NFJS Software Symposium Series 2011

Glu-ing the last Mile

Ken Sipe - @kensipe

About Speaker

http://kensipe.blogspot.com/
http://del.icio.us/kensipe
twitter: @kensipe
ken.sipe@gmail.com

Developer: Embedded, C++, Java, Groovy, Grails, C#, Objective C
Speaker: JavaOne 2009 Rock Star, JavaZone, NFJS, JAX
Microsoft MCP
Sun Certified Java 2 Architect
Master of Scrums
Agile Coach
Instructor: VisiBroker CORBA
Rational Rose, OOAD

Agenda

- Continuous Delivery
- What is Glu
- Glu Configuration
- Demo

Principles behind the Agile Manifesto

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Continuous Delivery!!!

If you had 1 line of code change in dev...

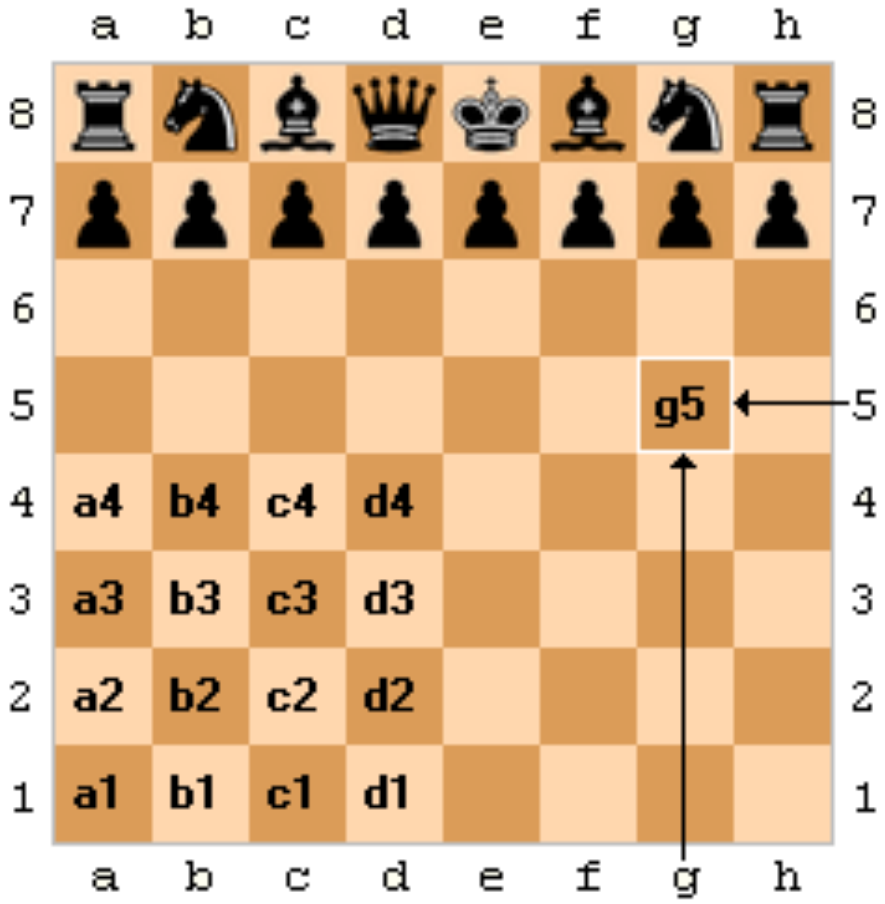
```
System.exit(0);
```

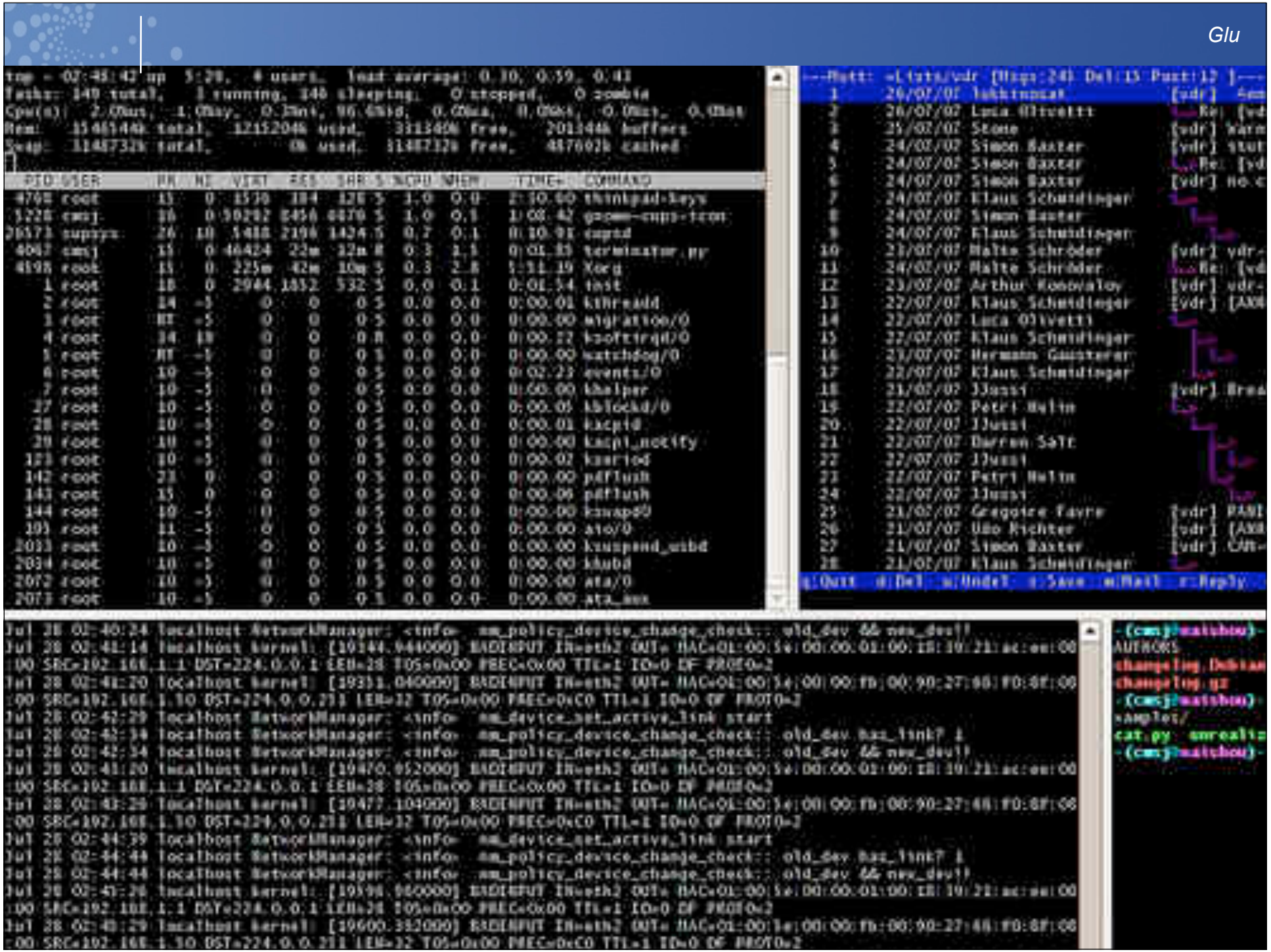
... how long would it take you to push that into production?

What does your production deployment night (or weekend :) look like?

- everyone is on call?
- midnight to 3 am?
- ???

