Viri

Remote execution of Python scripts

Every time you use Viri, God kills a sysadmin

Marc Garcia

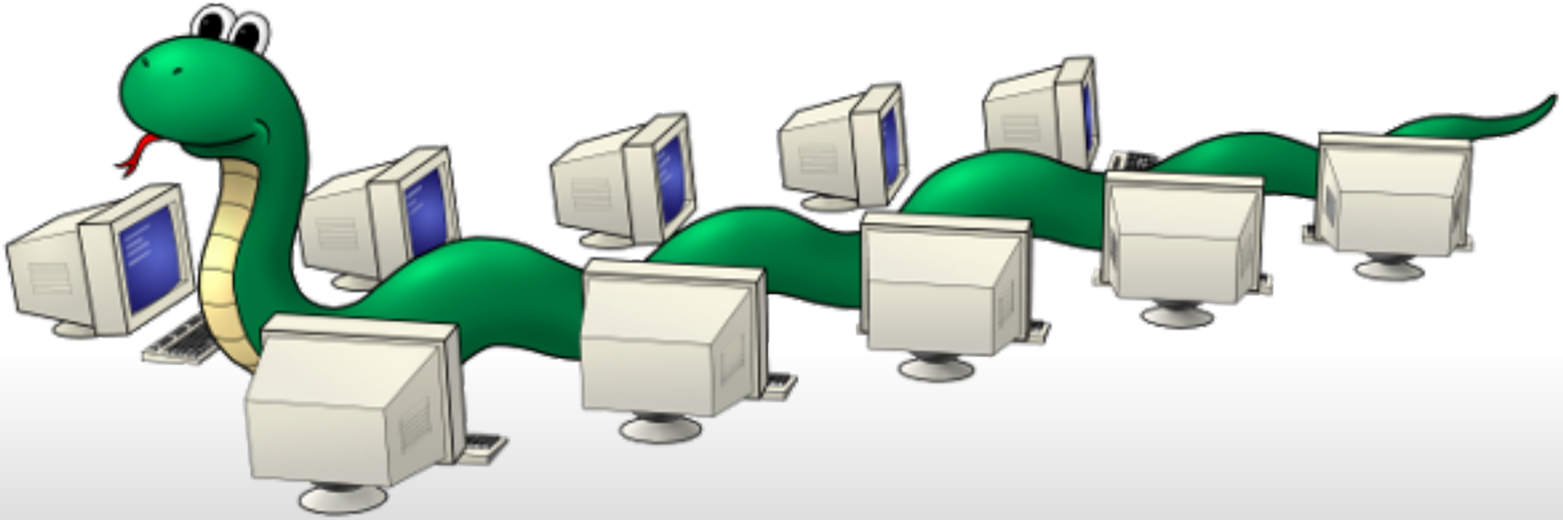
Overview

- **What is Viri?**
  - What is Viri?
  - What Viri does?
  - What can Viri be useful for
- **How is Viri?**
  - Technical overview
  - Viri components
  - Is Viri secure?
- **Using Viri**
  - Viri scripts
  - Basic Viri commands
  - Basic Viri options
  - Base script
  - Scheduling
- Project Status
- So, will God really kill sysadmins?

What is Viri?

