Jenkins: Humble servant of the Quality Assurance department

Using Jenkins as the backbone of a testing infrastructure

Petr Chytil & Michal Vaněk

AVAST Software a.s.

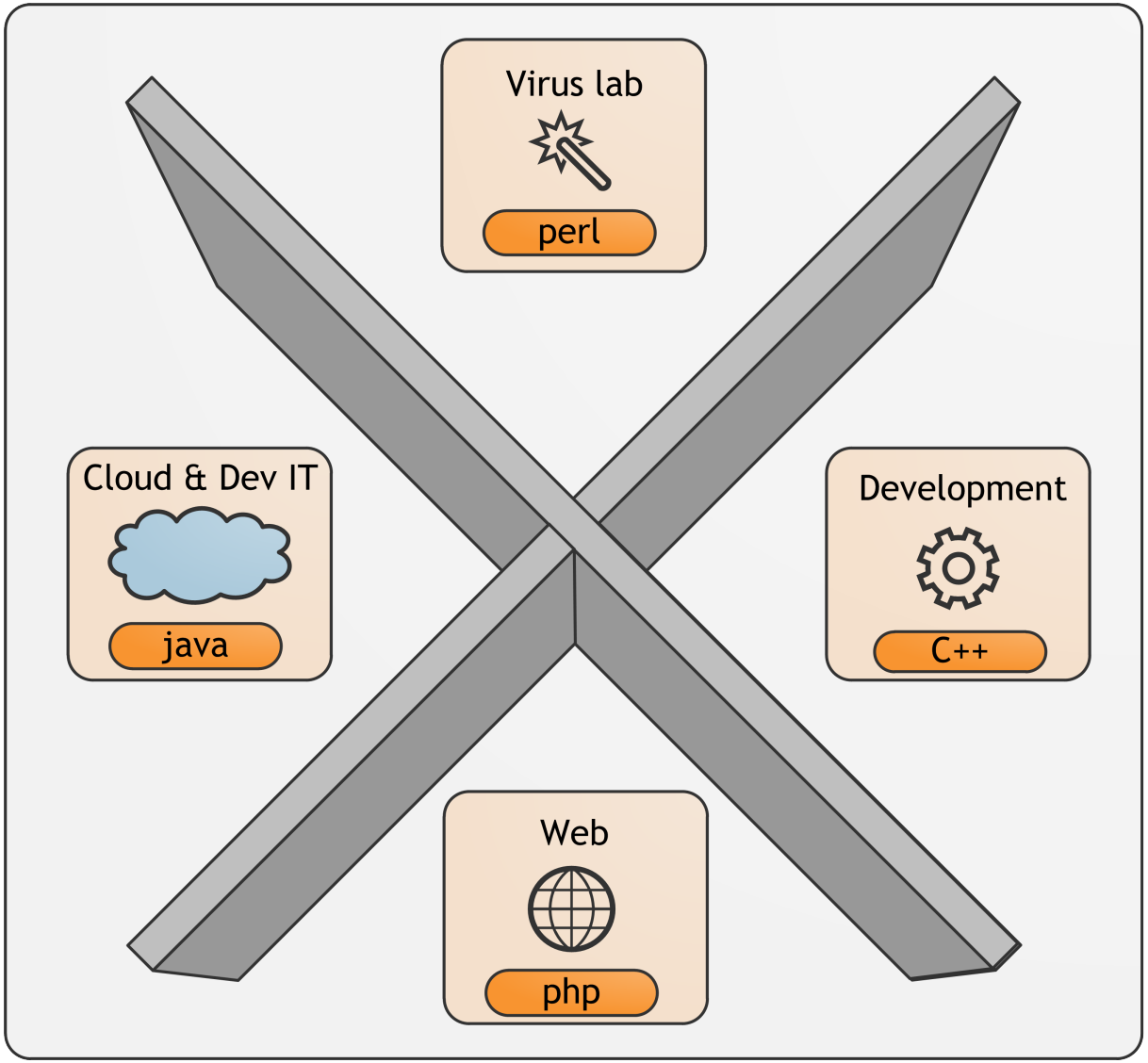
http://www.avast.com

avast! be free

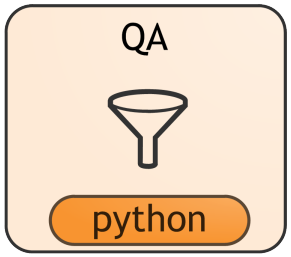
Case Study - Agenda

- What do we test at avast!
- How do we test it
  - Local execution
  - Remote execution
  - The Cloud
- Conclusion