Even with simple rules...
some things can be complex

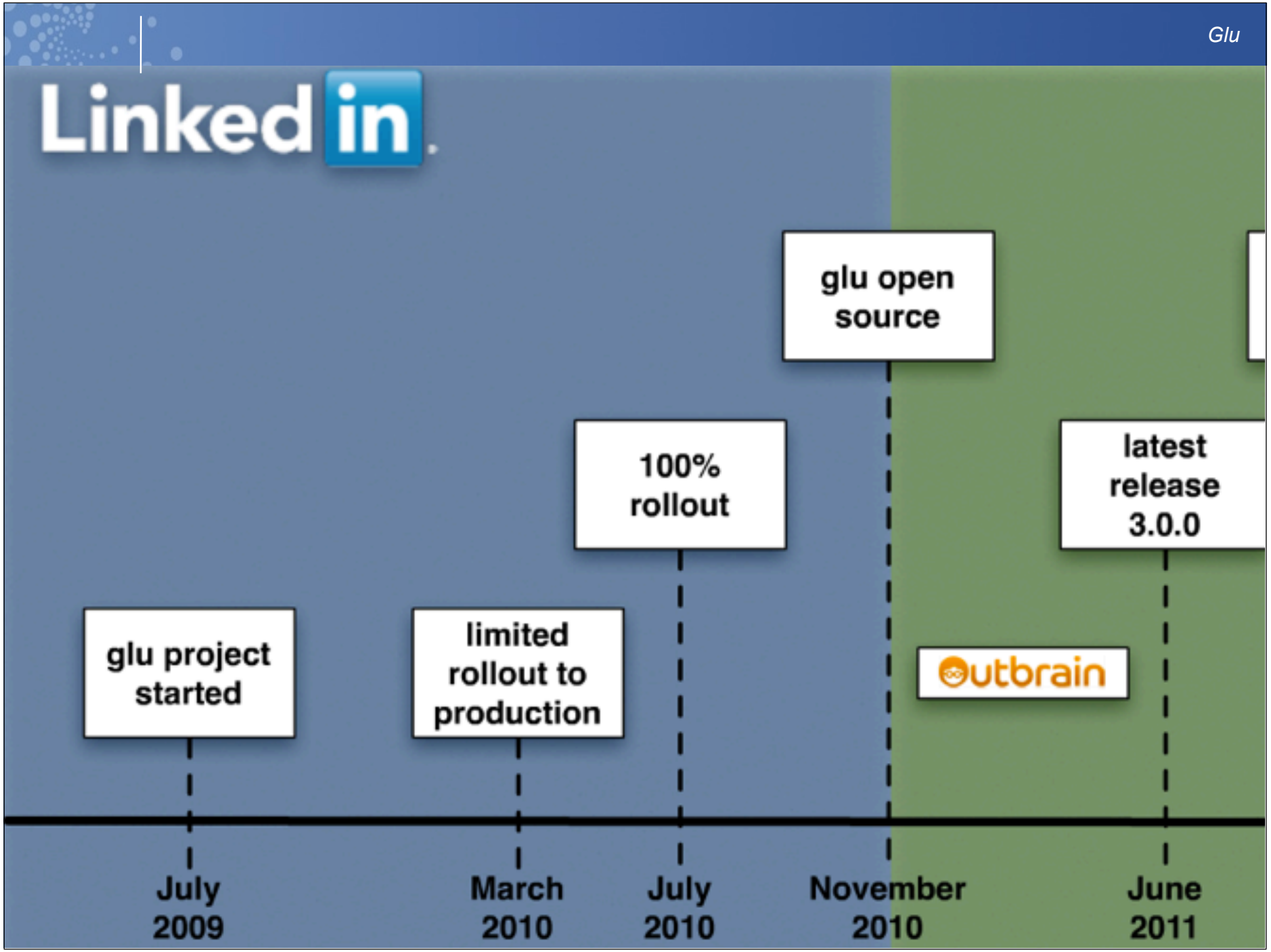

a b c d e f g h
8 7 6 5 4 3 2 1
g5
a4 b4 c4 d4
a3 b3 c3 d3
a2 b2 c2 d2
a1 b1 c1 d1


top - 02:48:42 up 5:20, 4 users, load average: 0.10, 0.59, 0.41
Tasks: 149 total, 1 running, 148 sleeping, 0 stopped, 0 zombie
Cpu(s): 2.0%us, 1.0%sy, 0.3%ni, 96.6%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1546144k total, 1215204k used, 331340k free, 201344k buffers
Swap: 1148732k total, 0k used, 1148732k free, 487692k cached
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
4798 root 15 0 1570 184 128 S 1.0 0.0 2:10.60 thinkpad-keys
5228 cmj 16 0 30292 8456 6870 S 1.0 0.1 1:08.42 gnome-cups-icon
26573 supriya 20 10 5488 2196 1424 S 0.7 0.1 0:10.91 cupsd
4067 cmj 15 0 46424 22m 12m R 0.3 1.1 0:01.83 terminator.py
4198 root 15 0 225m 42m 10m S 0.3 2.8 5:11.19 Xorg
1 root 18 0 2044 1052 532 S 0.0 0.1 0:01.54 init
2 root 34 -5 0 0 0 S 0.0 0.0 0:00.01 kthreadd
3 root RT -5 0 0 0 S 0.0 0.0 0:00.00 migration/0
4 root 34 19 0 0 0 R 0.0 0.0 0:00.22 ksoftirqd/0
5 root RT -5 0 0 0 S 0.0 0.0 0:00.00 watchdog/0
6 root 10 -5 0 0 0 S 0.0 0.0 0:02.23 events/0
7 root 10 -5 0 0 0 S 0.0 0.0 0:00.00 khelper
27 root 10 -5 0 0 0 S 0.0 0.0 0:00.01 kblockd/0
28 root 10 -5 0 0 0 S 0.0 0.0 0:00.01 kacpid
29 root 10 -5 0 0 0 S 0.0 0.0 0:00.00 kacpi_notify
141 root 10 -5 0 0 0 S 0.0 0.0 0:00.02 kseriod
142 root 21 0 0 0 0 S 0.0 0.0 0:00.00 pdflush
143 root 15 0 0 0 0 S 0.0 0.0 0:00.96 pdflush
144 root 10 -5 0 0 0 S 0.0 0.0 0:00.00 kswapd0
193 root 11 -5 0 0 0 S 0.0 0.0 0:00.00 aio/0
2033 root 10 -5 0 0 0 S 0.0 0.0 0:00.00 ksuspend_usbd
2034 root 10 -5 0 0 0 S 0.0 0.0 0:00.00 khubd
2072 root 10 -5 0 0 0 S 0.0 0.0 0:00.00 ata/0
2073 root 10 -5 0 0 0 S 0.0 0.0 0:00.00 ata_aux
--Mutt: =Lists/vdr [Msgs:24 Del:15 Post:12]---
1 26/07/07 Ilkka Tuomisaari [vdr]
2 26/07/07 Luca Olivetti Re: [vd
3 25/07/07 Stone [vdr] Warn
4 24/07/07 Simon Baxter [vdr] stut
5 24/07/07 Simon Baxter Re: [vd
6 24/07/07 Simon Baxter [vdr] no c
7 24/07/07 Klaus Schmidinger
8 24/07/07 Simon Baxter
9 24/07/07 Klaus Schmidinger
10 23/07/07 Malte Schröder [vdr] vdr-
11 24/07/07 Malte Schröder Re: [vd
12 23/07/07 Arthur Konovalov [vdr] vdr-
13 22/07/07 Klaus Schmidinger [vdr] [AM
14 22/07/07 Luca Olivetti
15 22/07/07 Klaus Schmidinger
16 23/07/07 Hermann Gausterer
17 22/07/07 Klaus Schmidinger
18 21/07/07 JJussi [vdr] Brea
19 22/07/07 Petri Helin
20 22/07/07 JJussi
21 22/07/07 Darren Salt
22 22/07/07 JJussi
23 22/07/07 Petri Helin
24 22/07/07 JJussi
25 21/07/07 Gregoire Favre [vdr] PAN
26 21/07/07 Udo Richter [vdr] [AM
27 21/07/07 Simon Baxter [vdr] CAM-
28 21/07/07 Klaus Schmidinger
q:Quit d:Del u:Undel s:Save m:Mail r:Reply
Jul 28 02:40:24 localhost NetworkManager: <info> nm_policy_device_change_check: old_dev && new_dev!
Jul 28 02:41:14 localhost kernel: [19144.944000] BADINPUT IN=eth2 OUT= MAC=01:00:5e:00:00:01:00:1f:3a:ac:ee:00
:00 SRC=192.168.1.1 DST=224.0.0.1 LEN=28 TOS=0x00 PREC=0x00 TTL=1 ID=0 DF PROTO=2
Jul 28 02:41:20 localhost kernel: [19351.040000] BADINPUT IN=eth2 OUT= MAC=01:00:5e:00:00:fb:00:90:27:46:f0:8f:00
:00 SRC=192.168.1.10 DST=224.0.0.251 LEN=32 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2
Jul 28 02:42:20 localhost NetworkManager: <info> nm_device_set_active_link: start
Jul 28 02:42:34 localhost NetworkManager: <info> nm_policy_device_change_check: old_dev has_link? 1
Jul 28 02:42:34 localhost NetworkManager: <info> nm_policy_device_change_check: old_dev && new_dev!
Jul 28 02:41:20 localhost kernel: [19470.952000] BADINPUT IN=eth2 OUT= MAC=01:00:5e:00:00:01:00:1f:3a:ac:ee:00
:00 SRC=192.168.1.1 DST=224.0.0.1 LEN=28 TOS=0x00 PREC=0x00 TTL=1 ID=0 DF PROTO=2
Jul 28 02:43:29 localhost kernel: [19477.104000] BADINPUT IN=eth2 OUT= MAC=01:00:5e:00:00:fb:00:90:27:46:f0:8f:00
:00 SRC=192.168.1.10 DST=224.0.0.251 LEN=32 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2
Jul 28 02:44:39 localhost NetworkManager: <info> nm_device_set_active_link: start
Jul 28 02:44:44 localhost NetworkManager: <info> nm_policy_device_change_check: old_dev has_link? 1
Jul 28 02:44:44 localhost NetworkManager: <info> nm_policy_device_change_check: old_dev && new_dev!
Jul 28 02:45:20 localhost kernel: [19598.960000] BADINPUT IN=eth2 OUT= MAC=01:00:5e:00:00:01:00:1f:3a:ac:ee:00
:00 SRC=192.168.1.1 DST=224.0.0.1 LEN=28 TOS=0x00 PREC=0x00 TTL=1 ID=0 DF PROTO=2
Jul 28 02:48:29 localhost kernel: [19600.392000] BADINPUT IN=eth2 OUT= MAC=01:00:5e:00:00:fb:00:90:27:46:f0:8f:00
:00 SRC=192.168.1.10 DST=224.0.0.251 LEN=32 TOS=0x00 PREC=0xC0 TTL=1 ID=0 DF PROTO=2
-(cmj)-(matthew)-
AUTHORS
changelog.Debian
changelog.gz
-(cmj)-(matthew)-
examples/
cat.py unreali
-(cmj)-(matthew)-

What is Glu?

