JUDCon

JBoss Users & Developers Conference

2010:Berlin

BoxGrinder

Marek Goldmann

Who’s Marek?

- A guy from Poland working for JBoss
- Part of the oddest team in JBoss
- BoxGrinder lead

Appliance is a preconfigured disk image (virtual machine) with operating system and all required applications to do specific job

Appliance examples with tasks

- **Database**
  - Storing data
- **Front-end**
  - Load balancing
- **Back-end**
  - Actual servers

Bake v. Fry

Bake: Produce a complete virtual machine offline, before first use.

Fry: Produce a complete virtual machine by booting a basic VM and then applying configuration.

Bake

Offline building

Quicker launch

Requires more storage

Not up-to-date

Fry

Online building (snapshots)

Longer launch

Less storage

More up-to-date

Bake!

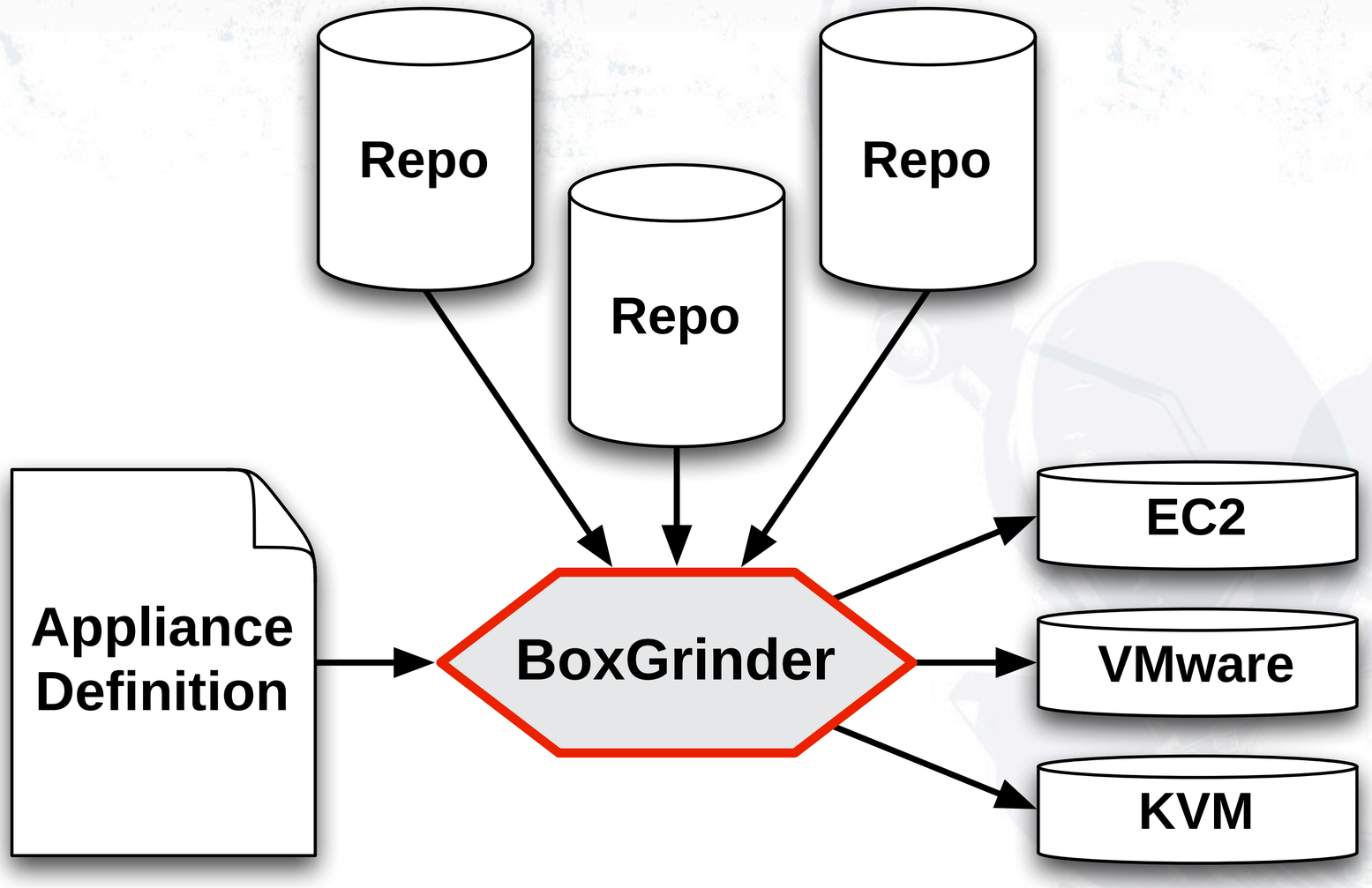
We think **baking** is The Right Way, especially for developers simply looking for reliable platforms.

