Best Practices for Python in the Cloud: *Lessons Learned @ActiveState*

Presented by:

Gisle Aas, Senior Developer, ActiveState

ActiveState
The Dynamic Language Experts

EUROPYTHON 2011
FLORENCE, JUNE 20–26

whoami?

Gisle Aas

- gisle@activestate.com
- @gisle
- github.com/gisle

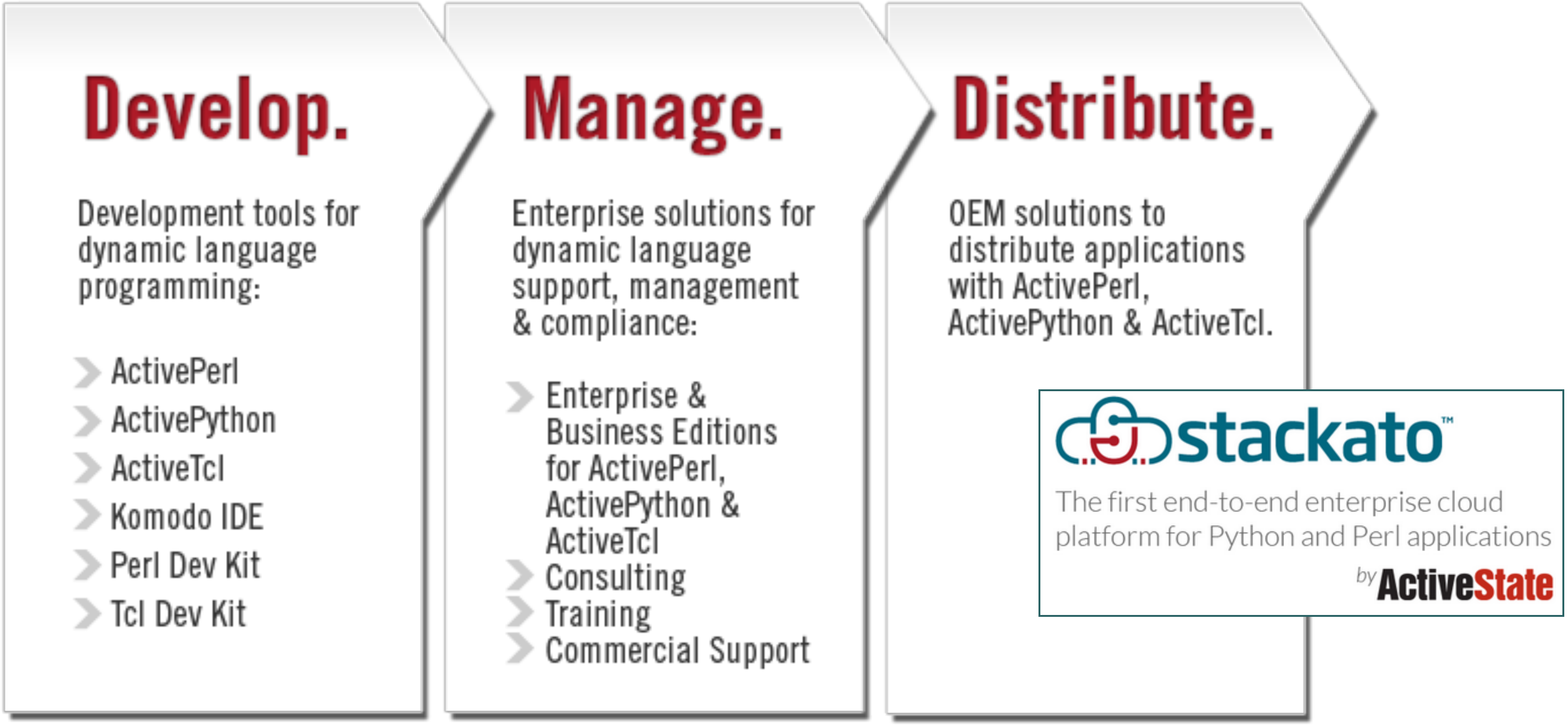
Agenda

- About ActiveState
- What’s a Cloud Anyway?
- Python in the Cloud
- The ActiveState Experience
- Best Practices
- Building for PaaS (Stackato)

About ActiveState

- Founded 1997
- 2 million developers, 97% of Fortune 1000 rely on ActiveState
- Development, management, distribution solutions for dynamic languages
- Core languages: Python, Perl, Tcl
- Other languages: PHP, Ruby, Javascript
- Related products: Komodo IDE, Stackato

ActiveState Solutions help developers



Develop.

Development tools for dynamic language programming:

- ActivePerl
- ActivePython
- ActiveTcl
- Komodo IDE
- Perl Dev Kit
- Tcl Dev Kit

Manage.

Enterprise solutions for dynamic language support, management & compliance:

- Enterprise & Business Editions for ActivePerl, ActivePython & ActiveTcl
- Consulting
- Training
- Commercial Support

Distribute.

OEM solutions to distribute applications with ActivePerl, ActivePython & ActiveTcl.

stackato™

The first end-to-end enterprise cloud platform for Python and Perl applications

by ActiveState

Just What Is A Cloud?