What is Glu?

- Deployment Automation Tool
  - similar to Puppet / Chef
- Model Based!
- Open Source
  - https://github.com/linkedin/glu
- Manages LinkedIns multi-day deployments

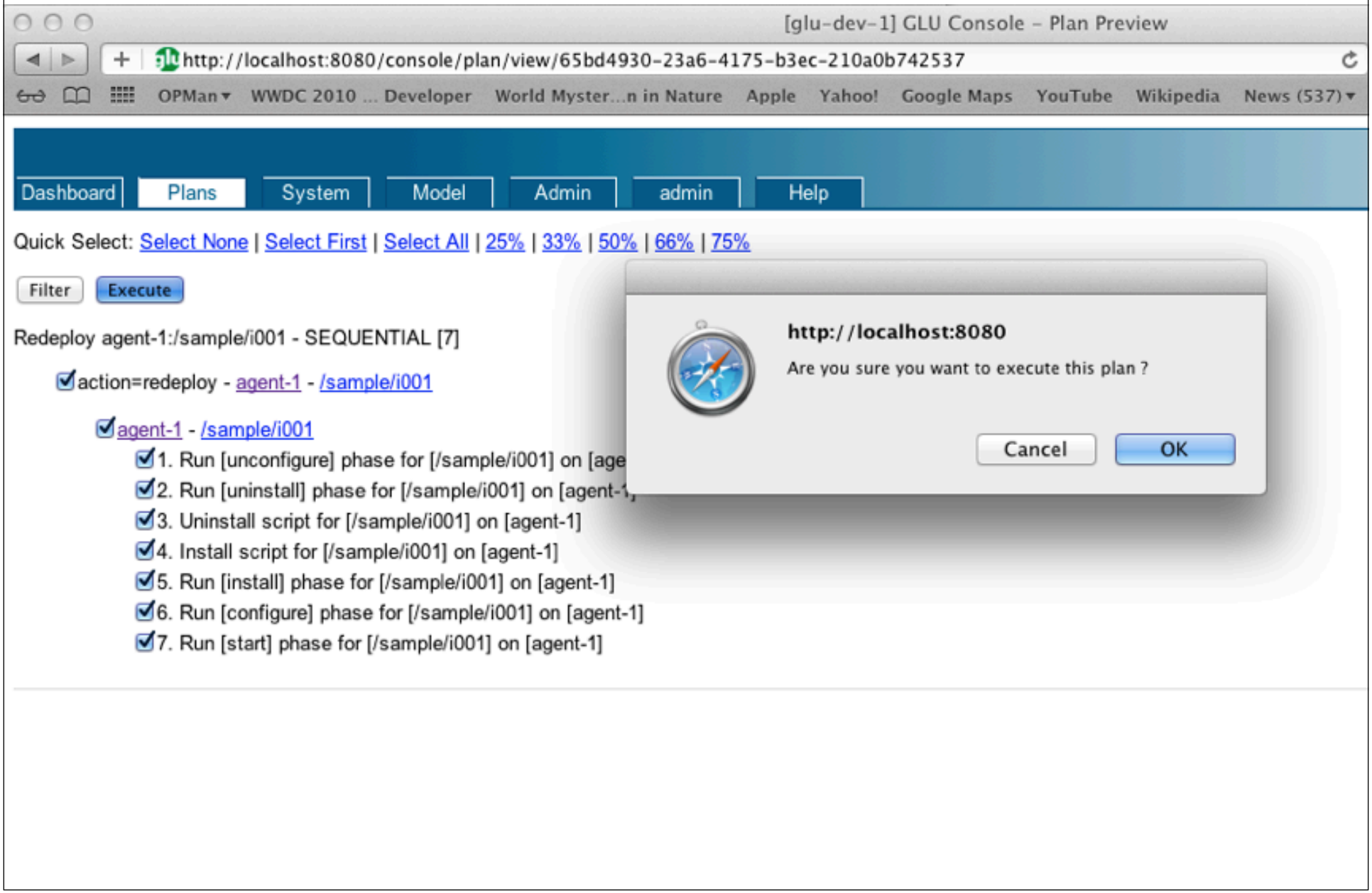

LinkedIn
glu project started
limited rollout to production
100% rollout
glu open source
outbrain
latest release 3.0.0
July 2009
March 2010
July 2010
November 2010
June 2011

Before Glu

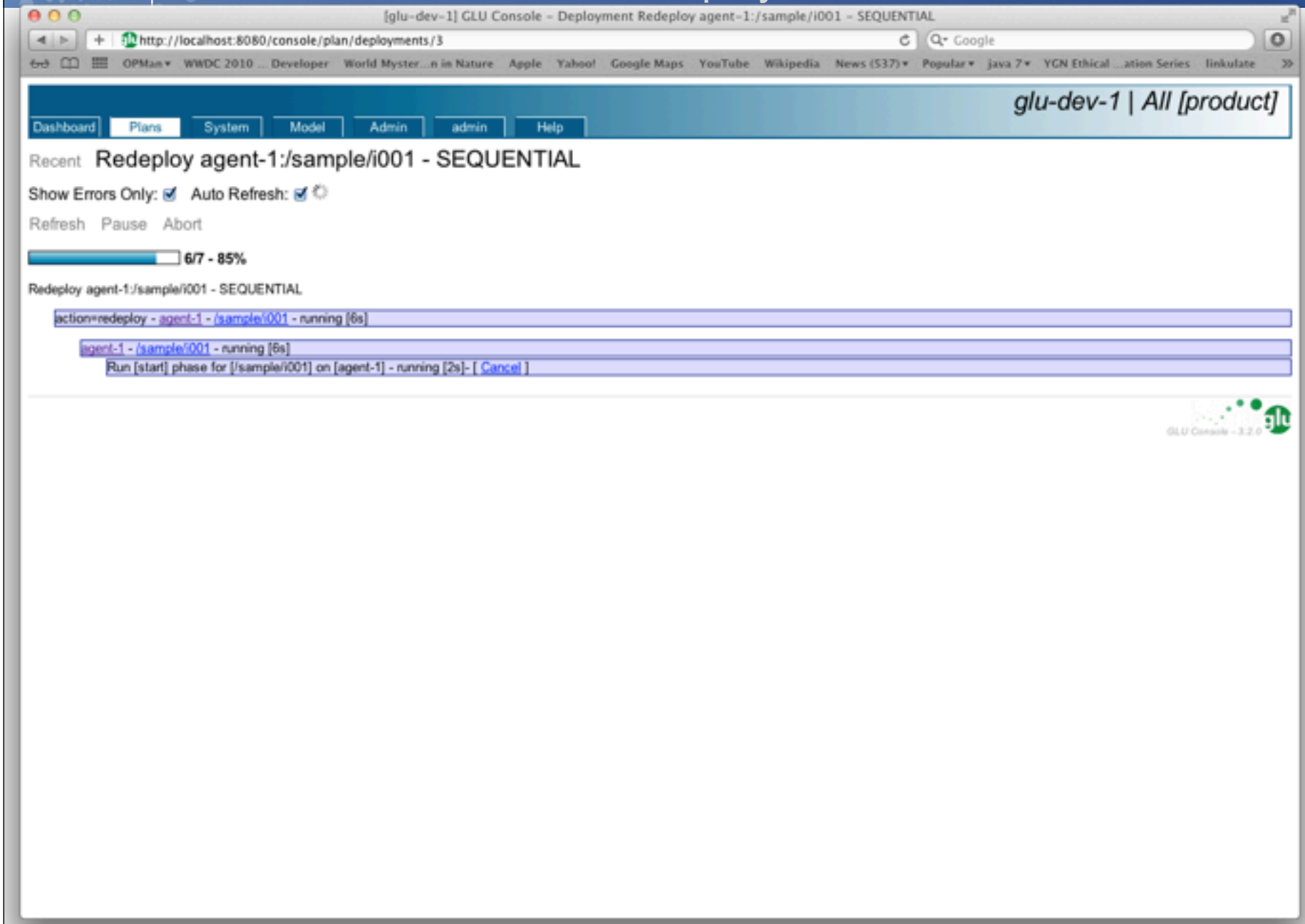
- Manual Deployments
  - ssh, scp
  - wiki / word doc instructional scripts
  - time-consuming
  - error prone

