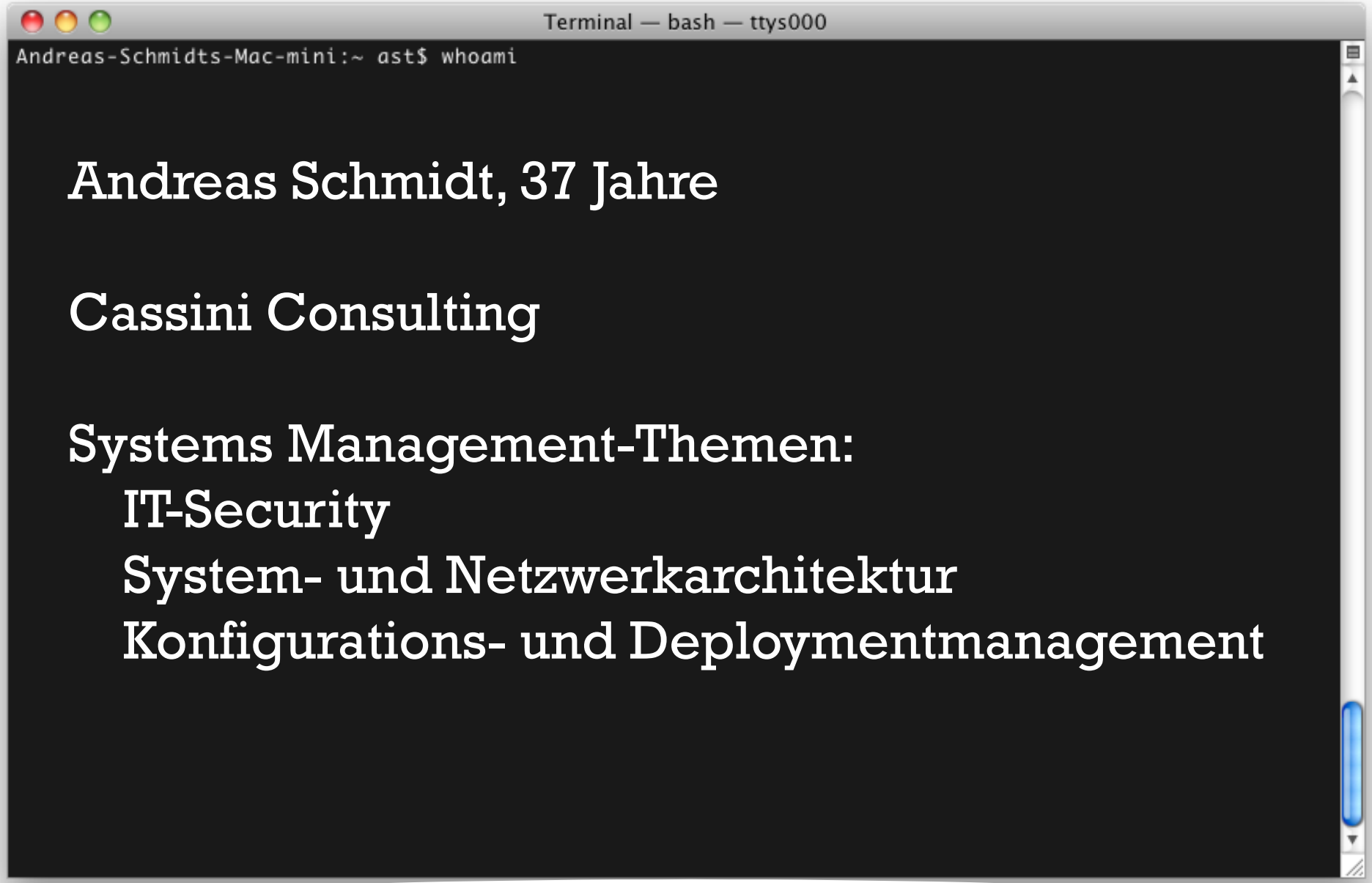
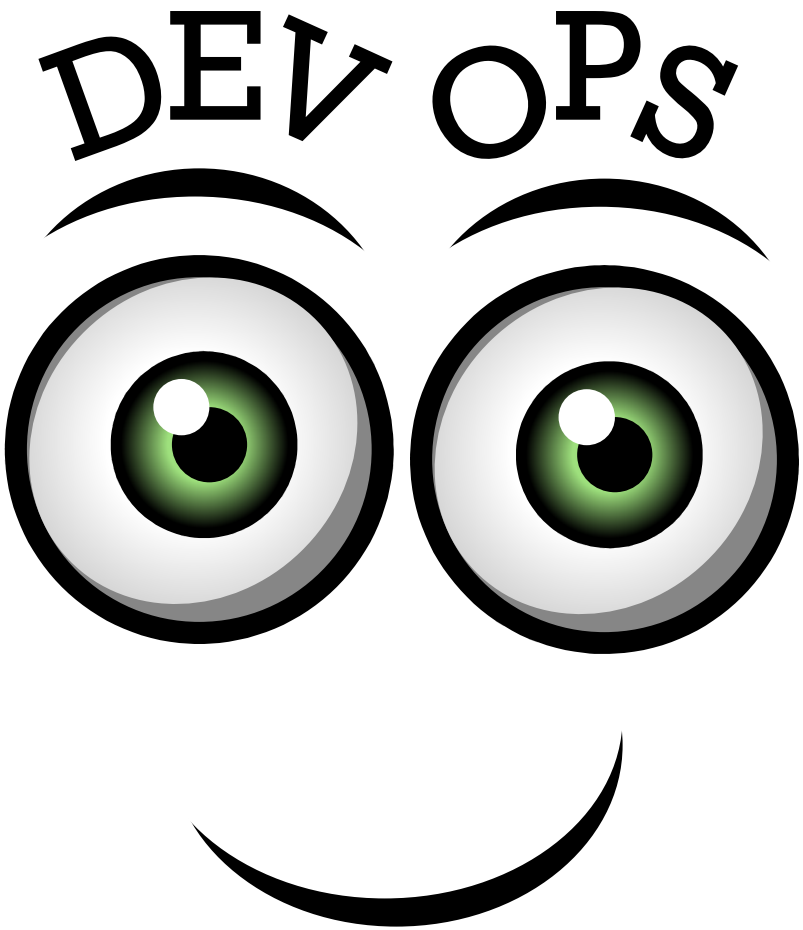
Parallelisierte Administration mit Marionette Collective