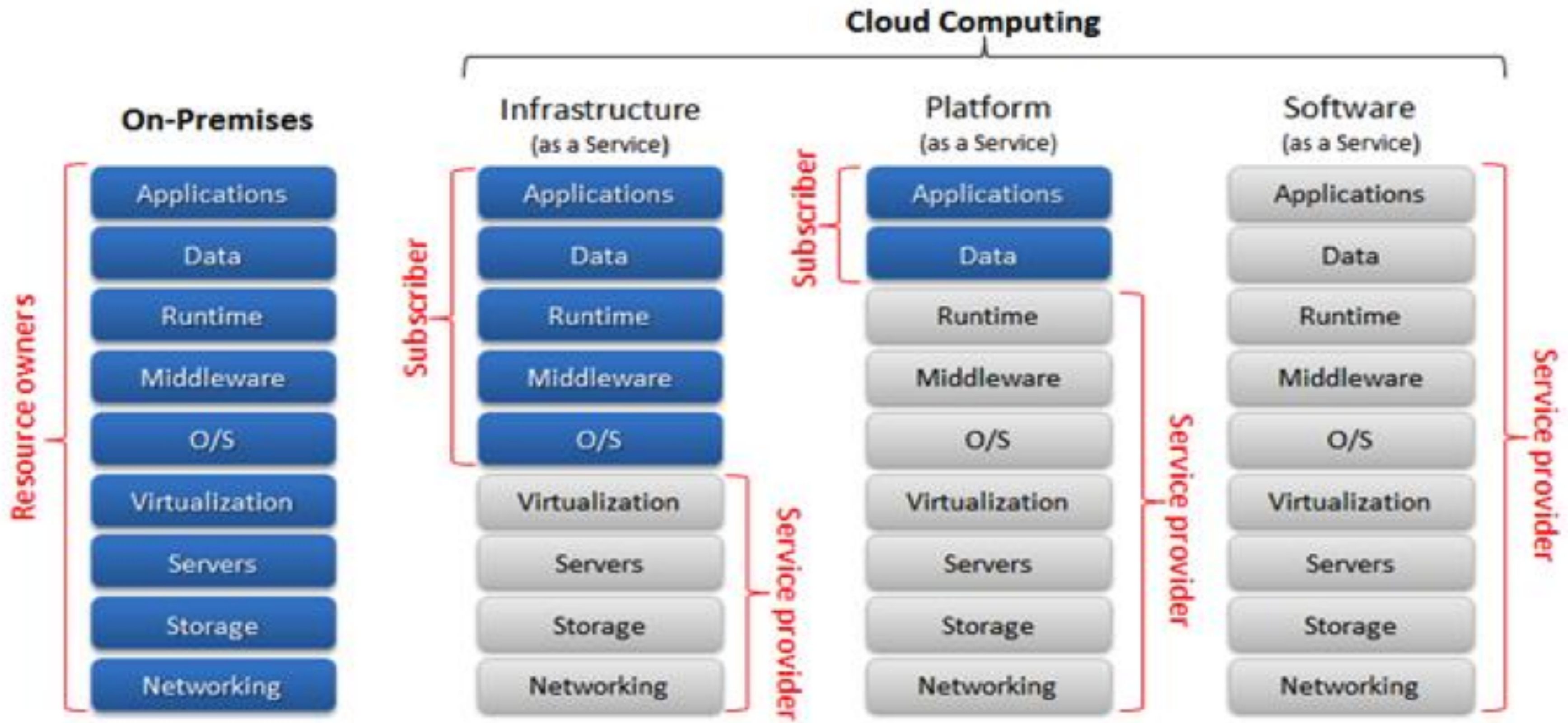
Cloud Computing definition (NIST)

Essential Characteristics:

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured Service

Cloud Computing definition (NIST)