Glu

- Addresses IT's operational pain points
- Deploy (and monitor) applications to an arbitrary large set of nodes:
  - minimizes human involvement
  - secure
  - reproducibility
- Prevents system drift
- Admin troubleshooting tool


[glu-dev-1] GLU Console - Plan Preview
http://localhost:8080/console/plan/view/65bd4930-23a6-4175-b3ec-210a0b742537
OPMan
WWDC 2010 ... Developer
World Myster...n in Nature
Apple
Yahoo!
Google Maps
YouTube
Wikipedia
News (537)
Dashboard
Plans
System
Model
Admin
admin
Help
Quick Select: Select None | Select First | Select All | 25% | 33% | 50% | 66% | 75%
Filter
Execute
Redeploy agent-1:/sample/i001 - SEQUENTIAL [7]
action=redeploy - agent-1 - /sample/i001
agent-1 - /sample/i001
1. Run [unconfigure] phase for [/sample/i001] on [age
2. Run [uninstall] phase for [/sample/i001] on [agent-1]
3. Uninstall script for [/sample/i001] on [agent-1]
4. Install script for [/sample/i001] on [agent-1]
5. Run [install] phase for [/sample/i001] on [agent-1]
6. Run [configure] phase for [/sample/i001] on [agent-1]
7. Run [start] phase for [/sample/i001] on [agent-1]
http://localhost:8080
Are you sure you want to execute this plan ?
Cancel
OK

Sit Back... and Watch the deployment

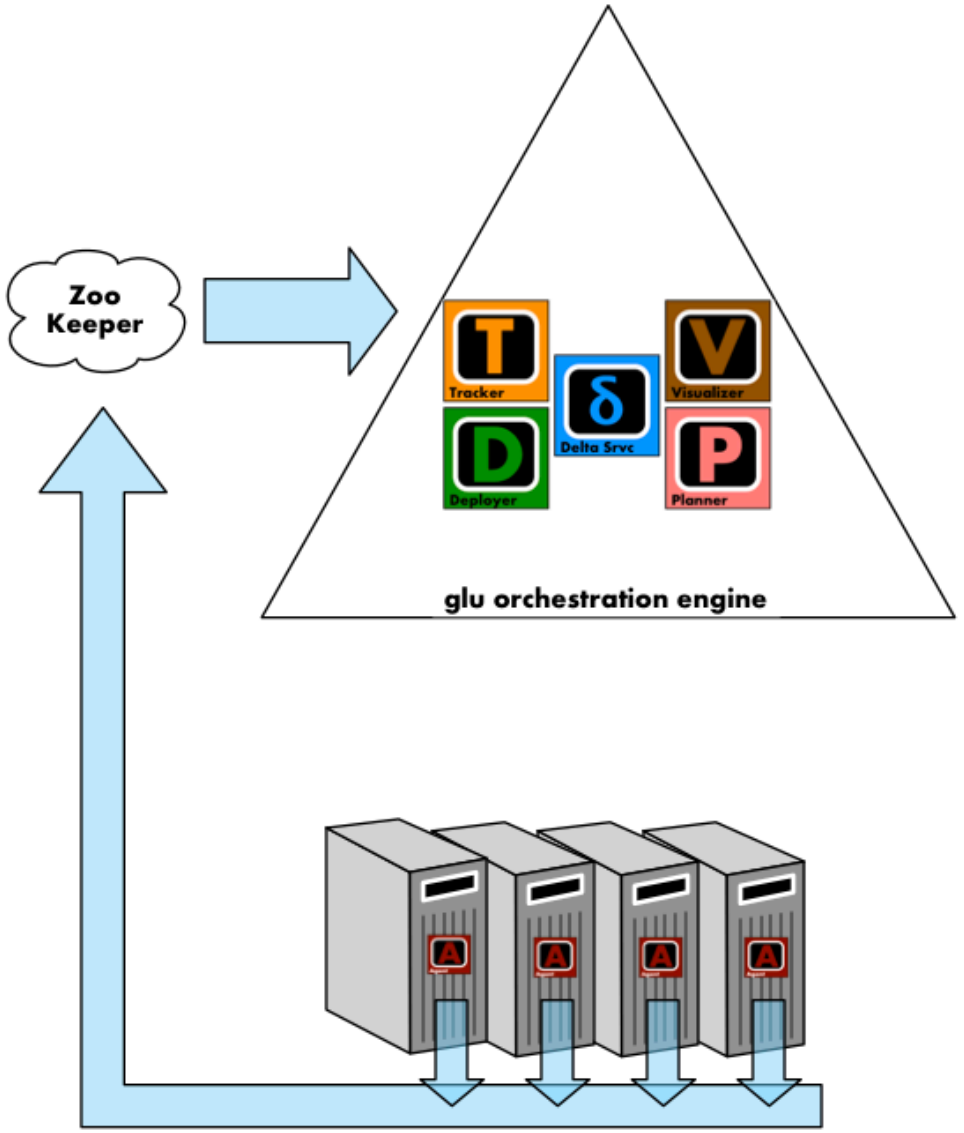


Rollout to Prod

- Glu started in July 2009
- LinkedIn production in March 2010
- by June 2011 @ LinkedIn:
  - 5 different ‘Fabrics’ (2 prod + 2 stg + 1 int. lab)
  - ~2650 nodes, ~9000 instances, ~300 services

Glu - Big Picture

- declarative model
  - compute the set of actions
  - ensure consistent ove time
  - detect and alert when there is a mismatch



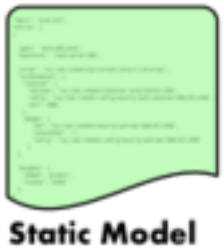
3 Components of Glu

Agents

Orchestration Engine

ZooKeeper

3 Concepts of Glu

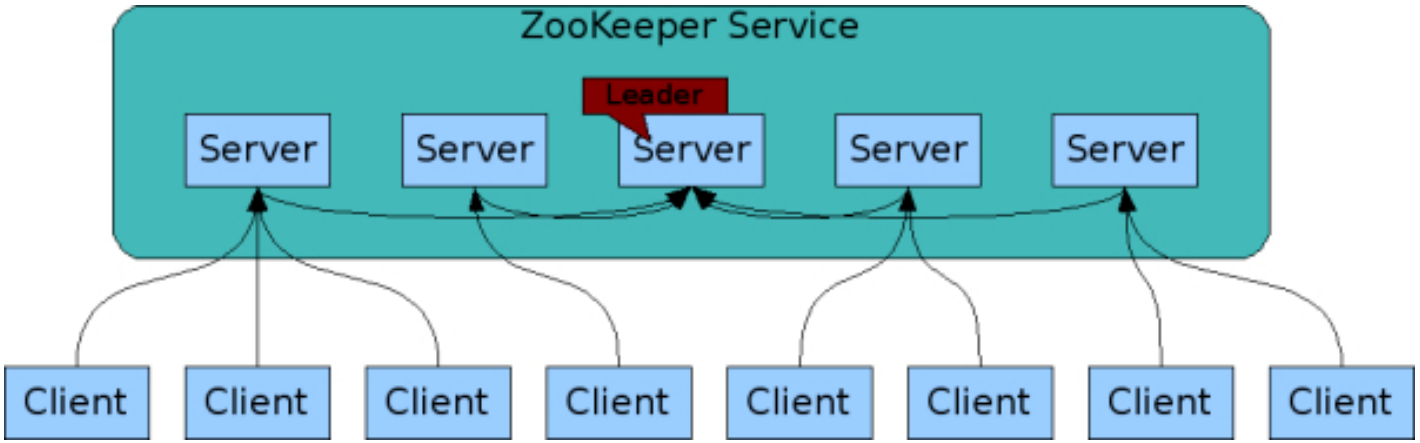


Static Model

Script

Live Model

- ZooKeeper
  - A Distributed Coordination Service
    - for Distributed Applications
  - Apache project
  - similar to a filesystem (think nfs)
    - + ‘directories’ can also contain data
    - + powerful watcher concept => notifications
  - Used in Glu to maintain the state of the system



- Replicated
- Fast
  - data is kept in-memory
  - “read-dominant”
- Simple API
- Guarantees
  - sequential consistency
  - atomicity
  - single system image
  - reliability
  - timeliness