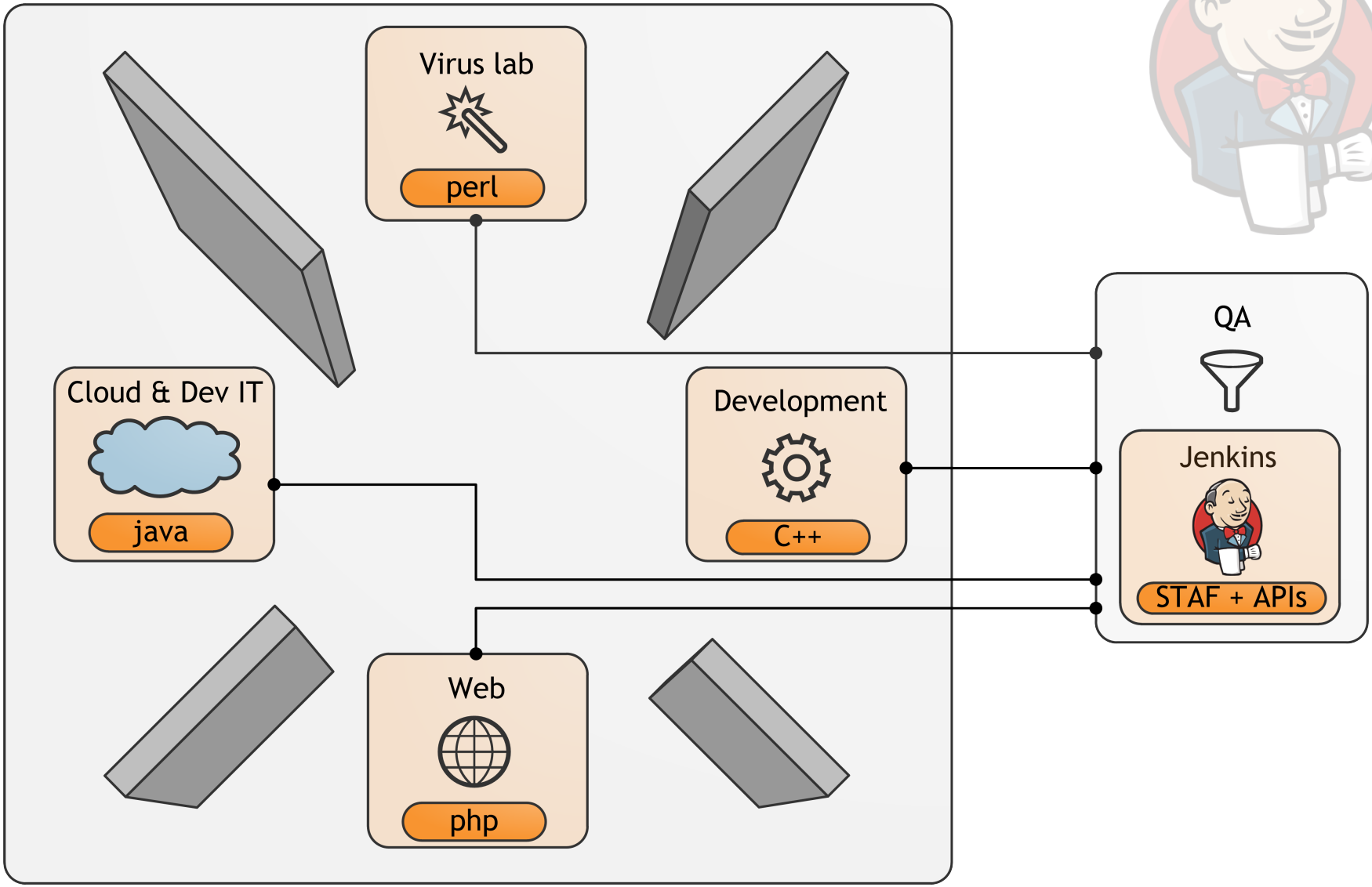
Department map



?



