

# glu

*deployment automation platform*

July 2011

Yan Pujante

 <http://www.linkedin.com/in/yan>

blog: <http://pongasoftware.com/blog/yan>

 @yanpujante

\* To see a video of this presentation given at Chicago devops, check this link:  
<http://devops.com/2011/07/09/glu-deployment-automation-video/>

# Video

- to see a video of this presentation given at Chicago devops, check this link:

<http://devops.com/2011/07/09/glu-deployment-automation-video/>



# A little bit about me...

- Software engineer (16 years experience)
- Software is my passion (28 years! TI-99/4A)
- Currently *not* working... for a boss... :)
  - glu, kiwidoc ([www.kiwidoc.com](http://www.kiwidoc.com))
- Worked @ LinkedIn for 8 years (founding team!)
  - Worked on a lot of infrastructure projects and early features (security, payment, graph, etc...)
  - Last (big) project was glu (main author/contributor/maintainer)



# Why glu ?

# Before glu...

:'(

```

Terminal — zsh — 80x24
-rw-r--r-- 1 ypujante wheel 2192 Jun 23 09:23 yjs201011091454.jar
(ypujante@beon:/tmp/ 4)ll 18:04:16
total 8
drwxr-xr-x 7 root wheel 306 Jun 23 09:54 ./
drwxr-xr-x 7 root wheel 238 Dec 5 2009 ../
-rw-r--r-- 1 ypujante wheel 8 Jun 23 09:06 icssuir503-
drwxr-xr-x 2 ypujante wheel 182 Jun 23 09:06 launch-3Nsuir/
drwxr-xr-x 2 ypujante wheel 182 Jun 23 09:06 launch-3tHn50/
drwxr-xr-x 2 ypujante wheel 182 Jun 23 09:06 launch-vlctv5/
drwxr-xr-x 2 ypujante wheel 182 Jun 23 09:06 launchd-171.WEIP2/
drwxr-xr-x 2 ypujante wheel 68 Jun 23 09:05 ssh-VFRzHnXHB/
-rw-r--r-- 1 ypujante wheel 2192 Jun 23 09:23 yjs201011091454.jar
(ypujante@beon:/tmp/ 5)ll 18:04:17
total 8
drwxr-xr-x 7 root wheel 306 Jun 23 09:54 ./
drwxr-xr-x 7 root wheel 238 Dec 5 2009 ../
-rw-r--r-- 1 ypujante wheel 8 Jun 23 09:06 icssuir503-
drwxr-xr-x 2 ypujante wheel 182 Jun 23 09:06 launch-3Nsuir/
drwxr-xr-x 2 ypujante wheel 182 Jun 23 09:06 launch-3tHn50/
drwxr-xr-x 2 ypujante wheel 182 Jun 23 09:06 launch-vlctv5/
drwxr-xr-x 2 ypujante wheel 182 Jun 23 09:06 launchd-171.WEIP2/
drwxr-xr-x 2 ypujante wheel 68 Jun 23 09:05 ssh-VFRzHnXHB/
-rw-r--r-- 1 ypujante wheel 2192 Jun 23 09:23 yjs201011091454.jar
(ypujante@beon:/tmp/ 6)ll 18:04:18
Terminal — zsh — 80x24
-rw-r--r-- 1 ypujante oadmin 544 Jun 22 09:20 org.linkedin.glu.packaging-all-3.8.8-SNAPSHOT/
drwxr-xr-x 18 ypujante oadmin 578 Jun 19 07:49 org.linkedin.glu.packaging-all-3.8.8.RC2-SNAPSHOT/
drwxr-xr-x 5 ypujante oadmin 374 May 5 09:06 org.linkedin.glu.packaging-set-up-2.2.3/
drwxr-xr-x 5 ypujante oadmin 374 May 13 09:11 org.linkedin.glu.packaging-set-up-2.2.3.0/
drwxr-xr-x 5 ypujante oadmin 374 May 13 08:15 org.linkedin.glu.packaging-set-up-2.3.0.dev.1-SNAPSHOT/
drwxr-xr-x 5 ypujante oadmin 374 May 16 14:59 org.linkedin.glu.packaging-set-up-2.3.1-SNAPSHOT/
drwxr-xr-x 5 ypujante oadmin 374 May 18 10:45 org.linkedin.glu.packaging-set-up-2.3.2-SNAPSHOT/
drwxr-xr-x 5 ypujante oadmin 374 May 20 09:29 org.linkedin.glu.packaging-set-up-2.4.0/
drwxr-xr-x 5 ypujante oadmin 374 May 24 09:15 org.linkedin.glu.packaging-set-up-2.4.1/
drwxr-xr-x 5 ypujante oadmin 374 Jun 22 09:29 org.linkedin.glu.packaging-set-up-3.0.0-SNAPSHOT/
drwxr-xr-x 5 ypujante oadmin 374 Jun 19 07:30 org.linkedin.glu.packaging-set-up-3.0.0.RC2-SNAPSHOT/
(ypujante@beon:/export/content/glu/ 3)ll 18:04:20
Terminal — zsh — 80x24
-rw-r--r-- 1 ypujante oadmin 544 Jun 22 09:20 org.linkedin.glu.packaging-all-3.8.8-SNAPSHOT/
drwxr-xr-x 18 ypujante oadmin 578 Jun 19 07:49 org.linkedin.glu.packaging-all-3.8.8.RC2-SNAPSHOT/
drwxr-xr-x 5 ypujante oadmin 374 May 5 09:06 org.linkedin.glu.packaging-set-up-2.2.3/
drwxr-xr-x 5 ypujante oadmin 374 May 13 09:11 org.linkedin.glu.packaging-set-up-2.2.3.0/
drwxr-xr-x 5 ypujante oadmin 374 May 13 08:15 org.linkedin.glu.packaging-set-up-2.3.0.dev.1-SNAPSHOT/
drwxr-xr-x 5 ypujante oadmin 374 May 16 14:59 org.linkedin.glu.packaging-set-up-2.3.1-SNAPSHOT/
drwxr-xr-x 5 ypujante oadmin 374 May 18 10:45 org.linkedin.glu.packaging-set-up-2.3.2-SNAPSHOT/
drwxr-xr-x 5 ypujante oadmin 374 May 20 09:29 org.linkedin.glu.packaging-set-up-2.4.0/
drwxr-xr-x 5 ypujante oadmin 374 May 24 09:15 org.linkedin.glu.packaging-set-up-2.4.1/
drwxr-xr-x 5 ypujante oadmin 374 Jun 22 09:29 org.linkedin.glu.packaging-set-up-3.0.0-SNAPSHOT/
drwxr-xr-x 5 ypujante oadmin 374 Jun 19 07:30 org.linkedin.glu.packaging-set-up-3.0.0.RC2-SNAPSHOT/
(ypujante@beon:/export/content/glu/ 4)ll 18:04:20
Terminal — zsh — 80x24
APSHOT
(ypujante@beon:~/content/glu/org.linkedin.glu.packaging-all-3.8.8-SNAPSHOT/ 4)
drwxr-xr-x 1 ypujante oadmin 11356 Nov 7 2010 LICENSE.txt
drwxr-xr-x 1 ypujante oadmin 26117 Apr 22 10:07 NOTICE.txt
drwxr-xr-x 1 ypujante oadmin 1447 Apr 25 00:36 README.ad
drwxr-xr-x 1 ypujante oadmin 487 Apr 22 14:41 README_FIRST.txt
drwxr-xr-x 1 ypujante oadmin 9208 Jun 20 14:28 RELEASE.rst
drwxr-xr-x 1 ypujante oadmin 12569 Nov 7 2010 RESTLET-2.0.1-EPL-LICENSE.html
drwxr-xr-x 1 ypujante oadmin 18858 Nov 7 2010 SIGAR-1.6.4-LICENSE.txt
drwxr-xr-x 5 ypujante oadmin 178 Jun 22 09:28 agent-cli/
drwxr-xr-x 4 ypujante oadmin 178 Jun 22 09:29 agent-server/
drwxr-xr-x 2 ypujante oadmin 374 Jun 22 09:28 bin/
drwxr-xr-x 4 ypujante oadmin 136 Jun 22 09:28 console-cli/
drwxr-xr-x 6 ypujante oadmin 204 Jun 22 09:28 console-server/
drwxr-xr-x 7 ypujante oadmin 374 Jun 22 09:29 org.linkedin.zookeeper-server-1.3.0/
drwxr-xr-x 5 ypujante oadmin 178 Jun 22 09:28 setup/
(ypujante@beon:/content/glu/org.linkedin.glu.packaging-all-3.8.8-SNAPSHOT/ 6)ll 18:04:20
Terminal — zsh — 80x24
-rw-r--r-- 1 ypujante oadmin 487 Jun 12 10:24 org.linkedin.util-core-1.7.8-
sources.jar.adoc
drwxr-xr-x 1 ypujante oadmin 32 Jun 12 10:22 org.linkedin.util-core-1.7.8-
sources.jar.adoc5
drwxr-xr-x 1 ypujante oadmin 48 Jun 12 10:22 org.linkedin.util-core-1.7.8-
sources.jar.shot
drwxr-xr-x 1 ypujante oadmin 153425 Jun 12 10:22 org.linkedin.util-core-1.7.8-
jar
drwxr-xr-x 1 ypujante oadmin 487 Jun 12 10:24 org.linkedin.util-core-1.7.8-
jar.adoc
drwxr-xr-x 1 ypujante oadmin 32 Jun 12 10:22 org.linkedin.util-core-1.7.8-
jar.adoc5
drwxr-xr-x 1 ypujante oadmin 48 Jun 12 10:22 org.linkedin.util-core-1.7.8-
jar.shot
drwxr-xr-x 1 ypujante oadmin 1839 Jun 12 10:22 org.linkedin.util-core-1.7.8-
pos
drwxr-xr-x 1 ypujante oadmin 487 Jun 12 10:24 org.linkedin.util-core-1.7.8-
pos.adoc
drwxr-xr-x 1 ypujante oadmin 32 Jun 12 10:22 org.linkedin.util-core-1.7.8-
pos.adoc5
drwxr-xr-x 1 ypujante oadmin 48 Jun 12 10:22 org.linkedin.util-core-1.7.8-
pos.shot
drwxr-xr-x 2 ypujante oadmin 182 Jun 12 10:24 target/
(ypujante@beon:~/linkedin/org.linkedin.util-core-1.7.8/ 9)ll 18:04:20
Terminal — zsh — 80x24
-rw-r--r-- 1 ypujante oadmin 15967 Nov 7 2010 NOTICE.txt
drwxr-xr-x 1 ypujante oadmin 3973 Dec 26 2010 README.ad
drwxr-xr-x 1 ypujante oadmin 659 Apr 30 11:18 RELEASE.ad
drwxr-xr-x 2 ypujante oadmin 284 Apr 30 11:11 bin/
drwxr-xr-x 2 ypujante oadmin 136 Apr 30 11:13 conf/
drwxr-xr-x 3 ypujante oadmin 182 Jun 12 10:47 data/
drwxr-xr-x 2 ypujante oadmin 272 Apr 30 11:13 lib/
drwxr-xr-x 2 ypujante oadmin 182 Apr 30 11:14 logs/
(ypujante@beon:~/content/linkedin-zookeeper/org.linkedin.zookeeper-server-1.3.0-
4)ll 18:05:56
total 72
drwxr-xr-x 7 ypujante oadmin 374 Apr 30 11:13 ./
drwxr-xr-x 6 ypujante oadmin 284 Apr 30 11:13 ../
drwxr-xr-x 1 ypujante oadmin 11358 Nov 7 2010 LICENSE.txt
drwxr-xr-x 1 ypujante oadmin 15967 Nov 7 2010 NOTICE.txt
drwxr-xr-x 1 ypujante oadmin 3973 Dec 26 2010 README.ad
drwxr-xr-x 2 ypujante oadmin 659 Apr 30 11:18 RELEASE.ad
drwxr-xr-x 2 ypujante oadmin 284 Apr 30 11:11 bin/
drwxr-xr-x 3 ypujante oadmin 182 Jun 12 10:47 data/
drwxr-xr-x 2 ypujante oadmin 272 Apr 30 11:13 lib/
drwxr-xr-x 2 ypujante oadmin 182 Apr 30 11:14 logs/
(ypujante@beon:~/content/linkedin-zookeeper/org.linkedin.zookeeper-server-1.3.0-
5)ll 18:05:56

