VirtShell
Framework para aprovisionamiento de soluciones virtuales

Carlos Alberto Llano R.

Escuela de Ingeniería de Sistemas y Computación
Maestría en Ingeniería con énfasis en Ingeniería de Sistemas y Computación

Director:
John Alexander Sanabria

August 25, 2016

Contents

Objetivos

General

Diseñar un framework web, que permita el aprovisionamiento de software automático, para ambientes virtualizados.

Objetivos

General

Diseñar un framework web, que permita el aprovisionamiento de software automático, para ambientes virtualizados.

Específicos

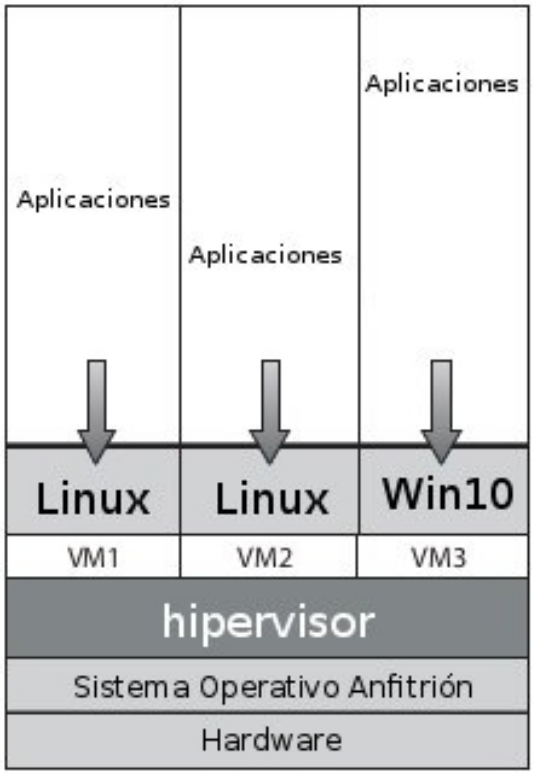
1. Evaluar diferentes técnicas y soluciones de aprovisionamiento que se utilizan en la actualidad.
2. Evaluar diferentes mecanismos de aprovisionamiento.
3. Realizar una ejemplificación del framework.

Técnicas de Virtualización trabajadas

Contenedores

Técnicas de Virtualización trabajadas

Contenedores



Máquinas Virtuales

Soluciones de aprovisionamiento evaluadas

- Fabric
- Chef
- Puppet
- Juju
- CFEngine
- Bcfg2
- Ansible
- Cobbler
- SmartFrog
- Amazon EC2
- Docker composer
- SaltStack
- Vagrant

Soluciones de aprovisionamiento evaluadas

Solución	Sop.Nubes	Curv.Apren	Crea	Aprov	API REST	Mult.SO
Chef	Todas	Alta	✓	✓	✓	✓
Juju	OpenStack MAAS	Baja	✓	✓	✗	✗
Puppet	La mayoria	Media	✗	✓	✗	✓
Ansible	Amazon OpenStack	Baja	✗	✓	✗	✓
Amazon EC2	Amazon	Media	✓	✓	✓	✓
Docker	✗	Media	✓	✓	✓	✓
Vagrant	✗	Baja	✓	✓	✗	✓

Selección del mecanismo de aprovisionamiento

Se evaluaron diferentes formas de escritura de scripts de aprovisionamiento

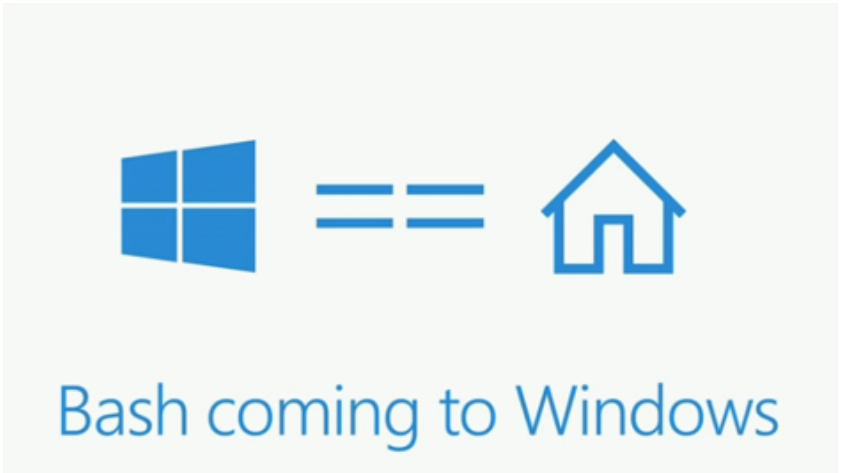
- json
- xml
- yaml
- archivos texto
- lenguajes de programación (ruby, python, etc.)
- bash