Bake, then fry

If you **bake** an image you can **fry** it then later too! Baked image can be your **start point**.

BoxGrinder

BoxGrinder is a family of tools to grind out **appliances** for various **platforms**


Repo
Repo
Repo
Appliance Definition
BoxGrinder
EC2
VMware
KVM

Appliance definition, huh?

- Plain text file – **YAML** format
- Very easy to understand, modify
- Inheritance (mixins)

Appliance example

```
name: back-end
version: 1
release: 1
summary: back-end appliance with JBoss AS 6
hardware:
  memory: 512
  partitions:
    "/":
      size: 2
appliances:
  - fedora-base
packages:
  includes:
    - jboss-as6
    - jboss-as6-cloud-profiles
    - java-1.6.0-openjdk
```

General information

must match the filename: back-end.appl

```
name: back-end
version: 1
release: 1
summary: back-end appliance with JBoss AS 6
```

Hardware

```
hardware:
  memory: 512
  partitions:
    "/":
      size: 2
```

512MB

2GB

Appliance Mix-ins

Mixing in **`fedora-base.appl`**

```
appliances:
  - fedora-base
```

Tangent: fedora-base.appl

```
name: fedora-base
summary: Basic Fedora OS
os:
  name: fedora
  version: 11
hardware:
  memory: 256
  partitions:
    "/":
      size: 1
packages:
  includes:
    - bash
    - kernel
    - grub
    - e2fsprogs
    - passwd
    - policycoreutils
    - chkconfig
    ...
```

Appliance Mix-ins

back-end.appl **fedora-base.appl**

overrides →

```
hardware:
  memory: 512
  partitions:
    "/":
      size: 2
```

```
hardware:
  memory: 256
  partitions:
    "/":
      size: 1
```

Contents

```
packages:
  includes:
    - jboss-as6
    - jboss-as6-cloud-profiles
    - java-1.6.0-openjdk
```

Contents

```
packages:
  includes:
    - jboss-as6
    - jboss-as6-cloud-profiles
    - java-1.6.0-openjdk
```

plus everything from **`fedora-base.appl`**

Let's **build** the appliance!

BoxGrinder

Build

Prepare your environment

Simple

Use meta appliance

Preconfigured appliance in various formats ready to build images (built with BoxGrinder of course!)