Department map with our friend, Jenkins



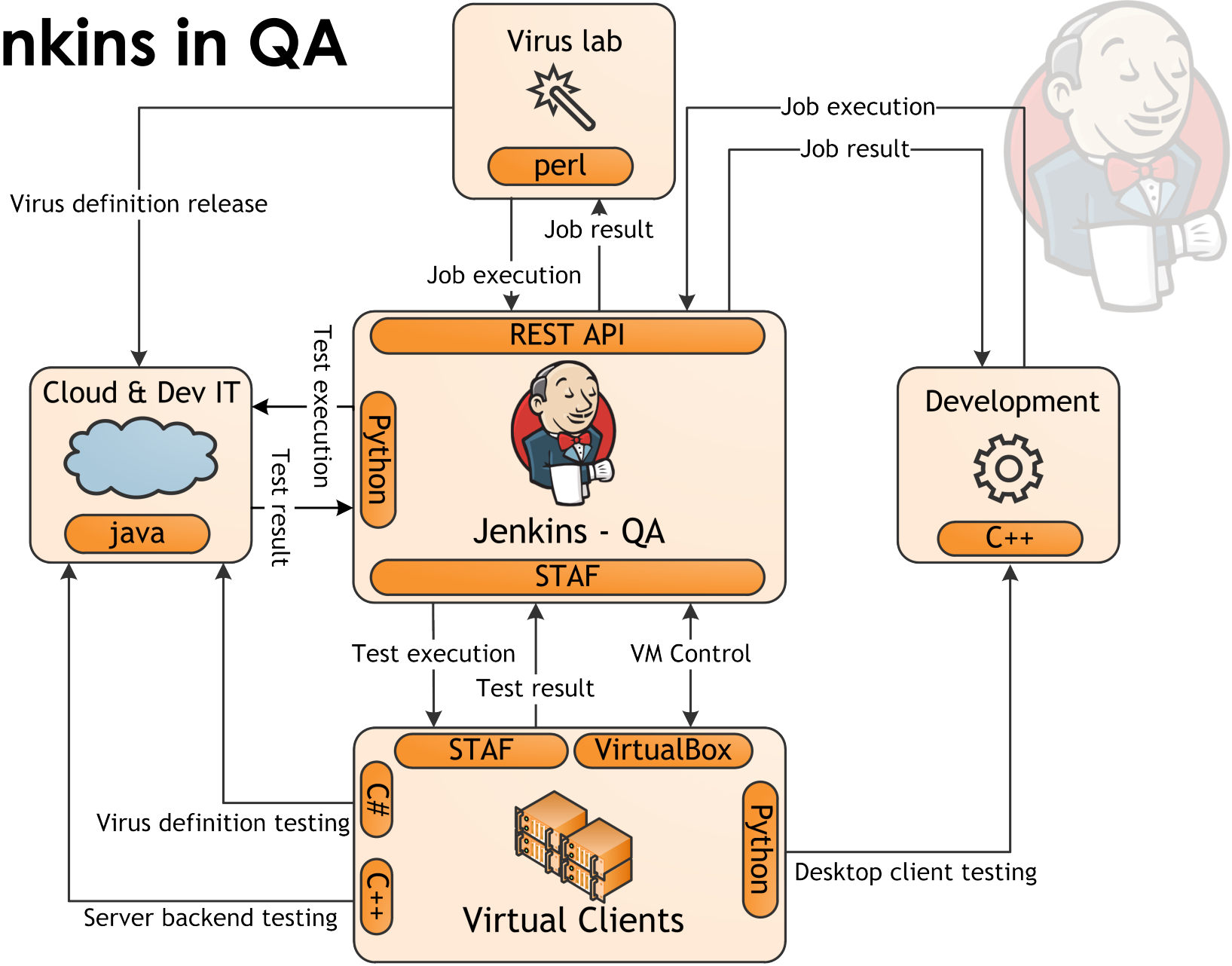
Test automation tasks

- Testing web services with REST API
- Server backend testing
- Functional testing of avast! antivirus desktop applications on multiple OS versions
  - Sanity suite
  - Smoke suite
  - Regression suite
  - Upgrade suite
- Virus definition validation
- Cloud features tests

Three different approaches

- Local execution
  - Web services and REST API
- Remote execution
  - Functional testing of antivirus desktop application
  - Virus definition validation
- Local cloud
  - Cloud features testing

Jenkins in QA



Local execution

- Web services and REST API
- Server infrastructure backend round tests
- Everything can be done out of the Jenkins build step
- Tests implemented in
  - Python
  - Robot
  - C++/C#