Glu Agent

- 1 agent / node
  - for every node
- Active Groovy-based process
- REST API
- Reports state to ZooKeeper

Glu Agent

- Belongs to **One** fabric
  - fabric is a group of agents
- Requires a unique name in the fabric
  - default canonical host name
- It is a glu Script Engine
  - installs and executes glu script
  - has an api

Connecting to an Agent

Groovy API:

```
agent.installScript(mountPoint: "/geo/i001",
                    scriptLocation: "http://host:port/glu/MyGluScript.groovy",
                    initParameters: [skeleton: "ivy:/skeleton/jetty/1.0"])
```

Command Line:

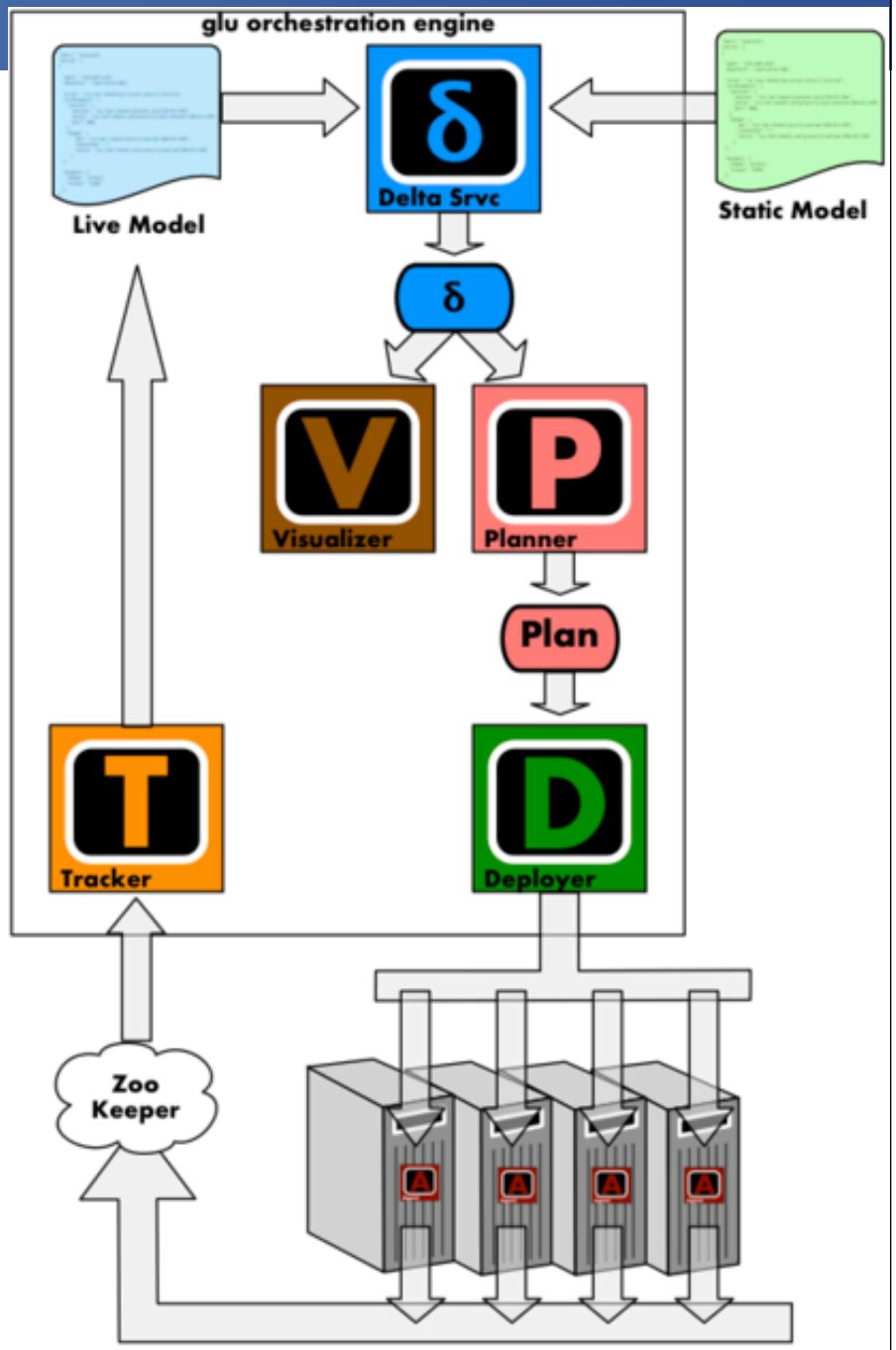
```
agent-cli.sh -s https://localhost:12906/ -m /geo/i001 -i http://host:port/glu/MyGluScript.groovy \
             -a "[skeleton: 'ivy:/skeleton/jetty/1.0']"
```

REST API:

```
PUT /mountPoint/geo/i001
{"args": {"scriptLocation": "http://host:port/glu/MyGluScript.groovy",
          "initParameters": {"skeleton": "ivy:/skeleton/jetty/1.0"} } }
```

Orchestration Engine

- Heart of glu
- Listens to ZooKeeper
- computes deltas
- orchestrates deployment plans
- Talks to agents
- runs inside a webapp
  - browser and REST interface



Orchestration Tracker

- Component responsible for listening to ZooKeeper
  - uses zookeeper watches
- Events
  - new agent added
  - an agent disappears
  - a script state changed
- Info used to create the live model

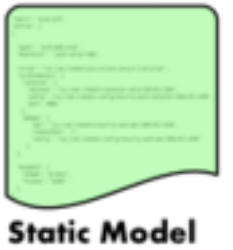
Orchestration Live Model

- Represents the current state in JSON


```
{
  "entries": [{
    "agent": "agent-1",
    "initParameters": {
      "port": 9000,
      "skeleton": "http://localhost:8080/glu/repository/tgzs/jett
      "webapps": [
        {
          "contextPath": "/cp1",
          "monitor": "/monitor",
          "war": "http://localhost:8080/glu/repository/wars/org.1
        },
        {
          "contextPath": "/cp2",
          "monitor": "/monitor",
          "war": "http://localhost:8080/glu/repository/wars/org.1
        }
      ]
    },
    "metadata": {
      "cluster": "c1",
      "container": {"name": "sample"},
      "currentState": "stopped",
```

Orchestration Static Model

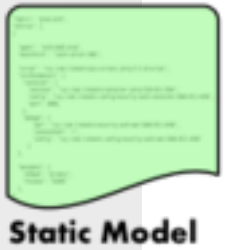
- Represents the the desired state of the fabric
- model
  - json
  - where to deploy
  - what and how to deploy



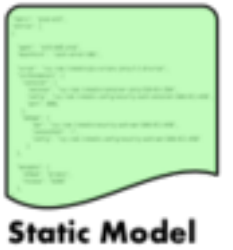

```
{
  "fabric": "glu-dev-1",
  "metadata": {}
  "agentTags": {},
  "entries": []
}
```

Orchestration Static Where?


```
"entries": [
  {
    "agent": "agent-1",
    "mountPoint": "/container",
    "script": "http://repository.prod/scripts/webapp-container-1.0.0.groovy",
    "initParameters": {
      "skeleton": "http://repository.prod/tgzs/jetty-7.2.2.v20101205.tgz",
      "config": "http://repository.prod/configs/search-container-config-2.1.0.json",
      "port": 8080
    }
  },
```



Orchestration Static Model What / How?



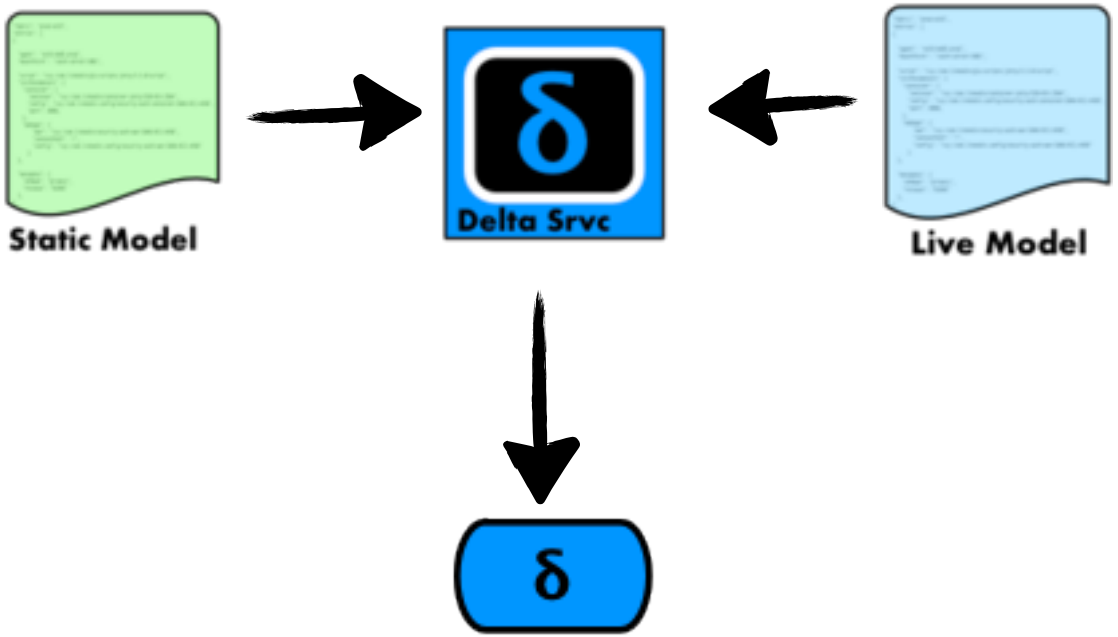

```
  {
    "agent": "agent-1",
    "mountPoint": "/webapp1",
    "parent": "/container",
    "script": "http://repository.prod/scripts/webapp-1.0.0.groovy",
    "initParameters": {
      "war": "http://repository.prod/wars/search-2.1.0.war",
      "contextPath": "/",
      "config": "http://repository.prod/configs/search-config-2.1.0.json"
    }
  }
],
```

