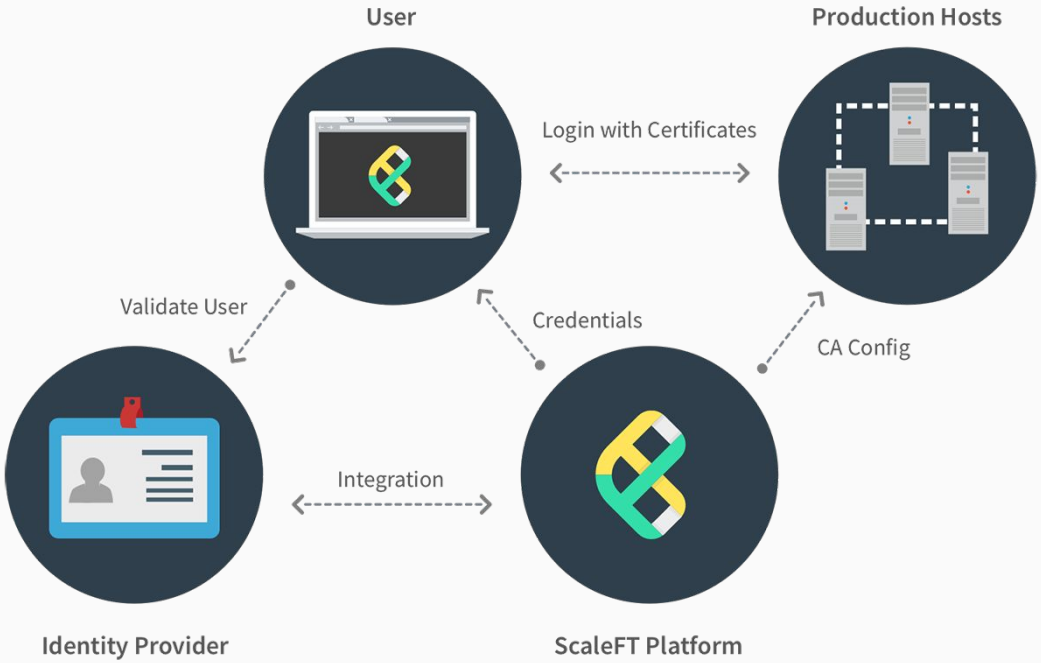
Navigating Compliance in a CoreOS World

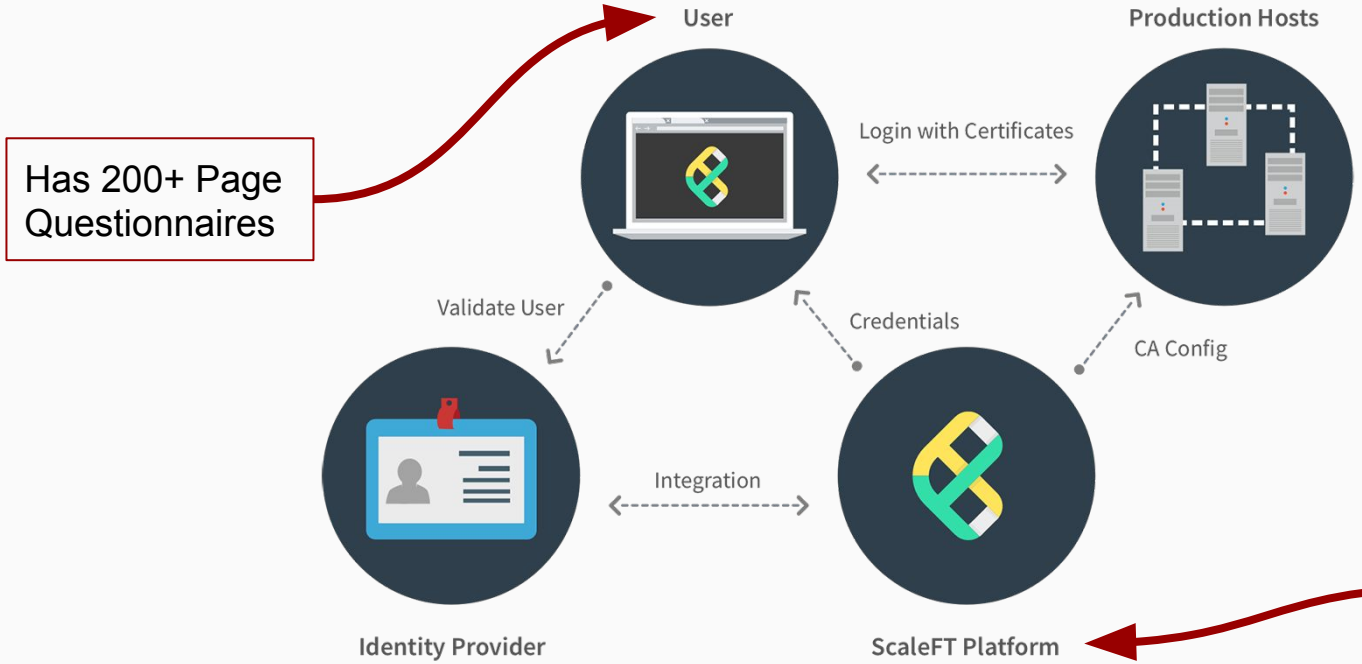
Paul Querna | @pquerna
CTO, ScaleFT

May 10, 2016

ScaleFT


User
Production Hosts
Login with Certificates