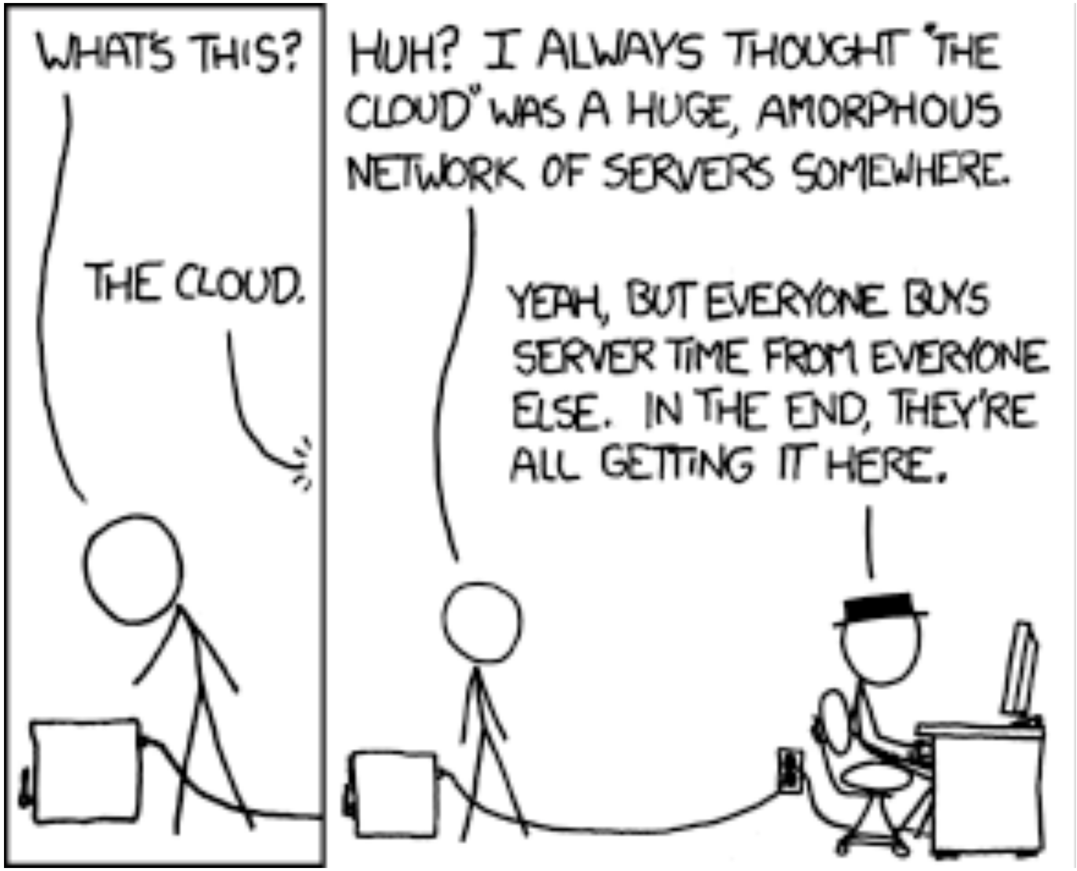
Service models: IaaS, PaaS, SaaS



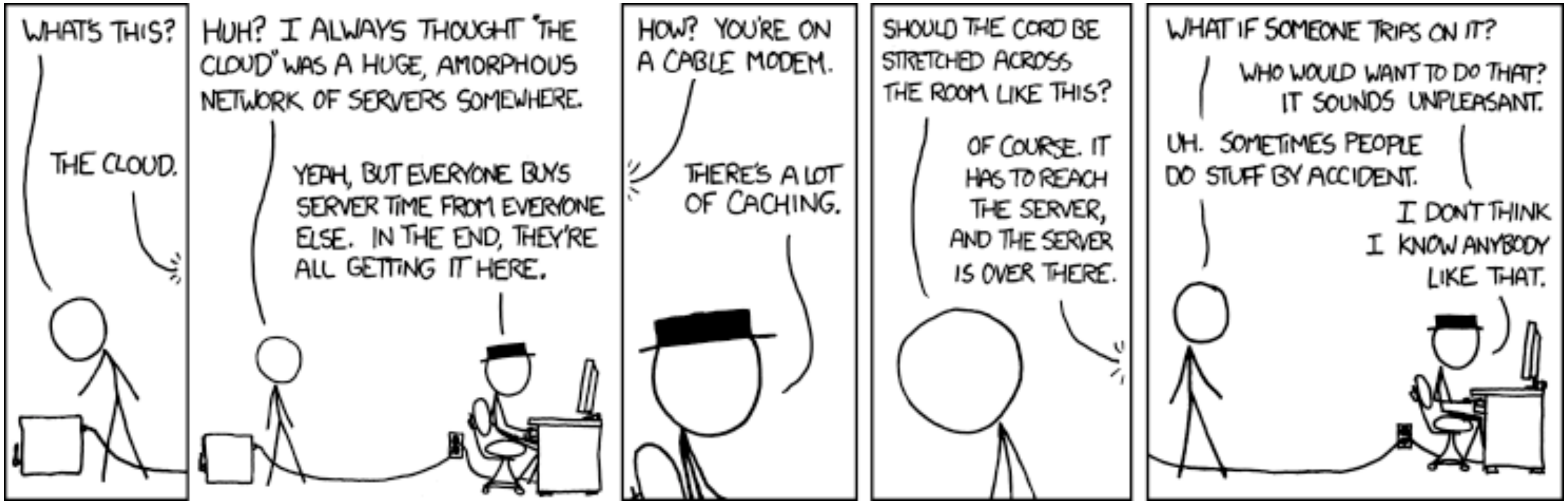
Cloud Computing definition (NIST)

Deployment models:

- Private cloud
- Community cloud
- Public cloud
- Hybrid cloud


WHAT'S THIS?
THE CLOUD.
HUH? I ALWAYS THOUGHT "THE CLOUD" WAS A HUGE, AMORPHOUS NETWORK OF SERVERS SOMEWHERE.
YEAH, BUT EVERYONE BUYS SERVER TIME FROM EVERYONE ELSE. IN THE END, THEY'RE ALL GETTING IT HERE.