```



deployment automation  
platform

# Before glu...

- Operations performs manual deployment:
  - ssh, rcp, etc...
  - non shared manually edited scripts
- extremely time-consuming
- error prone

# glu project

- Address operations pain points
- Deploy (and monitor) applications to an arbitrary large set of nodes:
  - efficiently
  - with minimum/no human interaction
  - securely
  - in a reproducible manner
- ensure consistency over time (prevent drifting)
- detect and troubleshoot quickly when problems arise



# After...

glu-dev-1 | All [product]

Dashboard Plans System Model Admin admin Help

Quick Select: [Select None](#) | [Select First](#) | [Select All](#) | [25%](#) | [33%](#) | [50%](#) | [66%](#) | [75%](#)

[Filter](#) [Execute](#)

Deploy - Fabric [glu-dev-1] - PARALLEL [12]

Deploy - Fabric [glu-dev-1] - PARALLEL

[agent-1 - /sample/i001](#)

- 1. Install script for [/sample/i001] on [agent-1]
- 2. Run [install] phase for [/sample/i001] on [agent-1]
- 3. Run [configure] phase for [/sample/i001] on [agent-1]
- 4. Run [start] phase for [/sample/i001] on [agent-1]

[agent-1 - /sample/i002](#)

- 1. Install script for [/sample/i002] on [agent-1]
- 2. Run [install] phase for [/sample/i002] on [agent-1]
- 3. Run [configure] phase for [/sample/i002] on [agent-1]
- 4. Run [start] phase for [/sample/i002] on [agent-1]

[agent-1 - /sample/i003](#)

- 1. Install script for [/sample/i003] on [agent-1]
- 2. Run [install] phase for [/sample/i003] on [agent-1]
- 3. Run [configure] phase for [/sample/i003] on [agent-1]
- 4. Run [start] phase for [/sample/i003] on [agent-1]



The page at [localhost:8080](http://localhost:8080) says:  
Are you sure you want to execute this plan ?

[Cancel](#) [OK](#)

Click me!

# After...

:)

glu-dev-1 | All [product]

Dashboard Plans System Model Admin admin Help

Recent Deploy - Fabric [glu-dev-1] - PARALLEL

Show Errors Only:  Auto Refresh:  

Refresh Pause Abort

 0/12 - 0%

Deploy - Fabric [glu-dev-1] - PARALLEL

Deploy - Fabric [glu-dev-1] - PARALLEL - running [1s]

agent-1 - /sample/i001 - running [1s]  
Install script for [/sample/i001] on [agent-1] - running [1s]- [ Cancel ]  
Run [install] phase for [/sample/i001] on [agent-1] - not started  
Run [configure] phase for [/sample/i001] on [agent-1] - not started  
Run [start] phase for [/sample/i001] on [agent-1] - not started

agent-1 - /sample/i002 - running [1s]  
Install script for [/sample/i002] on [agent-1] - running [1s]- [ Cancel ]  
Run [install] phase for [/sample/i002] on [agent-1] - not started  
Run [configure] phase for [/sample/i002] on [agent-1] - not started  
Run [start] phase for [/sample/i002] on [agent-1] - not started

agent-1 - /sample/i003 - running [1s]  
Install script for [/sample/i003] on [agent-1] - running [1s]- [ Cancel ]  
Run [install] phase for [/sample/i003] on [agent-1] - not started  
Run [configure] phase for [/sample/i003] on [agent-1] - not started  
Run [start] phase for [/sample/i003] on [agent-1] - not started

Nothing to do here...  
Sit back and enjoy!



# After...

:D

glu-dev-1 | All [product]

Dashboard Plans System Model Admin admin Help

Recent Deploy - Fabric [glu-dev-1] - PARALLEL

Show Errors Only:  Auto Refresh:

**12/12 - 100%**

Deploy - Fabric [glu-dev-1] - PARALLEL

Deploy - Fabric [glu-dev-1] - PARALLEL - 16s

agent-1 - /sample/i001 - 16s

- Install script for [/sample/i001] on [agent-1] - 1s
- Run [install] phase for [/sample/i001] on [agent-1] - 3s
- Run [configure] phase for [/sample/i001] on [agent-1] - 2s
- Run [start] phase for [/sample/i001] on [agent-1] - 8s

agent-1 - /sample/i002 - 12s

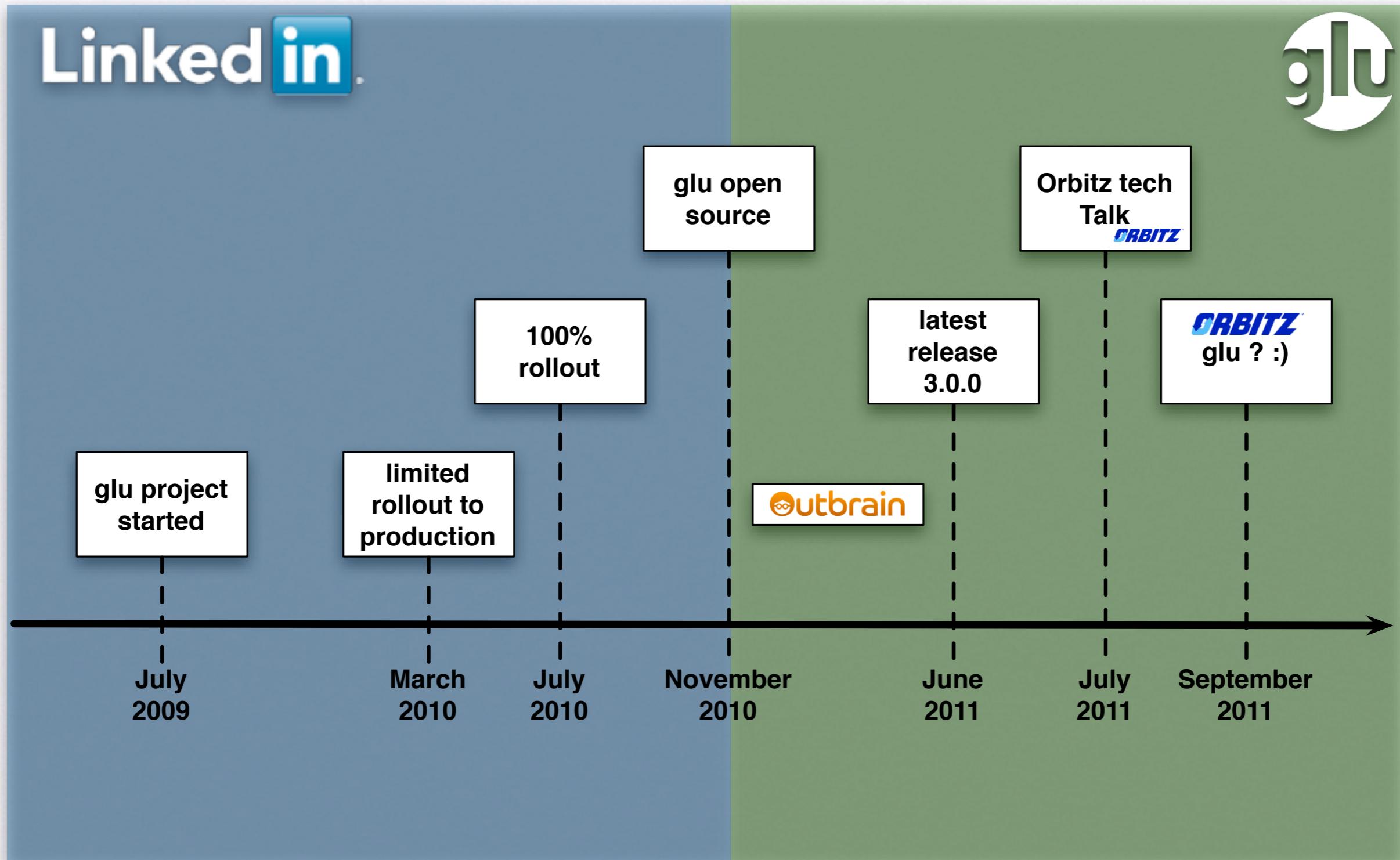
- Install script for [/sample/i002] on [agent-1] - 2s
- Run [install] phase for [/sample/i002] on [agent-1] - 2s
- Run [configure] phase for [/sample/i002] on [agent-1] - 1s
- Run [start] phase for [/sample/i002] on [agent-1] - 5s

