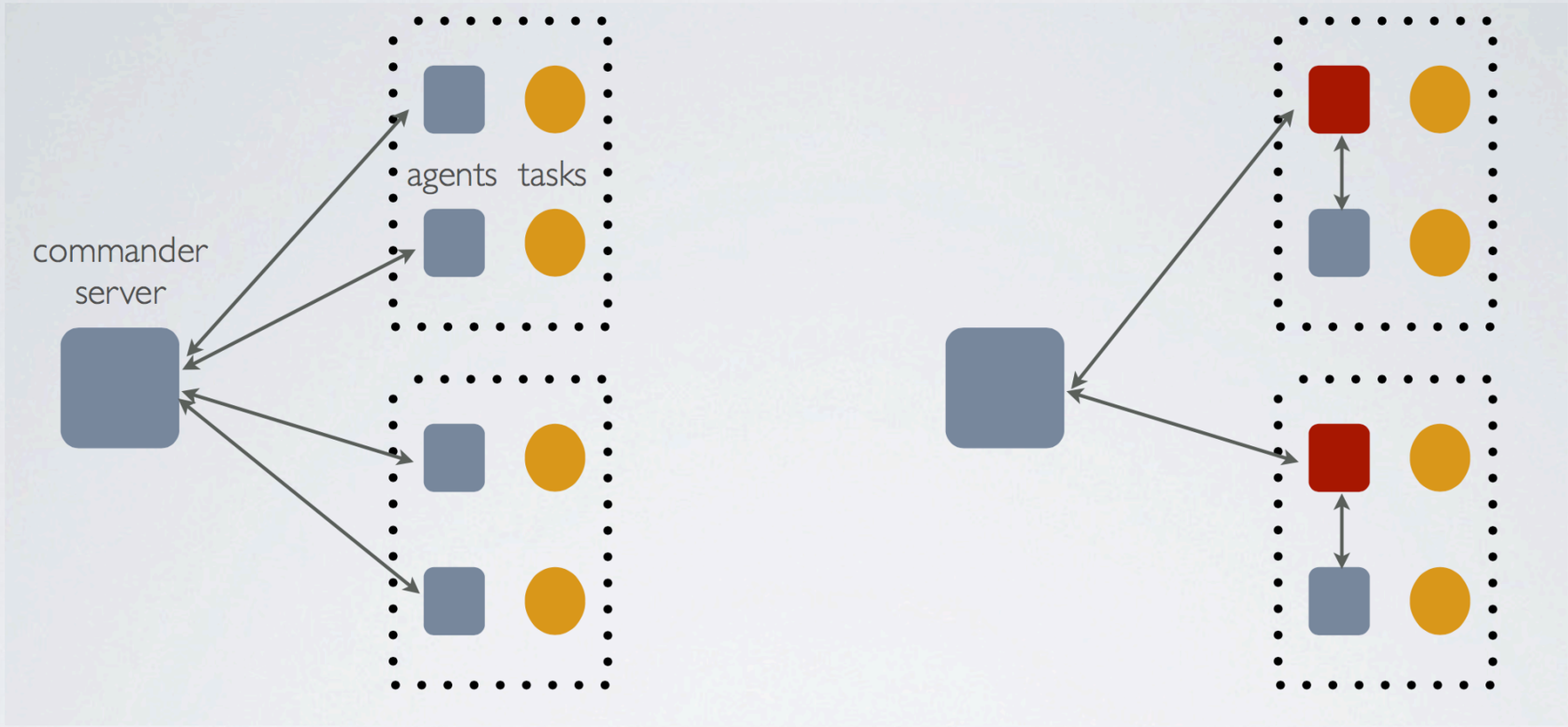
For more information refer to DDS documentation:
http://dds.gsi.de/documentation.html

Plugins



1. dds-commander starts a plugin based on the dds-submit parameter,
2. plugin connects back to dds-commander,
3. plugin receives submission details,
4. plugins takes WN package and deploys it to WNs.