The Cloud


WHAT'S THIS?
THE CLOUD.
HUH? I ALWAYS THOUGHT "THE CLOUD" WAS A HUGE, AMORPHOUS NETWORK OF SERVERS SOMEWHERE.
YEAH, BUT EVERYONE BUYS SERVER TIME FROM EVERYONE ELSE. IN THE END, THEY'RE ALL GETTING IT HERE.
HOW? YOU'RE ON A CABLE MODEM.
THERE'S A LOT OF CACHING.
SHOULD THE CORD BE STRETCHED ACROSS THE ROOM LIKE THIS?
OF COURSE. IT HAS TO REACH THE SERVER, AND THE SERVER IS OVER THERE.
WHAT IF SOMEONE TRIPS ON IT?
WHO WOULD WANT TO DO THAT? IT SOUNDS UNPLEASANT.
UH. SOMETIMES PEOPLE DO STUFF BY ACCIDENT.
I DON'T THINK I KNOW ANYBODY LIKE THAT.

http://xkcd.com/908/

Python in the Cloud

Komodo

Pyramid

MySQL

Django

Boto

PyPM

Pip

Stackato

Bottle

Scientific

ActivePython

Redis

Financial

Postgres

Flask

Who is using Python in the Cloud?

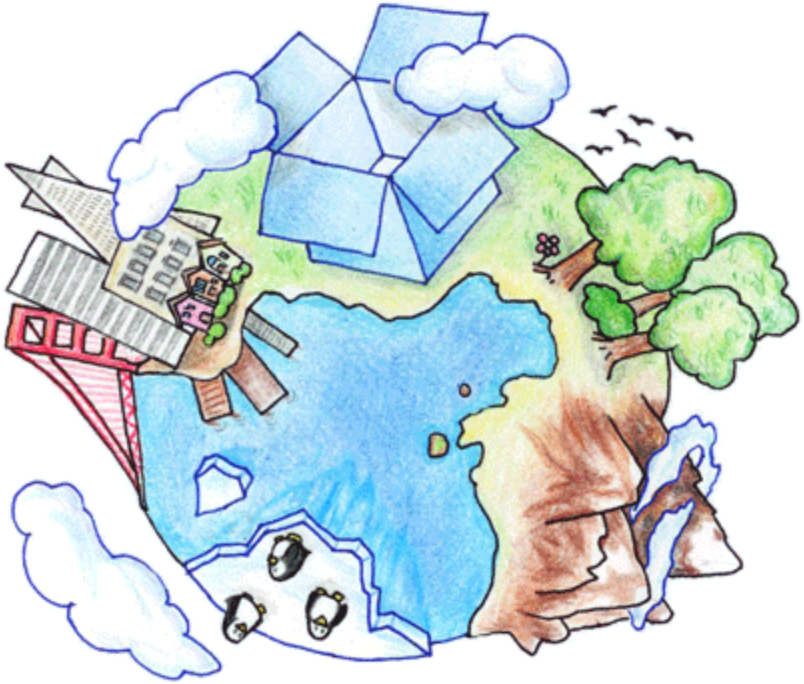
Dropbox

WebFilings™

TRUECar®

Changing How Cars are Sold

reddit



even michaelmoore.com uses python!

What ActiveState Learned along the way…

Our Cloud Experience

Firefly

Make your life easier.

Enlightened project management and collaboration for software development

Projects | Blog

Public Cloud

application drupal extension javascript komodoedit perl php project python

ActiveState Firefly

What is it for?

What does it do?

Firefly News

Workspace is now Firefly!

→

amazon web services™

→

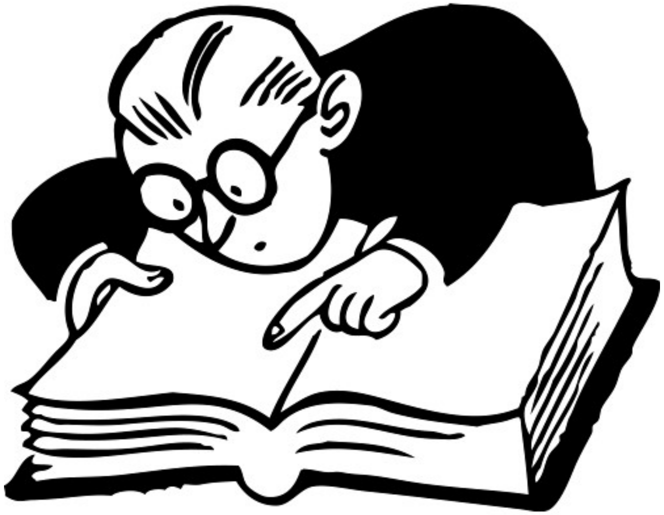
stackato™

The first end-to-end enterprise cloud platform for Python and Perl applications

by ActiveState

Our Best Practices Guidelines

1. Simple
2. Isolated
3. Disposable
4. Relocatable
5. Portable
6. Open



keep it simple.

Keep Your App Simple

- Keep your code and concepts simple
- Use simplest database that can possibly work
- LAMP Stacks work

ActivePython AMI LAMP Stack

for Python Web Applications

- Ubuntu 10.04LTS 64-bit
- Nginx
  - For static content and route web-traffic to multiple worker instances
- Apache (2.2.16), running mod_wsgi (3.3.7)
- ActivePython, with PyPM packaging manager
- virtualenv (1.5.1) & virtualenvwrapper
- Django (1.2.3)
- SQLite3, MySQL 5.1, memcached

- State of the art … 1 year ago!
  - The art changes quickly in the cloud

Make Deploying Apps Simple

- Developers just want to Develop
- Developers want to leverage familiar tools
  - VCS, IDE, etc.
- Resolve Dependencies Automatically
- Developers do ***not*** want to be sysadmins
  - No firewall management
  - No database installs
  - No network management

Manage your Packages!