agent-1 - /sample/i003 - 11s

- Install script for [/sample/i003] on [agent-1] - 2s
- Run [install] phase for [/sample/i003] on [agent-1] - 2s
- Run [configure] phase for [/sample/i003] on [agent-1] - 1s
- Run [start] phase for [/sample/i003] on [agent-1] - 4s



# History of glu



# Rollout to production

- glu project started in July 2009
- Initial rollout to LinkedIn production in March 2010
- Gradual until full rollout in July 2010
- As of June 2011 LinkedIn glu numbers:
  - 5 different 'fabrics' (2 prod + 2 stg + 1 int. lab)
  - ~2650 nodes, ~9000 instances, ~300 services
- LinkedIn working on 'glu on the desktop' (dev)



# glu open source

- Before I left LinkedIn, open sourced glu (~3 months effort)
  - 1.0.0 released in November 2010
  - 2.0.0 released in February 2011 (tagging)
  - 3.0.0 released in June 2011 (parent/child)
  - (~ 20 releases total... smaller releases)



# glu interest

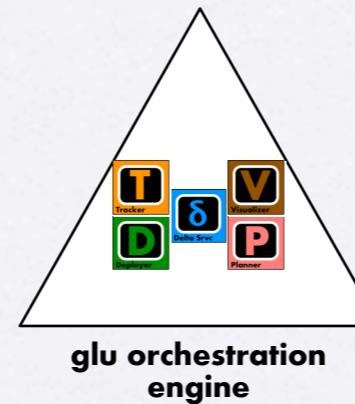
- since 11/2010, glu has generated a lot of interest
  - oubrain.com is using glu (integrated in CI!)
  - companies interested in glu: Orbitz, Netflix, GigaSpaces, Rearden Commerce, etc...
  - some academic use (Budapest university)
  - a lot of 'followers' on github
  - lots of downloads



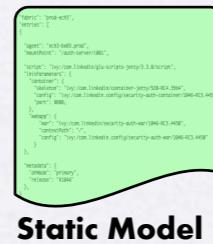
# Architecture

# Components/Concepts

- 3 physical components



- 3 concepts



Static Model



Live Model



# ZooKeeper



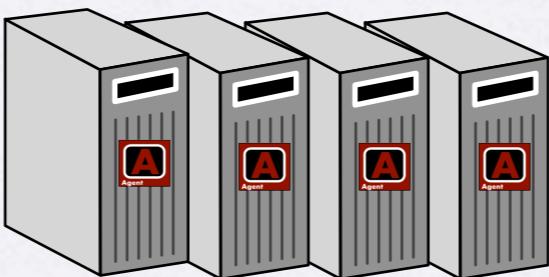
- 1 ZooKeeper cluster (3 or 5 instances enough)
- ZooKeeper is an Apache project
- similar to a (networked) filesystem (think nfs)
  - + 'directories' can also contain data
  - + ephemeral nodes
  - + powerful watcher concept => notifications
- ZooKeeper is used to maintain the state of the system



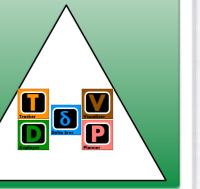
# glu Agent



- 1 agent per node => as many agents as there are nodes
- agent is active process (groovy)
- (secure) REST API
- Reports its state to ZooKeeper

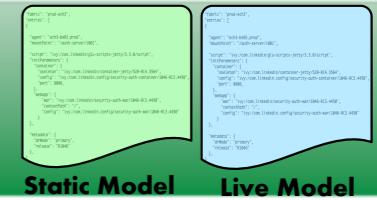


# glu orchestration engine



- 1 orchestration engine
- runs inside a webapp
  - offers both browser and REST interface
- Listens to ZooKeeper events (to compute 'live state')
- Talks to the agents

# Static/Live Model



- model is a json document which describes
    - where to deploy
    - what and how to deploy
  - “Static” is what you want
  - “Live” is what is actually deployed/running



# Static Model: Where ?

```
{  
    "fabric": "prod-chicago",  
    "entries": [ {  
        "agent": "node01.prod",  
        "mountPoint": "/search/i001",  
  
        "script": "http://repository.prod/scripts/webapp-deploy-1.0.0.groovy",  
        "initParameters": {  
            "container": {  
                "skeleton": "http://repository.prod/tgzs/jetty-7.2.2.v20101205.tgz",  
                "port": 8080,  
            },  
            "webapp": {  
                "war": "http://repository.prod/wars/search-2.1.0.war",  
                "contextPath": "/"  
            } } ] } } }
```

- “agent” => node which runs this agent
- “mountPoint” => unique key
- can deploy more than 1 ‘thing’ per agent

# Static Model: What / How ?

```
{  
  "fabric": "prod-chicago",  
  "entries": [{  
    "agent": "node01.prod",  
    "mountPoint": "/search/i001",  
  
    "script": "http://repository.prod/scripts/webapp-deploy-1.0.0.groovy",  
    "initParameters": {  
      "container": {  
        "skeleton": "http://repository.prod/tgzs/jetty-7.2.2.v20101205.tgz",  
        "port": 8080,  
      },  
      "webapp": {  
        "war": "http://repository.prod/wars/search-2.1.0.war",  
        "contextPath": "/"  
      }  
    }  
  }]  
}
```

- “script” => instructions about what ‘deploy’ means
- “initParameters” => parameters provided to the script

# glu Script

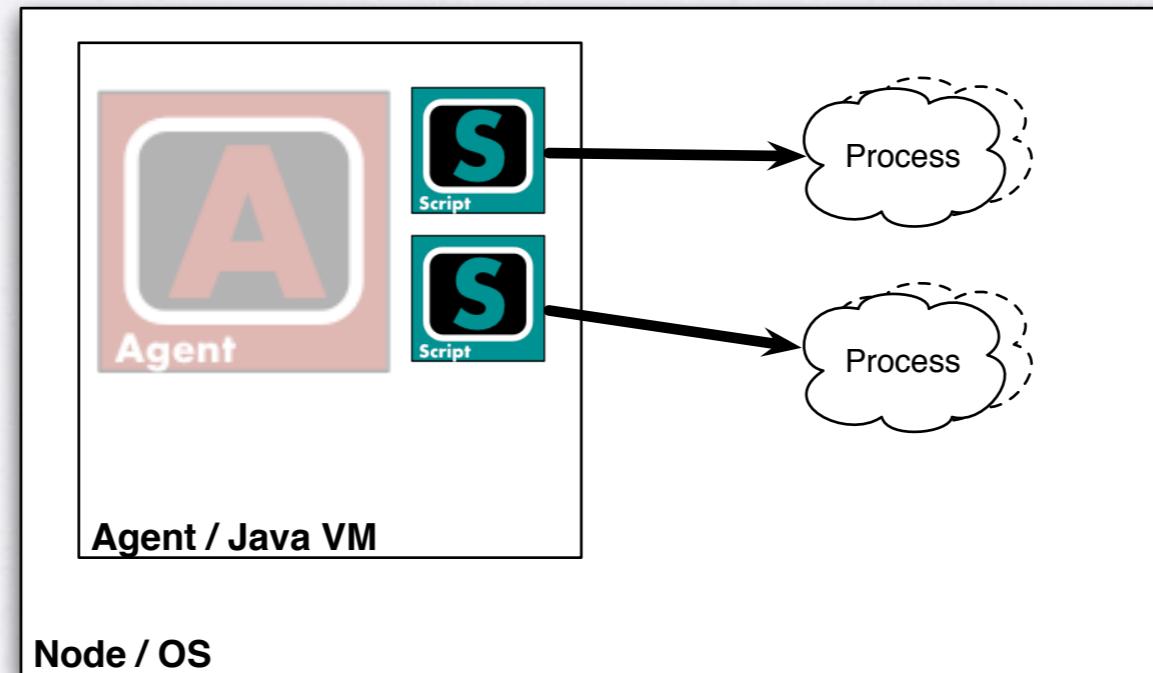


```
class MyGluScript
{
    def port
    def pid

    def install      = { /* install code */ }
    def configure   = { /* configure code */ }
    def start        = { /* start code */ }
    def stop         = { /* stop code */ }
    def unconfigure = { /* unconfigure code */ }
    def uninstall   = { /* uninstall code */ }
}
```

