Introduction to a Virtualization Testsuite

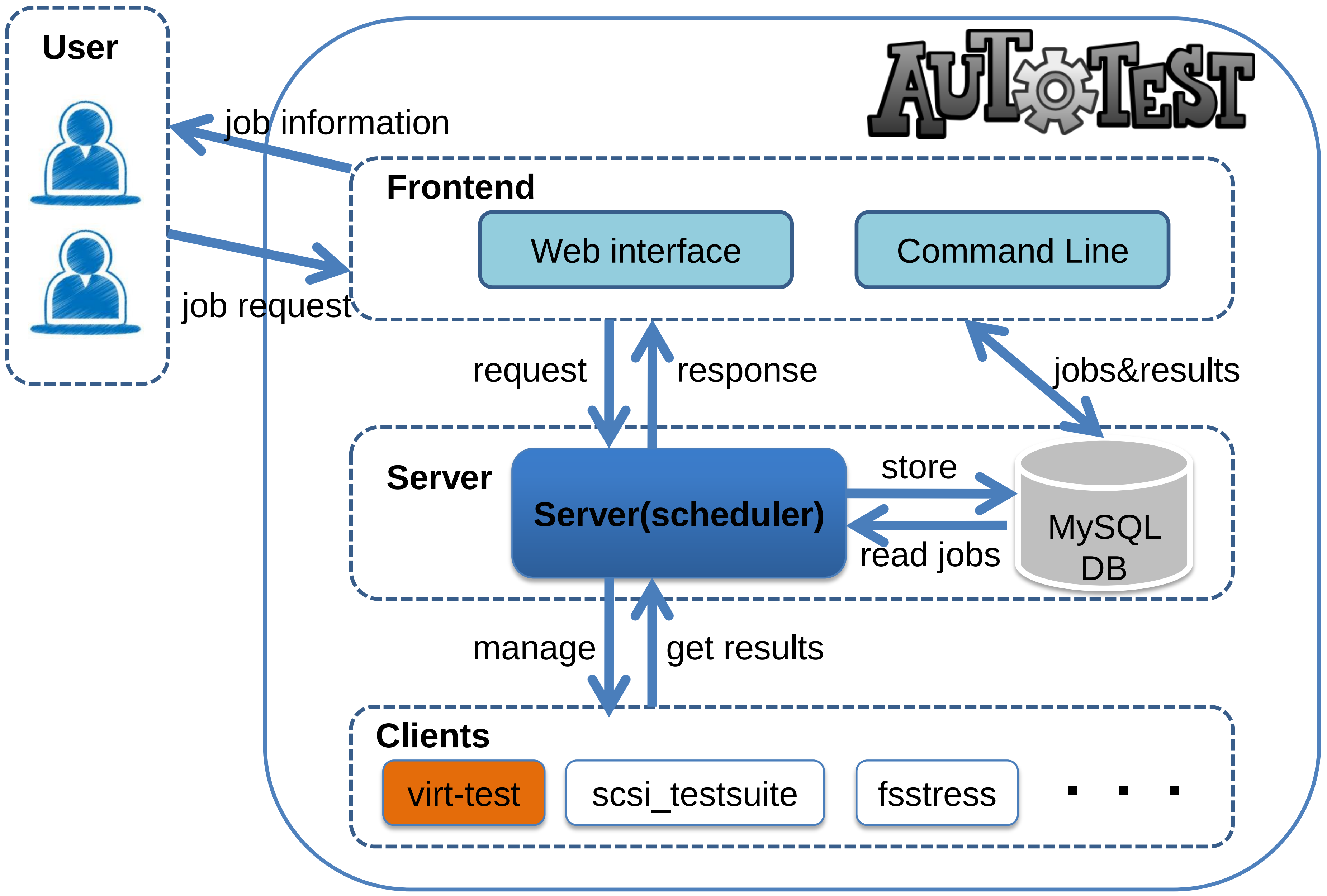
--Based on Autotest Testing Framework

Yu Mingfei
yumingfei@cn.fujitsu.com

Agenda

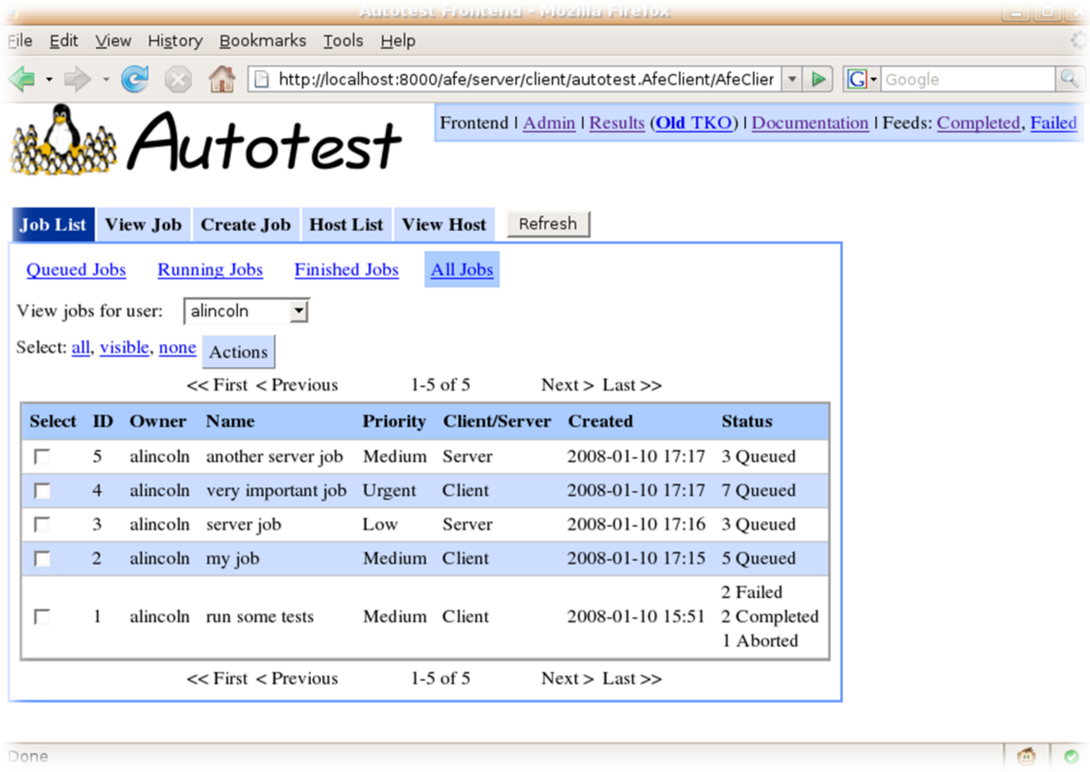
1. What’s Autotest: Overview & Features
2. Virtualization testsuite: Virt-test
    - Why Virt-test
    - Virt-test : Overview & Features
    - Runner : Run tests
    - Provider : Write tests
3. Future work

Autotest Overview



Autotest Frontend

- Frontend: user interface
  - browsing existing jobs
  - viewing job details
  - submitting new jobs
  - managing hosts