http://www.jboss.org/boxgrinder/downloads/build/meta-appliance.html

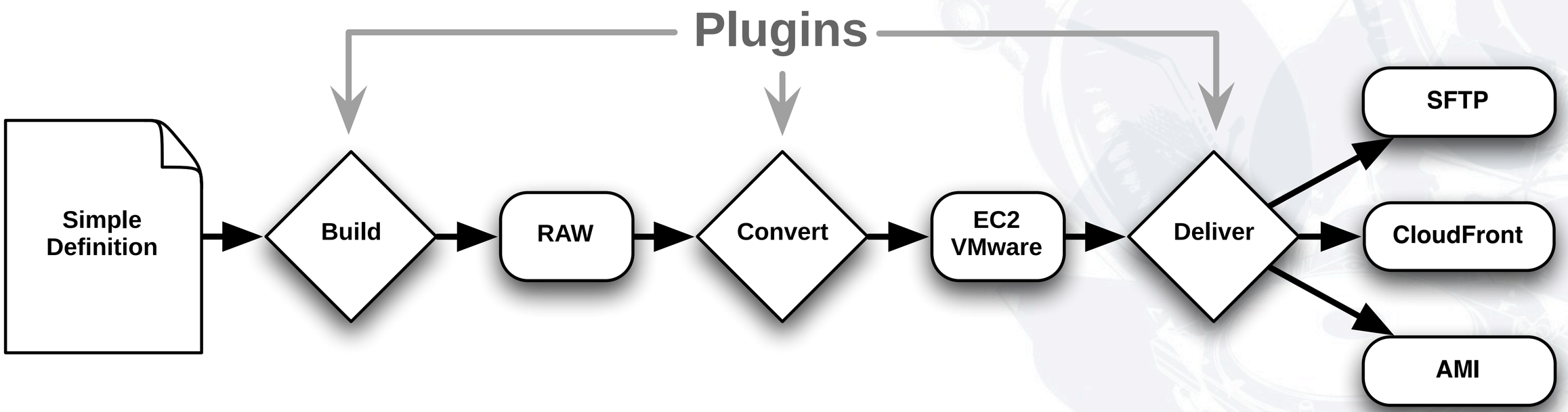
Harder

Install required software

appliance-tools, libguestfs, qemu, rubygems

http://community.jboss.org/wiki/BoxGrinderBuildPreparingEnvironmentToBuildImages

Build process



BoxGrinder Build installation

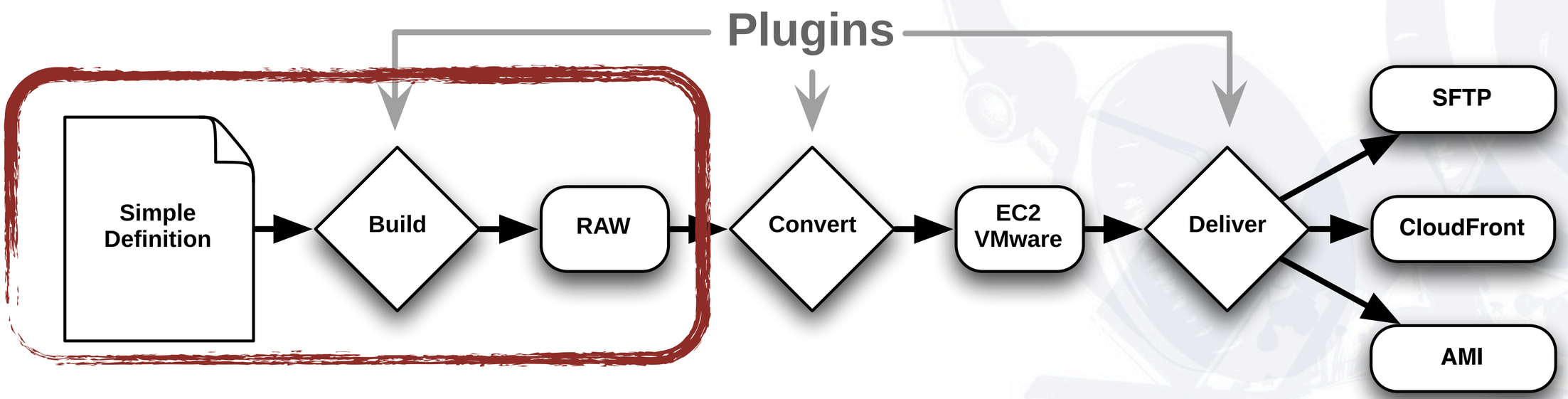
```
gem install boxgrinder-build
```

...and some plugins

```
gem install boxgrinder-build-fedora-os-
plugin boxgrinder-build-ec2-platform plugin
boxgrinder-build-s3-delivery-plugin
```