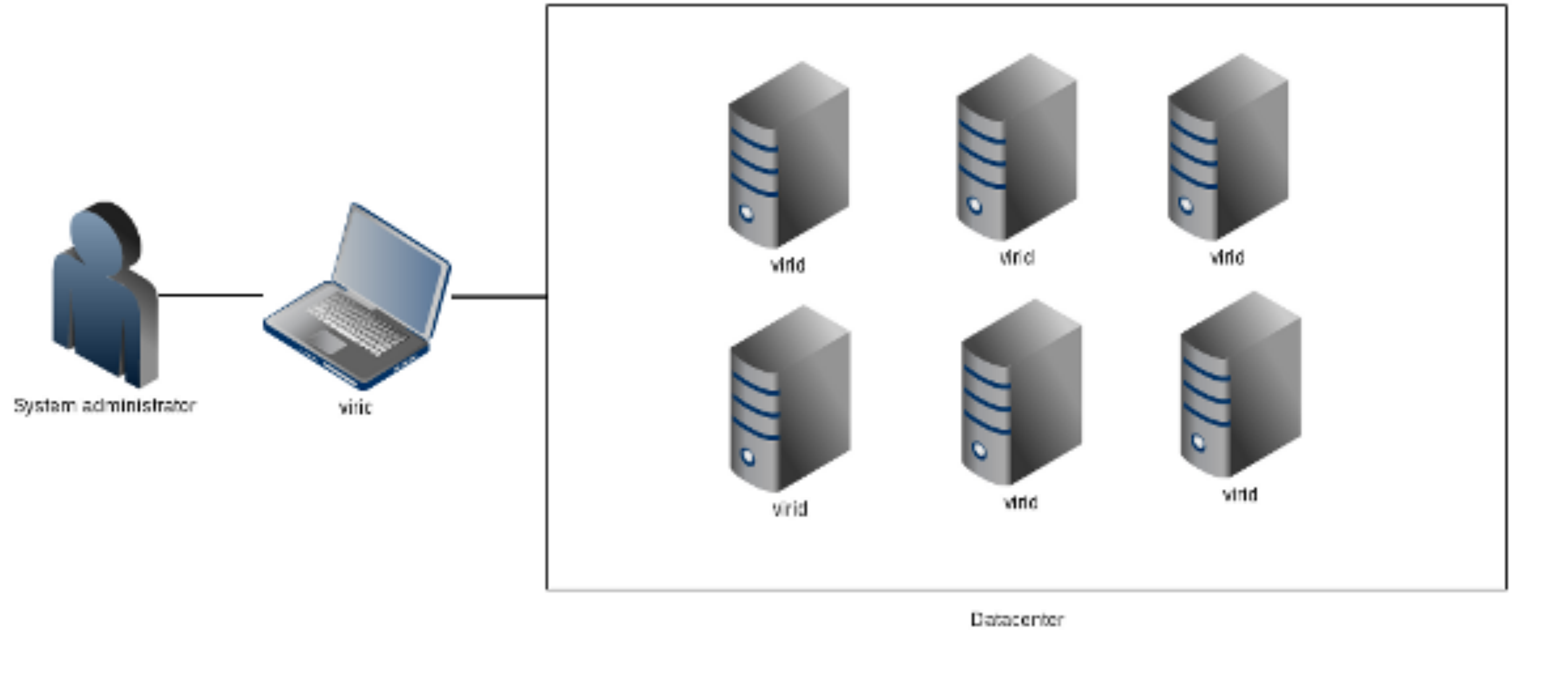
What is Viri?

In short: Viri is an application to administrate datacenters (hosts) using **Python**.



What Viri does?

- Automation of tasks over a large set of computers using **Python scripts**.
  - ○ Script deployment
  - ○ Transfer of required data files
  - ○ On-demand or scheduled execution
  - ○ Recording execution history



What can Viri be useful for?

Real world examples:

- **Gather system data and send to a central location**:
  - System (Architecture, OS, etc)
  - Network (IP addresses, networks, etc)
  - User access
  - Log information
  - *Whatever you can get from a Python script*
- **Implement actions that require per host operations**:
  - Add users to all hosts /root/.ssh/authorized_keys
  - Changes to network configuration
  - Complex operations:
    - Download source tarball from Internet
    - Compile code
    - Perform benchmarking tests
    - Submit results to a website

How is Viri?

Technical overview

- **Python 3.1**
- No other dependencies
- XML-RPC for client/server communication
- TLS for communications
- Custom Cron server

- Viri is 100% Python, so it is multiplatform.
- Python 3.1 is not available for most UNIX systems.
- Viri has to provide Python 3 packages where not available.
- So far, Viri is distributed for:
  - Debian 6
  - RHEL/CentOS 5
  - *Other systems soon*

Viri components

virid

Daemon running on remote hosts

- Receives scripts and data
- Records history
- Returns results
- Controls exceptions