Selección del mecanismo de aprovisionamiento

Selección del mecanismo de aprovisionamiento


```
#!/bin/bash
```

Selección del mecanismo de aprovisionamiento



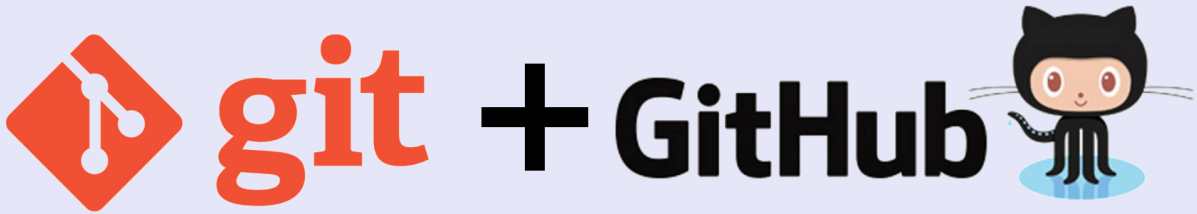
Selección del mecanismo de aprovisionamiento

Se evaluaron diferentes formas de guardar los scripts en el sistema

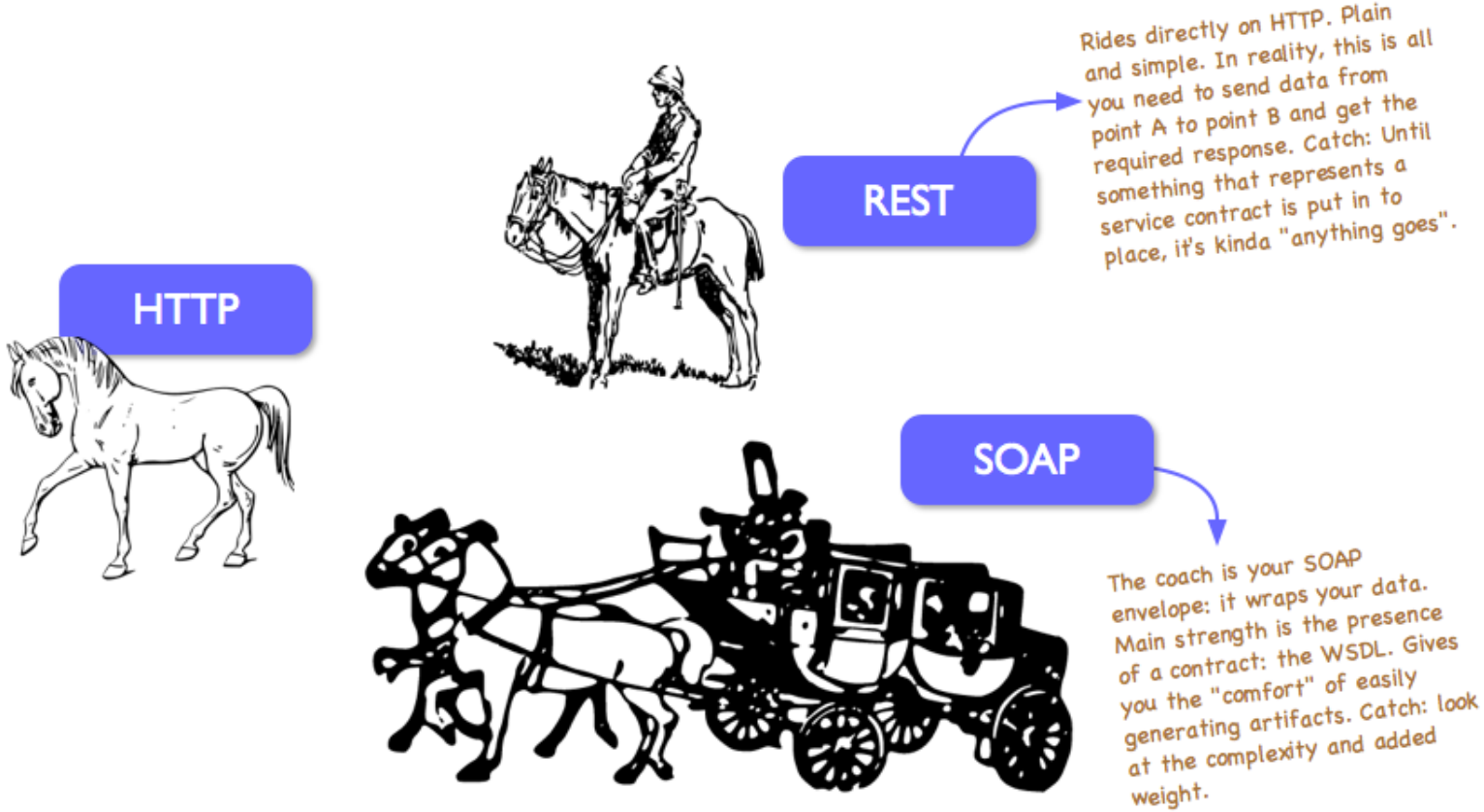
- repositorio central propio?
- enviarlo en el momento del aprovisionamiento?