Local execution – example

- Job configuration
  - Add build parameters
    - `SERVERS=192.168.1.3 192.168.1.4`
  - Set build trigger
    - `@hourly`
  - Configure build step
    - `add execute shell`
    - `trigger child job execution`
  - Publish test results
    - `jUnit`
    - `Robot Framework plugin`

Local execution – example

- Test script example

testRating.py:

```
class  Test_getRatingTestCase(unittest.TestCase):

    def test_getRating(self):
        ratingData = 'www.avast.com'
        rating = int(getRating(ratingData, WebRepServer))
        expectedRating = 100
        assert rating == expectedRating
if __name__ == '__main__':
    unittest.main()
```

- Jenkins Execute shell

```
for server in ${SERVERS}; do
  ./testRating.py -s ${server} -l ${WORKSPACE}/
WebRepRating_${server}_log.xml
```

Local execution – test results evaluation

- jUnit – pyUnit

☑ Publish JUnit test result report

Test report XMLs

Fileset 'includes' setting that specifies the generated raw XML report files, such as 'myproject/target/test-reports/*.xml'. Basedir of the fileset is the workspace root.

☑ Retain long standard output/error

- Robot framework Jenkins plugin

☑ Publish Robot Framework test results

Directory of Robot output: results

Path to directory containing robot xml and html files (relative to build workspace)

Log/Report link

Name of log or report file to be linked on jobs front page

Thresholds for build result

% 90.0

% 100.0

☑ Use thresholds for critical tests only

Remote execution

- Goals of remote execution
  - Multi environment testing
    - Same tests, different OS versions
    - Need for virtualization
    - One machine cannot host enough virtual machines
  - Remote process execution
    - Virtual machine control
    - Test execution
    - Test result gathering

Remote execution – solution

- Executor:
  - Jenkins
- Virtualization technology:
  - VirtualBox
- Remote execution technology:
  - STAF framework
    - Used for remote VirtualBox machine control
    - Allows remote test execution and results gathering

(STAF)

Remote execution – STAF and VirtualBox configuration

- Install STAF framework on both host and guest machines
  - `All in one installer`
- Configure staf.cfg on guest machine, set trust machine option
  - `trust machine tcp://10.0.2.2 level 5`
- Set port forwarding on VirtualBox machine for STAF ports
  - `From VirtualBox settings`
  - `Forward STAF port 6500 to other port on a host machine`

Remote execution – example

- Start remote virtual machine

```
$> staf $IP process start command  "VBoxManage
startvm $MACHINE_NAME -  type headless"
```

  – Alternative: Jenkins VirtualBox plugin?

- Run remote tests

```
$> staf $IP process start command  "nosetests /
tests/*.py " WAIT  RETURNSTDOUT STDERRTOSTDOUT
```

Remote execution – example

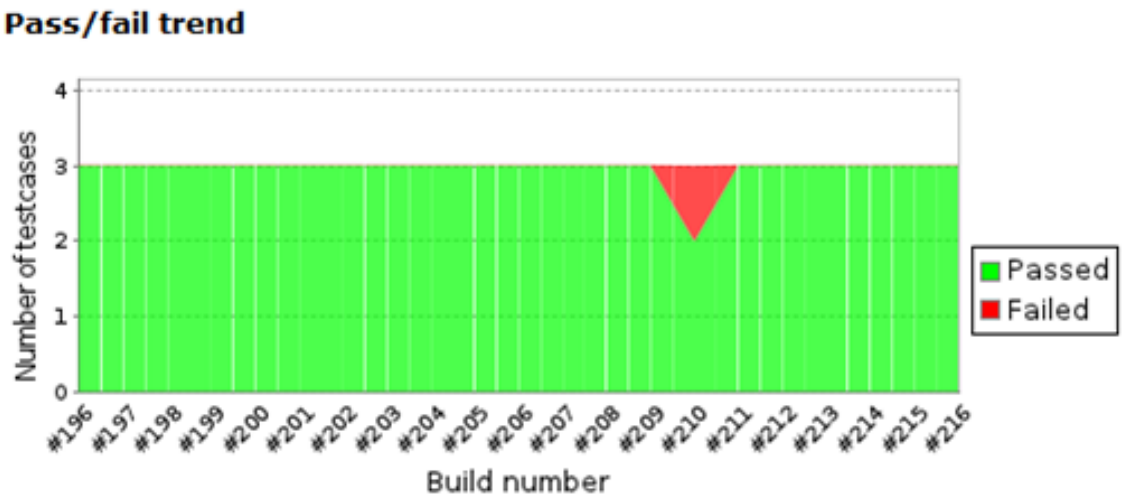
- 1) revert to a clean snapshot and start the virtual machine on a remote computer
  - `staf $IP process start command "VBoxManage startvm $MACHINE_NAME -type headless"`
- 2) wait
  - `wait_for_machine_start.py $IP 300`
- 3) execute tests directly on the remote virtual machine
  - `staf $IP process start command "/tests/runTests.cmd " WAIT RETURNSTDOUT STDERRTOSTDOUT`
- 4) test result evaluation
  - `staf $IP FS copy directory '\test_results' todirectory $WORKSPACE`