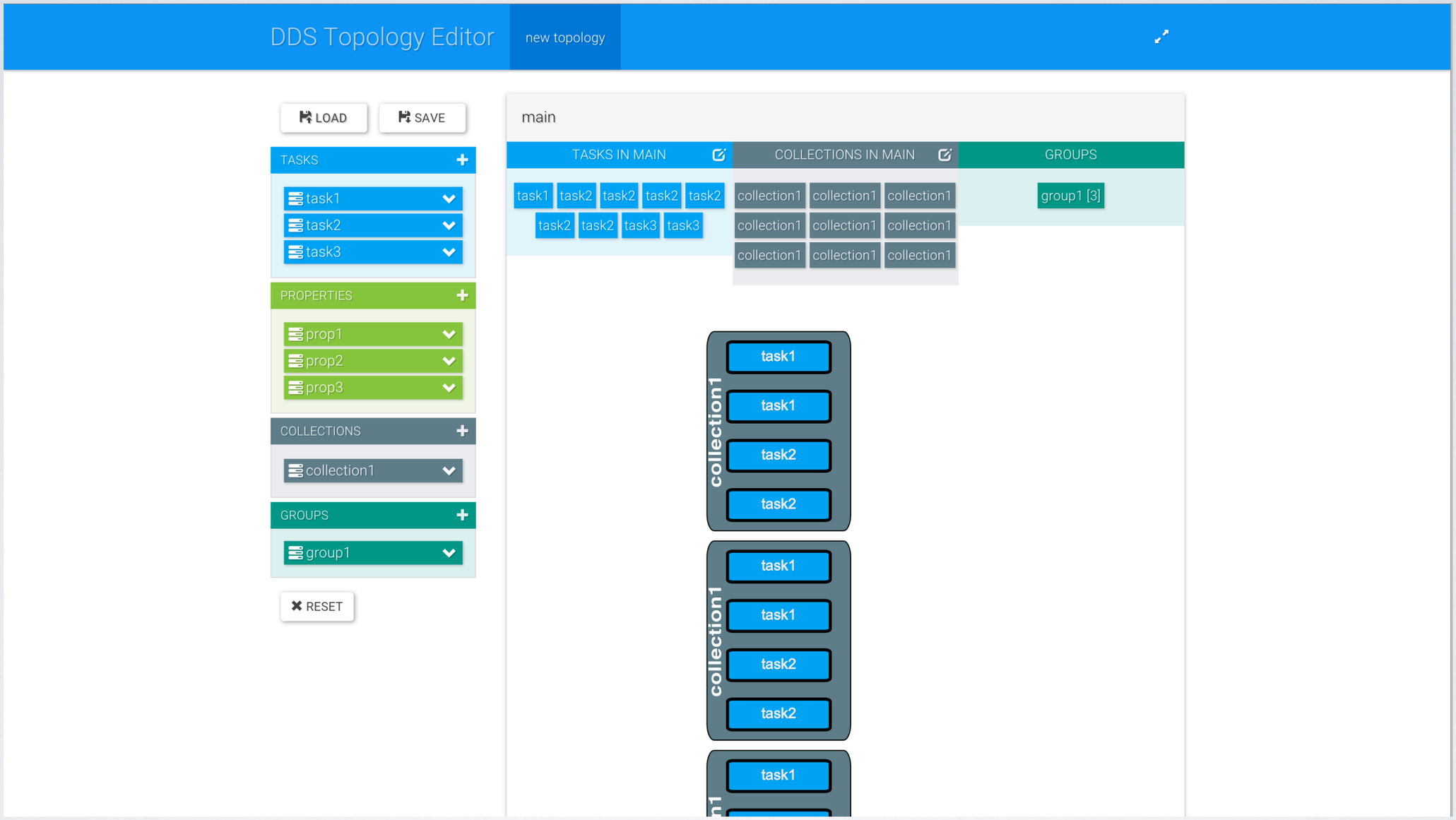
Master agent feature



1. DDS Commander will have one connection per host,
2. master agents will act as dummy proxy services, no special lo gic will be put on them except key-value propagation inside c ollections,
3. key-value will be either global or local for a collection

Work in progress. Expected in the next DDS release 1.2.