Autotest Frontend - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8000/afe/server/client/autotest.AfeClient/AfeClier
Google
Autotest
Frontend | Admin | Results (Old TKO) | Documentation | Feeds: Completed, Failed
Job List
View Job
Create Job
Host List
View Host
Refresh
Queued Jobs
Running Jobs
Finished Jobs
All Jobs
Actions
Done

View jobs for user: alincoln

Select: all, visible, none

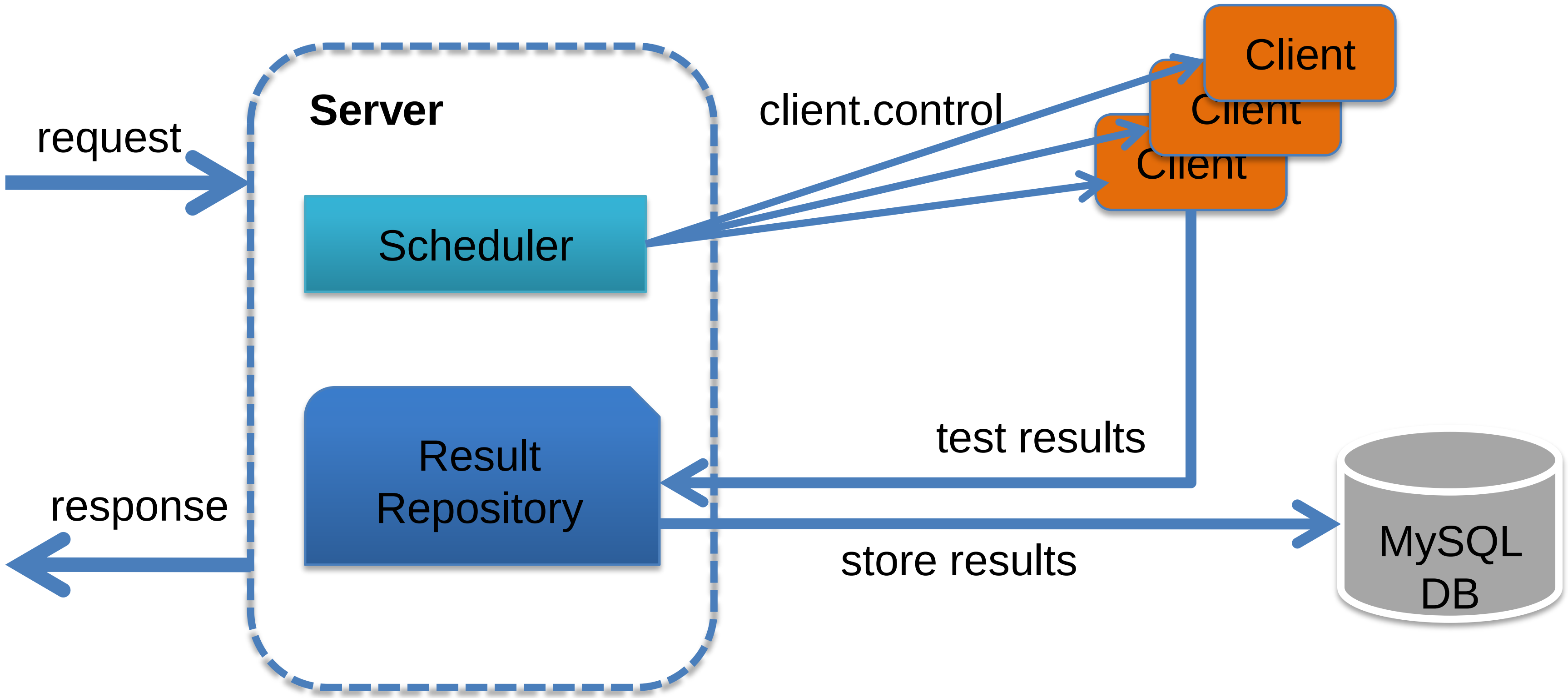
<< First < Previous 1-5 of 5 Next > Last >>

Select	ID	Owner	Name	Priority	Client/Server	Created	Status
☐	5	alincoln	another server job	Medium	Server	2008-01-10 17:17	3 Queued
☐	4	alincoln	very important job	Urgent	Client	2008-01-10 17:17	7 Queued
☐	3	alincoln	server job	Low	Server	2008-01-10 17:16	3 Queued
☐	2	alincoln	my job	Medium	Client	2008-01-10 17:15	5 Queued
☐	1	alincoln	run some tests	Medium	Client	2008-01-10 15:51	2 Failed 2 Completed 1 Aborted