- Use package managers such as
  - yum and aptitude for tools such as Apache
  - PyPM and Pip for Python package dependencies
- Use pre-built package repositories
  - *PyPM* pulls from a single central repository of pre-built binaries that have been tested to work on a number of platforms.
  - *pip* pulls resources from multiple sources, potentially delaying an install if one of the dependencies is temporarily inaccessible.

Keep it Isolated

Create in Isolated Environments

- Compartmentalize and isolate your build environment
- Dominant tools are virtualenv and virtualenvwrapper
  - http://pypi.python.org/pypi/virtualenv
  - http://www.doughellmann.com/projects/virtualenvwrapper/.
  - Easily setup and switch between multiple, isolated python environments
- Developers like the isolation
  - Fearlessly install new tools for experimentation without corrupting other environments
  - Allows incompatible version dependencies

Enable Security by Isolation

- The only truly “trusted” solution:

Security by Isolation!

Natural borders, like OS system boundaries are the strongest security borders you can create for application deployment.

http://theinvisiblethings.blogspot.com/2008/09/three-approaches-to-computer-security.html

A Choice of Solutions

- FreeBSD Jail
- Solaris Containers
- IBM AIX Partitions
- Linux-VServer
- Parallels Virtuozzo
- OpenVZ
- LXC
- Full OS virtualization

Make it Disposable

Disposable Instances

- Treat instances as temporary, disposable things that can easily be rebuilt from scratch
  - Manage state changes from a build script
  - Avoid logging-in via ssh to run installs or to edit configurations
- Tools like Fabric are designed to handle multi-step, multi-machine deployment
  - A fabfile becomes executable documentation of how to build your instance, run tests, and bring the system live
- Simplifies software updates or altering configurations
- Useful for adjusting instance sizes or replication for load-balancing

Disposable Data

- What parts of your data is disposable
- How to persist data
  - Locally managed data
  - Off-system data stores
- Mirroring
  - For high-availability
  - For safety
- Backups … need I say more?

Make it Relocatable

Relocatable Components

- Communicate through sockets
- Use distinct hostnames, no IP addresses
- Make cloning easy that allows retargeting hostnames
- Separate primary services (e.g. db and web app)
  - Allows cloning the right services for scaling
  - Backups can happen only for key components
  - Fail-over and recovery become easier to implement
  - Nginx front-end should be able to run separately from the underlying web server(s)