Selección del mecanismo de aprovisionamiento

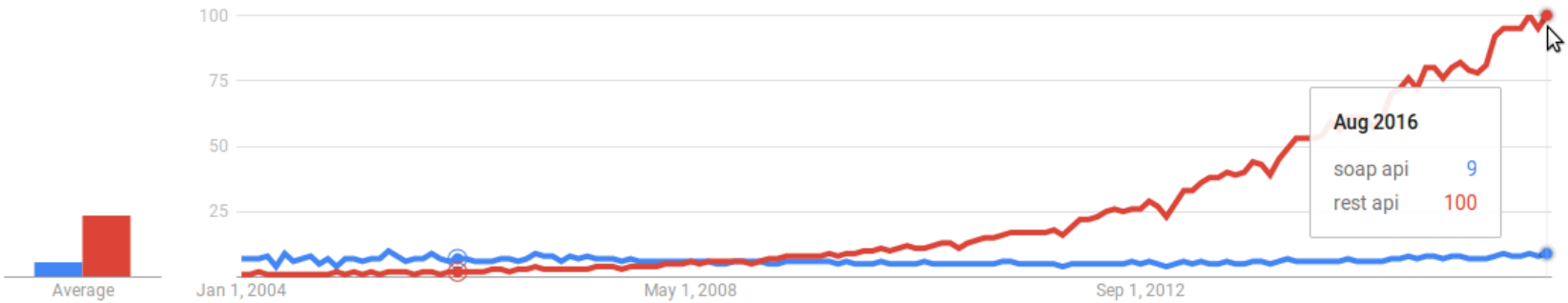
Solución



Evaluación y selección de un estilo arquitectural



Evaluación y selección de un estilo arquitectural

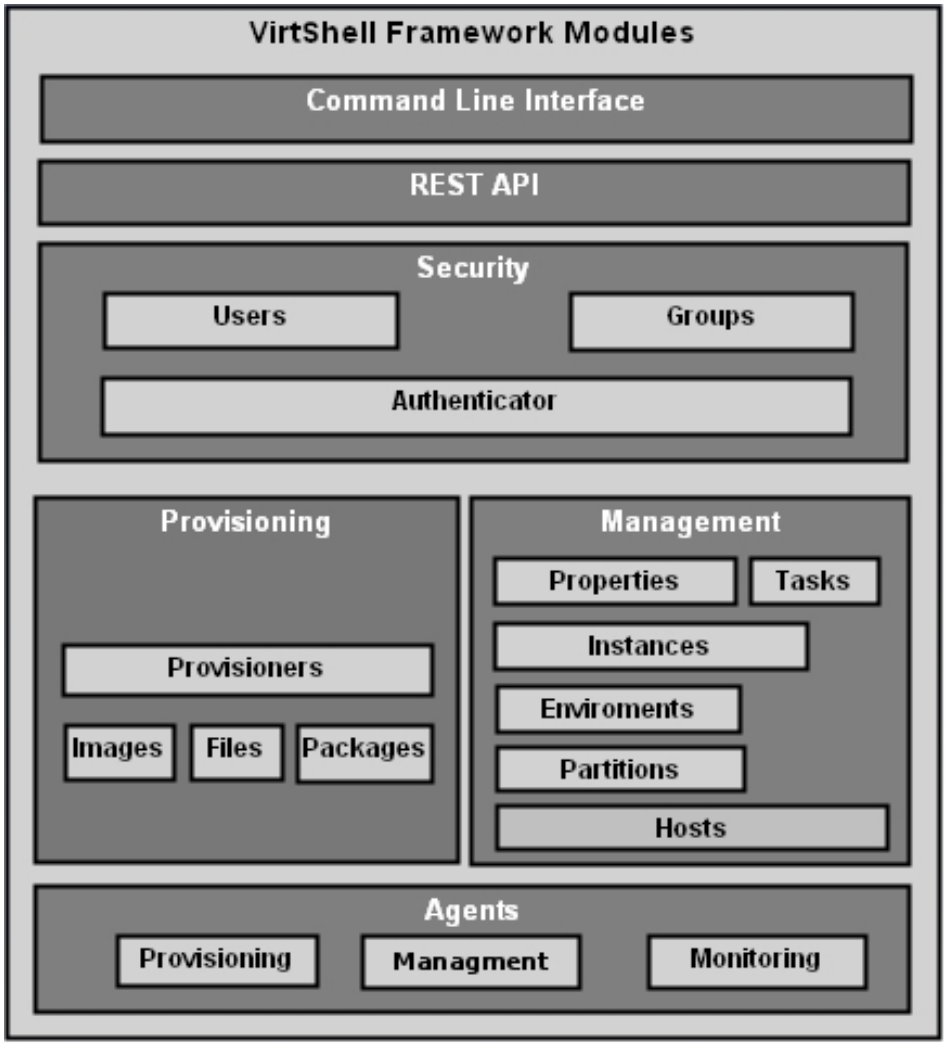


Fuente: Google Trends

Evaluación y selección de un estilo arquitectural