Andreas Schmidt



guug FFG 2012

cassini TECHNOLOGY GUIDANCE



Infrastructure
Development

What‘s on for today?

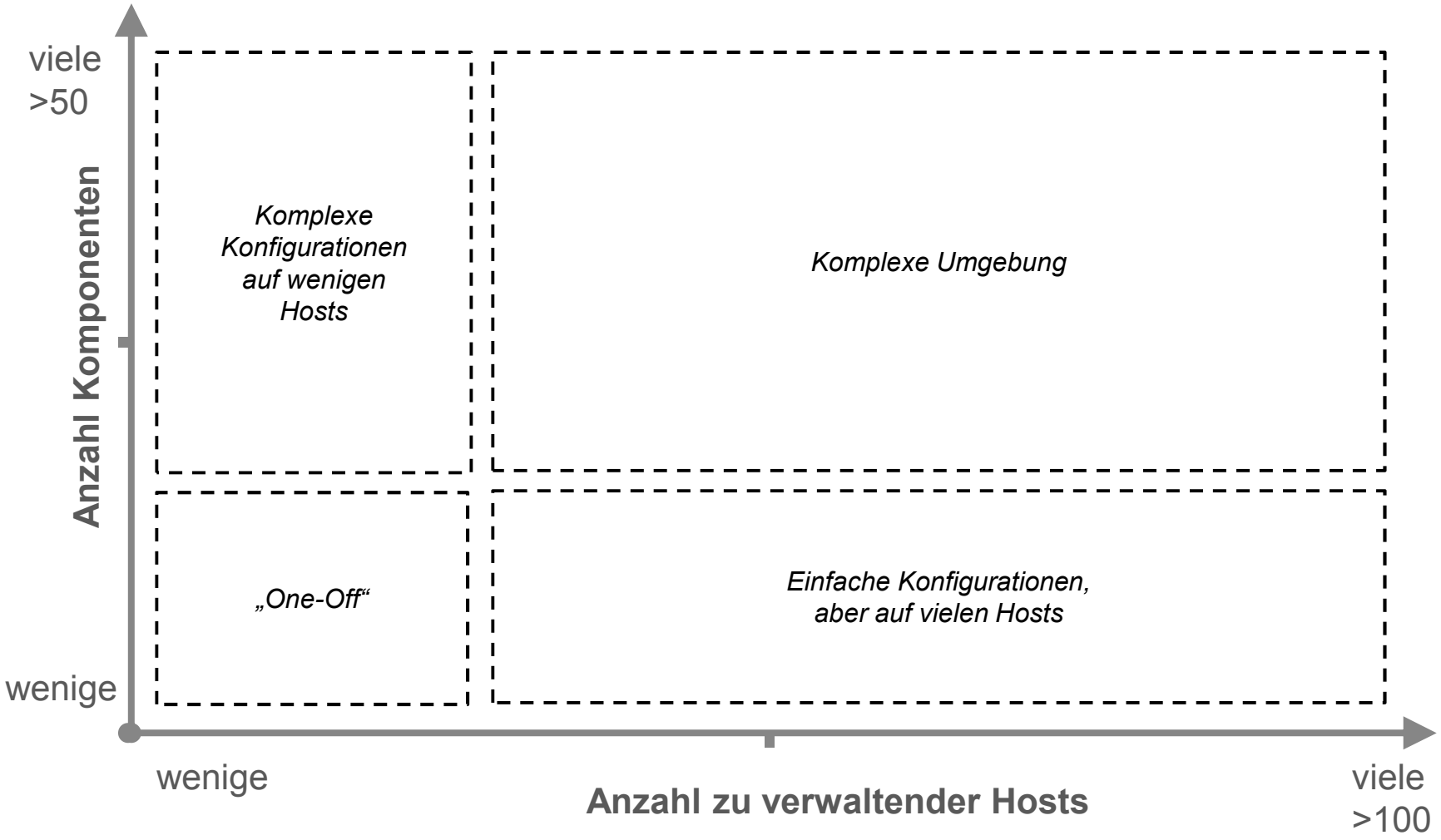
Herausforderungen beim Aufbau und Betrieb größerer Umgebungen