Orchestration Static Model Example


```
"entries": [
  {
    "agent": "agent-1",
    "mountPoint": "/container",
    "script": "http://repository.prod/scripts/webapp-container-1.0.0.groovy",
    "initParameters": {
      "skeleton": "http://repository.prod/tgzs/jetty-7.2.2.v20101205.tgz",
      "config": "http://repository.prod/configs/search-container-config-2.1.0.json",
      "port": 8080
    }
  },
  {
    "agent": "agent-1",
    "mountPoint": "/webapp1",
    "parent": "/container",
    "script": "http://repository.prod/scripts/webapp-1.0.0.groovy",
    "initParameters": {
      "war": "http://repository.prod/wars/search-2.1.0.war",
      "contextPath": "/",
      "config": "http://repository.prod/configs/search-config-2.1.0.json"
    }
  }
],
```

Orchestration Delta Service

- Computes the delta between live and static model
- Zookeeper
  - IF connection lost last report used
  - accuracy information is provided


Static Model
Delta Srvc
Live Model
δ

Orchestration Visualizer

mountPoint:3	I:3	E:2	agent:1	tags:4	status:3
/sample/i001	1	1	agent-1	frontend osx webapp	NOT running
/sample/i002	1	0	agent-1	frontend osx webapp	running
/sample/i003	1	1	agent-1	backend osx webapp	NOT deployed

Orchestration Planner

- Turns deltas into a deployment plan
- Plan
  - List of instructions

Deploy: Fabric [glu-dev-1]

Deploy: Fabric [glu-dev-1] - SEQUENTIAL [5] | Select this plan

Deploy: Fabric [glu-dev-1] - SEQUENTIAL

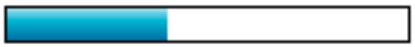
type=deploy - fabric=glu-dev-1

agent-1 - /sample/i001
1. Start agent-1:/sample/i001 on agent-1

agent-1 - /sample/i003
2. Install script for installation agent-1:/sample/i003 on agent-1
3. Install agent-1:/sample/i003 on agent-1
4. Configure agent-1:/sample/i003 on agent-1
5. Start agent-1:/sample/i003 on agent-1

Orchestration Deployer

- Turns deltas into a deployment plan



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

type=deploy - fabric=glu-dev-1 - running [3s]

agent-1 - /sample/i001 - running [3s]

Start agent-1:/sample/i001 on agent-1 - running [3s]- [Cancel]

agent-1 - /sample/i003 - running [3s]

Install script for installation agent-1:/sample/i003 on agent-1 - 481ms

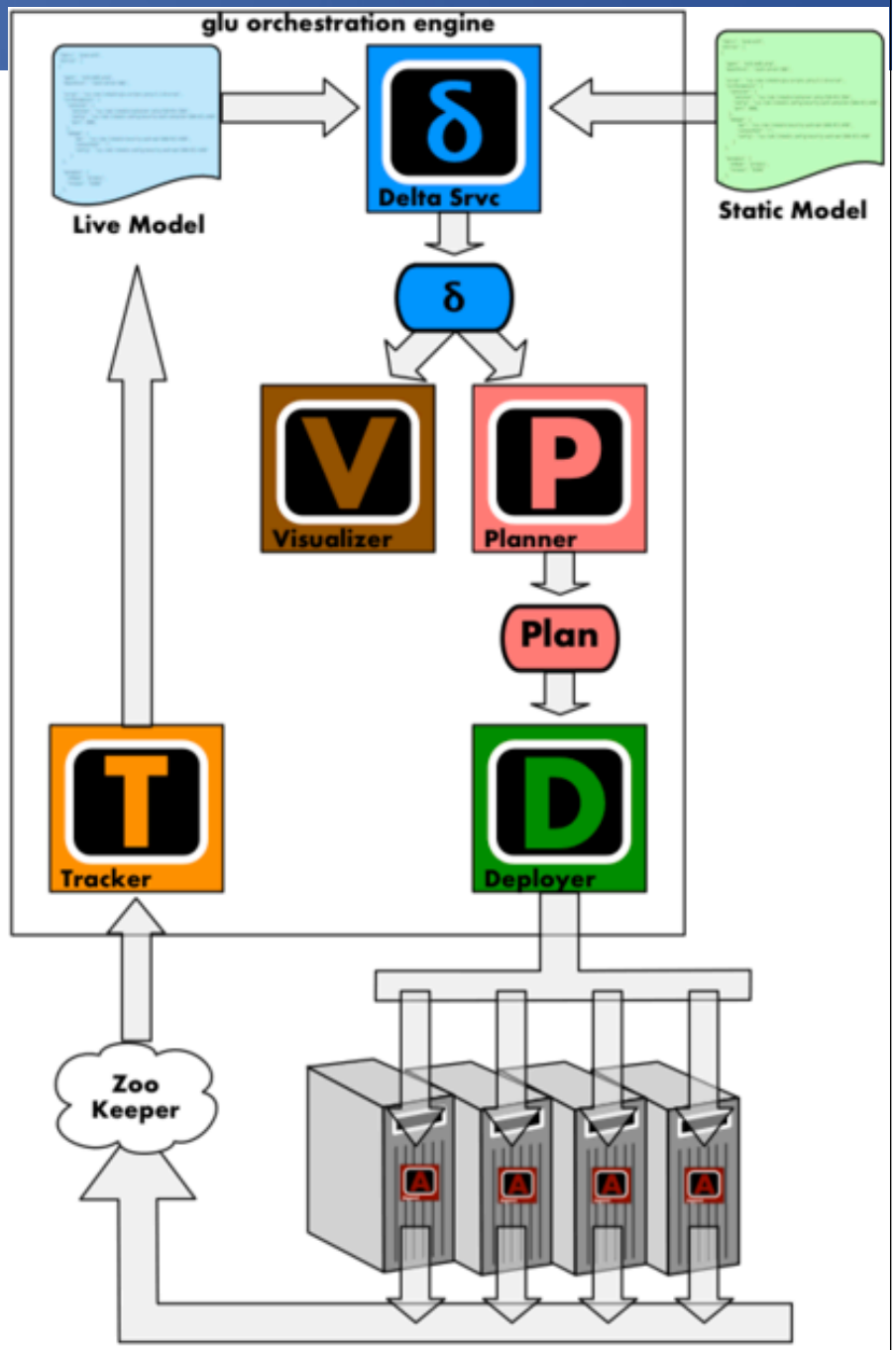
Install agent-1:/sample/i003 on agent-1 - 2s

Configure agent-1:/sample/i003 on agent-1 - running [716ms]- [Cancel]

Start agent-1:/sample/i003 on agent-1 - not started

```
2011-04-11 14:29:24,586 [INFO] - gluconsole.rest.Client - status url = plan/56e22137/execution/4
 60% [**************************************************************************************>
```