Remote execution - evaluation

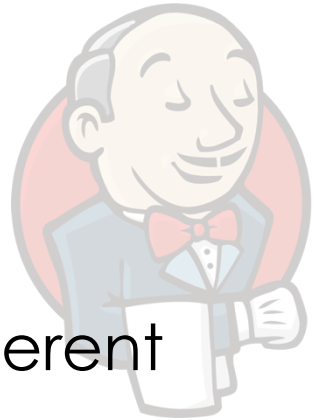
- Evaluate the jUnit, pyUnit or Robot framework test results in the Jenkins job

All Testcases

Name
Registration
Free Express Install with Chrome
Verify Avast Installation



Remote execution – job hierarchy



- Requirement of multiple products and on different OSes
- Solution: One Master job and multiple Child jobs

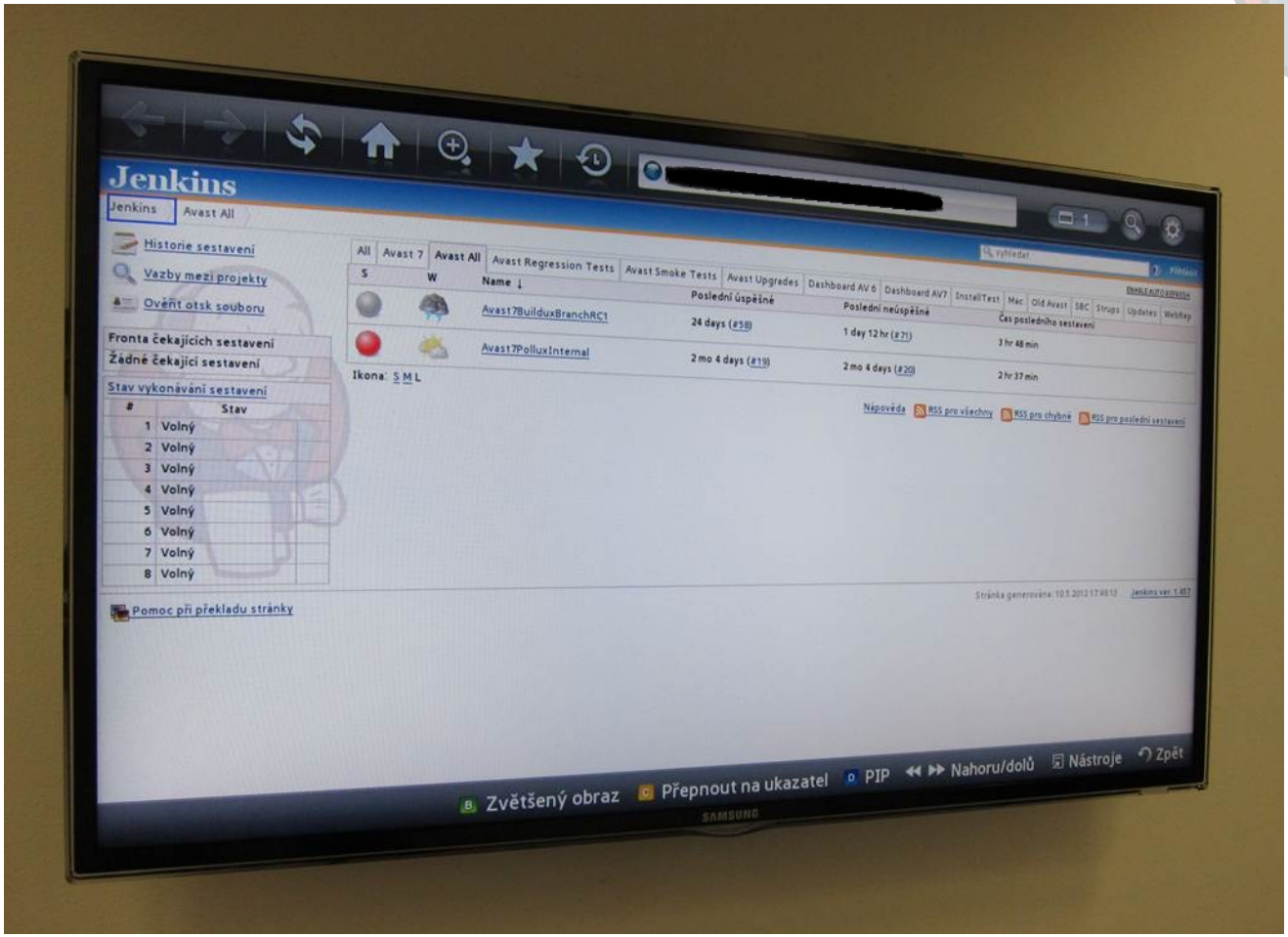
- Master job
  - Application of Parameterized Trigger Plugin
  - Trigger child job as a build step
- Child jobs with parameters
- Re-using child jobs in various master jobs
  - One common child job for desktop app installation
  - Different set of tests executed from different master jobs
  - Testing of staging/production builds