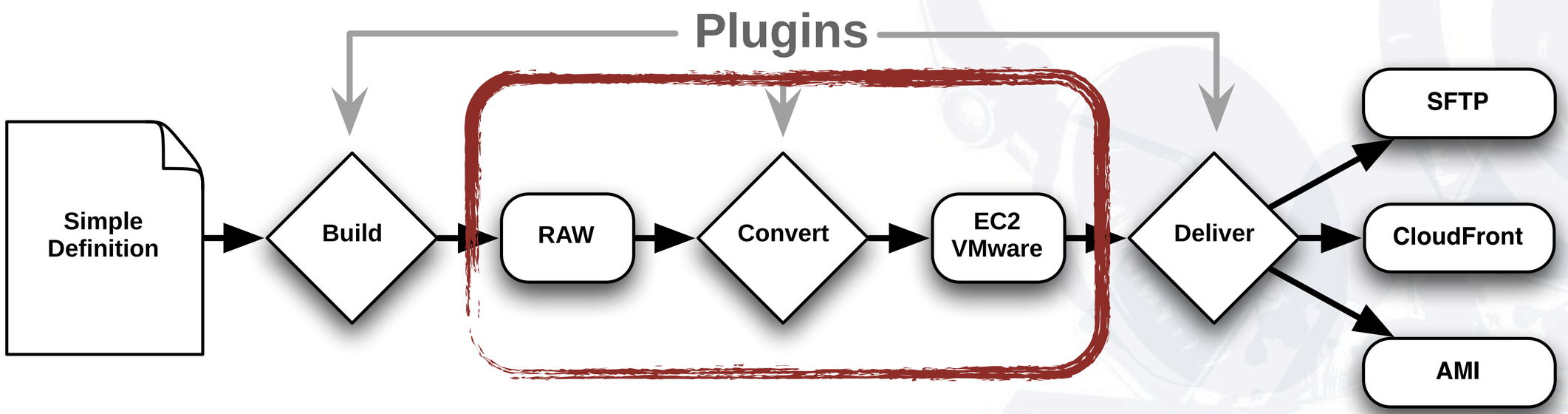
Task: build back-end AMI

Step 1: create base image



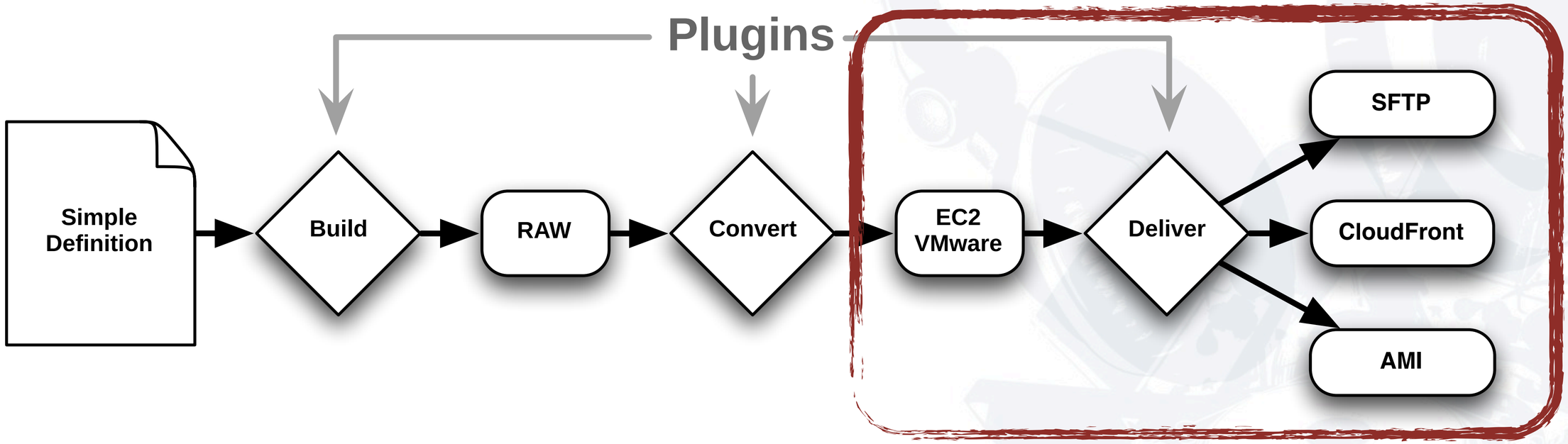

```
boxgrinder-build back-end.appl
```

Step 2: convert it to EC2 type




```
boxgrinder-build back-end.appl -p ec2
                               -p vmware
                               ...
```

Step 3: deliver it as AMI




```
boxgrinder-build back-end.appl -p ec2 -d ami
                                      -d ebs
                                      -d s3
                                      ...
```

Of course you can run the command just **once** with same result!

```
boxgrinder-build back-end.appl -p ec2 -d ami
```

BoxGrinder Build features

- **Supported OSes: Fedora, CentOS, RHEL**
- **Support for various platforms: EC2** (S3-backed and EBS-backed too!), **KVM**, **VMware**
- **Many delivery options: local**, **SFTP**, **S3** or **CloudFront** as tarred image, **AMI**
- **Cross-arch builds**: producing i386 images on x86_64 host
- Caching downloaded resources (RPM's)
- **Fast** – from .appl to registered AMI: **15 minutes**.