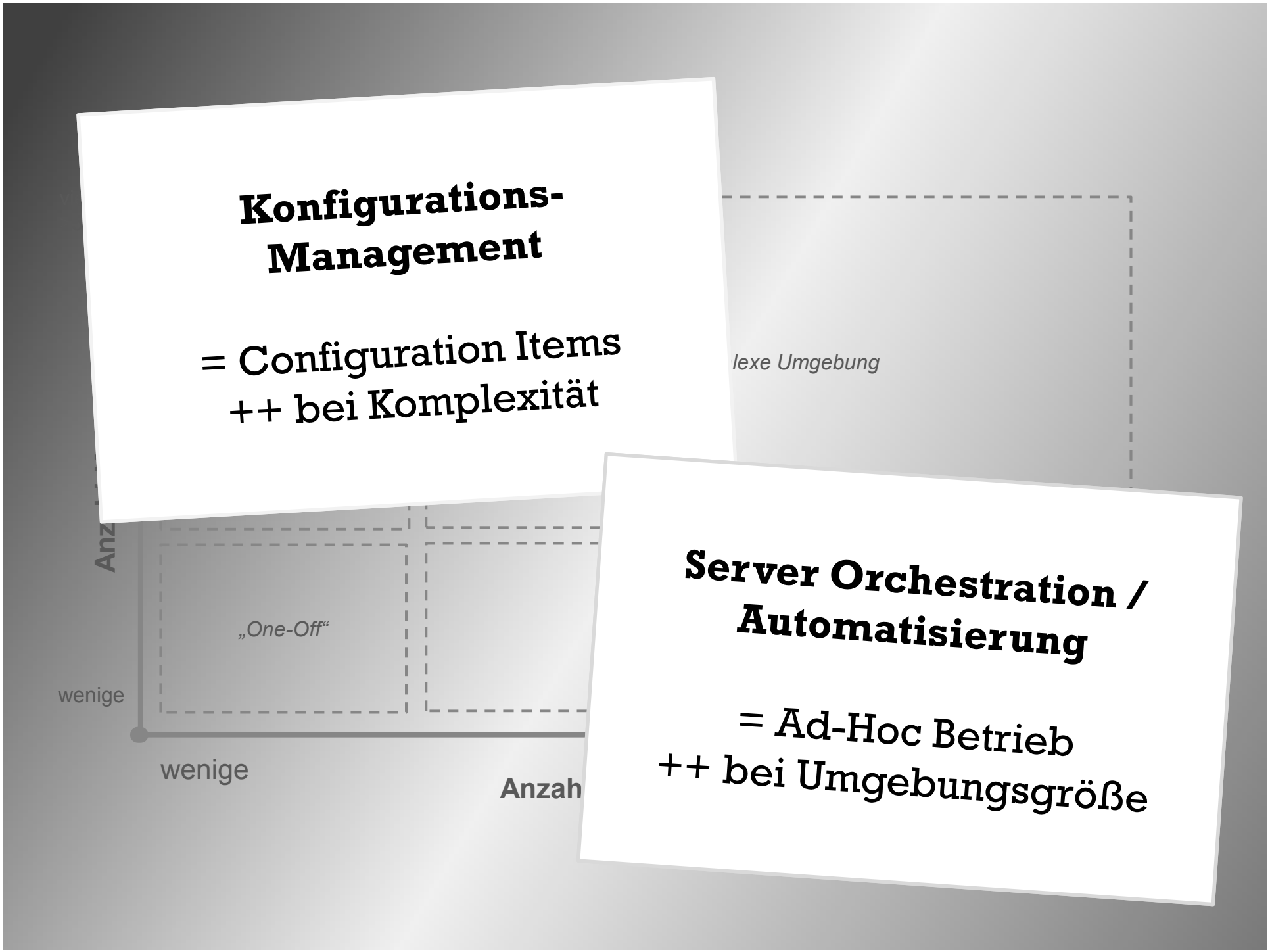
Was ist Marionette Collective?

Wofür kann man es einsetzen?

Wieso ist Marionette Collective zur Administration hilfreich?

Auswirkungen von Umgebungsgröße und -komplexität




Konfigurations-
Management
= Configuration Items
++ bei Komplexität
lexe Umgebung
Anz
„One-Off“
wenige
wenige
Anzah
Server Orchestration /
Automatisierung
= Ad-Hoc Betrieb
++ bei Umgebungsgröße

“ssh in a for loop is not a solution”

- Luke Kanies

guug FFG 2012

Pro Puppet (APress): Page xvi

Verbindungsaufbau

Adressierung durch Hostnamen

```
ssh www17 "sudo su - -c \"/sbin/iptables -L
-n\"" | awk '/ACCEPT/ { print $4 }' | sort -u
```

„Give me an API or give me death“

http://theagileadmin.com/2010/07/22/give-me-an-api-or-give-me-death/

Real Life
System Administration

Aktiviere ein Content-Update auf 200 Servern

- gleichzeitig

Räume das Cache-Verzeichnis /tmp/cx0 auf

- aber nur auf CentOS-Nodes mit einer IP-Adresse in der DMZ

Starte alle Tomcats durch

- auf denen die Applikation iWTF deployed ist.

(Zuordnung dynamisch)

Wie erreiche ich eine größere Anzahl Hosts möglichst gleichzeitig?

Wie kann ich Kommandos sicher gestalten und dokumentieren?

Wie finde ich heraus, welche Hosts für meine Abfrage relevant sind?

Performante Ausführung

Wiederverwendbare, verständliche Schnittstelle

Flexible Adressierung der Zielhosts

Aktiviere ein Content-Update auf 200 Servern - gleichzeitig

```
$ HOSTS=`cat ./contenthosts.txt`

$ for h in $HOSTS; do
    ssh $h "curl -d \"ctx_act_id=4711\"
              http://127.0.0.1:7676/activate_content
                 && echo OK on $h"
done;
```

 =307 sec.