Remote execution – jobs hierarchy, test results evaluation

- Request to be able to see all failed tests in all jobs of the hierarchy
- Dashboard view
  - Unable to display job hierarchy
- Downstream build view
  - Incompatible with Parameterized Trigger plugin

What's your solution?

Remote execution – test result presentation



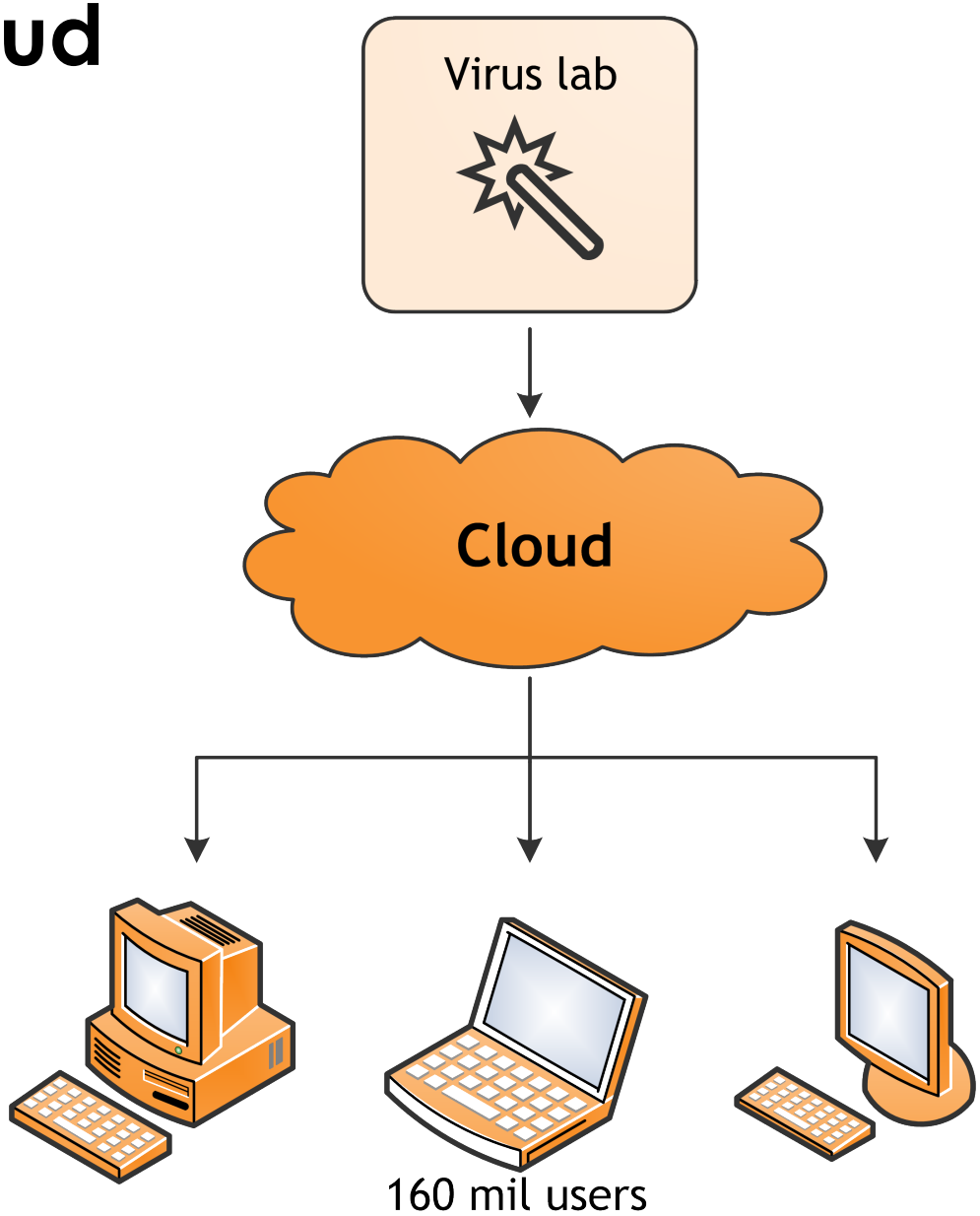
Remote execution – Jenkins slave as an alternative solution