<< First < Previous 1-5 of 5 Next > Last >>

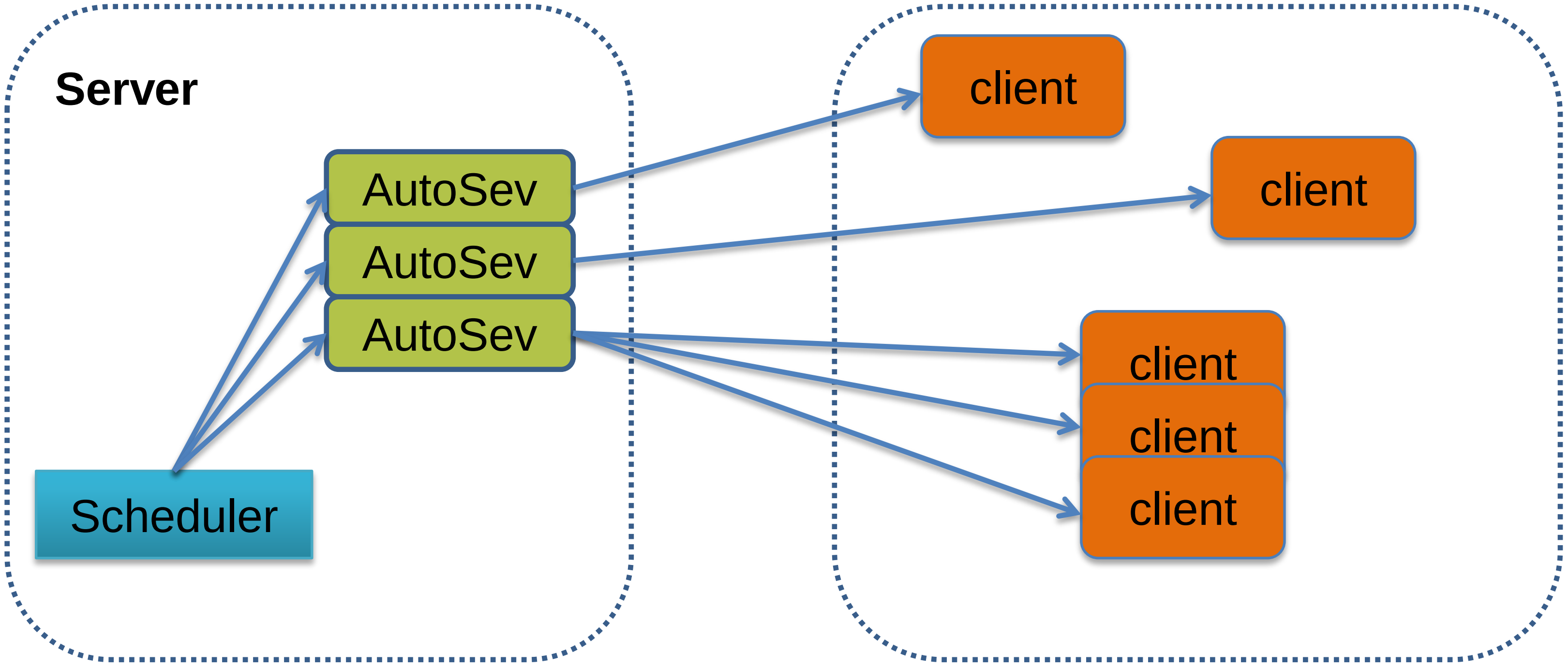
Autotest Server

- Server: control center



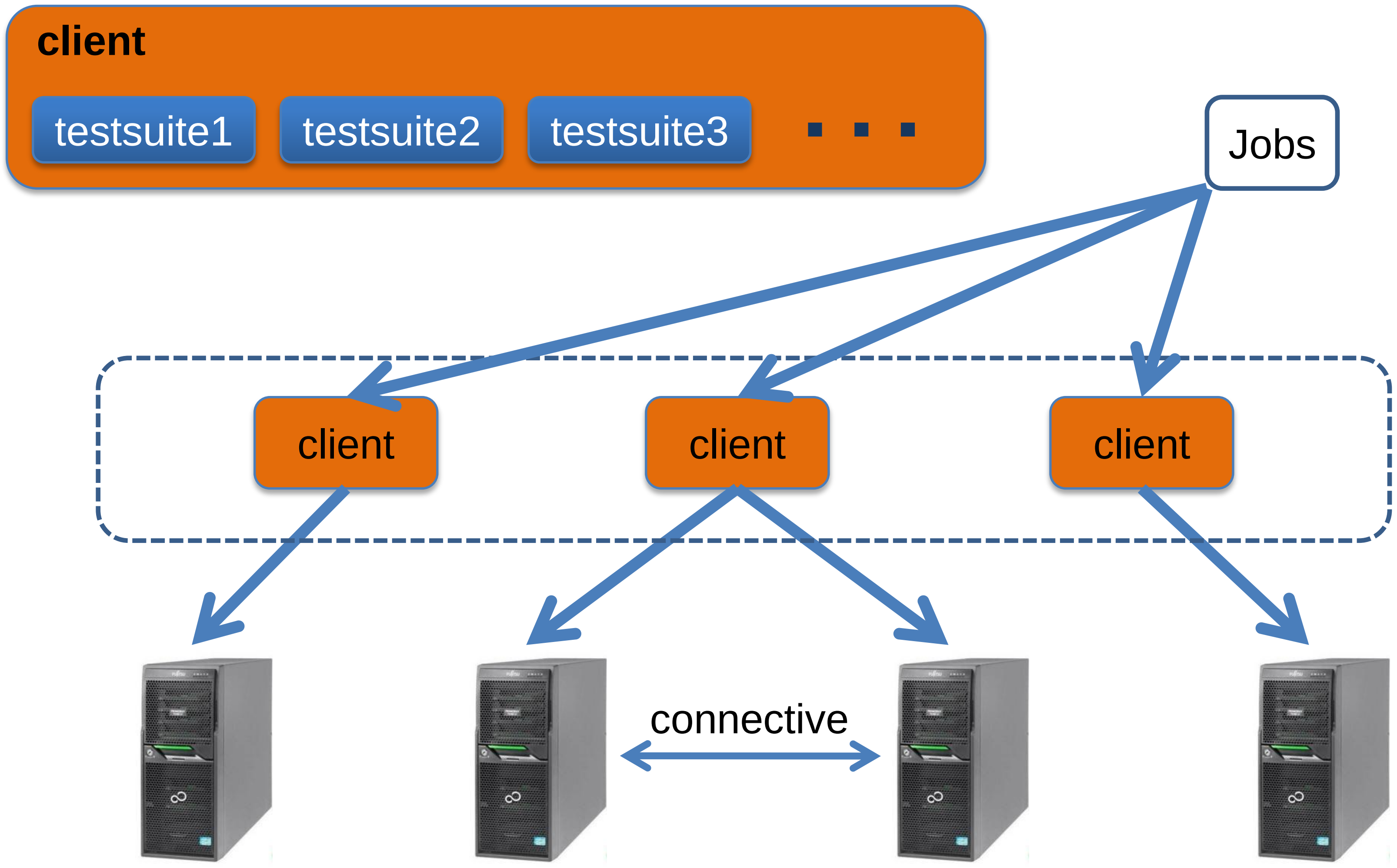
Autotest Scheduler

- Scheduler: trigger job for clients



Autotest Client

- Client: tests engine



Agenda

1. What’s Autotest: Overview & Features
2. Virtualization testsuite: Virt-test
    - Why Virt-test
    - Virt-test : Overview & Features
    - Runner : Run tests
    - Provider : Write tests
3. Future work

Why Virt-test

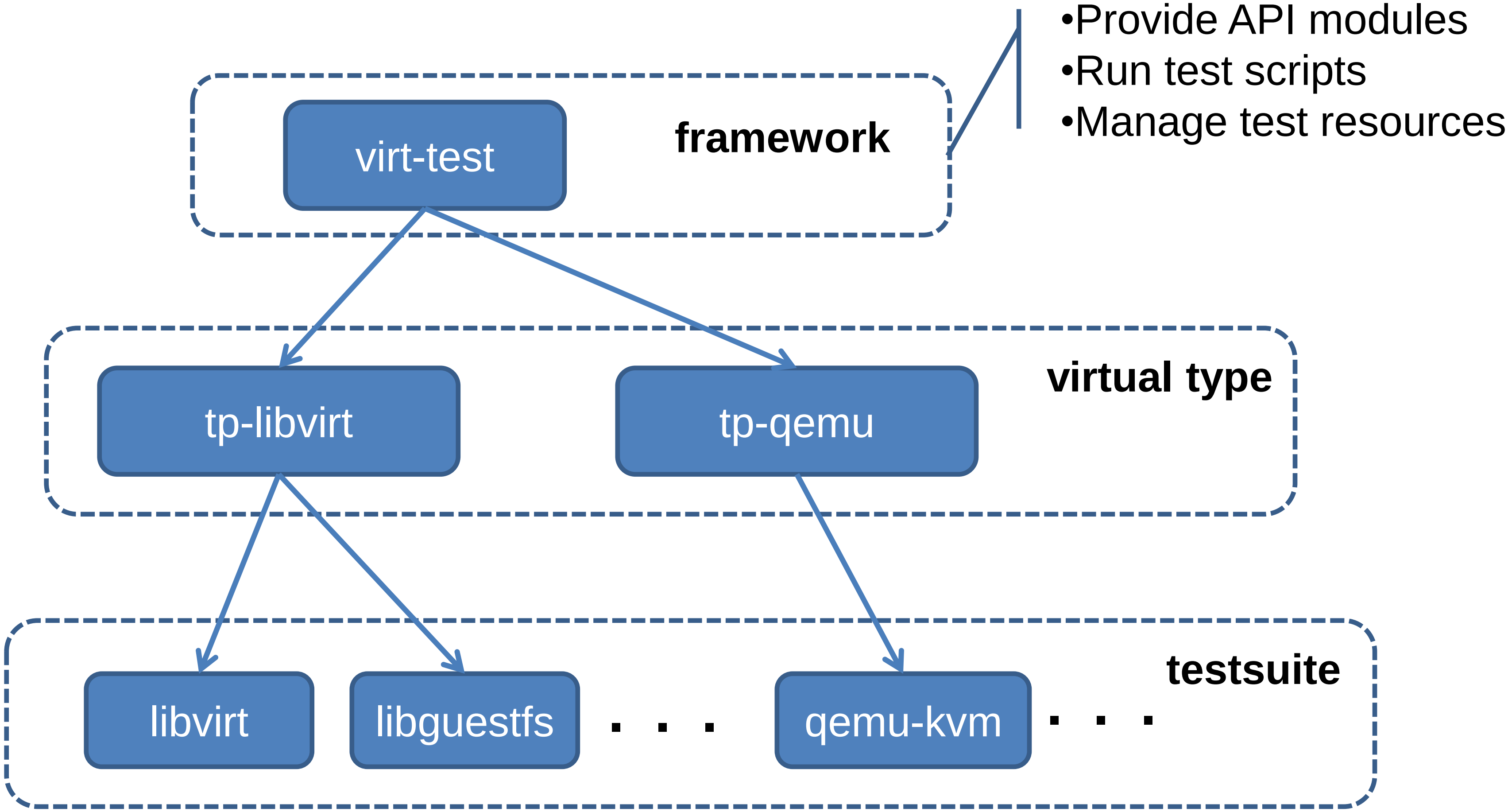
Virtualization:

- Ample functions
- Various Virtual Machines
- Increasing new features

A testsuite can do tests:

- Fully
- Automatically
- Expandability

Autotest client : Virt-test



- Provide API modules
- Run test scripts
- Manage test resources

Parameter references

Problem: Massive parameters

- Device types : IDE, virtio, scsi…
  formats : qcow2, raw…
- Storage types : directory, filesystem, logical, iscsi…
- Network types : bridge, nat…
- Command options

Solution : Cartesian Configuration

Cartesian Configuration

Example:

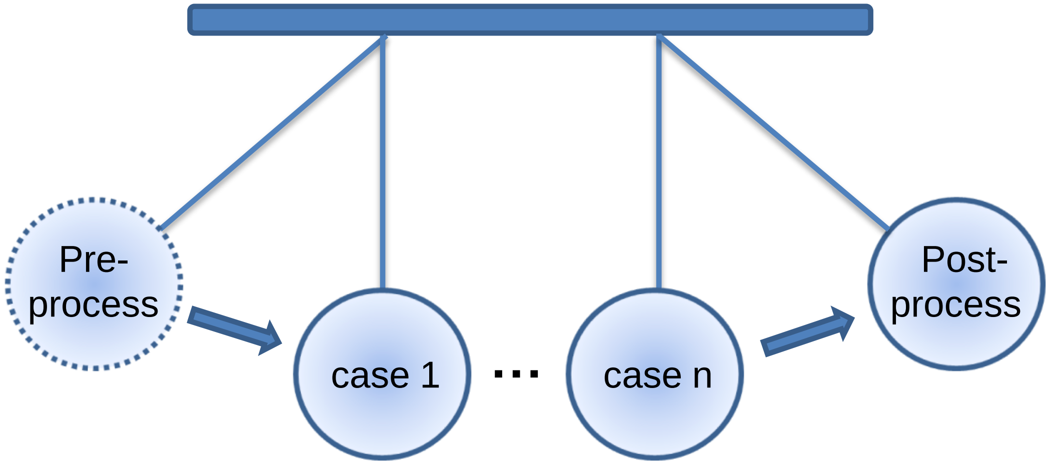
```
variants:
    - IDE:
        disk_type = ide
    - virtio:
        disk_type = virtio
    - scsi:
        disk_type = scsi
variants:
    - directory:
        storage_type = dir
    - filesystem:
        storage_type = fs
    - logical:
        storage_type = logical
variants:
    - bridge:
        network_type = bridge
    - NAT:
        network_type = nat
```

2 * 3 * 3

Dict1: bridge.directory.IDE
Dict2: bridge.directory.virtio
Dict3: bridge.directory.scsi
Dict4: bridge.filesystem.IDE
Dict5: bridge.filesystem.virtio
Dict6: bridge.filesystem.scsi
Dict7: bridge.logical.IDE
Dict8: bridge.logical.virtio
Dict9: bridge.logical.scsi
Dict10: NAT.directory.IDE
Dict11: NAT.directory.virtio
Dict12: NAT.directory.scsi
Dict13: NAT.filesystem.IDE
Dict14: NAT.filesystem.virtio
Dict15: NAT.filesystem.scsi
Dict16: NAT.logical.IDE
Dict17: NAT.logical.virtio
Dict18: NAT.logical.scsi

Pre&Post Processes

- Initialize Resources
- Setup and Cleanup Services
- Prepare Environment



OS support(Linux)