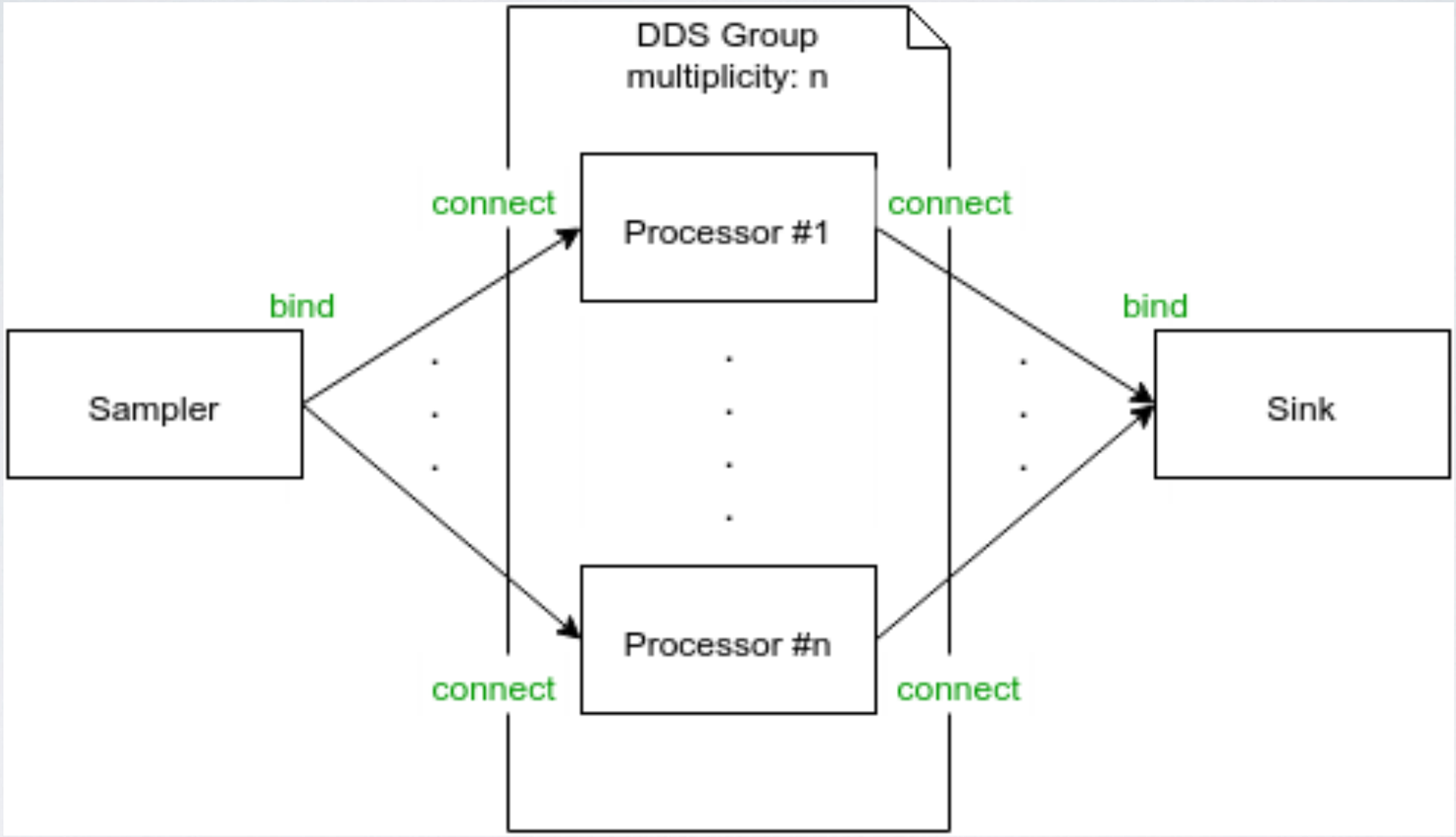
Topology editor



http://rbx.github.io/DDS-topology-editor/

Demo by Alexey Rybalchenko and Aleksandar Rusinov

FairRoot example with DDS



https://github.com/FairRootGroup/FairRoot/tree/master/examples/MQ/3-dds

Demo by Alexey Rybalchenko

Summary

- Current stable release - DDS v1.0 (2015-11-20, http://dds.gsi.de/download.html)
- Home site: http://dds.gsi.de
- User's Manual: http://dds.gsi.de/documentation.html
- Continuous integration: http://demac012.gsi.de:22001/waterfall
- Source Code:
  https://github.com/FairRootGroup/DDS
  https://github.com/FairRootGroup/DDS-user-manual
  https://github.com/FairRootGroup/DDS-web-site
  https://github.com/FairRootGroup/DDS-topology-editor