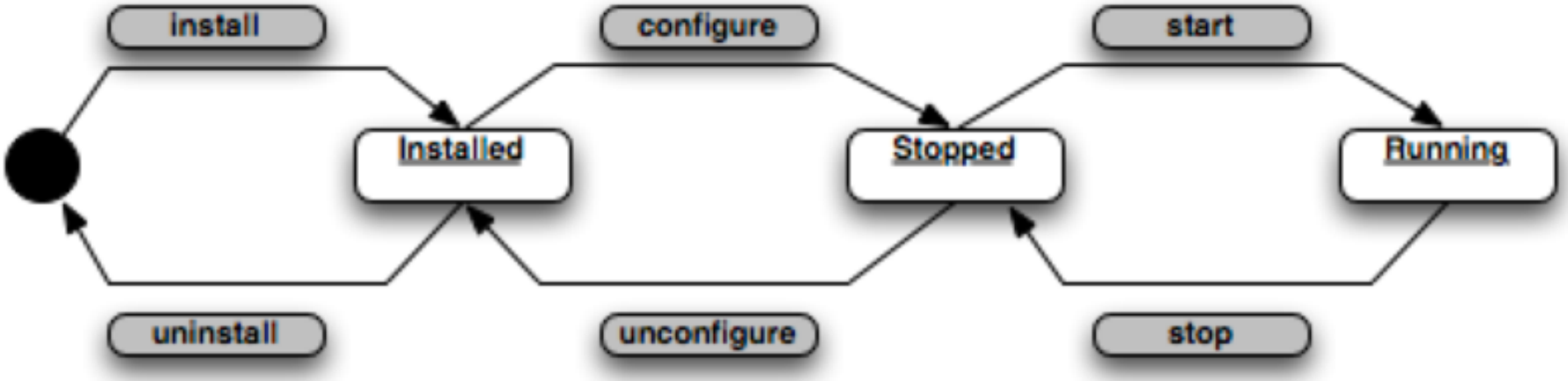
Orchestration Engine



- Provides instructions:
  - install
  - configure
  - start
  - stop
  - unconfigure
  - uninstall


```
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 */ }
}
```

glu Script State Machine



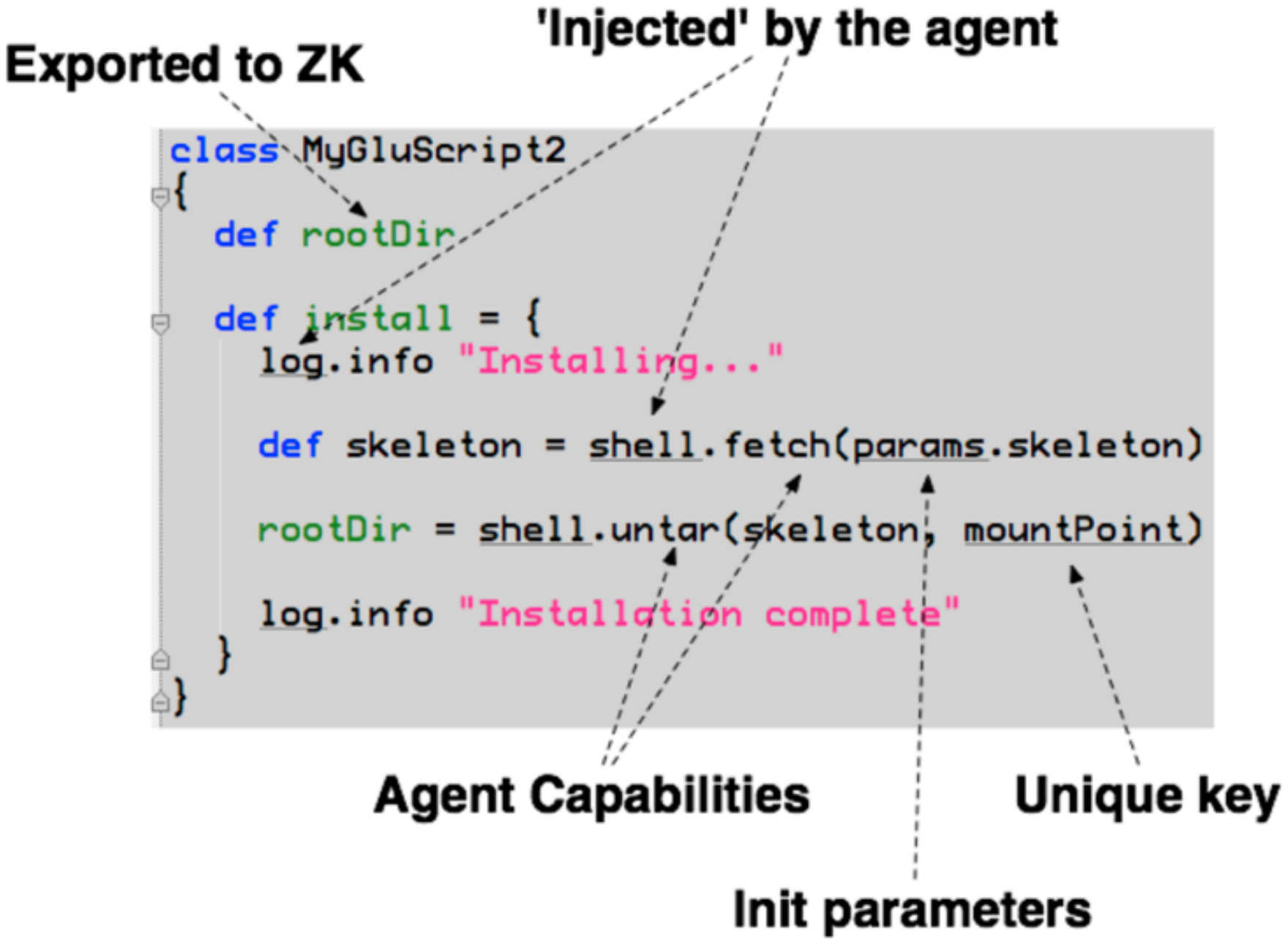

```
def static DEFAULT_TRANSITIONS =
[
  NONE: [[to: 'installed', action: 'install']],
  installed: [[to: 'stopped', action: 'configure'], [to: 'NONE', action: 'uninstall']],
  stopped: [[to: 'running', action: 'start'], [to: 'installed', action: 'unconfigure']],
  running: [[to: 'stopped', action: 'stop']]
]
```

glu script capabilities

name	Usage
log	write log messages
mountPoint	point the script is mounted to
params	access to initParameters
parent	access to parent glu script
shell	shell capabilities (mv, ls)
shell.env	sys env variables
stateManager	manage state
state	shortcut to current state
timers	schedule / cancel timers

Ex: https://github.com/linkedin/glu/blob/master/scripts/org.linkedin.glu.script-jetty/src/main/groovy/JettyGluScript.groovy

Example glu Script



Unit Testing glu script

- GluSriptBaseTest


```
class TestMyGluScript extends GluScriptBaseTest
{
  public void setUp() {
    super.setUp()
    initParameters = [ p1: 'v1' ]
  }

  // this method is not required if you follow the conventions
  public String getScriptClass() {
    return MyGluScript.getClass().getName()
  }

  public void testHappyPath() {
    deploy()
    undeploy()
  }
}
```

Gradle setup for testing


```
// gradle format
dependencies {
  compile "org.linkedin:org.linkedin.util-groovy:1.7.0"
  compile "org.linkedin:org.linkedin.glu.agent-api:3.1.0"
  groovy  "org.codehaus.groovy:groovy:1.7.5"

  testCompile "org.linkedin:org.linkedin.glu.scripts-test-fwk:3.1.0"
  testCompile "junit:junit:4.4"
}
```