Notes

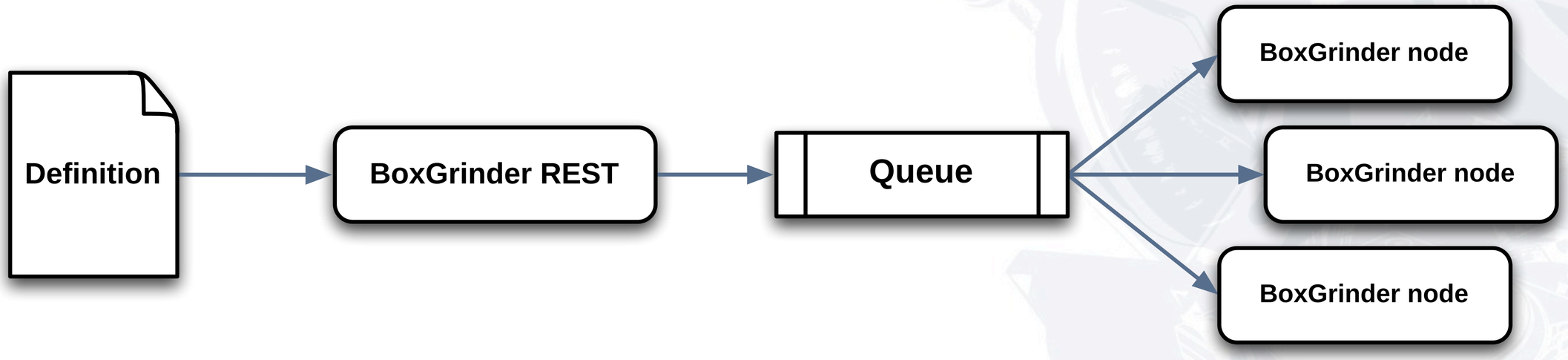
- If you're building AMI's – **do it on EC2** – this will safe your time (uploading to S3 from your local machine isn't fun...)
- Building **EBS-backed AMI's requires** to run BoxGrinder on EC2

BoxGrinder

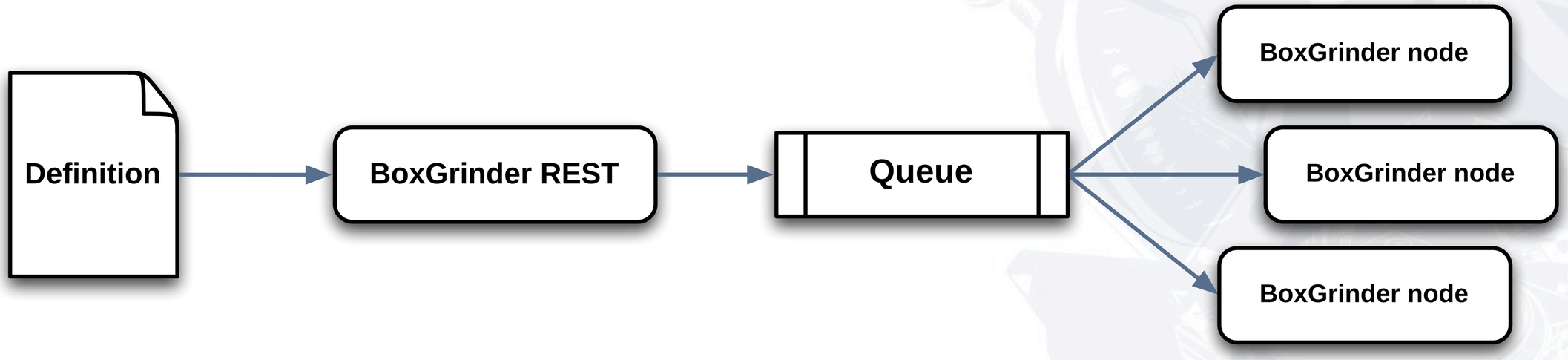
REST

RESTful API for driving farms of BoxGrinder-Build nodes.

BoxGrinder-REST



BoxGrinder-REST



Runs on **TorqueBox**, using **HornetQ**

BoxGrinder-REST

- Queue per each supported platform
- Multiple builders per queue possible
- Builders launching on-demand soon
  - Using Deltacloud
- Supports for multiple conversions
- Supports multiple deliveries

BoxGrinder-REST

- Command-line interface
- Web Client (BoxGrinder-Studio?)
- IDE integration (JBoss Tools?)

Thanks!

Q&A

Resources

http://github.com/stormgrind/	# Code
http://jboss.org/boxgrinder/	# Home page
http://cloudpress.org/	# Blog
#boxgrinder	# IRC
@boxgrinder	# Twitter
@marekgoldmann	
@bobmcwhirter	