Fuente: Google Trends

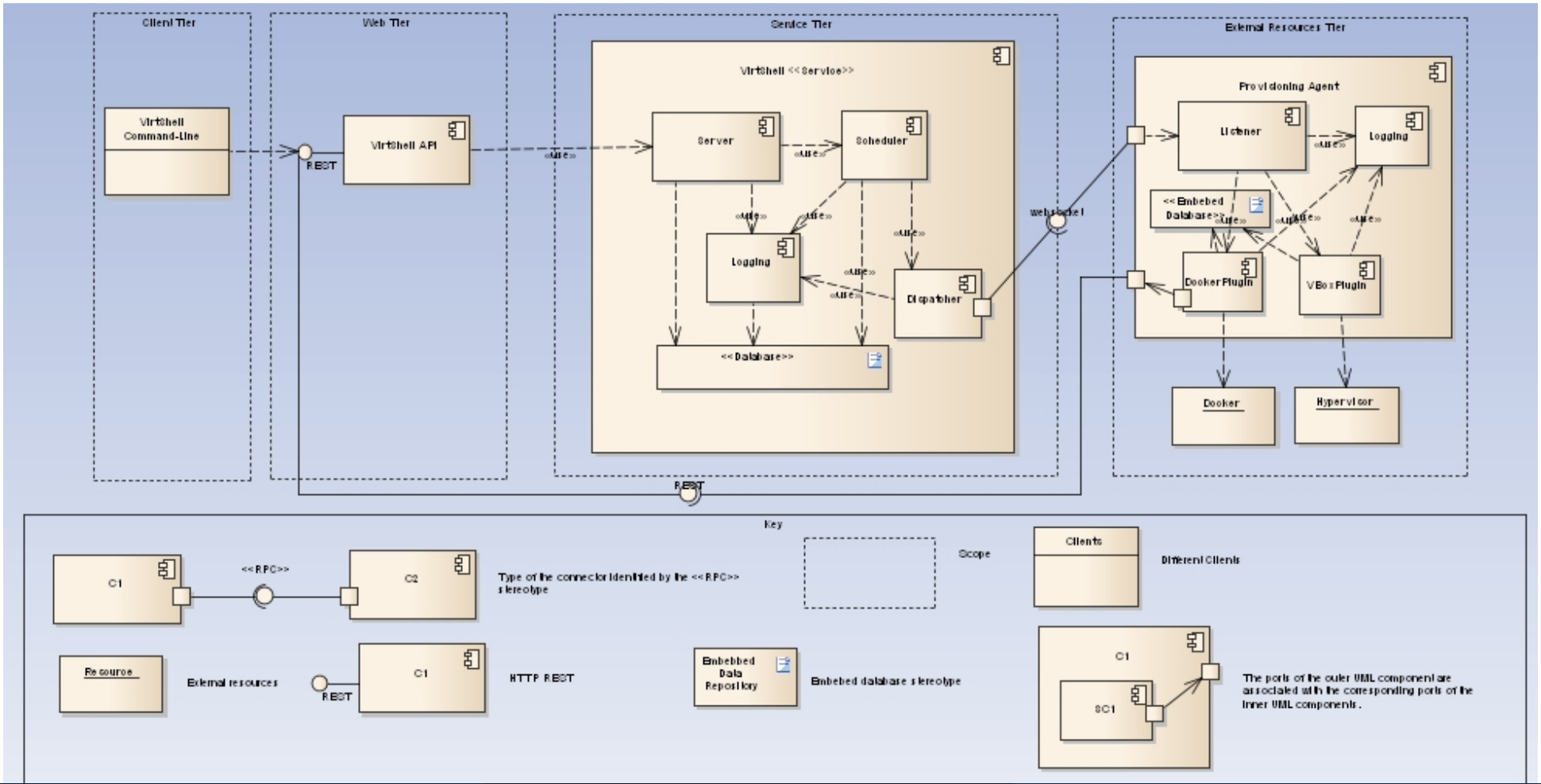
Framework


VirtShell Framework Modules
Command Line Interface
REST API
Security
Users
Groups
Authenticator
Provisioning
Provisioners
Images
Files
Packages
Management
Properties
Tasks
Instances
Enviroments
Partitions
Hosts
Agents
Provisioning
Managment
Monitoring