Aktiviere ein Content-Update auf 200 Servern - gleichzeitig

```
$ mco rpc     --agent appcontent --action activate \
              --arg contentid=4711 \
              --discovery-timeout=5 --timeout=3 \
              -T produktion
```

 =3,1 sec.

Marionette Collective

„Server Orchestration Framework“

„Programmatic execution of system administration actions on clusters of servers“

Agent-basiert, ruby (1.8.5+)

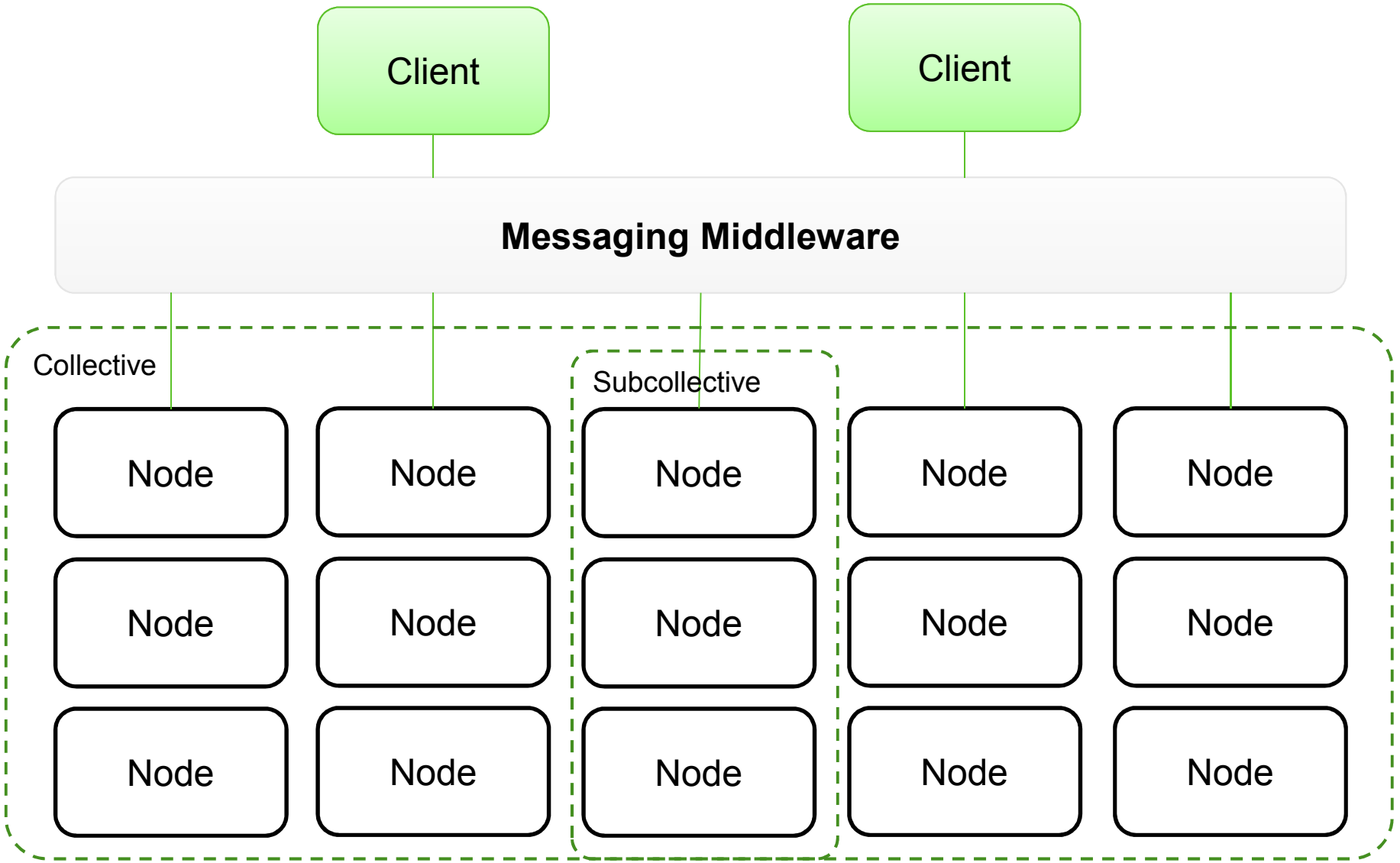
Puppet Labs

Apache 2 License

Broadcast Paradigm

Publish-Subscribe Middleware
(z.B. ActiveMQ oder RabbitMQ)