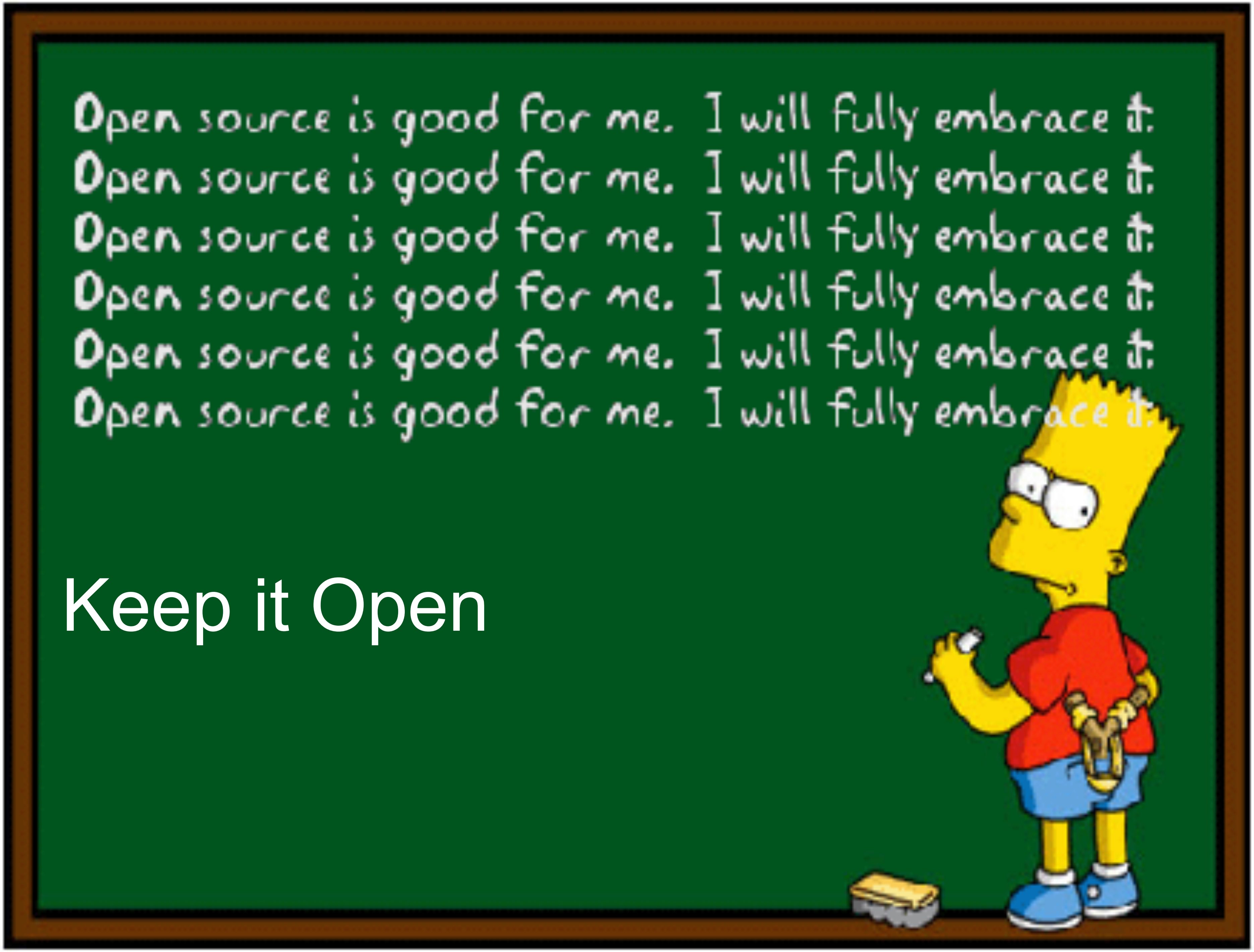
Make it Portable

Portability

- Between IaaS providers
  - Amazon, Linode, RackSpace, …
- Between PaaS systems
- Minimize Dependencies
- Rely on the web app framework
  - It provides many abstractions already

Object Relational Mappers

- Abstracts working with databases
- Change in response to load or storage needs
- Decouple the database from the application
  - An adapter between different styles of database access
- Design for scalability from the outset


Open source is good for me. I will fully embrace it.
Open source is good for me. I will fully embrace it.
Open source is good for me. I will fully embrace it.
Open source is good for me. I will fully embrace it.
Open source is good for me. I will fully embrace it.
Open source is good for me. I will fully embrace it.