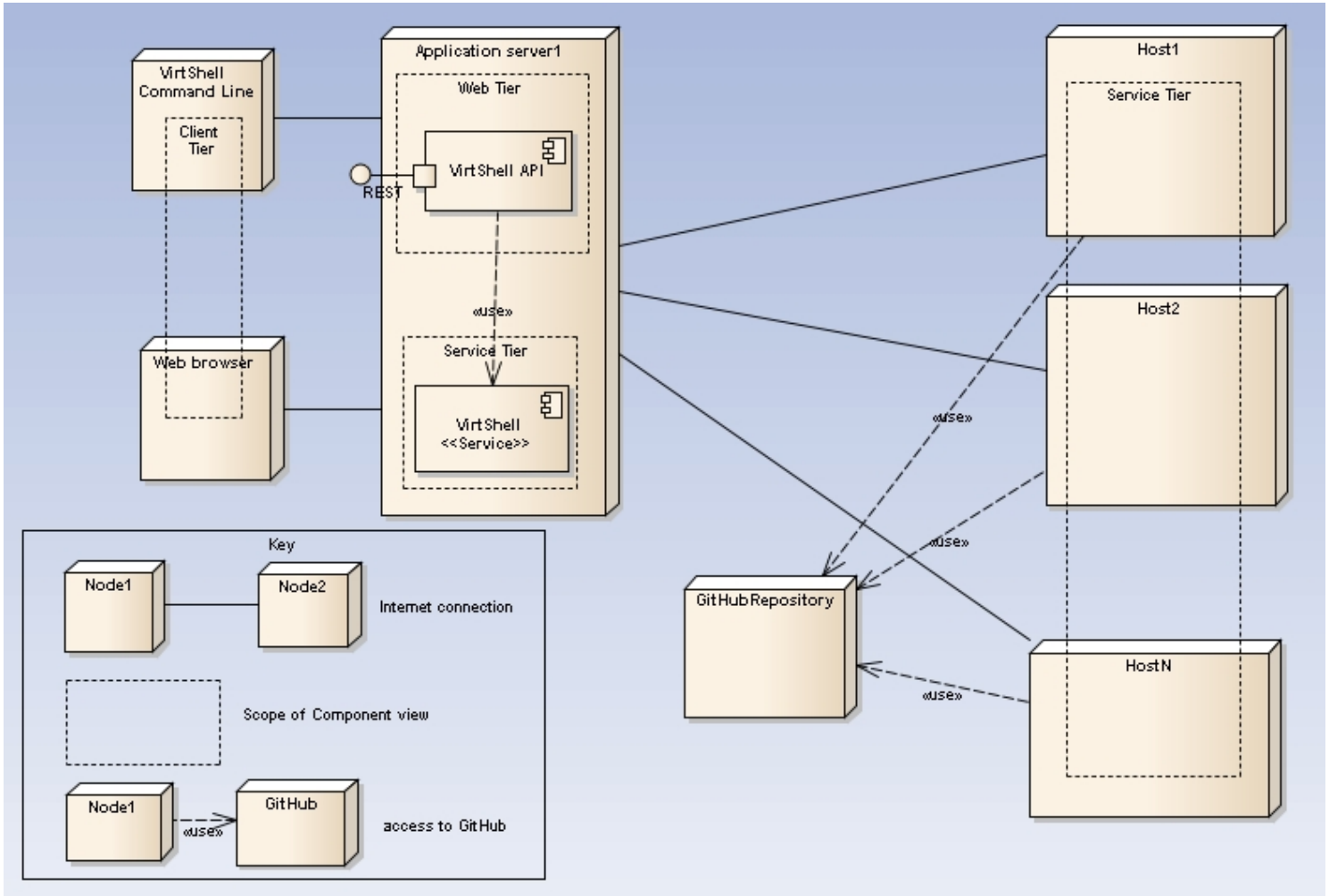
Características del Framework

- Programable
- Repetible
- Modular
- Seguro
- Extensible
- Inyección de dependencias virtuales
- Interoperable