Architektur von Marionette Collective



Zentrale Metadaten

Hostlisten oder LDAP

Vergessene Server

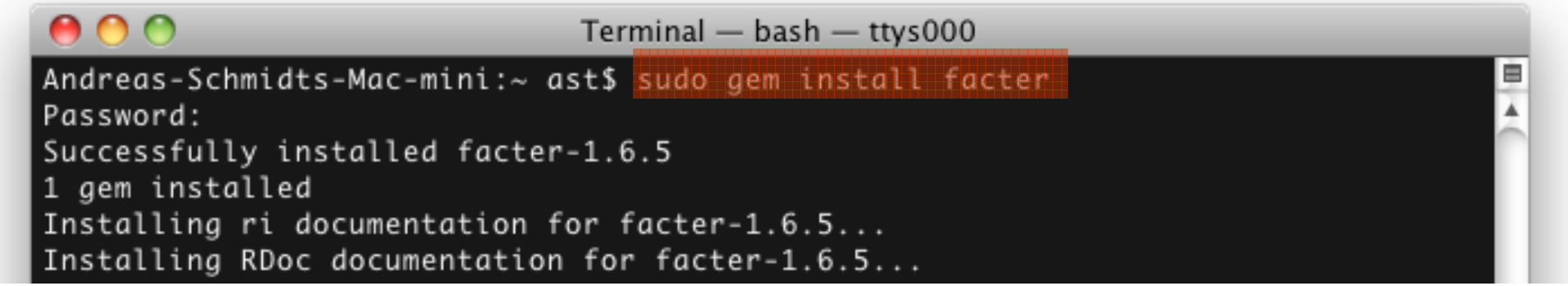
Missverständnisse & Kommunikationsfehler

Verteilte Metadaten

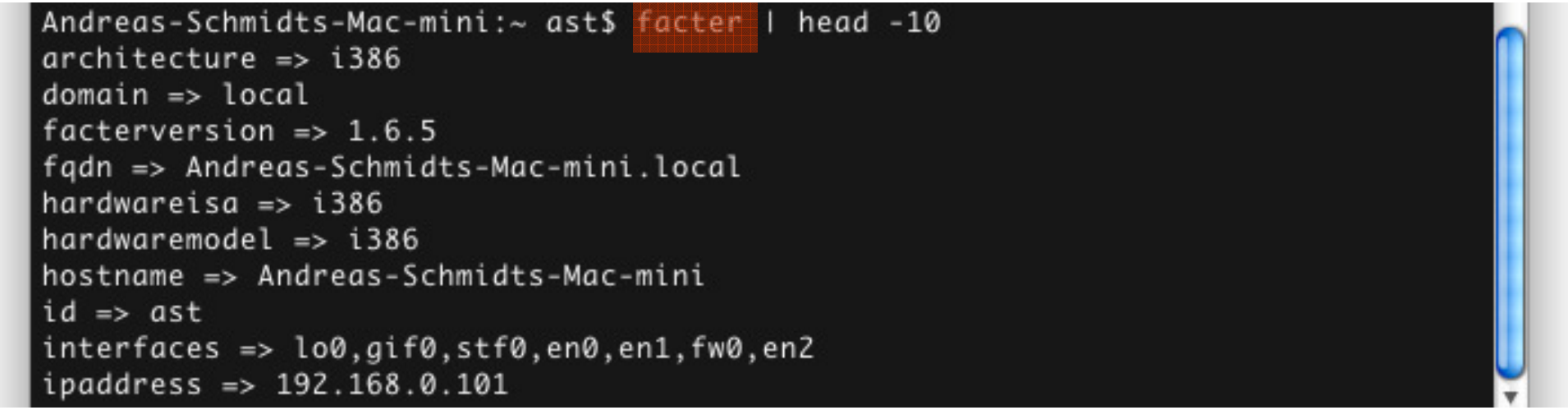
Der Host erhält die Identität durch das was er ist und was er enthält.

Immer aktuell

Ich facter, also bin ich



[...]

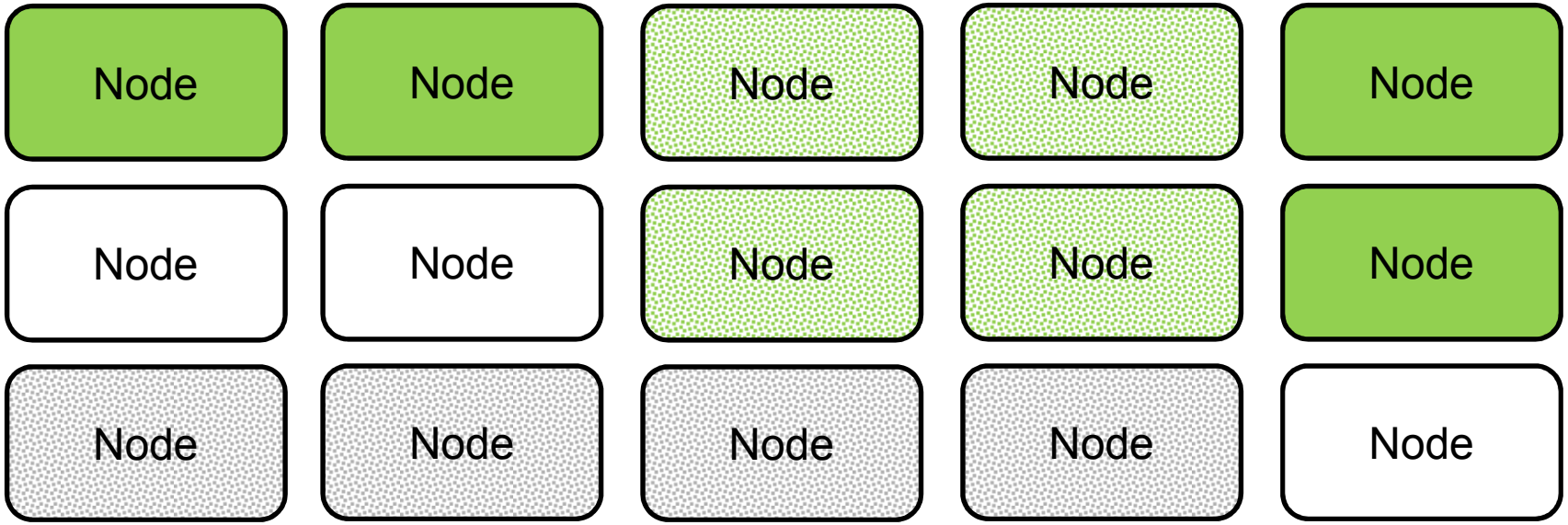


[...]