- groovy class which defines
  - a set of ‘phases’ (install, start, etc...) backed by a state machine
  - properties (exported to ZooKeeper)
- glu does not dictate what goes in each ‘phase’

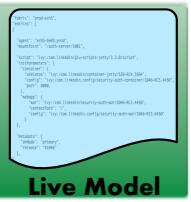
# glu Script runtime



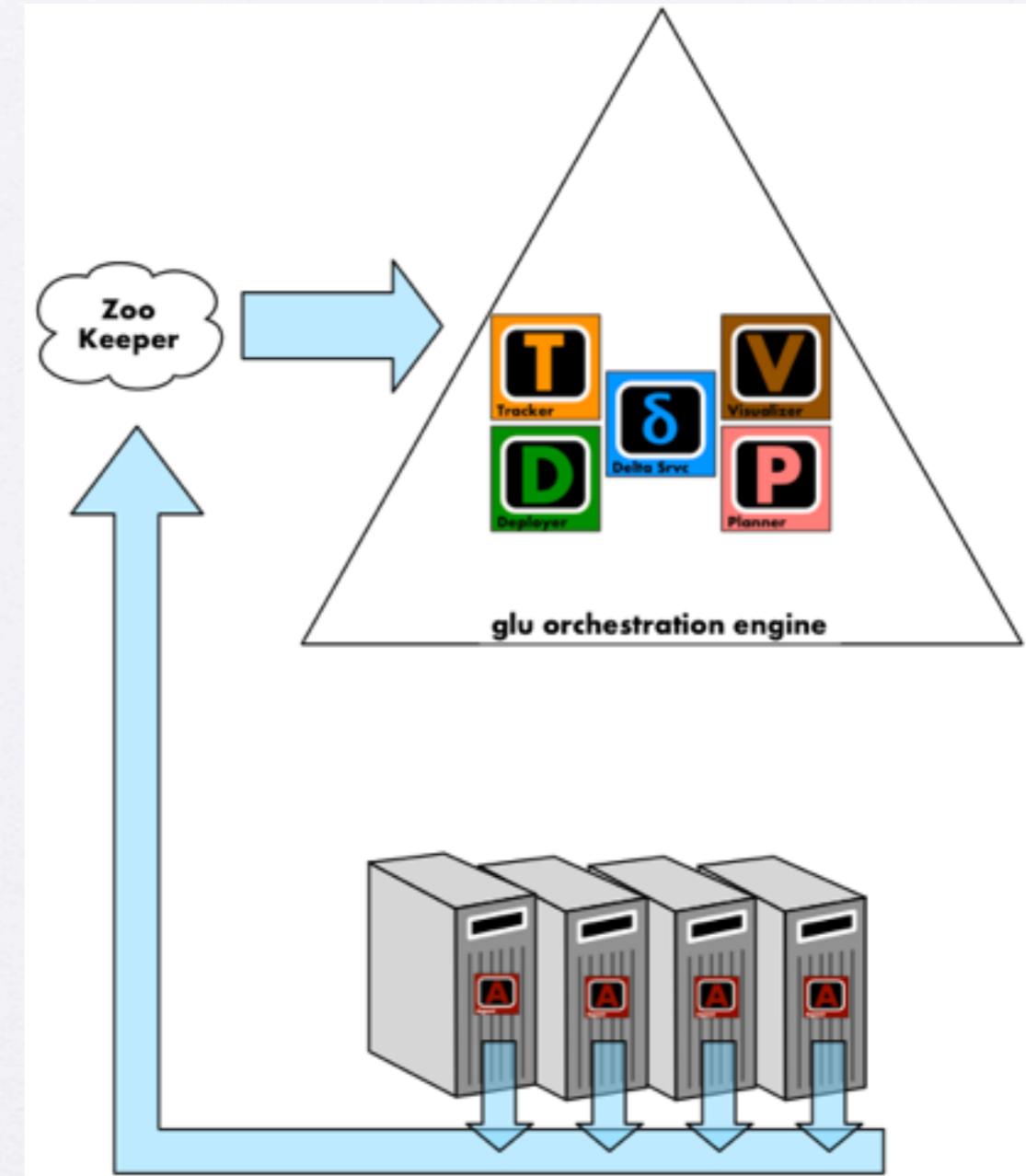
- glu Script code runs inside the (java) VM of the agent
- in general, a glu Script will spawn external processes (ex: webapp container, memcached, etc...) but it is not a requirement!

# How does it all work ?

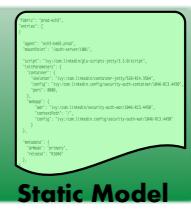
# Live Model



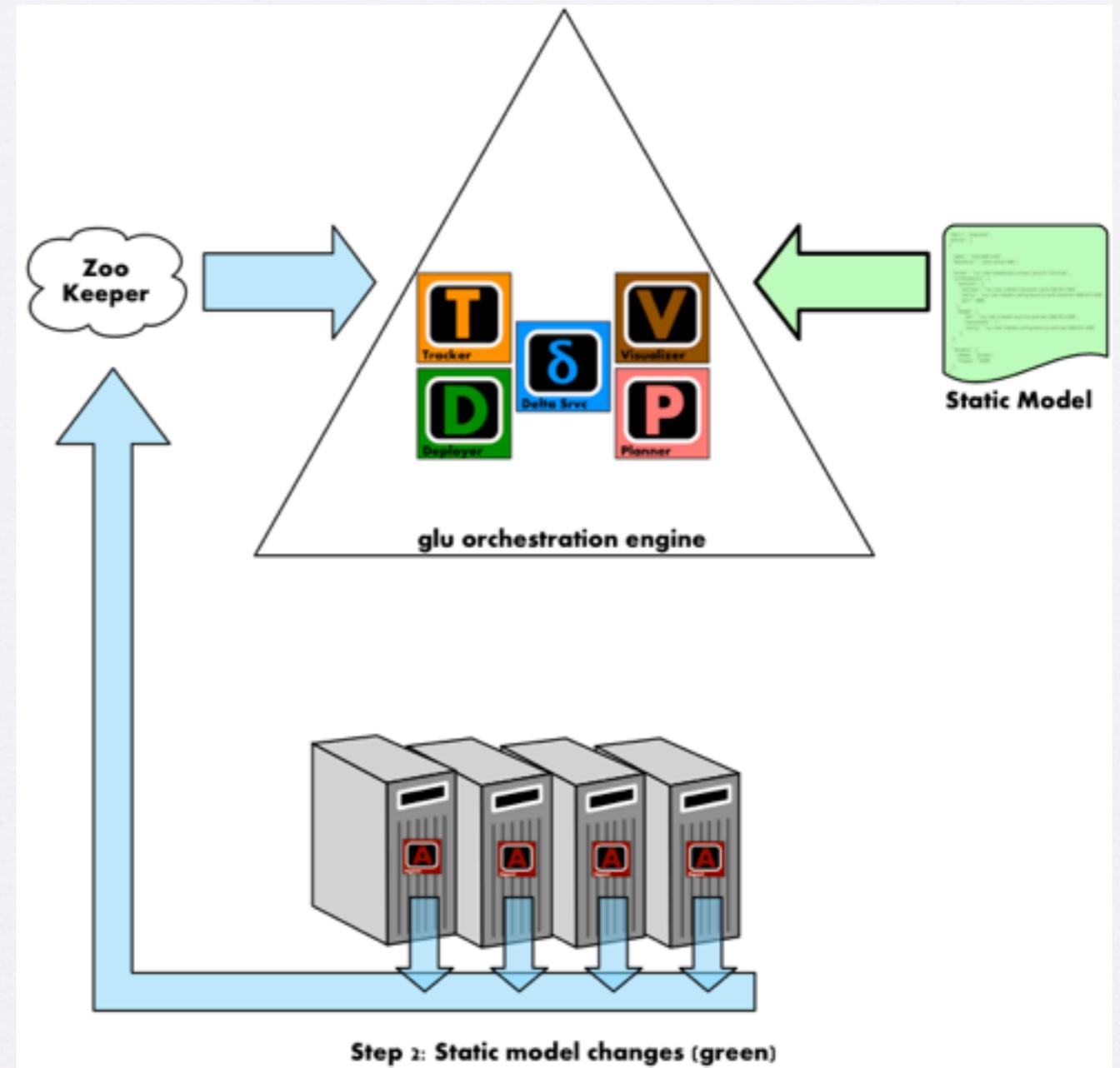
- each agent reports its state to ZooKeeper
- the orchestration engine listens to ZooKeeper and builds the 'live' model