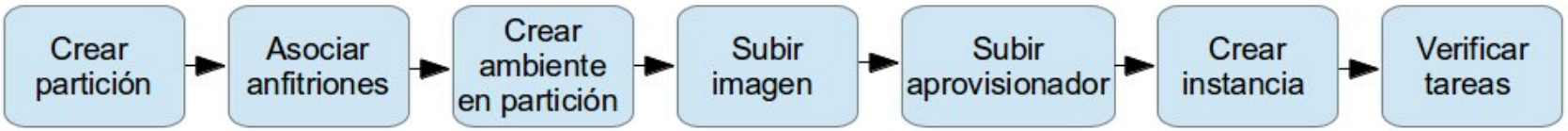
Vista de componentes



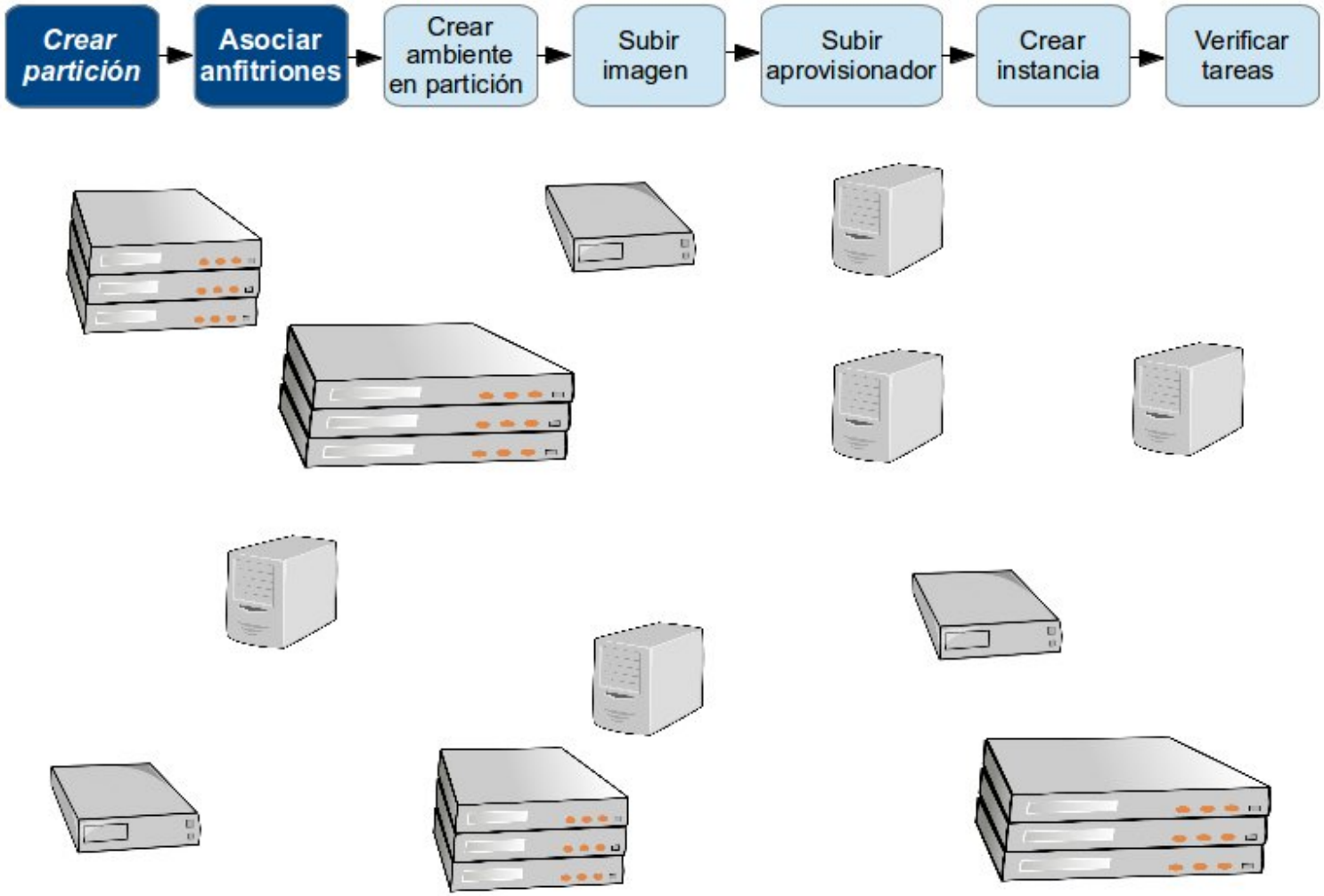
Vista de deployment



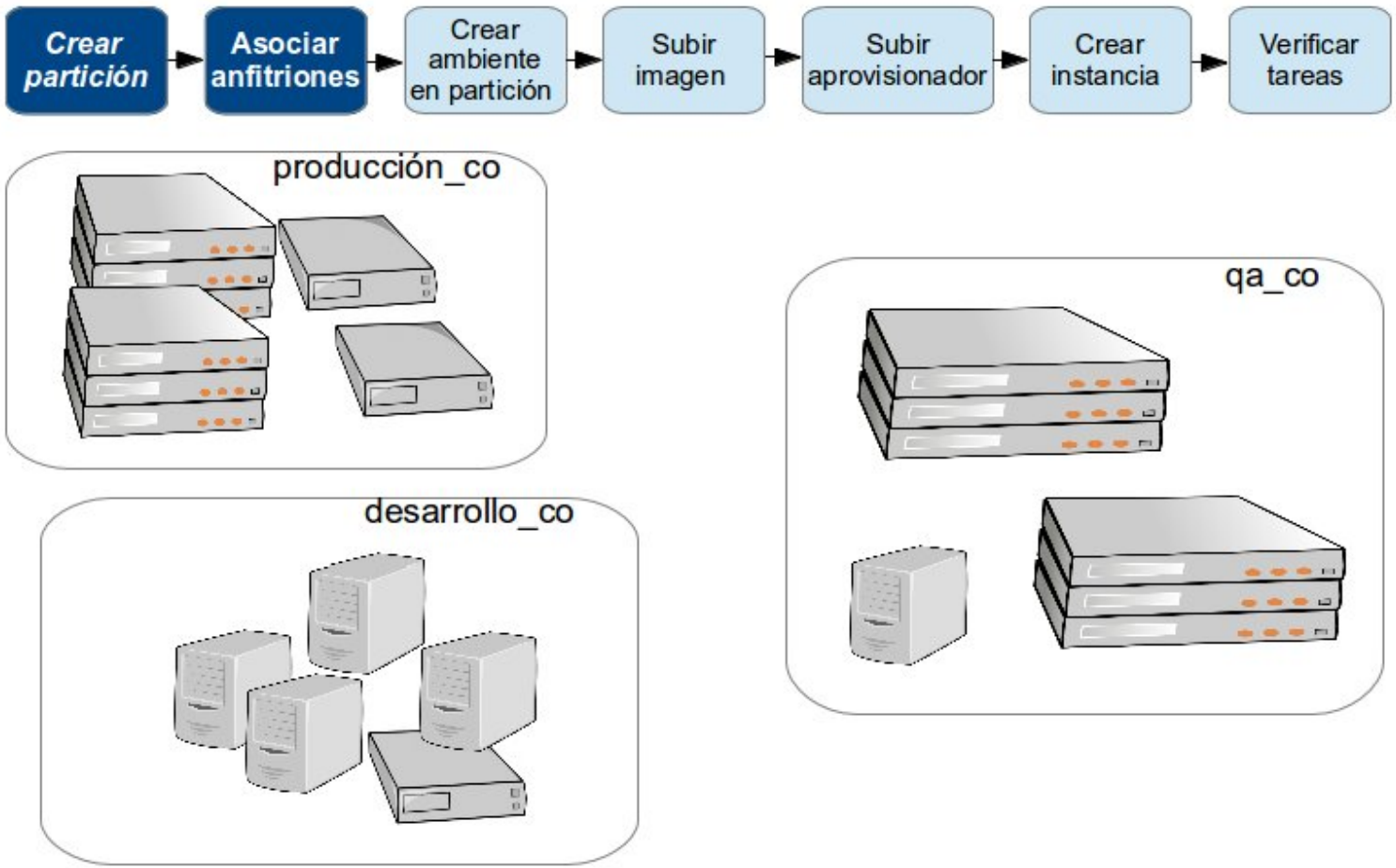
Flujo de aprovisionamiento



Particiones y Anfitriones



Particiones y Anfitriones



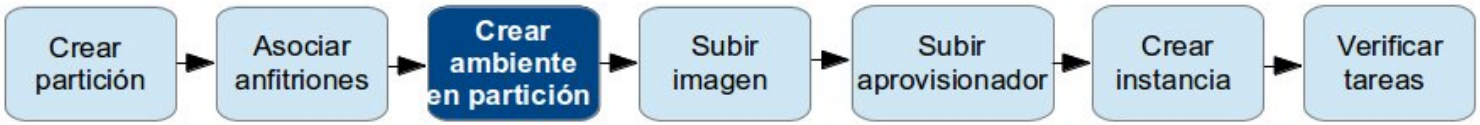
Creación de una partición (Ejemplo)

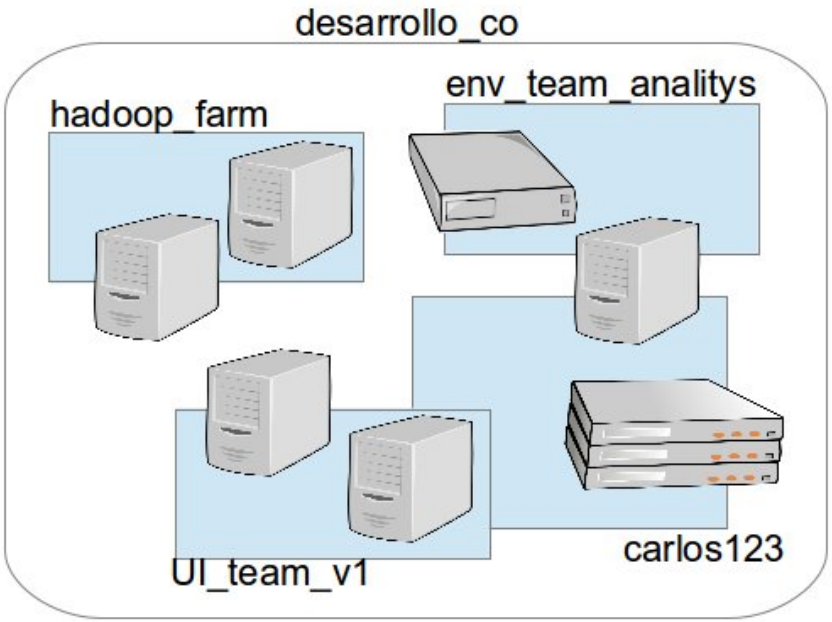
```
curl -X POST http://virtshellsrv:80/partitions/
    -d "{\"name\":\"development_co\",
        \"description\":\"Collection of servers oriented to
            development team in Colombia.\"}"
    -H "accept:application/json" | jq .
```

Asociación de un anfitrión a una partición (Ejemplo)

```
curl -X POST http://virtshellsrv:80/hosts/
    -d "{\"name\": \"host-server-01\",
        \"os\": \"ubuntu-14.04.4-amd64\",
        \"memory\": \"2GB\",
        \"partition\":\"development_co\",
        \"type\": \"GeneralPurpose\",
        \"local_ipv4\": \"192.168.56.101\",
        \"drivers\": [\"docker\", \"lxc\"]}"
    -H "accept:application/json" | jq .
```

Ambientes de trabajo