- Most release distros

ubuntu

debian

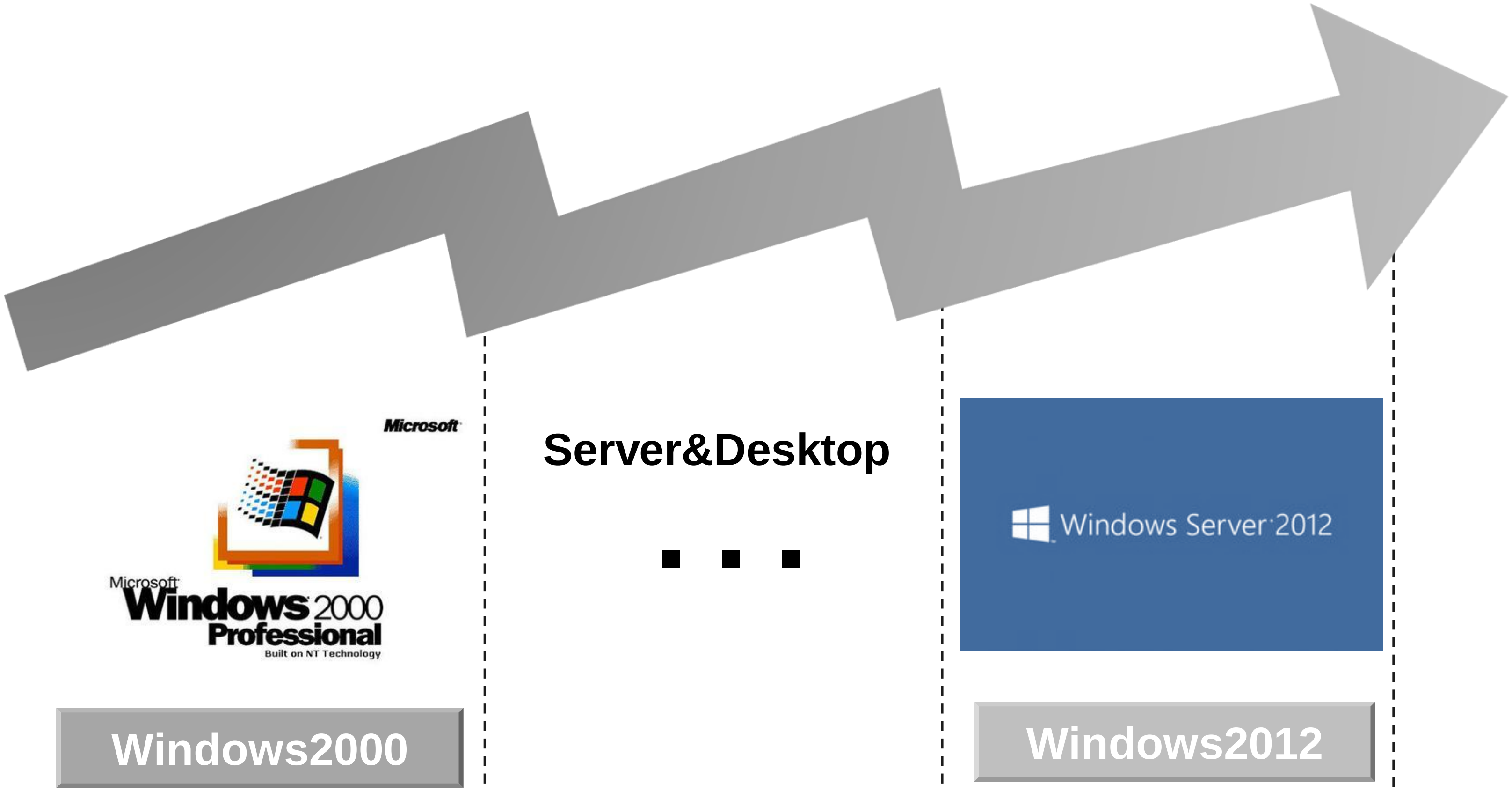
openSUSE

redhat

OS support(Linux)

- Just enough OS: JeOS
  - Based on Fedora
  - Less than 200Mib after compress
  - Average booting time is 5s
  - Customizable functions

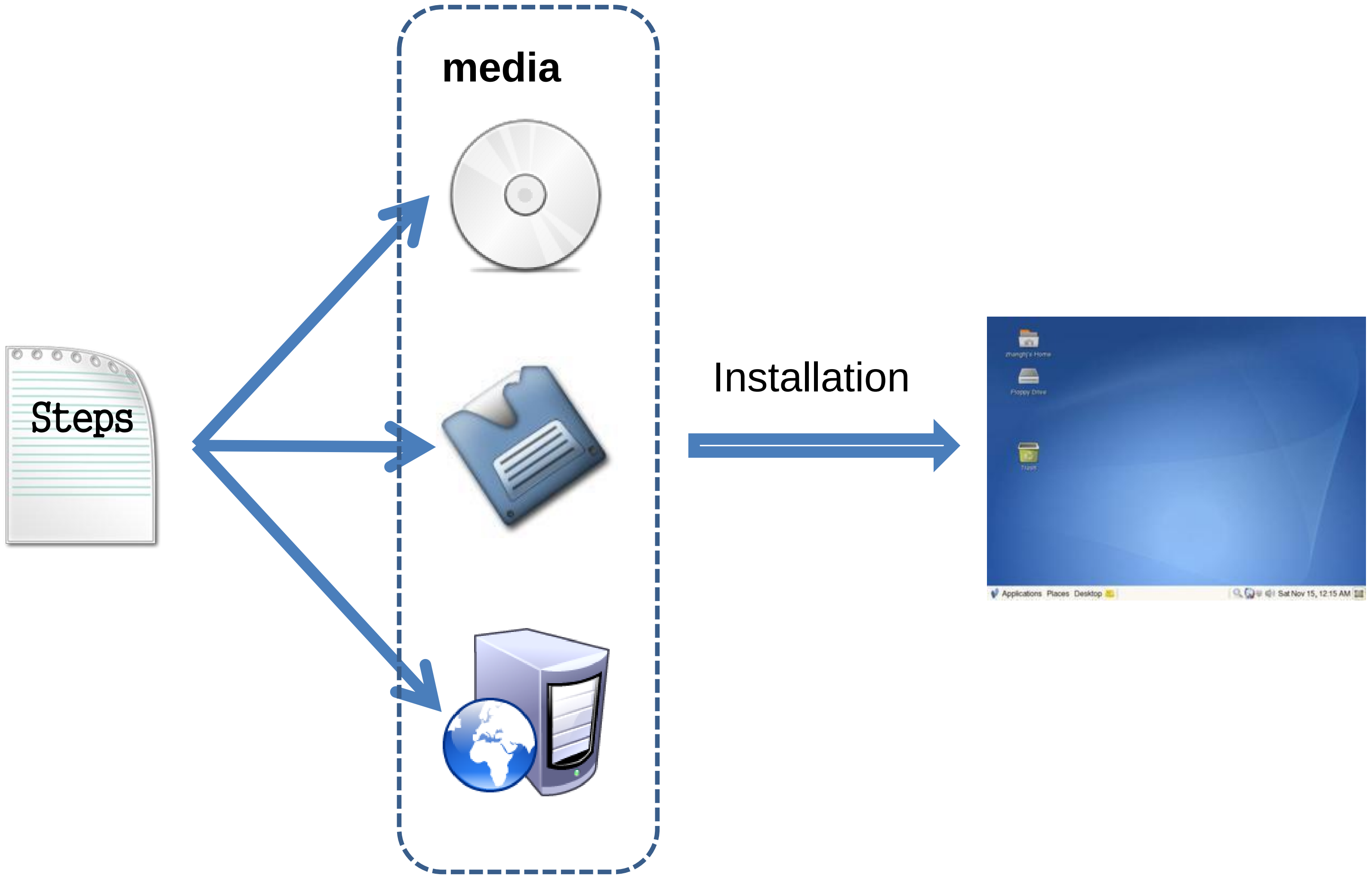
OS support(Windows)