Validate User
Credentials
CA Config
Integration
Identity Provider
ScaleFT Platform

ScaleFT



Has 200+ Page Questionnaires

Runs CoreOS

+ COMPLIANCE

+ COMPLIANCE

Fun! New!

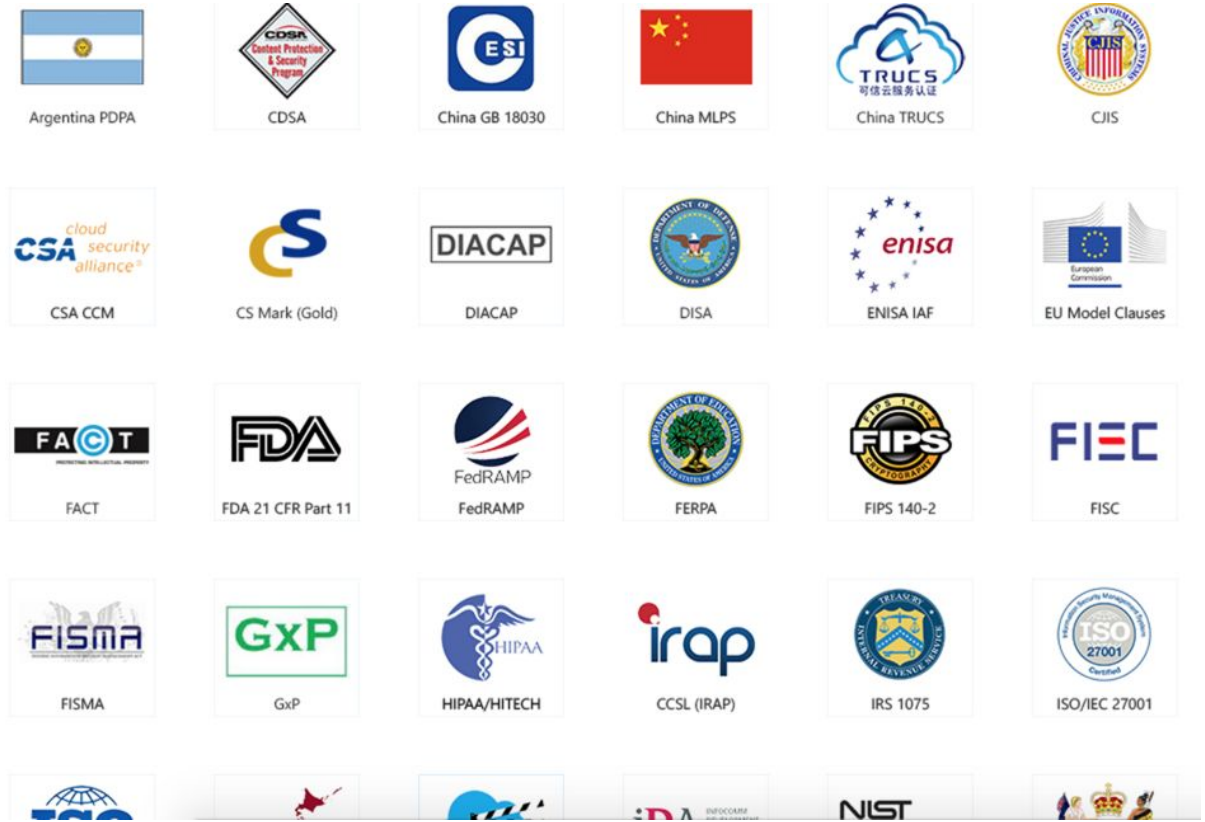
Not Fun! Old!