Ambientes de trabajo (Ejemplo)

```
curl -X POST http://virtshellsrv:80/enviroments/
    -d "{\"name\":\"development\",
        \"description\":\"Development enviroment\",
        \"partition\": \"development_co\",
        \"users\": [{\"login\": \"development_user\"},
                    {\"login\": \"guest\"}]}"
    -H "accept:application/json" | jq .
```

Imágenes

De dos Tipos

- ISO
- Templates

Imágenes (ISO)

`https://github.com/CALlanoR/ubuntu-unattended`

Imágenes (ISO)

```
curl -sv -X PUT \
    -H 'accept: application/json' \
    -H "Content-Type: text/plain" \
    -H 'X-VirtShell-Authorization: UserId:Signature' \
    -d '{"name": "ubuntu_server_14.04.2_amd64",
         "type": "iso",
         "os": "ubuntu",
         "timezone": "America/Bogota",
         "key": "/home/callanor/.ssh/id_rsa.pub",
         "preseed_url": "https://<host>:<port>/api/virtshell/v1/
             files/seeds/seed_ubuntu14-04.txt",
     "download_url": "http://releases.ubuntu.com/raring/ubuntu
         -14.04-server-amd64.iso"}' \
   'http://virtshellsrv:8080/api/virtshell/v1/image' | jq .
```