- Jenkins slave node instead of STAF framework
- More complicated Jobs execution on snapshot reverted virtual machines
- Cannot reboot slave machine during job run
- Higher system performance requirements than STAF in Desktop environment

The Cloud

- Goal
  - Integrate smoke tests into existing viruslab infrastructure
  - Build infrastructure implemented in perl
  - Every 30 minutes, small package of virus definitions is distributed to 160 milion users
    - But, do not break their computers

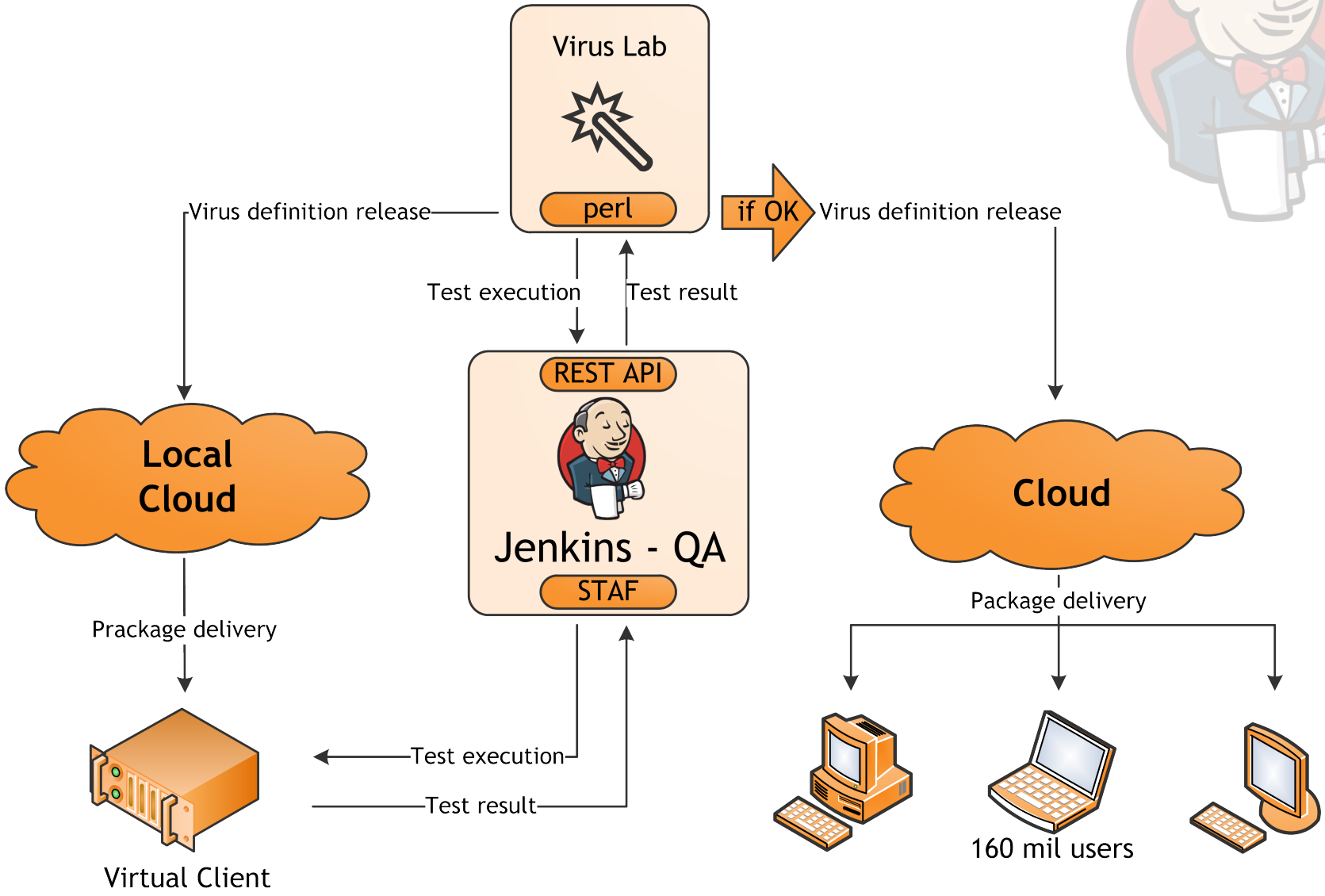
The Cloud



The Cloud - solution

- Combination of everything that we have learned
- Remote test execution
- Integration of Jenkins job into the perl build script using Jenkins REST API

The Cloud - solution



The Cloud – REST API in perl example

```
my $srvurl = 'http://our.jenkins.server:8080', $job = 'OurJobName';
my $security_token = 'SomeSecretStuff', $uagent = new LWP::UserAgent;
my $req = HTTP::Request->new(GET => "$srvurl/job/$job/build?token=$security_token");

# Get next build number
my $job_req = HTTP::Request->new(GET => "$srvurl/job/$job/api/json");
my $job_res = $uagent->request($job_req);
my $job_info = from_json( $job_res->content );
my $next_build_number=$job_info->{"nextBuildNumber"};

# Run the job on Jenkins
my $res = $uagent->request($req);
if (not $res->is_success) { die "Failed to run the jenkins job: ".$res->status_line."\n"; }
sleep(10); # Wait for the job to start (job execution in jenkins takes a while)