Available tests

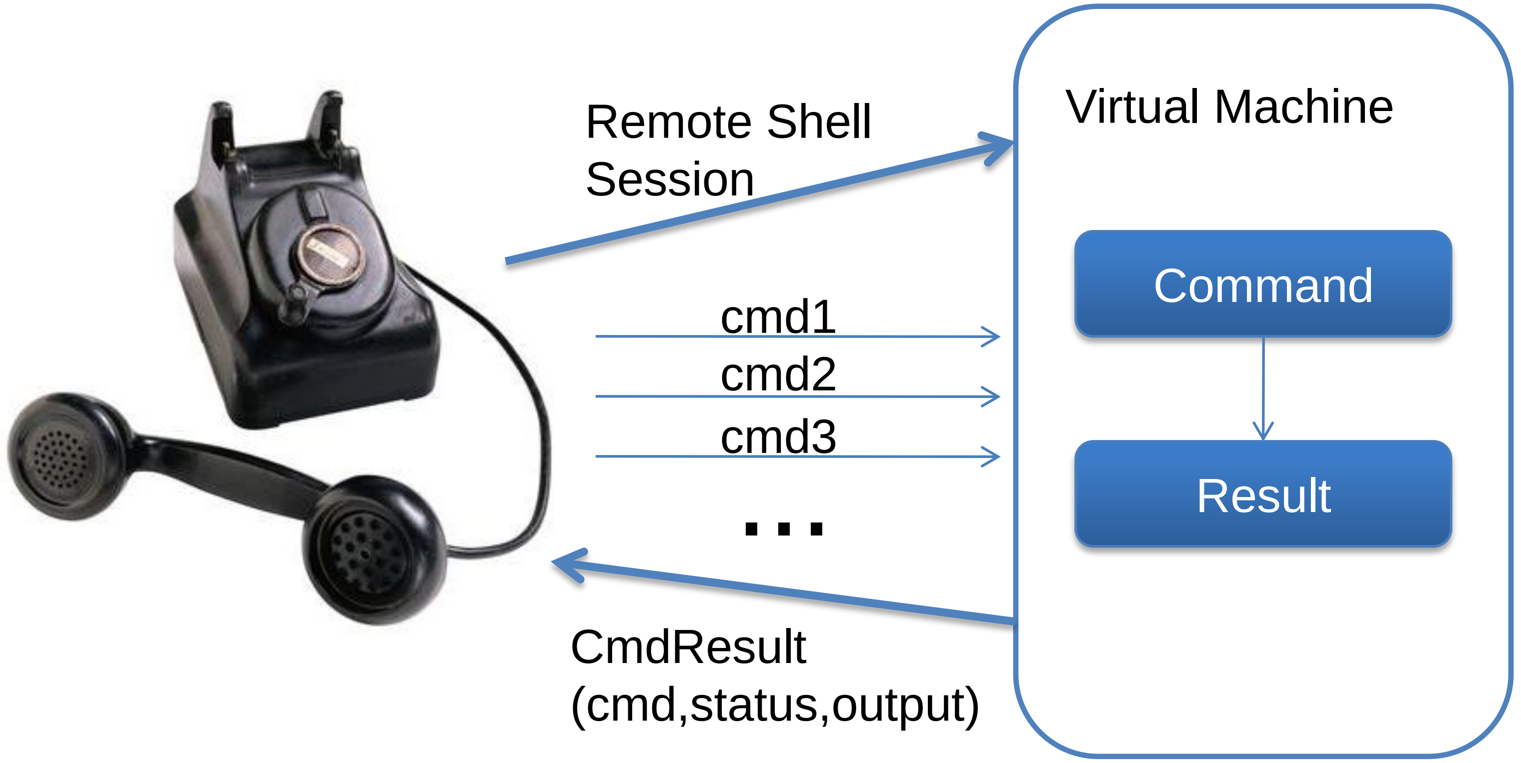
- Qemu-kvm
- Openvswitch
- Libvirt
- Libguestfs
- V2V

Unattended installation



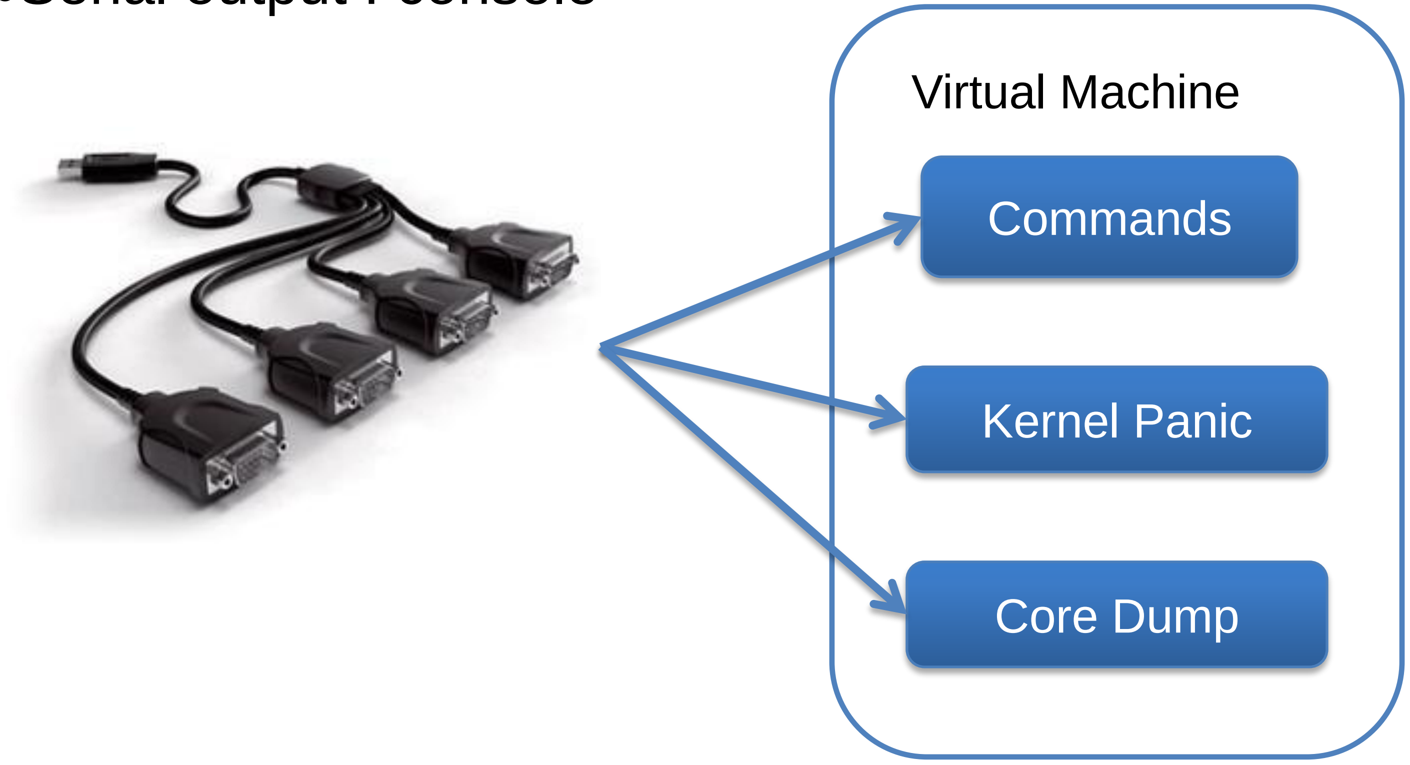
Non-interactive login

- Session output : ssh, nc



Non-interactive login

- Serial output : console



Requirements

OS
- Linux

Virtualization
- qemu-kvm
- libvirt

Packages
- Autotest framework
- Virt-test framework
- Test providers
- tcpdump/nc/p7zip

install →

Hardware

Intel or AMD
VT supported

Agenda

1. What’s Autotest: Overview & Features
2. Virtualization testsuite: Virt-test
   - Why Virt-test
   - Virt-test : Overview & Features
   - Runner : Run tests
   - Provider : Write tests
3. Future work