Keep it Open

6 Simple Rules to Live By

- Simple
- Isolated
- Disposable
- Relocatable
- Portable
- Open Source

Working with a PaaS Platform

The Stackato Experience

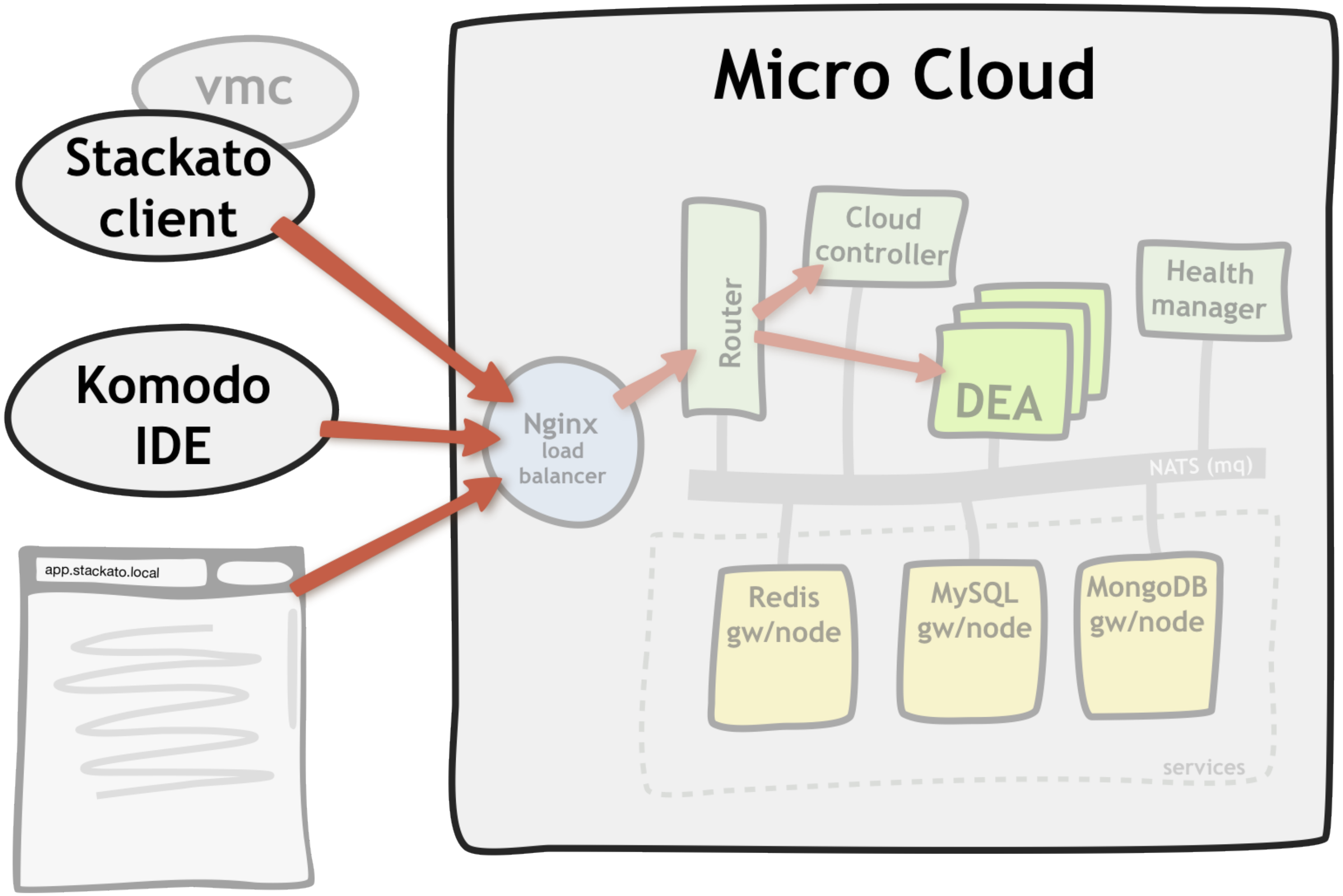
Stackato in 6 Simple Rules

- Simple
  - 3 step application deployment
  - Automatic dependency resolution
- Isolated
  - Each instance get their own environment
  - Security by isolation
- Disposable
  - Reusable VMs
  - Code separated from instances

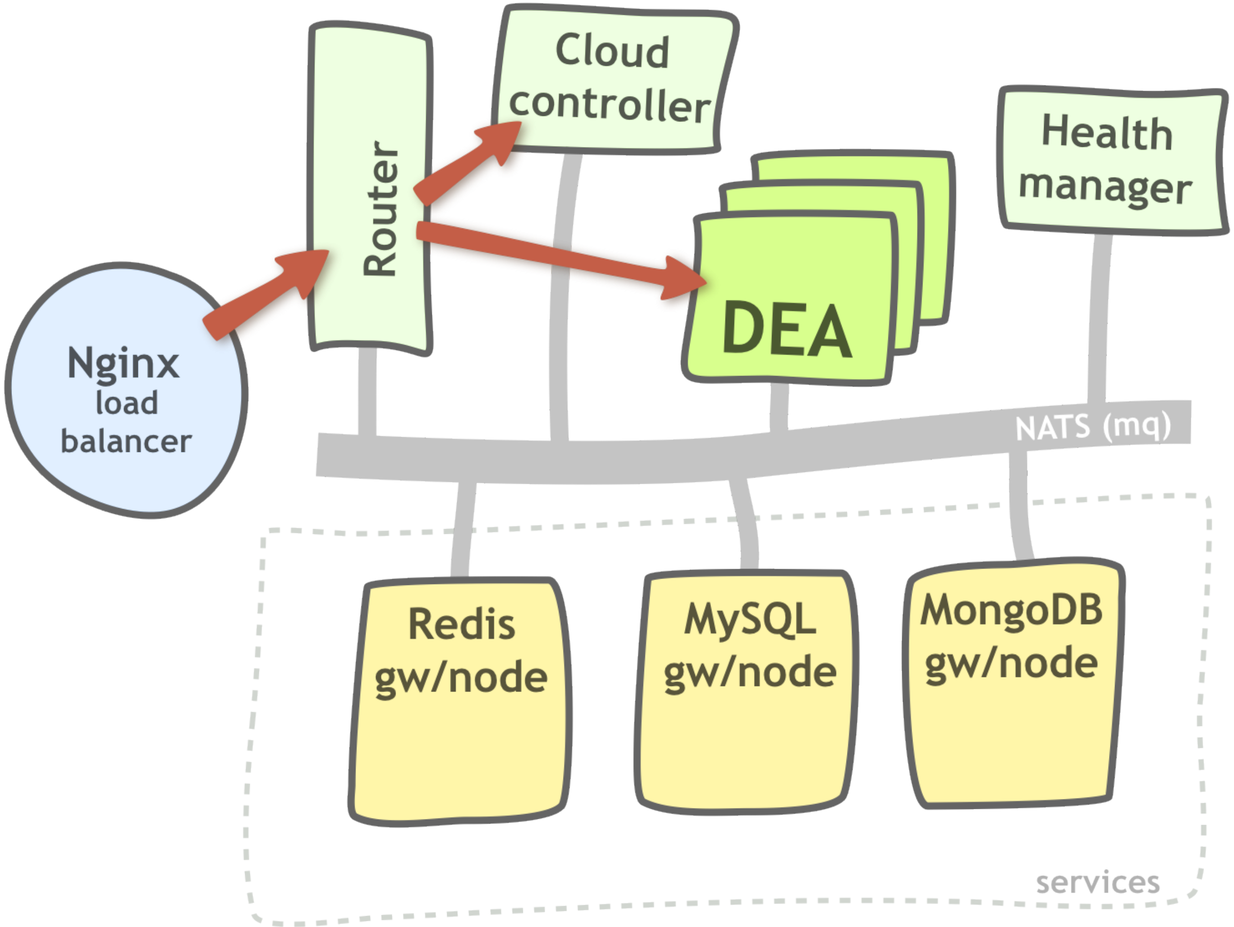
Stackato in 6 Simple Rules

- Relocatable
  - Concept of "Services"
- Portable
  - Options of VMs or AMI
  - Both with the VM and code by design
- Open Source
  - Based on Cloud Foundry
  - LAMPy stack

Stackato Client and VM



Stackato Architecture



Stackato from a Developer Perspective

- 3 Step App Deployment
  - Install the 'stackato' client with pypm
  - Set the 'target' URL
  - Run 'stackato push’
- Client bundles the application and pushes it to the server
- Config files in the application directory tell the cloud controller what additional software is required
- The application environment is assembled, and the application is deployed to one or more worker nodes

Any Questions?