# Static Model

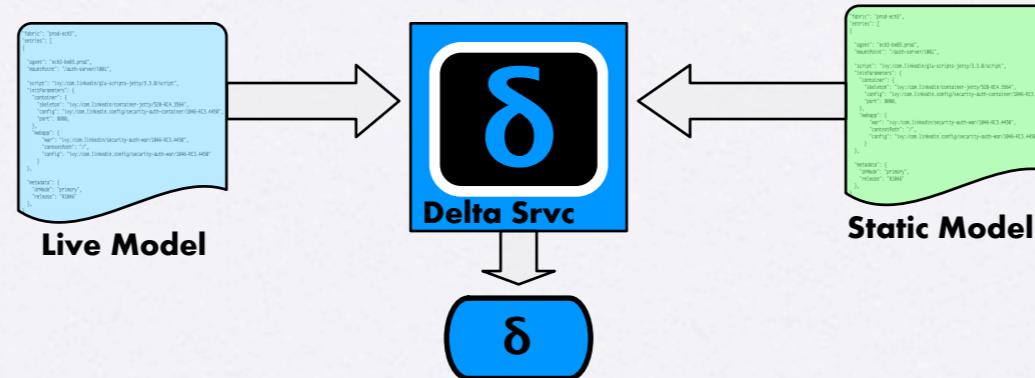


- the 'static' model is loaded in the orchestration engine



# Delta Computation

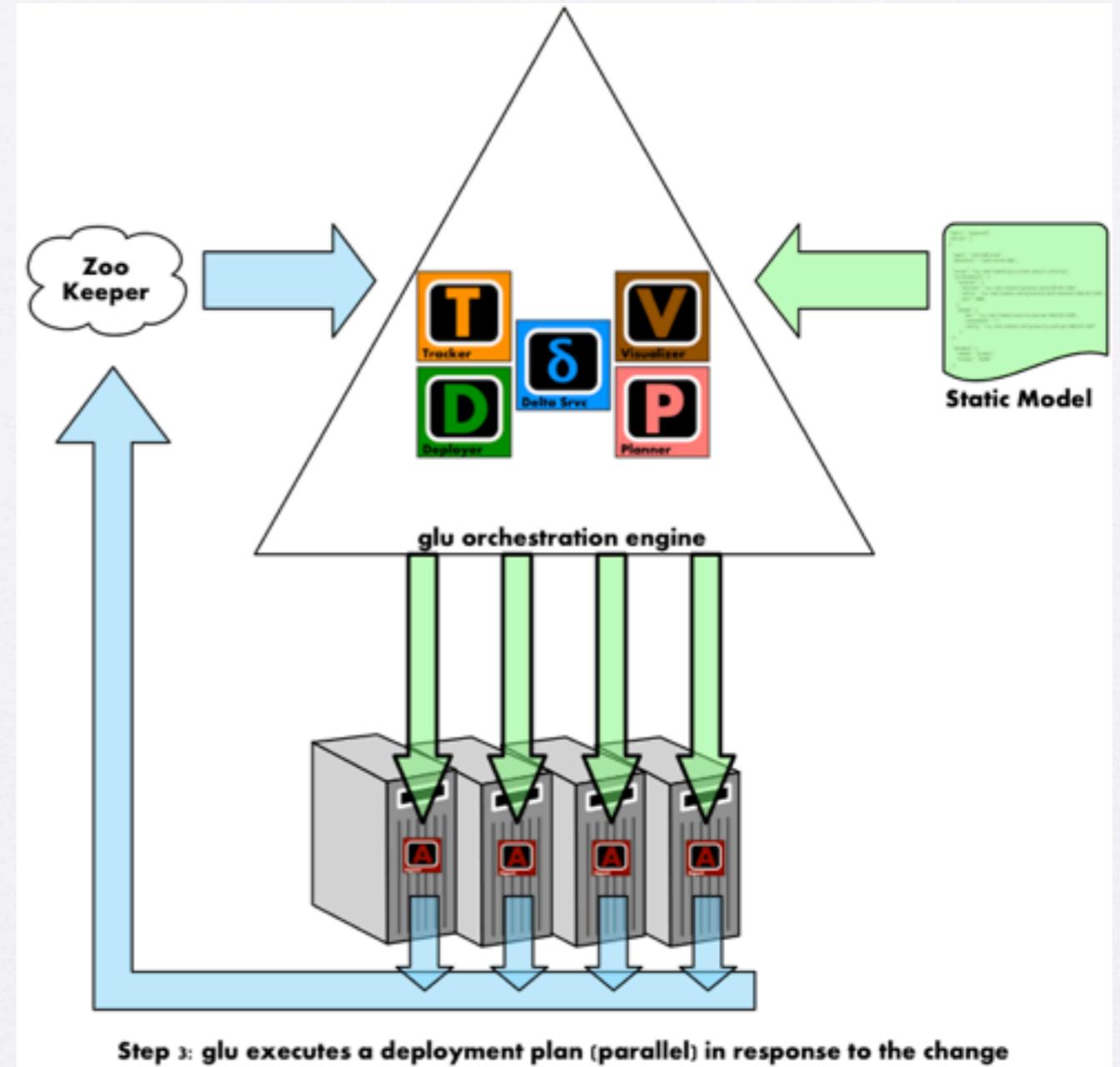
8



- orchestration engine computes a delta by comparing the static model and the live model
  - “desired” state vs “current” state

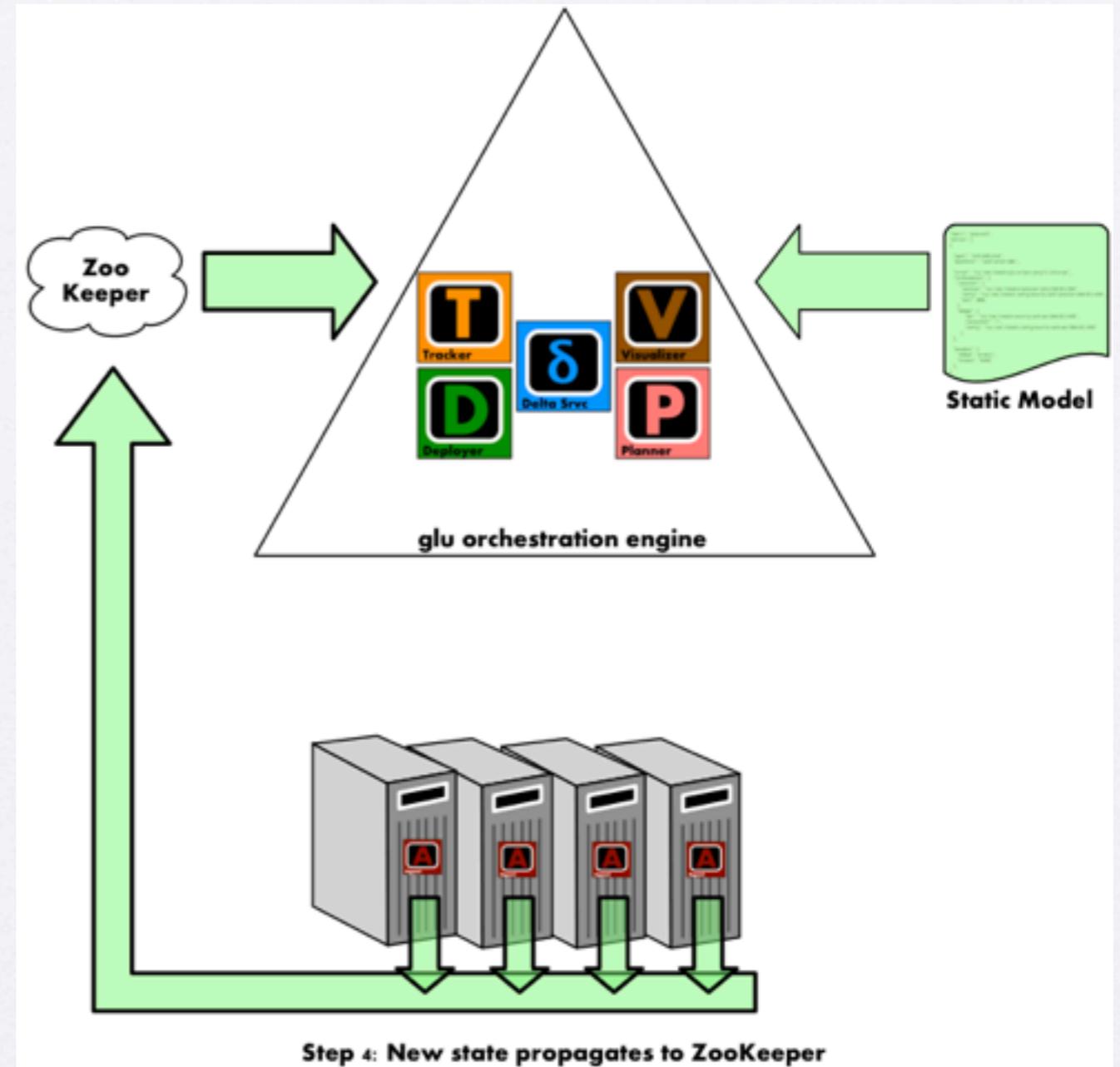
# deployment plan

- delta is used to compute a deployment plan
- orchestration engine sends commands (REST) to the appropriate agents