Virt-test: Runner

Bootstrap
Check requirements

Configurations
Set and Update test parameters

List & Run tests
Plenty options to help tests

Virt-test: Runner

- Bootstrap

  ```
  # ./run  -t libvirt --bootstrap
  ```

  1. Check necessary packages
  2. Download JeOS according need
  3. Create test configurations

- Configurations

  ```
  # ./run  -t libvirt --update-config
  ```

  1. Parameters for installing VMs
  2. Setting for special test

Virt-test: Runner

- List & Run tests

1. Get tests

```
# ./run  -t libvirt  --list-tests
```

2. Run tests

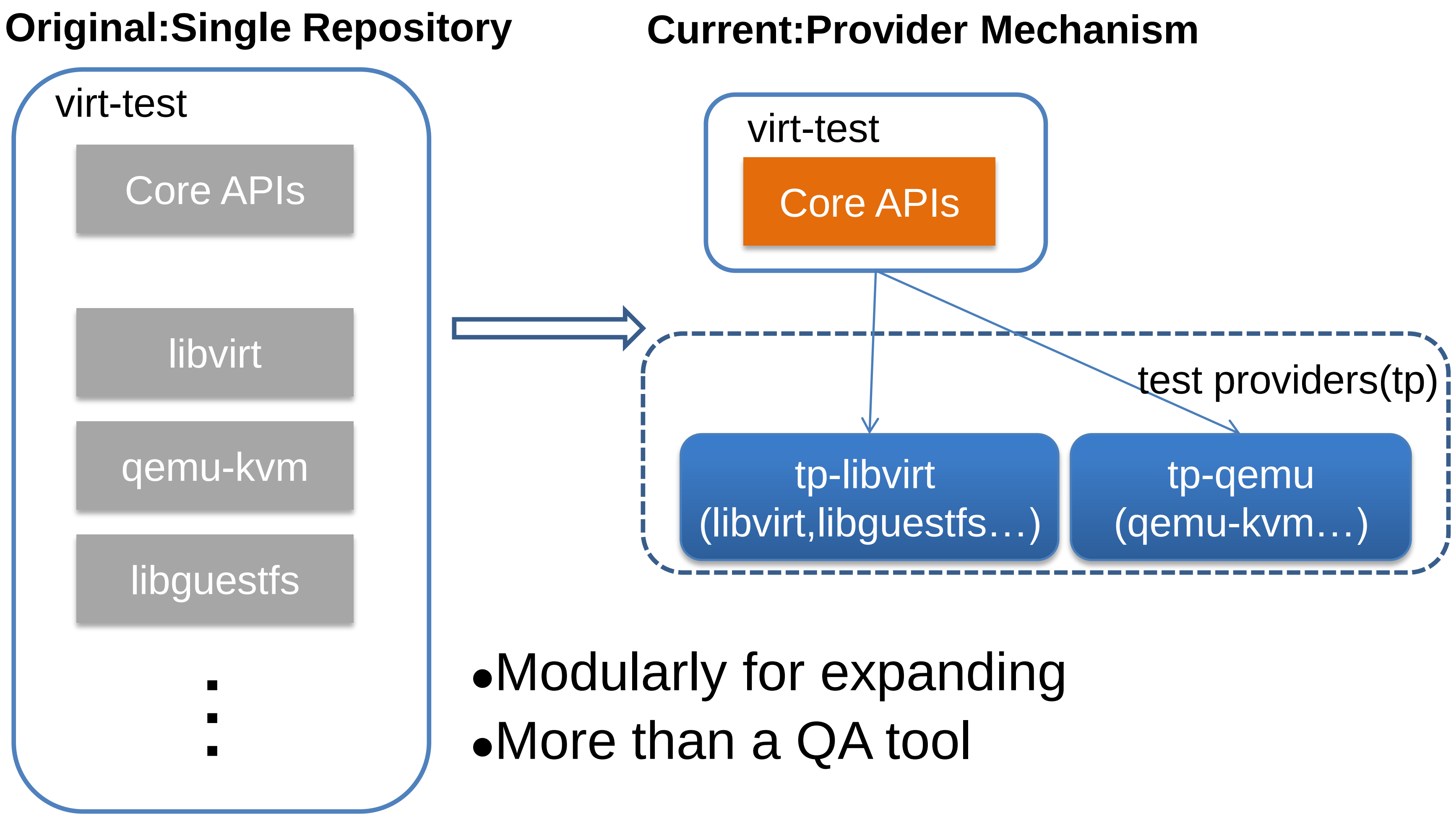
```
# ./run  -t libvirt  --tests “install virsh.list uninstall”
```

PASS FAIL SKIP

Agenda

1. What’s Autotest: Overview & Features
2. Virtualization testsuite: Virt-test
    - Why Virt-test
    - Virt-test : Overview & Features
    - Runner : Run tests
    - Provider : Write tests
3. Future work

Test providers



- Modularly for expanding
- More than a QA tool

Provider Configurations

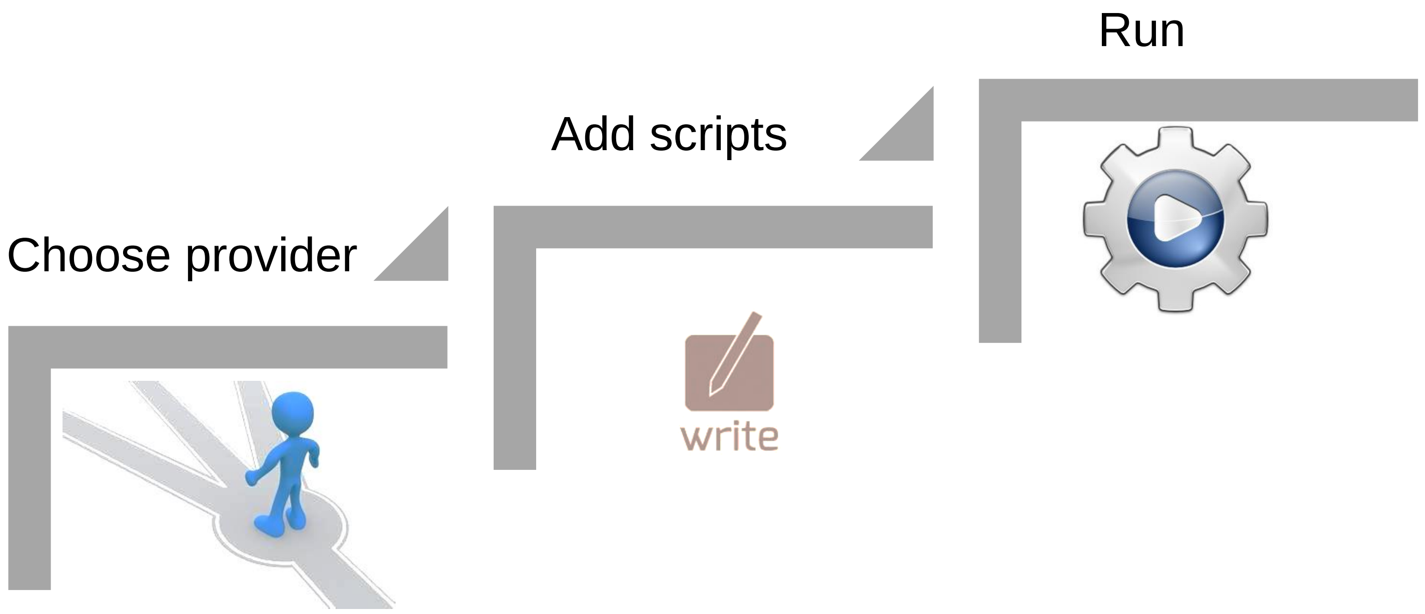
test provider layout

```
.
- backend_1         -> Backend name.
   |-- cfg            -> Test config directory.
   |-- deps           -> Auxiliary files
   |-- provider_lib   -> Shared libraries among tests.
   `-- tests          -> Python test files.
      `-- cfg         -> Config files for tests.
```

```
# Provider URI
[provider]
uri: git://git-provider.com/repo.git
# Directory of backends
[backend]
subdir: foo
```

provider configurations

Add tests(exist provider)