# Get the status of the job in a loop until result is found
my $status_req = HTTP::Request->new(GET => "$srvurl/job/$job/$next_build_number/api/json");
my $build_result, $finished = 0;
while ( $finished != 1 ) {
  my $status_res = $uagent->request($status_req);
  my $build_info = from_json( $status_res->content );
  if ( defined($build_info->{"result"}) ){
    $build_result = $build_info->{"result"};
    $finished = 1;
  }
  sleep(6);
}
```

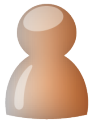
The Cloud – Job Hierarchy

- Every 30 minutes, Jenkins master job is executed by the perl script, package gets validated and is released to users
- The master job runs set of child jobs on various OS versions
- Child jobs evaluate test results by parsing their jUnit results
- Perl build script continues only if the master job status is SUCCESS

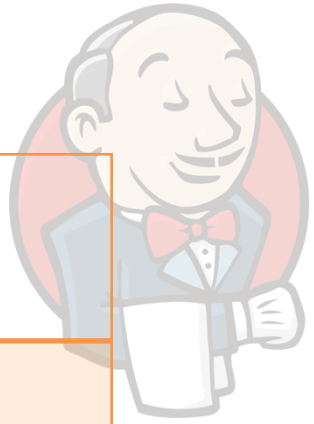
Conclusion

- Jenkins allows even functional testing of desktop applications or distant cloud systems
- It is worth executing even simple tests by Jenkins
- All this you get for free
  - Test result view
  - Scheduled job triggering
  - Integration using Jenkins API
- Easy integration with other development departments without infrastructure changes

Jenkins is your humble servant!



Thank You To Our Sponsors



Platinum Sponsor	CloudBees
Gold Sponsors	JFrog, CLOUDANT
Silver Sponsors	LIFERAY, SendGrid Email Delivery. Simplified., CLOUDSMITH
Bronze Sponsors	WAN DISCO

Links

- http://www.avast.com
- http://staf.sourceforge.net
- https://www.virtualbox.org
- http://jenkins-ci.org
- http://code.google.com/p/robotframework
- http://pyunit.sourceforge.net