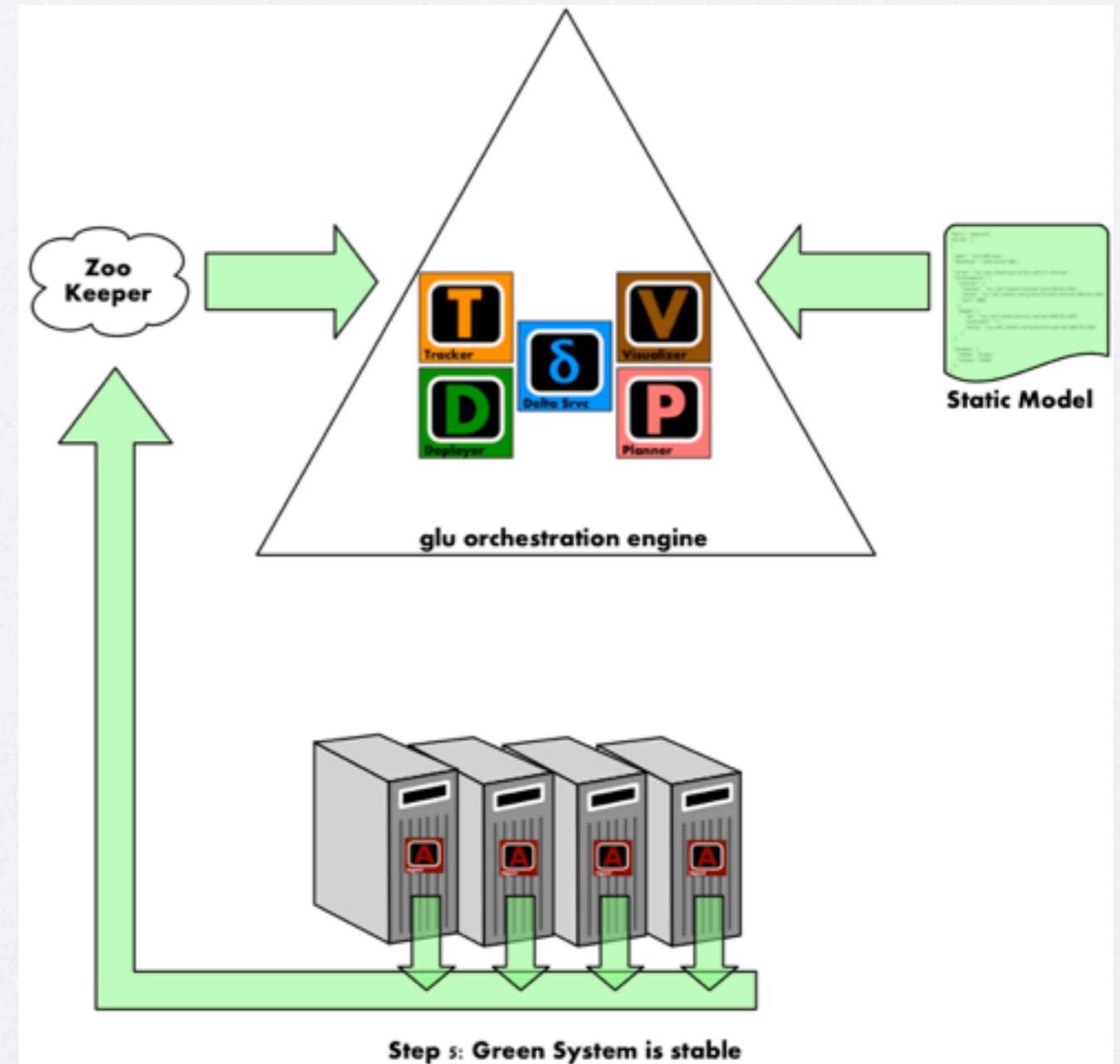
# Live Model updated

- as the agents run the commands they update their state in ZooKeeper



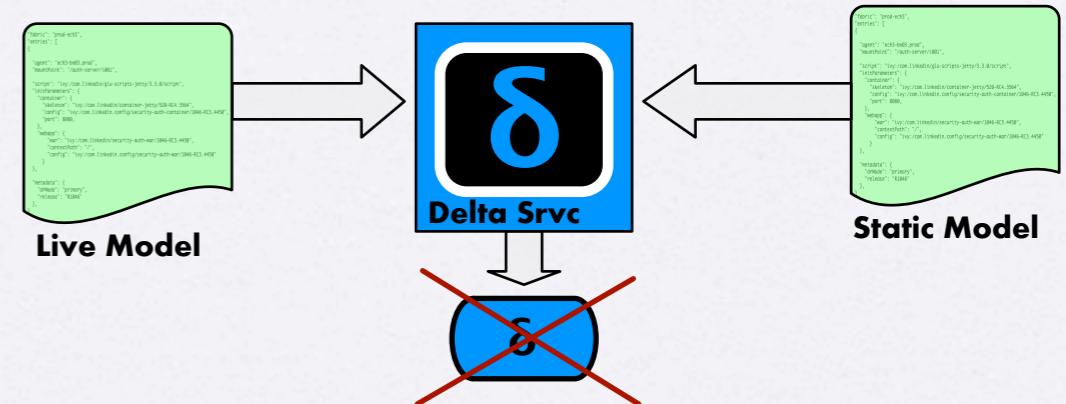
# System Stable

- The live model and the static model match
  - => no more delta

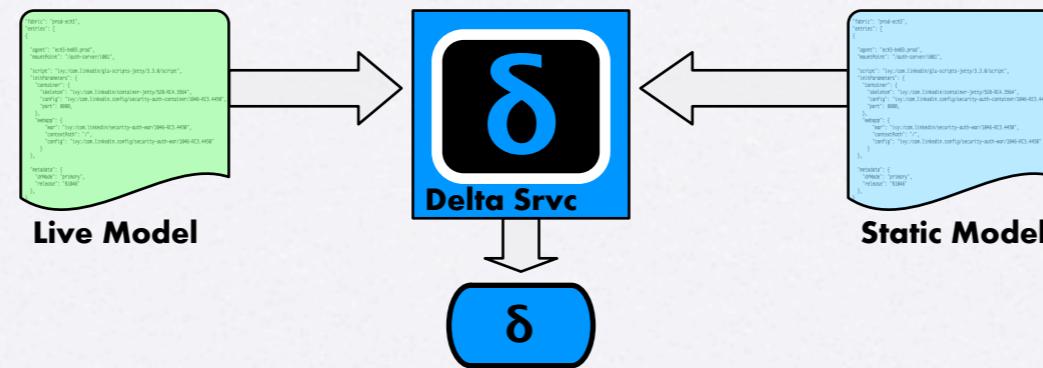


# System Stable (no delta)

- remains stable until:
  - static model changes (ex: new version of software)
  - live model changes (ex: hardware crash)

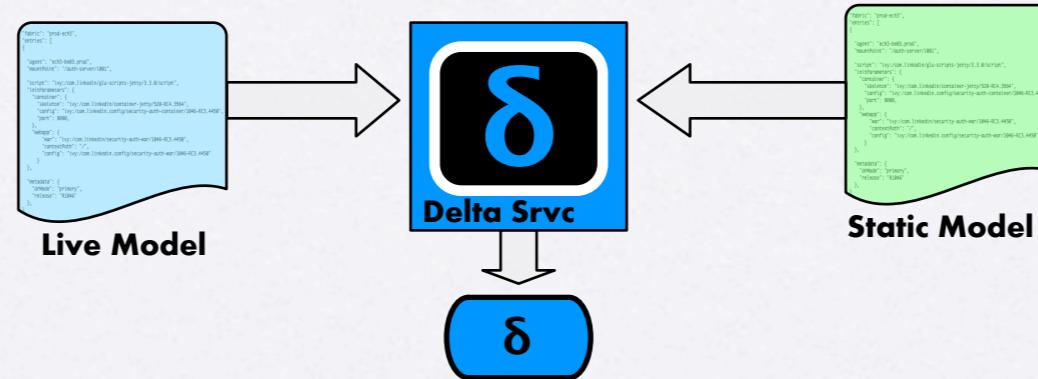


# Static Model Changes



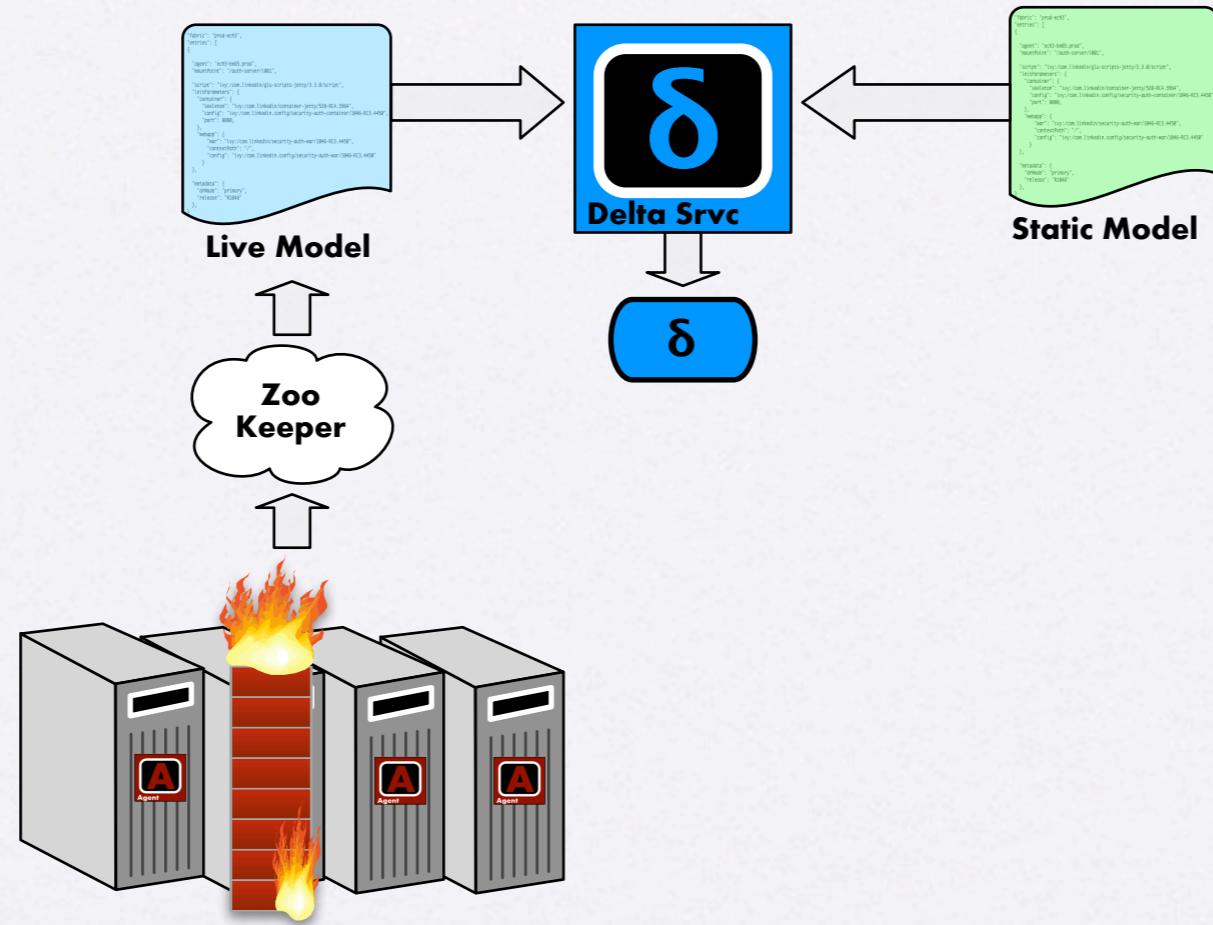
- Static model changes
  - ex: new version of software, new node, etc...
  - => delta => deploy/upgrade software, provision new nodes

# Live Model Changes



- Live Model changes
  - ex: hardware crash, bad behavior, high load, etc...
  - => delta => monitoring!