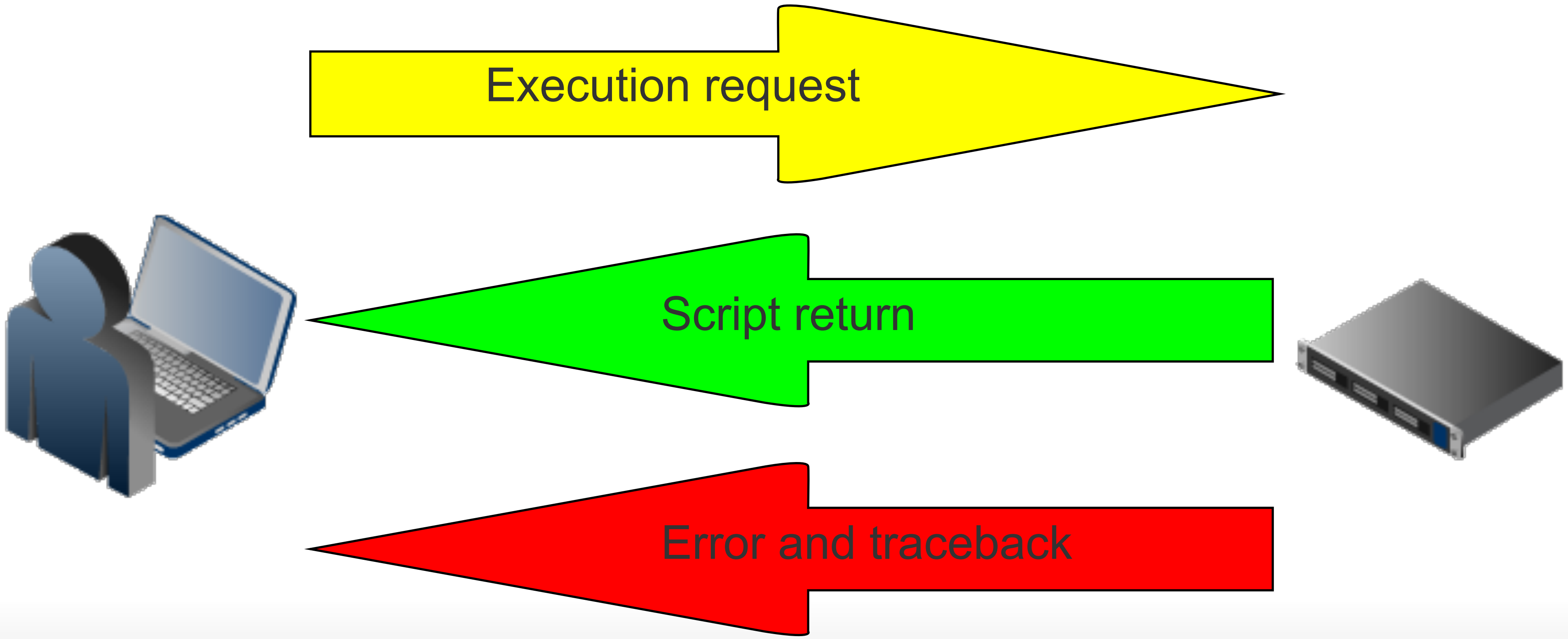
viric

User interface, command line utility.

```
viric execute test.py --host=10.0.0.9
```

Can be integrated with third-party apps.