Thank you! Questions?

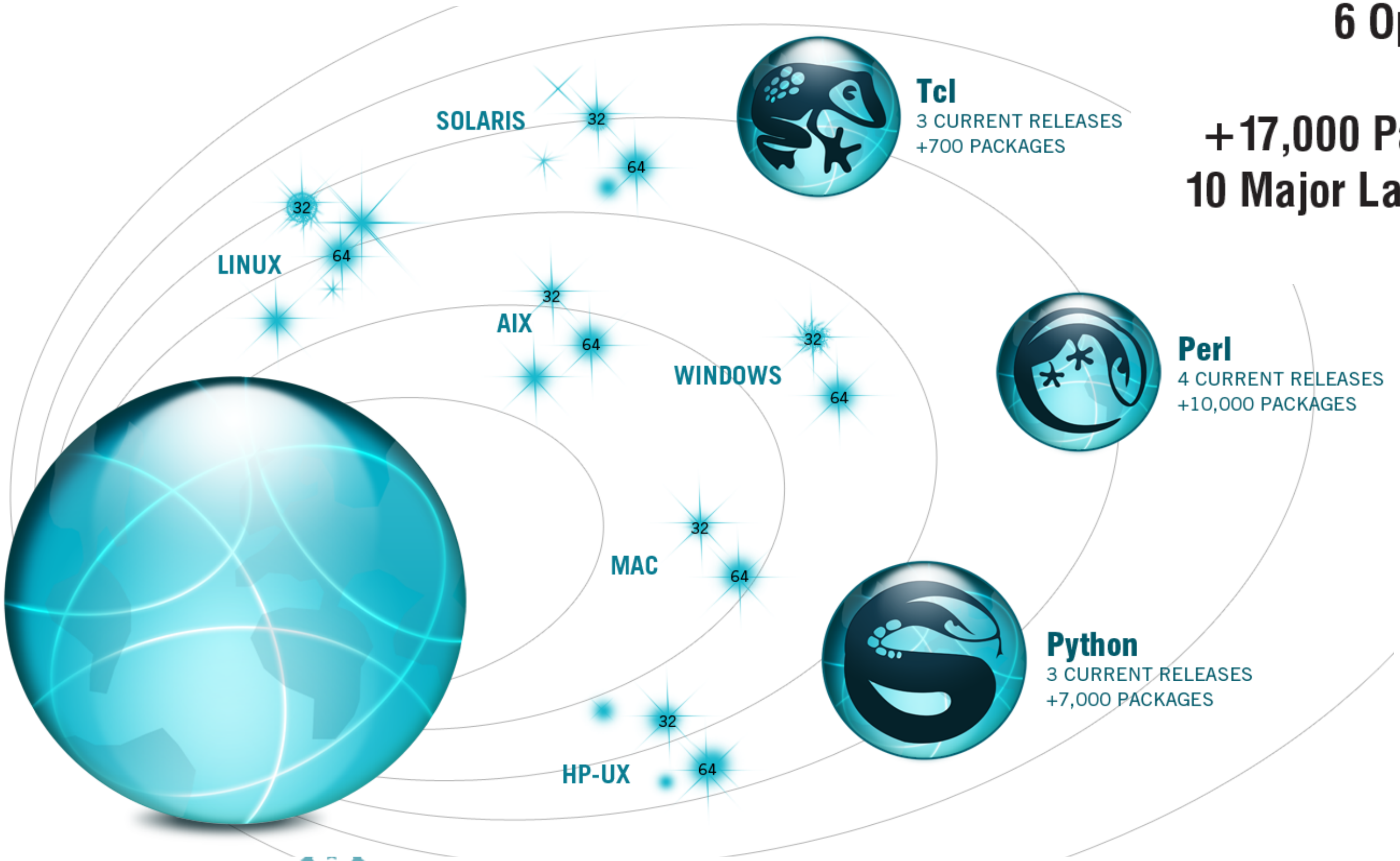
- Questions?

- Next Steps:
  - Find out more about Stackato
    - activestate.com/cloud

  - Request Information:
    - Business-Solutions@activestate.com
    - 1-866-510-2914

We've Got Your Universe Covered

6 Operating Systems
16 Architectures
+17,000 Packages/Modules
10 Major Language Releases



Tcl
3 CURRENT RELEASES
+700 PACKAGES

Perl
4 CURRENT RELEASES
+10,000 PACKAGES

Python
3 CURRENT RELEASES
+7,000 PACKAGES

EUROPYTHON 2011
FLORENCE, JUNE 20–26

ActiveState
The Dynamic Language Experts