glu Script Runtime

- glu Script code
  - runs in a JVM
  - can spawn external processes

glu Security

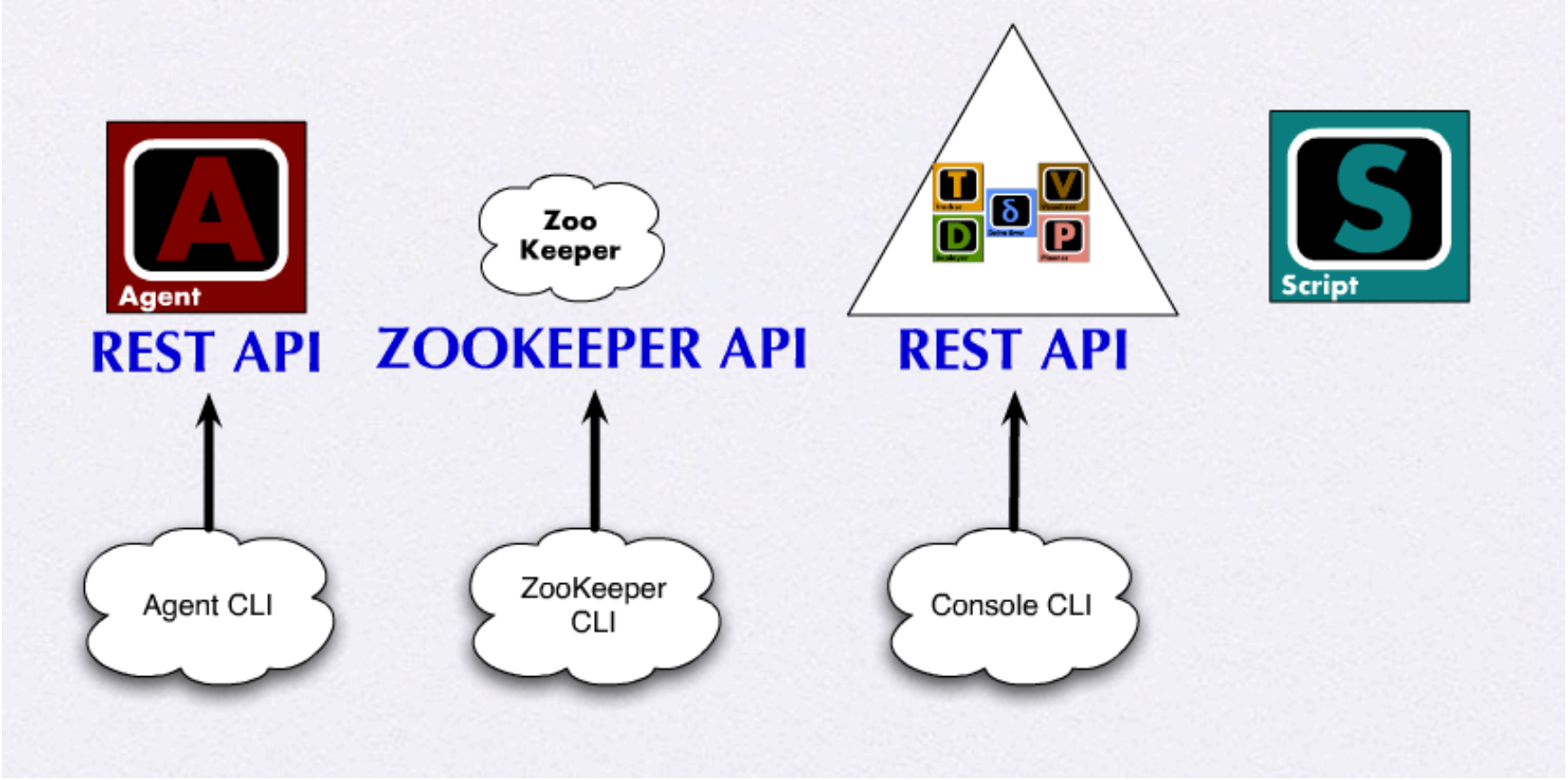
- User must authenticate
  - LDAP
  - glu
- Agent REST API behind HTTPS
- Every ‘change’ is audited in the audit log

Customizing glu

glu is more than a tool

- glu is highly customizable
- it is a **platform** on which you can build your own deployment or monitoring solution

glu Connection Points



- CLI
- REST API

Customizing the Agent

- Could be viewed as a remote script engine with a secure REST API
  - even without ZooKeeper

- ZooKeeper is independently accessible
  - build your own listeners / watchers
  - use AgentsTracker library
  - build a monitoring solution

Customizing the Orchestration

- Integrate with CI directly
  - outbrain.com
- Build your own UI

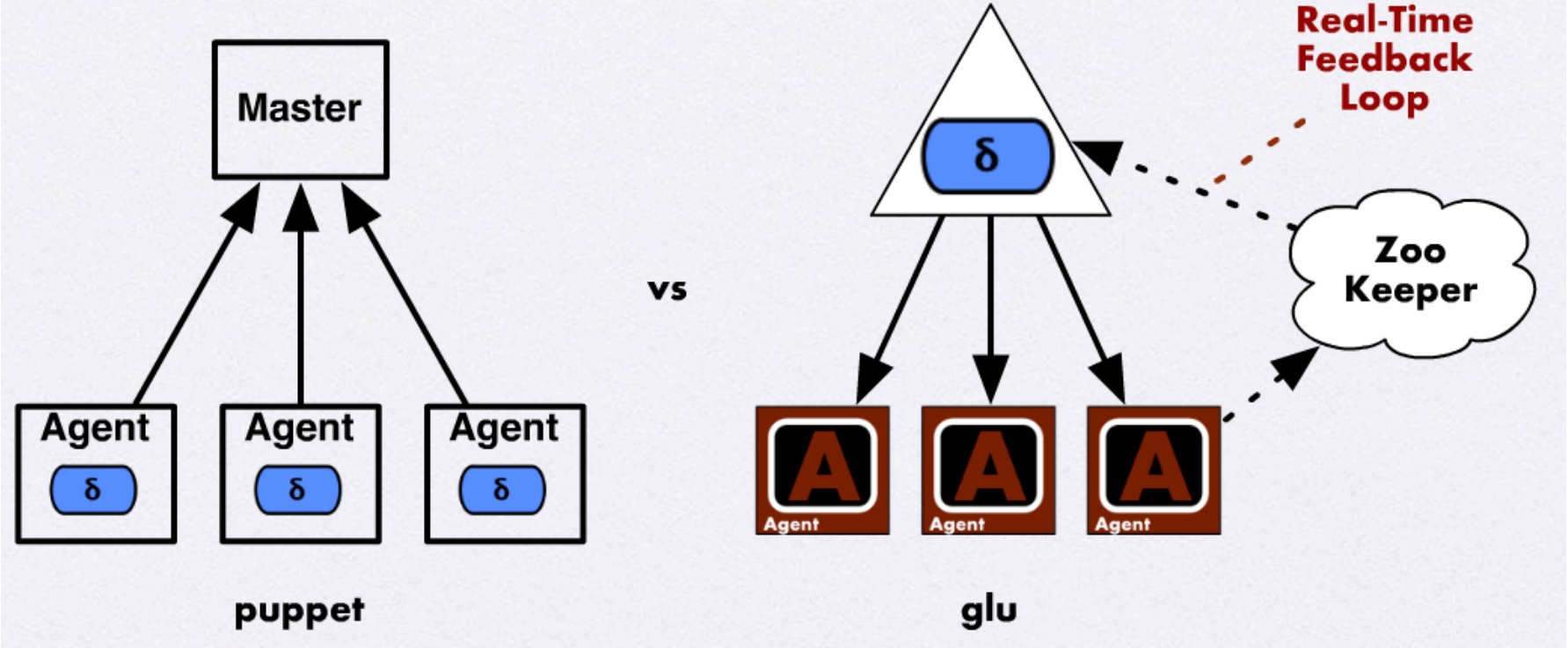
More glu Customization

- Powerful tagging and filtering
- Query Language
  - slide and dice the model

glu vs. puppet

- Intrinsically similar concepts
  - desire model vs current model
  - declarative approach
- Difference
  - puppet == ruby
  - glu == groovy

glu vs. puppet: orchestration



- delta computation
  - 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
    - users, groups, packages
  - stable doesn’t change often
- glu is good at:
  - provisioning dynamic applications
  - changes often, real-time failure detection
  - bounce

glu can use puppet

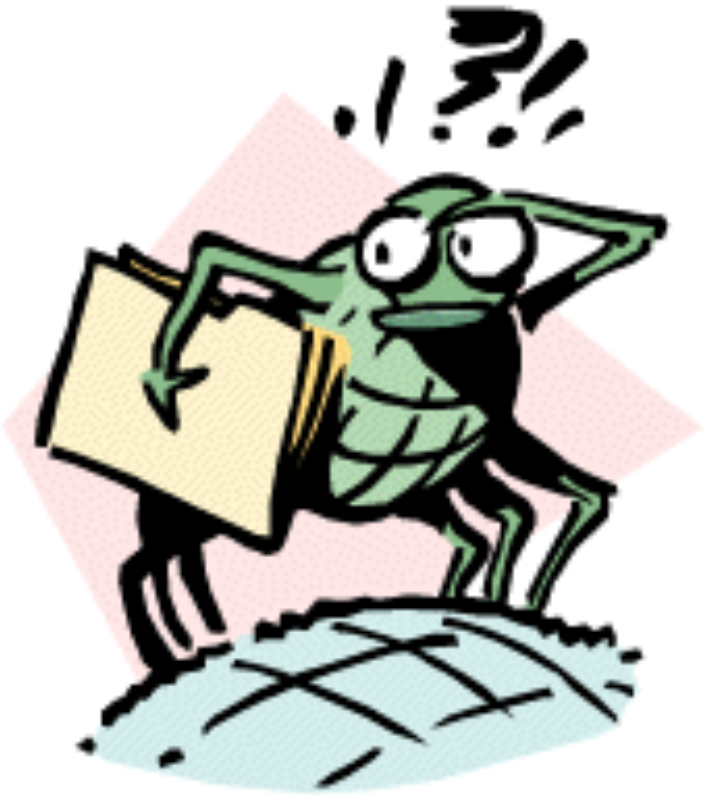

```
class PuppetGluScript {
  def puppetManifest

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

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

Demo

Summary

- Yan:
  - www.pongasoft.com/blog/yan
- ken.sipe@gmail.com
- twitter: @kensipe

Questions



- **Please Fill Out Surveys**

ken.sipe@gmail.com

twitter: @kensipe