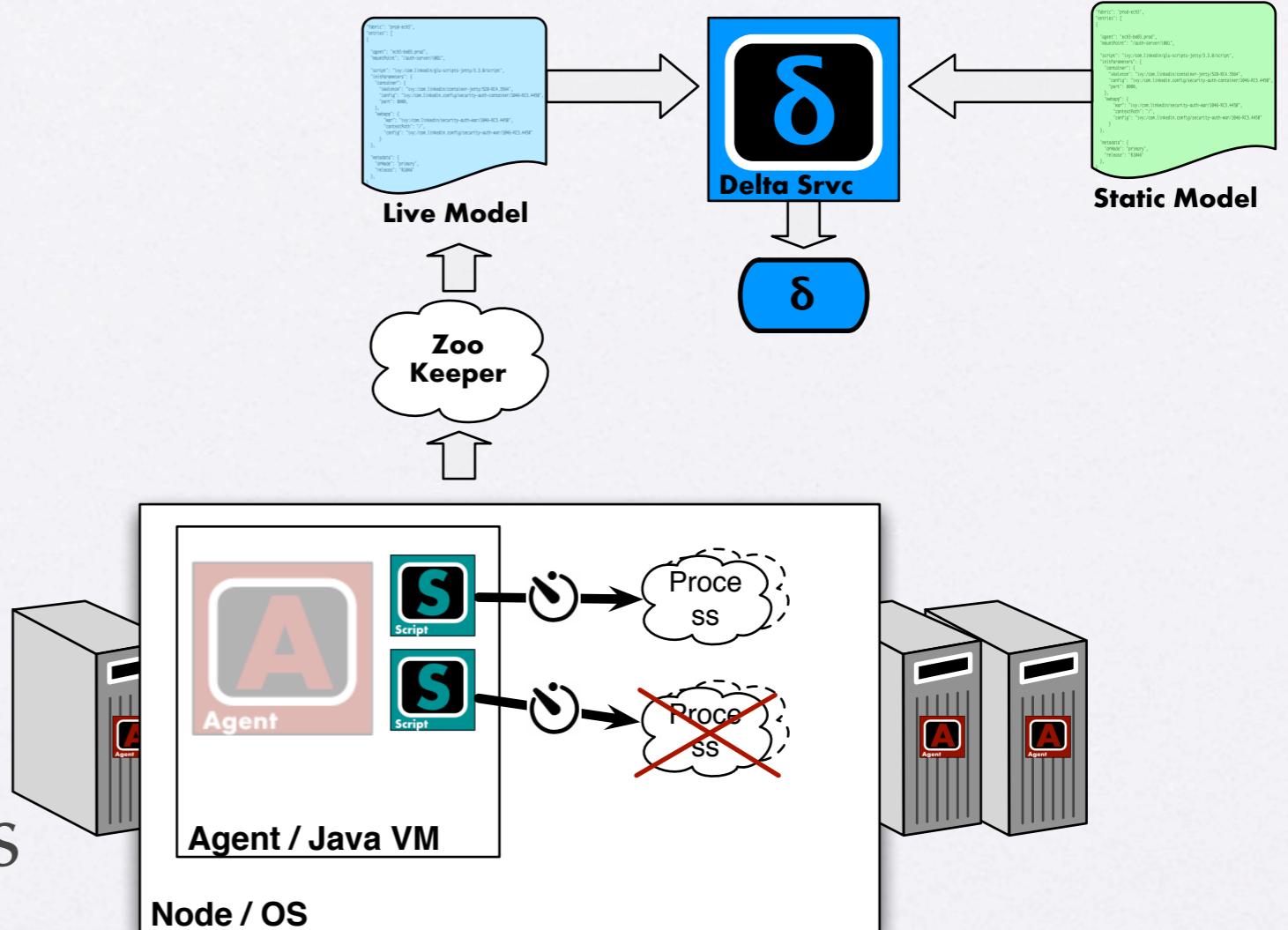
# Monitoring: built-in



- agent registers a ZooKeeper ephemeral node
- => when agent disappears, state changes!

# Monitoring: add-on

- script runs in “active” agent
- agent has “timer” capability
  - =>script can also monitor what it starts and change state when failure detected



# Monitoring: advanced

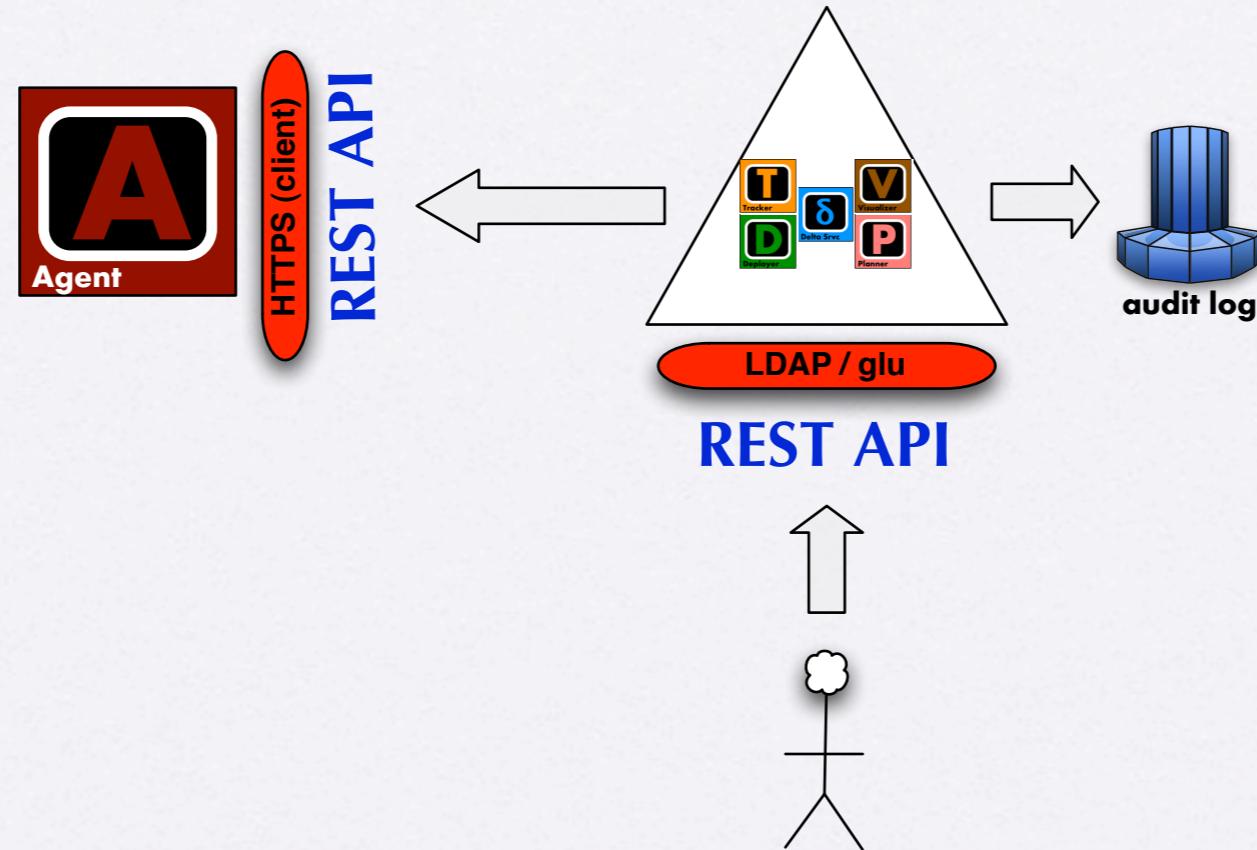
- You can even build a full monitoring solution on top of glu
- Not enough time/space here :)
- Check out my blog (source examples included!) @  
<http://www.pongasoft.com/blog/yan/categories/glu/>

mountPoint:4	I:4	E:1	agent:1	status:2
/monitor	1	1	agent-1	<span style="color: blue;">ERROR</span> High load detected...
/sample/i001	1	0	agent-1	running
/sample/i002	1	0	agent-1	running
/sample/i003	1	0	agent-1	running

# What about security ?



# Security



- User must authenticate (LDAP and/or glu)
- Agent REST API is ‘protected’ behind **HTTPSS** with client auth
- Every ‘change’ is audited in the audit log

# Live Demo...

\* You can see the live demo in the presentation given at Chicago devops (starts around 27:00):  
<http://devops.com/2011/07/09/glu-deployment-automation-video/>