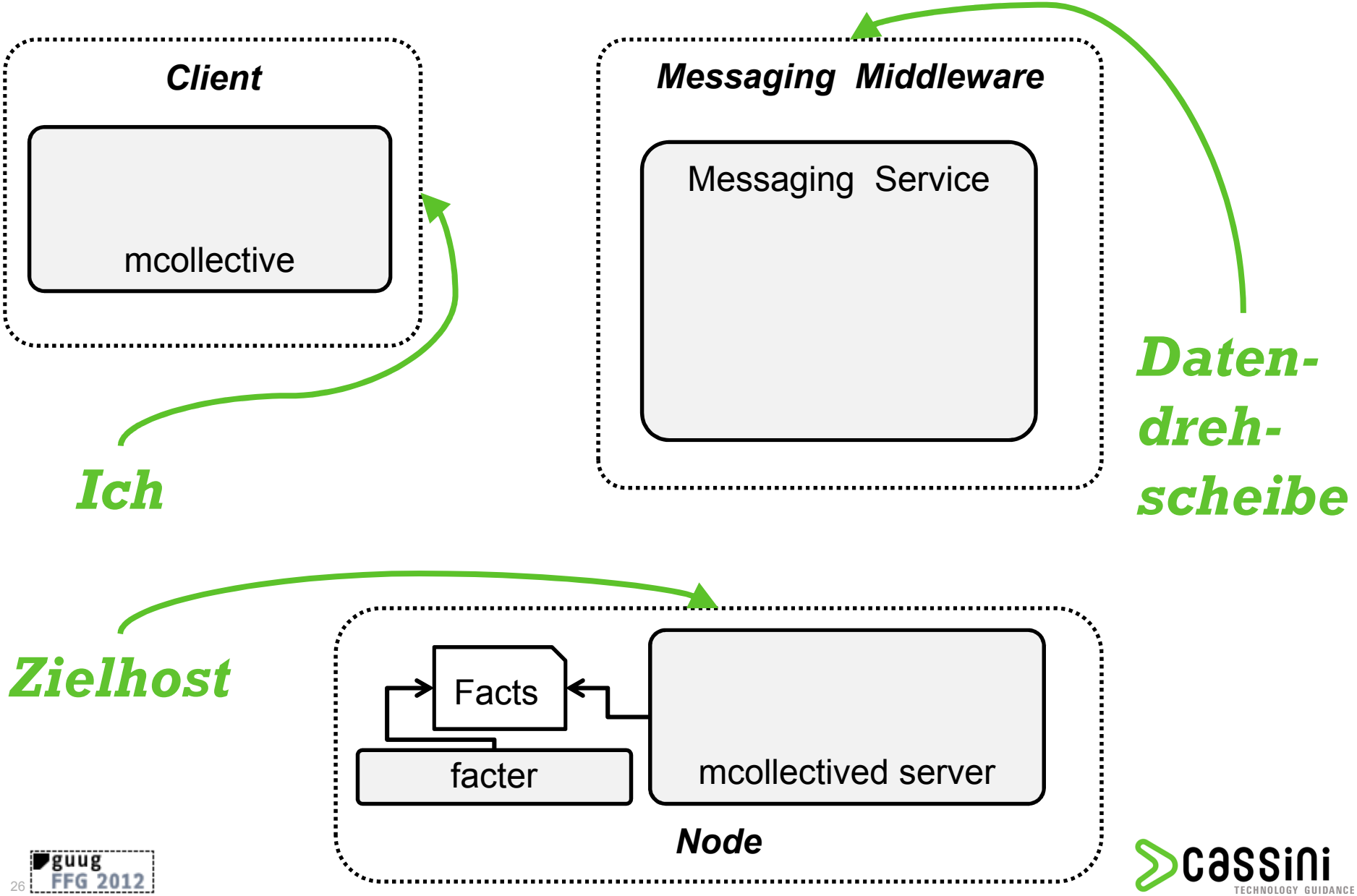
Adressierung der Zielhosts über Facts

```
--with-fact    ipaddress_eth1=~^192.168.10.[0-9]+$

--with-class   mytomcat
```