Add tests(new provider)

1. Provider Layout
   - Test scripts
   - Configurations

```
tp_lxc
  |-- lxc
      |-- cfg
      `-- tests
          `-- cfg
```

2. Plug into virt-test
   - provider configurations
   - backend configurations

```
[provider]
uri: git://lxc/tp-lxc.git
[backend]
subdir: lxc
```

```
virt-test
  |-- backends
      |-- libvirt
      |-- lxc
          |--cfg
```

Agenda

1. What’s Autotest: Overview & Features
2. Virtualization testsuite: Virt-test
   - Why Virt-test
   - Virt-test : Overview & Features
   - Runner : Run tests
   - Provider : Write tests
3. Future work

Future work

- A fully Libvirt testsuite
    - Tests for virsh relative commands
    - Tests for libguestfs tools
    - Tests for V2V
- Support more virtualization types
    - Linux Container
- Bug fix & Enhancements

Thank you!

Q&A

Contact

- yumingfei@cn.fujitsu.com
- lmr@redhat.com
- MainPage: https://github.com/autotest/virt-test.git
- Virt test devel list: virt-test-devel@redhat.com

Appendix A

Cartesian Configuration

Statements:

● Keys and values

```
key1 = value1
key2 = value2
  …
```

● Variants

```
variants:
    - block1:
        key1 = value1
        key2 = value2
          …
```

Blocks Relationship
AND : block1..block2
FOLLOWED-BY :
block1.block2

Example:

```
main_vm = vm1
variants:
    - domname:
        vm_ref = domname
    - domid:
        vm_ref = domid
variants:
    - running:
        start_vm = yes
    - shutoff:
        start_vm = no
variants:
    - normal_test:
```

two cases: normal_test..domname
one case: normal_test.running.domid

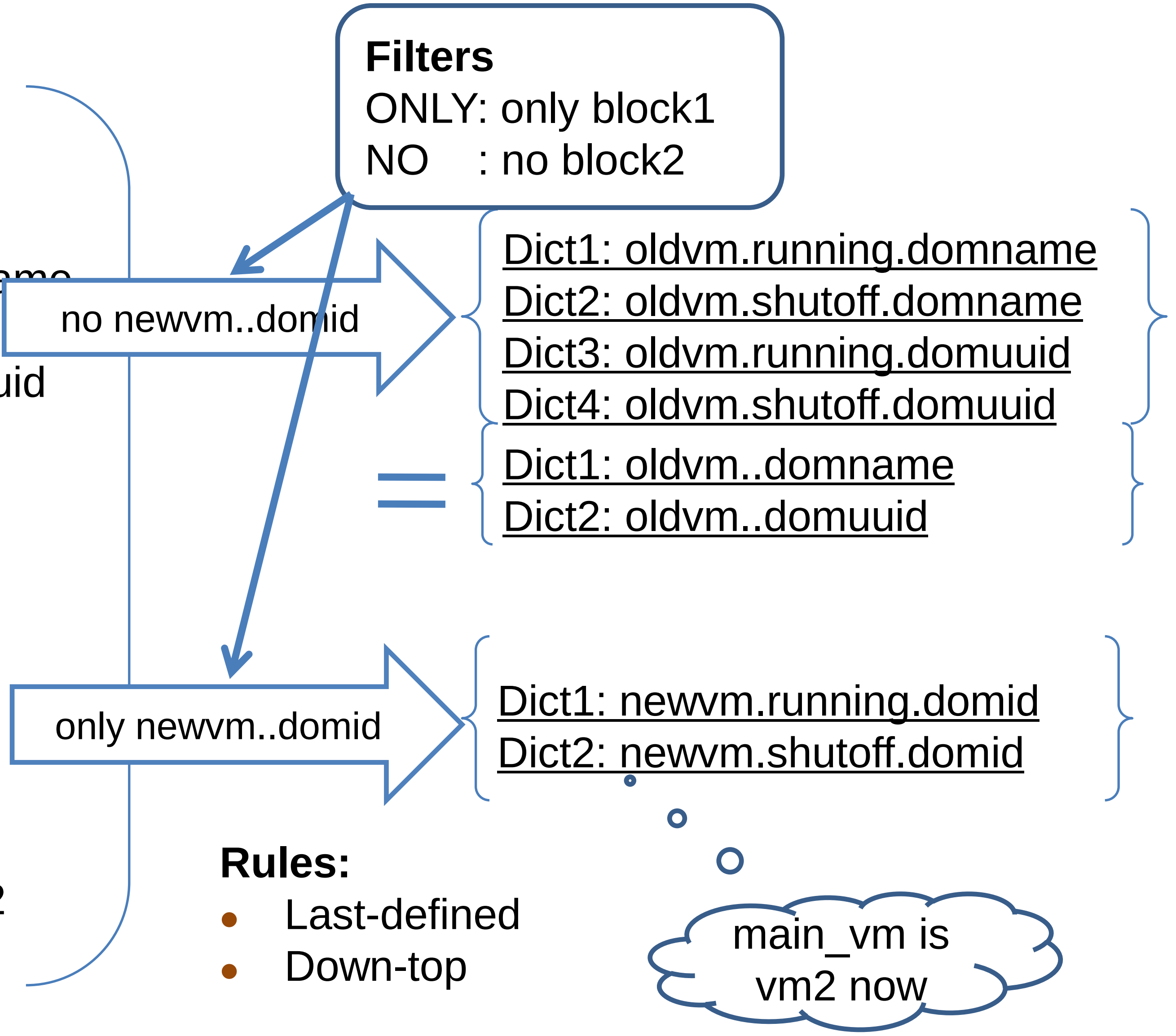
Appendix A

Cartesian Configuration

Example:

```
main_vm = vm1
variants:
    - domname:
        vm_ref = domname
    - domuuid:
        vm_ref = domuuid
    - domid:
        vm_ref = domid
variants:
    - running:
        start_vm = yes
    - shutoff:
        start_vm = no
variants:
    - newvm:
        main_vm = vm2
    - oldvm:
```

Filters
ONLY: only block1
NO : no block2



Dict1: oldvm.running.domname
Dict2: oldvm.shutoff.domname
Dict3: oldvm.running.domuuid
Dict4: oldvm.shutoff.domuuid

Dict1: oldvm..domname
Dict2: oldvm..domuuid

Dict1: newvm.running.domid
Dict2: newvm.shutoff.domid

Rules:
- Last-defined
- Down-top