Execution workflow



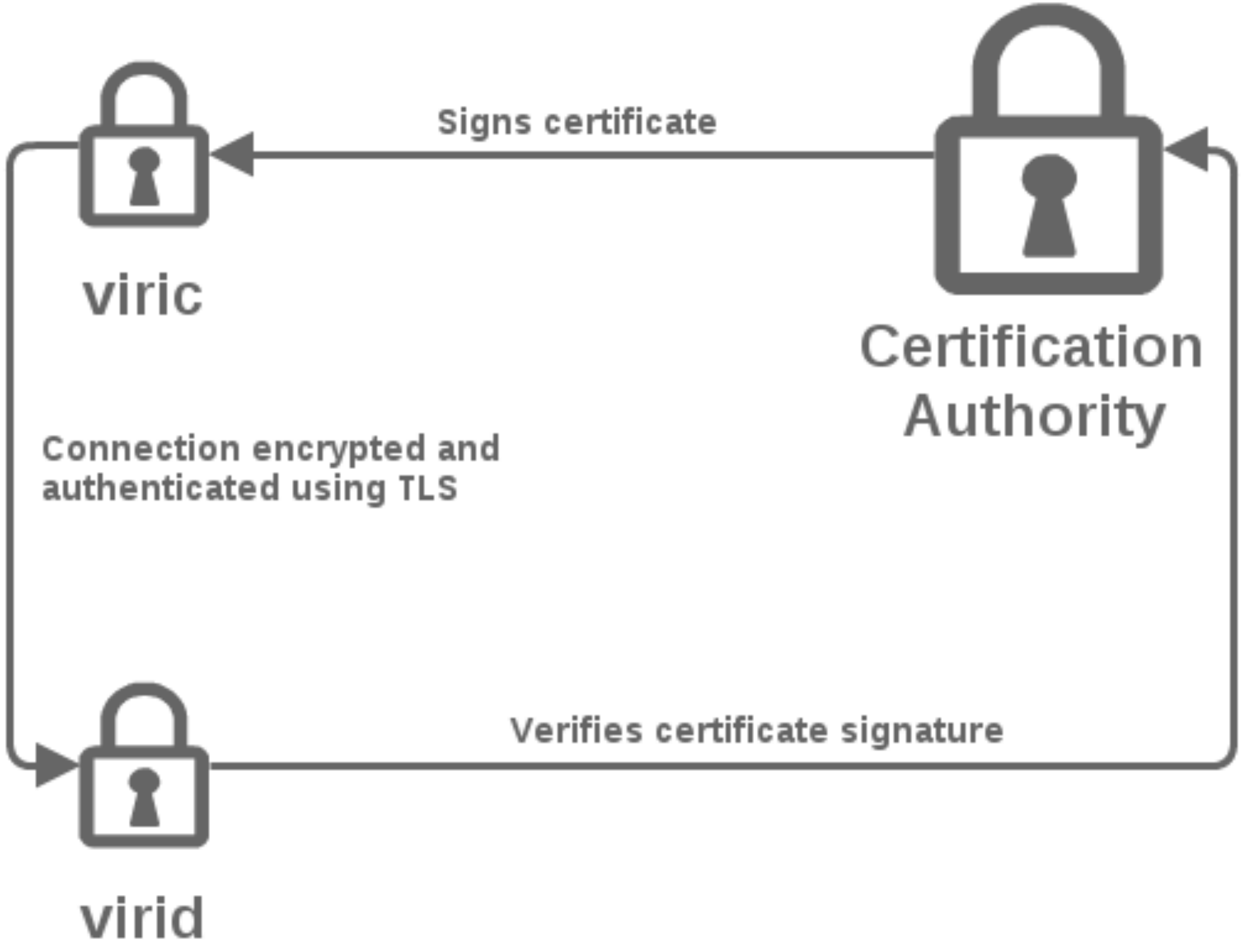
Execution request

Script return

Error and traceback

Is Viri secure?

Communication is **encrypted** using TLS.
Viri daemon requires **authentication** using a PKI.


Signs certificate
viric
Certification Authority
Connection encrypted and authenticated using TLS
Verifies certificate signature
virid

Using Viri

Viri scripts

```
import os

class ViriScript:
    hello_file = '/tmp/viri.hello'

    def say_hello(self):
        with open(self.hello_file) as f:
            f.write('Viri was here!\n')

    def run(self):
        if not os.path.isfile(self.hello_file):
            self.say_hello()
            return 'Viri said hello'
        else:
            return 'Viri has already been here'
```

Basic viric commands

viric COMMAND [OPTIONS]

- help
  - Show usage information
- ls
  - Show installed scripts
  - Show copied data files
- put
  - Send scripts / data files
- get
  - Downloads scripts / data files
- **execute**
  - Executes a script

Basic viric options

viric COMMAND [OPTIONS]

- --host
  - Remote host IP or domain
- --port
  - Remote port (Default is 6808)
- --data
  - On some commands like ls, put or get, specifies that the operation is for data files instead of scripts.

Base script

Special __base__.py *script*:

```
class ViriScript:
    def custom_log(self, msg):
        with open('/tmp/viri.custom_log', 'a') as f:
            f.write('%s\n' % msg)
```

```
./viric put __base__.py --host=10.0.0.9
```

All scripts inherit from it:

```
class ViriScript:
    def run(self):
        # do something
        self.custom_log('I did something')
```

Scheduling

Special __crontab__ *data* file:

```
./viric put --data __crontab__ --host=10.0.0.9
```

Cron syntax (using script id):

```
# daily at midnight
0 0 * * * 99154c826fca745be859c6481a5f87631e4b2b78

# Just once, on January 1st, 2015 at 9:00
0 9 1 1 * 2015 99154c826fca745be859c6481a5f87631e4b2b78
```

Project Status

Viri is in **BETA** status:

- Fully functional
- Unit tests
- Documentation
- Existing packages:
  - Debian 6
  - RedHat/CentOS 5
- Not widely used **yet**

Future plans

- More packages: Windows, FreeBSD, Solaris, etc
- SQLite for virid data management
- Performance revision
- Man page
- Contribute XML-RPC code back to Python

So, will God really kill sysadmins?

No! He will convert them in Python ninjas, and they will write excellent Viri scripts.