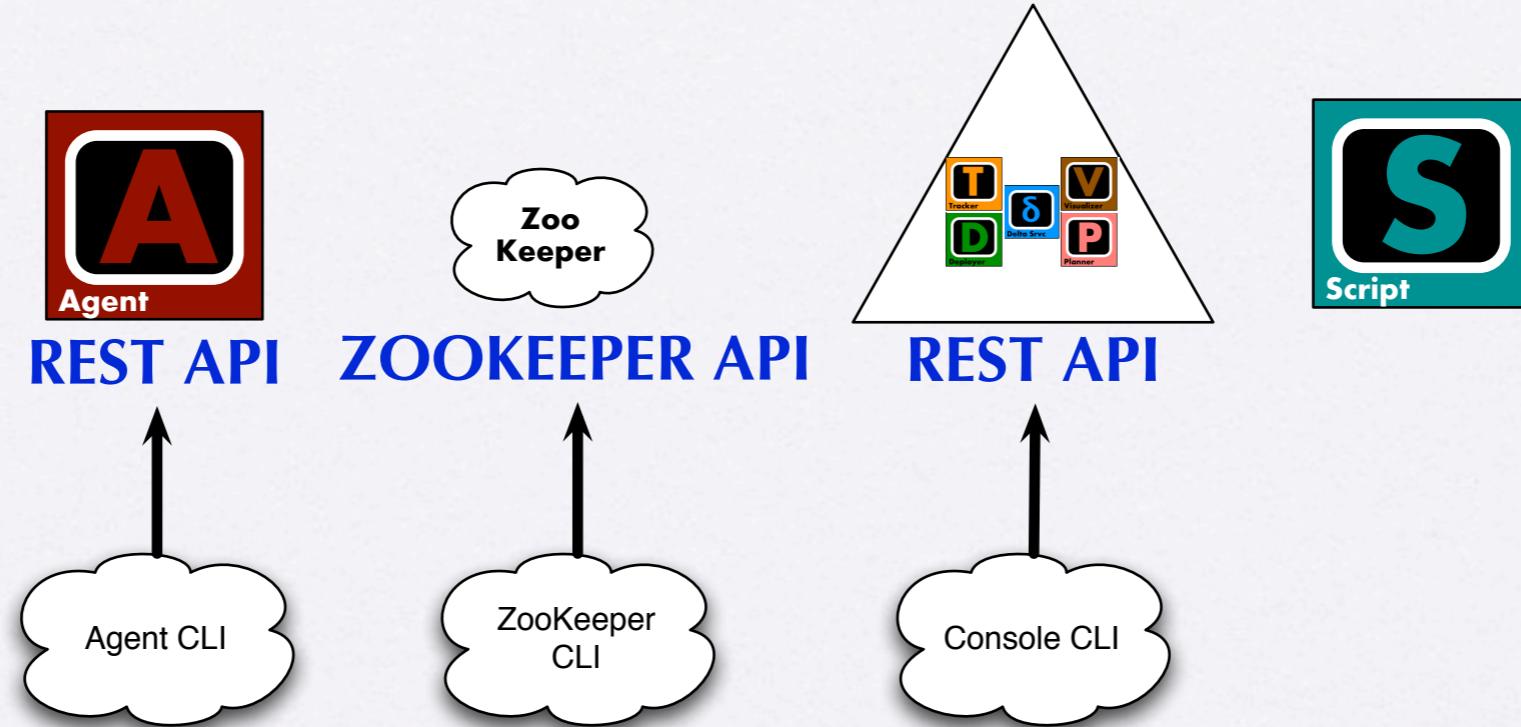
# glu as a platform

# glu is more than a tool

- glu is a tool with a lot of customization points
- it is also a **platform** on top of which you can build your **own** deployment (and optionally monitoring) solution



# APIs



- Agent CLI and Console CLI are mostly wrappers/examples around the REST API
- => you can use the REST API directly or use the CLI

# glu Script



- A glu script is any code you want (groovy/java) made easier by agent capabilities (but you don't have to use them!)
- **shell.exec** capability allow you to write your script in any language you want (will be 'promoted' native soon...)

```
class RubyGluScript {
    def install = {
        shell.exec("./ruby/install.rb")
    }
    def start = {
        shell.exec("./ruby/start.rb")
    }
}
```

# Agent



- One way to look at the agent: script engine remotely accessible through a (secure) REST API
  - => can also be used on its own (no ZooKeeper or orchestration engine)

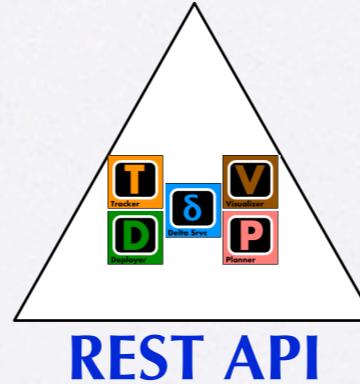
# ZooKeeper



## ZOOKEEPER API

- ZooKeeper is independently accessible
  - => can build your own listeners/watchers directly
  - => use AgentsTracker library which comes with glu  
(check the blog for more details)
  - Ex: build a monitoring solution

# Orchestration Engine



- For example, you can integrate your CI directly with glu by using the orchestration engine REST api (ex: outbrain.com)
- Although very customizable, you can also build your own UI if you do not like the one that comes with glu

# Much more...

- Powerful tagging/filtering feature allow to create concepts that glu does not know about (ex: webapp, frontend, cluster, etc...)
- Query language allows you to slice & dice the models
  - => build higher level constructs (like dynamic node assignment)



# glu vs puppet

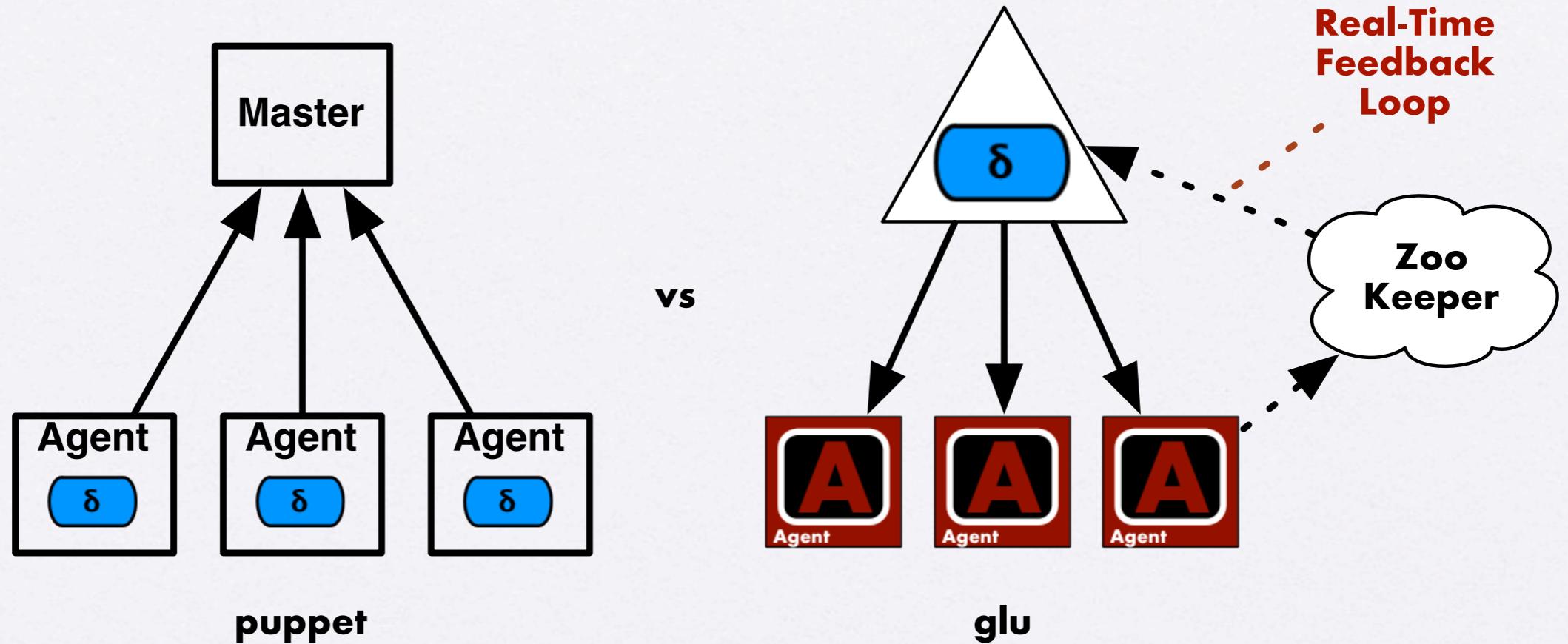
- \* Disclaimer: I have spent 2 years with glu (I wrote it :-) ) and 1 day with puppet...



# glu vs puppet

- Great news: intrinsically similar concepts
  - ‘desired’ vs ‘current’!
  - declarative approach
- Minor difference:
  - puppet is ruby vs glu is groovy/java

# glu vs puppet: orchestration



- delta computation / orchestration takes place at a different level
  - => glu can orchestrate across nodes
  - => glu delta is system wide (and real-time)

# glu vs puppet: conclusion

- puppet is very good at ***configuring*** the infrastructure of a machine (users, groups, packages, etc...)
  - => static/stable does not change often
- glu is very good at ***provisioning*** dynamic applications on an ensemble of machines (the system)
  - => changes often, real-time failure detection (monitoring), “bounce”, etc...



# glu can use puppet :)

```
class PuppetGluScript {
    def puppetManifest

    def install = {
        // download manifest
        puppetManifest = shell.fetch(params.puppetManifestURI)
    }

    def start = {
        // execute manifest
        shell.exec("puppet apply ${puppetManifest}")
    }
}
```



# References

# References

- glu source: [github.com/linkedin/glu](https://github.com/linkedin/glu) (links to all you need)
- blog: [www.pongasoft.com/blog/yan](http://www.pongasoft.com/blog/yan)
- twitter: @glutweets