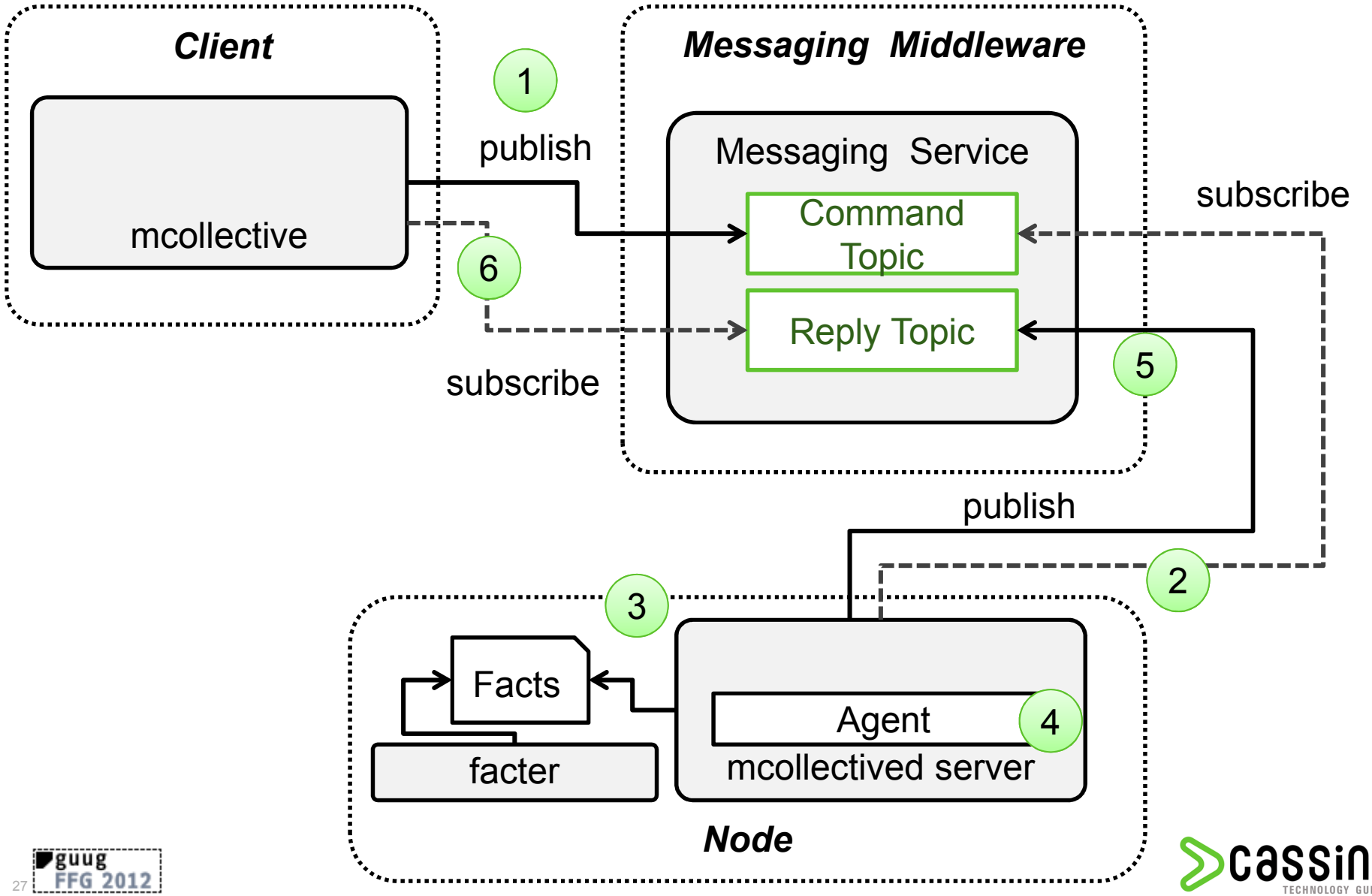
Komponenten eines Collectives



guug FFG 2012

cassini TECHNOLOGY GUIDANCE

Kommunikation zwischen Client und Node



Beispiel: Ping

```
$ mco ping -T test

web01a                   time=128.38 ms
app01b                   time=131.07 ms
web01b                   time=134.14 ms
app02a                   time=147.42 ms
app02b                   time=152.98 ms
web01c                   time=159.73 ms
[...]

---- ping statistics ----
24 replies max: 549.12 min: 128.38 avg: 315.18
```

Collective

Beispiel: Ping

Fact

```
$ mco ping -T test --with-fact hostname=~web

web01a                  time=133.90 ms
web01b                  time=149.21 ms
web01c                  time=163.08 ms

---- ping statistics ----
3 replies max: 163.08 min: 133.90 avg: 148.73
```

Agent

=

Data Description Language (DDL)

+

Agent Code

Beispiel: NRPE-Agent // DDL

selbstbeschreibend

```
action "runcommand", :description => "Run a NRPE command" do
    input :command,
          :prompt => "Command",
          :description => "NRPE command to run",
          :type => :string,
          :validation => '^[a-zA-Z0-9_-]+$',
          :optional => false,
          :maxlength => 50

    output :output,
           :description => "Output from the Nagios plugin",
           :display_as => "Output"
[...]
```

Validierung

https://github.com/puppetlabs/mcollective-plugins/blob/777597f9904e4a8e744d7735415ec517b7a5d5e7/agent/nrpe/agent/nrpe.ddl

Beispiel: NRPE-Agent // Agent Code

```
module MCollective
  module Agent
    class Nrpe<RPC::Agent
      action "runcommand" do
        validate :command, :shellsafe

        command = plugin_for_command(request[:command])

[...]
        reply[:exitcode] = run(command[:cmd],
                               :stdout => :output, :chomp => true)

        case reply[:exitcode]
        when 0
          reply.statusmsg = "OK"
[...]
        when 2
          reply.fail "CRITICAL"
[...]
        else
          reply.fail "UNKNOWN"
        end
[...]
```

https://github.com/puppetlabs/mcollective-plugins/blob/777597f9904e4a8e744d7735415ec517b7a5d5e7/agent/nrpe/agent/nrpe.rb