+

~~COMPLIANCE~~

RISK

Many Standards for Many Purposes



https://www.microsoft.com/en-us/trustcenter/Compliance


PCi
Security
Standards Council
™

- Controls (*think: things to reduce risk*):
  - Policies / documentation
  - **Technical**

User Management on CoreOS

User Management Controls

- Unique User IDs
- Role based Permissions
- Lifecycle Management

First Strategy

1. Put everything into cloud-config

Put everything into cloud-config

```
#cloud-config

users:
- name: paul.querna
  shell: /bin/bash
  groups:
  - sudo
  - docker
  sudo:
  - ALL=(ALL) NOPASSWD:ALL
  ssh-authorized-keys: [ssh-rsa AAAAB….
      pquerna@GraphiteModerated.local]
```

"cloud-init... there are a number of hurdles..."

Alex Crawford
2015 CoreOS Fest

Hurdles

- Go code to generate YAML
  - Users, fetching keys from git
  - Inline script rendering
  - systemd unit files
- Reboots
  - Deleted user, comes back!
- Changes
  - Lifecycle of configurations (including users) != lifecycle of servers

Attempt Two

1. Put “bootstrap” script in cloud-config
   (from zero today, try Ignition?)
2. Use Ansible for post-boot management

Bootstrap

```
#cloud-config

write_files:
- path: /opt/bin/bootstrap-cc.sh
  permissions: "0755"
  owner: root
  content: |-
    #!/bin/bash
    ...
coreos:
  units:
  - name: bootstrap-cc.service
    command: start
    content: |
      [Unit]
      Description=bootstrap runcmd
      [Service]
      Type=oneshot
      RemainAfterExit=yes
      ExecStart=/opt/bin/bootstrap-cc.sh
```

Ansible on CoreOS Linux

- Python…. Is not in the base system.
  - PyPy portable: github.com/squeaky-pl/portable-pypy
  - ln -s bin/pypy /opt/bin/python
  - Tell ansible where python is:

    ```
    [coreos:vars]
    ansible_python_interpreter="/opt/bin/python"
    ```
- Ansible basically* works!
  - Shell, Users, File
- Future: rkt fly?

Agents on CoreOS

First Strategy

1. Docker in systemd

- Namespaces
- Mounting the universe
- Systemd integration (lack of)

Outside of containers

1. Ansible: untar into /opt
2. Ansible: creates systemd unit file

- Great for Go & self contained things

Round 3: rkt (fly)

- Tried 12 months ago for all uses: Pain
- Tried 60 days ago w/ fly stage1: Yay!

acbuild: pretty easy?

```
# Start the build with an empty ACI
acbuild --debug begin

# Name the ACI
acbuild --debug set-name scaleft.com/sftd

# Copy the app to the ACI
acbuild --debug copy "${INPUT_SFTD}" /scaleft/bin/sftd

# Set correct file permissions and owner
chmod 0755 .acbuild/currentaci/rootfs/scaleft/bin/sftd
chown 0:0 .acbuild/currentaci/rootfs/scaleft/bin/sftd

# Run sftd
acbuild --debug set-exec -- /scaleft/bin/sftd

for m in ${MOUNT_DIRS}; do
    acbuild mount add "${m}" "/${m}"
done

acbuild --debug write --overwrite "${OUTPUT_FILE}"
```

User Management: Via Agent

- Dogfooding our own Agent
- ScaleFT Server Daemon manages users
- Runs via rkt fly and a systemd unit
- www.scaleft.com/docs/sftd-coreos

Logs on CoreOS

Log Controls

- User identification (see User Management)
- Action
- Timestamp
- Prevent modification
- Ship to central server

Log Management

- systemd-journald: yay
- This is mostly about journal vs classic syslog
- More systemd journal integrations happening every day

First Strategy

1. journalctl -o json
2. shell script to upload to s3

Round 2: In progress

- journalbeat in rkt fly:
  - Pulls from journal using CGO bindings
  - Cursor integration
  - github.com/mheese/journalbeat
- ACI build:
  - github.com/authclub/journalbeat-aci

Updates on CoreOS

Updates Controls

- Change control / documented approval procedures
- If Anti-virus, auto-updates: +1
- If not: Anti-virus: ?

Auto Updates

Here's how you turn off CoreOS Linux's original feature:

```
echo REBOOT_STRATEGY=off | sudo tee -a /etc/coreos/update.conf
```

See also:

```
update_engine_client -status

update_engine_client -update
```

CoreUpdate by CoreOS

Thanks!

@pquerna
paul@scaleft.com

paul.querna.org/slides