Imágenes (Template)

```
FROM ubuntu:14.04
MAINTAINER Carlos Llano <carlos_llano@hotmail.com>

RUN locale-gen en_US.UTF-8
RUN dpkg-reconfigure locales

RUN sed 's/#$ModLoad imudp/$ModLoad imudp/' -i /etc/rsyslog.conf
RUN sed 's/#$UDPServerRun 514/$UDPServerRun 514/' -i /etc/rsyslog.conf
RUN sed 's/#$ModLoad imtcp/$ModLoad imtcp/' -i /etc/rsyslog.conf
RUN sed 's/#$InputTCPServerRun 514/$InputTCPServerRun 514/' -i /etc/rsyslog.conf
EXPOSE 514/tcp 514/udp
CMD ["/usr/sbin/rsyslogd", "-dn", "-f", "/etc/rsyslog.conf"]

RUN apt-get update -y
RUN apt-get install -y openssh-server git

RUN mkdir /var/run/sshd
RUN useradd -s /bin/bash -m virtshell
RUN echo "virtshell:virtshell" | chpasswd
RUN echo "virtshell ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

RUN echo "root:virtshell" | chpasswd
RUN sed -i "s/PermitRootLogin without-password/PermitRootLogin yes/" /etc/ssh/sshd_config

RUN sed "s@session\s*required\s*pam_loginuid.so@session optional pam_loginuid.so@g" -i /etc/pam.d/sshd

ENV NOTVISIBLE "in users profile"
RUN echo "export VISIBLE=now" >> /etc/profile

EXPOSE 22
CMD ["/usr/sbin/sshd", "-D"]

RUN apt-get install -y python3-setuptools python3-pip
RUN pip3 install virtshell_commands

RUN echo  'export LC_ALL=C' >> ~/.bashrc
CMD ["source .bashrc"]

RUN apt-get install -y supervisor
RUN mkdir -p /var/log/supervisor
RUN printf '[supervisord]\nnodaemon=true\n\n' >> /etc/supervisor/conf.d/supervisord.conf
RUN printf '[program:sshd]\nautostart=true\ncommand=/usr/sbin/sshd -D' >> /etc/supervisor/conf.d/supervisord.conf

CMD /usr/bin/supervisord -c /etc/supervisor/conf.d/supervisord.conf
```

Universidad del Valle

Primero el archivo (Ejemplo - Template)

```
curl -sv -X POST \
    -H 'accept: application/json'
    -H "Content-Type: multipart/form-data"
    -F "permissions=xwrxwrxwr"
    -F "file=@/home/callanor/Documents/Tesis/Repositories/VirtShell
        /virtshell_server/tests/files/dockerfile_ubuntu_server_14
        .04"
    http://virtshellsrv:80/files/  | jq .
```

Luego la imágen (Ejemplo - Template)

```
curl -sv -X POST http://virtshellsrv:80/images/
    -d "{\"name\":\"ubuntu_server_14.04_amd64\",
        \"type\":\"docker-container\",
        \"container_resource\":\"http://192.168.56.103/file/
            dockerfile_ubuntu_server_14.04\"}"
    -H 'accept: application/json' | jq .
```

Aprovisionadores (BASH Puro)

https://github.com/CALlanoR/VirtShell_Provisioner_Simple_WebSite_Example

Aprovisionadores (BASH + VirtShell Commands)

https://github.com/CALlanoR/VirtShell_Generic_Provisioner_Simple_WebSite_Example

Tipos de instancias

Tres Tipos

- VirtualBox
- Docker
- LXC
- Amazon (Under Construction...)

Creación de una instancia (Ejemplo)

```
curl -X POST http://virtshellsrv:80/instances/
    -d "{\"name\": \"website2\",
        \"memory\": 1024,
        \"cpus\": 1,
        \"hdsize\": \"2GB\",
        \"description\": \"WebServer\",
        \"enviroment\": \"development\",
        \"provisioner\": \"generic_simple_web_site\",
        \"host_type\": \"GeneralPurpose\",
        \"driver\": \"docker\"}"
    -H "accept:application/json" | jq .
```

Chequeo de tareas

Consultar una tarea (Ejemplo)

```
curl -s http://192.168.56.103:80/tasks/b9bc6d72-cf78-4c92-bc34-
    c06809d4d52b | jq .
```

Demo

Experiencia y Evaluación

En las pruebas realizadas VirtShell demostró que parece ser una herramienta útil para aprovisionar software de manera sencilla y fiable.

La experiencia adquirida con la primera versión es la siguiente:

- *VirtShell funciona*.
- *Aprovisionar ambientes virtuales via web usando scripts escritos en el lenguaje que prefiera si es posible*.
- *El aprovisionamiento de máquinas virtuales o contenedores es prácticamente el mismo*.

Experiencia y Evaluación

La primera versión de VirtShell fue desarrollada en el lenguaje Python (versión 3)

Se encuentra alojada en el repositorio git: `https://github.com/janutechnology/VirtShell`.

Esta versión inicial aun no esta terminada y se encuentra en continuo desarrollo para lograr tener todas las funcionalidades funcionando.

Conclusiones

Recomendaciones

Preguntas?