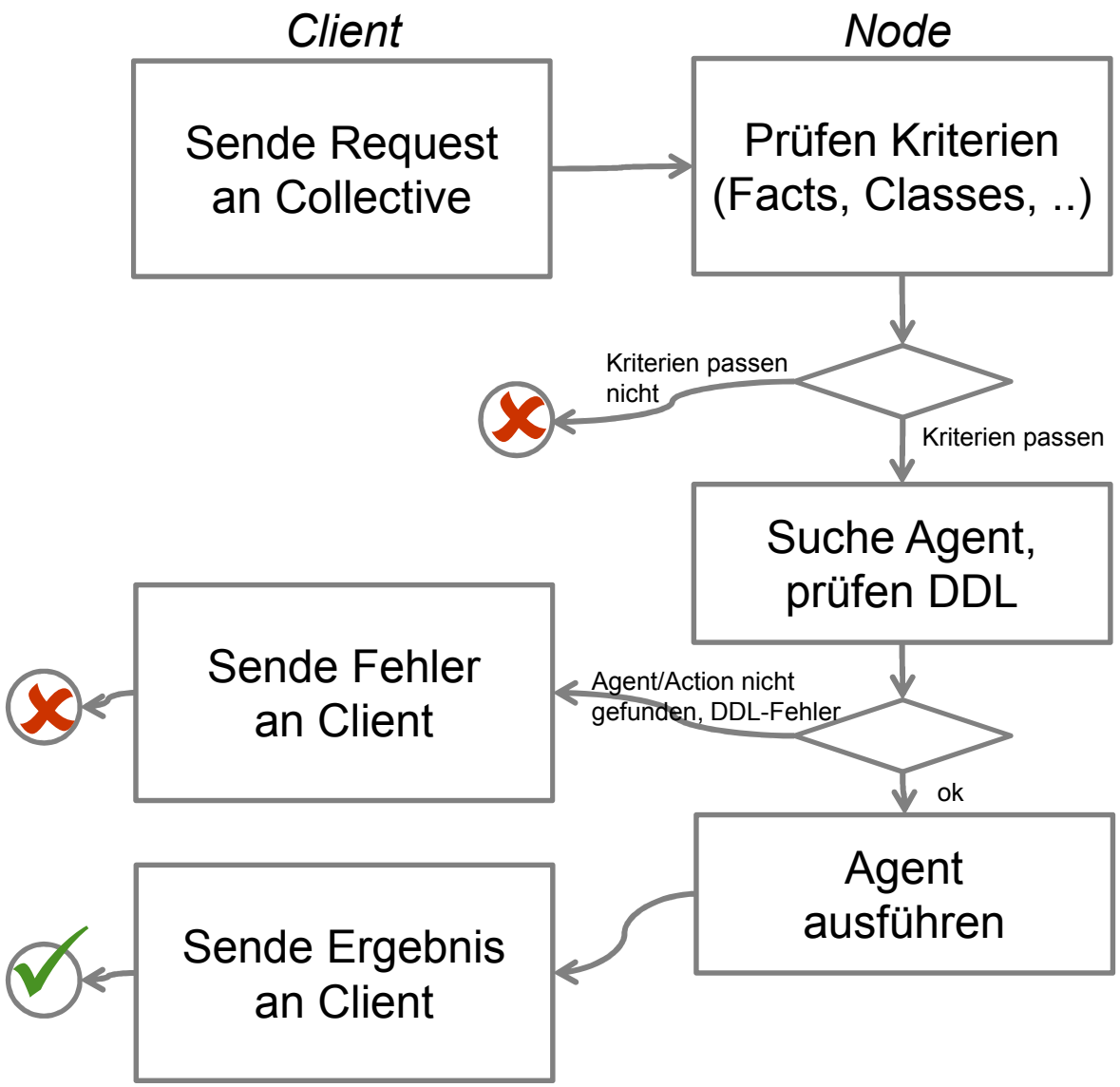
Räume das Cache-Verzeichnis /tmp/cx0 auf - aber nur auf CentOS- Nodes mit einer IP-Adresse in der DMZ

```
action "cleanup", :description => "clean up cache" do

   input :cachepart,
         :prompt      => "Cache Part",
         :description => "What part of cache to clean",
         :type        => :string,
         :validation  => '^[a-zA-Z\-_\d]+$',
         :optional    => false,
         :maxlength   => 30
[...]
```

```
$ mco rpc     --agent cachemgr --action cleanup \
              --arg cachepart=cx0 \
              -T produktion \
              --with-fact operatingsystem=CentOS \
              --with-fact ipaddress=~192.168.10
```

Ablauf Request / Response



Starte alle Tomcats durch

- auf denen die Applikation iWTF deployed ist.

Custom Fact

```
Facter.add(„tomcat_app_iwtf") do
  setcode do
    File.exists? „/usr/libexec/tomcat/webapps/iwtf.war“
  end
end
```

```
$ mco rpc     --agent service --action restart \
              --arg service=tomcat \
              -T produktion -v \
              --with-fact tomcat_app_iwtf=true
```

Wobei hilft mir MCollective?

Ein API ist der Kommandozeile überlegen

Dashboards und GUIs

Dokumentation

Qualität

Wiederverwendbarkeit

Vereinfachung

Abstraktion

Rechte- und Rollentrennung
(z.B. 2nd / 3rd level)

Parallelität

Geschwindigkeit

Skalierbarkeit

Cassini Consulting GmbH
Technology Guidance

Andreas Schmidt
twitter @aschmidt75
andreas.schmidt@cassini.de

Halskestraße 46
40880 Ratingen
Deutschland

T +49 (0)21 02 94 34 737
F +49 (0)21 02 94 34 738
visit www.cassini.de

guug FFG 2012

cassini TECHNOLOGY GUIDANCE