



LOLCATS.COM

# Izolacja i generalizacja: jak pomagają administratorowi?

Stanisław „dozzie” Klekot

Dni Wolnego Oprogramowania  
3 marca 2012

izolacja

## izolacja

Oddzielenie od siebie dwóch elementów. Elementy pracują niezależnie, komunikując się dobrze znanym interfejsem.

## izolacja

Oddzielenie od siebie dwóch elementów. Elementy pracują niezależnie, komunikując się dobrze znanym interfejsem.

## generalizacja

## izolacja

Oddzielenie od siebie dwóch elementów. Elementy pracują niezależnie, komunikując się dobrze znanym interfejsem.

## generalizacja

*Stworzenie* dla zadania narzędzia, które umie trochę więcej niż potrzeba, a jest skonfigurowane do tego zadania.  
Na przykład `cron` albo `inetd`.

# Generalizacja: parę przykładów

powszechnie znane

# Generalizacja: parę przykładów

## powszechnie znane

- crond



# Generalizacja: parę przykładów

## powszechnie znane

- crond
- inetd

# Generalizacja: parę przykładów

## powszechnie znane

- crond
- inetd
- stunnel

# Generalizacja: parę przykładów

## powszechnie znane

- crond
- inetd
- stunnel
- RRDtool (collectd)

# Generalizacja: parę przykładów

## powszechnie znane

- crond
- inetd
- stunnel
- RRDtool (collectd)
- broker *AMQP*

# Generalizacja: parę przykładów

## powszechnie znane

- crond
- inetd
- stunnel
- RRDtool (collectd)
- broker *AMQP*
- shush

# Generalizacja: parę przykładów

## powszechnie znane

- crond
- inetd
- stunnel
- RRDtool (collectd)
- broker *AMQP*
- shush
- Fluentd

# Generalizacja: parę przykładów

## powszechnie znane

- crond
- inetd
- stunnel
- RRDtool (collectd)
- broker *AMQP*
- shush
- Fluentd

# Generalizacja: parę przykładów

## powszechnie znane

- crond
- inetd
- stunnel
- RRDtool (collectd)
- broker *AMQP*
- shush
- Fluentd

## stworzone przeze mnie



# Generalizacja: parę przykładów

## powszechnie znane

- crond
- inetd
- stunnel
- RRDtool (collectd)
- broker *AMQP*
- shush
- Fluentd

## stworzone przeze mnie

- cfgen

# Generalizacja: parę przykładów

## powszechnie znane

- crond
- inetd
- stunnel
- RRDtool (collectd)
- broker *AMQP*
- shush
- Fluentd

## stworzone przeze mnie

- cfgen
- cronbuilder

# Generalizacja: parę przykładów

## powszechnie znane

- crond
- inetd
- stunnel
- RRDtool (collectd)
- broker *AMQP*
- shush
- Fluentd

## stworzone przeze mnie

- cfgen
- cronbuilder
- xmlrpcd

# Generalizacja: jak to działa?

*reusable code blocks*

# Generalizacja: jak to działa?

## *reusable code blocks*

- sprawdzone (*bugless*)

## *reusable code blocks*

- sprawdzone (*bugless*)
- dobrze poznane (przewidywalne)

# Generalizacja: jak to działa?

## *reusable code blocks*

- sprawdzone (*bugless*)
- dobrze poznane (przewidywalne)
- więcej osób zna/ma szansę znać

## *reusable code blocks*

- sprawdzone (*bugless*)
- dobrze poznane (przewidywalne)
- więcej osób zna/ma szansę znać
- klocki używają klocków



zyski

## zyski

- mniej błędów i przeoczeń

## zyski

- mniej błędów i przeoczeń
- łatwiejsze projektowanie (duże *black boxes*)

## zyski

- mniej błędów i przeoczeń
- łatwiejsze projektowanie (duże *black boxes*)
- łatwiejsze planowanie zmian

## zyski

- mniej błędów i przeoczeń
- łatwiejsze projektowanie (duże *black boxes*)
- łatwiejsze planowanie zmian
- łatwiejsze śledzenie postępów

# Dlaczego warto izolować elementy?

Po co izolacja?

# Dlaczego warto izolować elementy?

Po co izolacja?

Łatwo wymienić na lepszy pomysł

# Dlaczego warto izolować elementy?

## Po co izolacja?

Łatwo wymienić na lepszy pomysł

- cfengine + cgen



# Dlaczego warto izolować elementy?

## Po co izolacja?

Łatwo wymienić na lepszy pomysł

- cfengine + cfgen
- cfengine + logi z wykonania

# Dlaczego warto izolować elementy?

## Po co izolacja?

Łatwo wymienić na lepszy pomysł

- cfengine + cfgen
- cfengine + logi z wykonania
- cfengine + SVN/git/whatever

# Dlaczego warto izolować elementy?

## Po co izolacja?

Łatwo wymienić na lepszy pomysł

- cfengine + cfgen
- cfengine + logi z wykonania
- cfengine + SVN/git/whatever
- antyprzykład: Puppet + puppetmaster

# Dlaczego warto izolować elementy?

## Po co izolacja?

Łatwo wymienić na lepszy pomysł

- cfengine + cfgen
- cfengine + logi z wykonania
- cfengine + SVN/git/whatever
- antyprzykład: Puppet + puppetmaster

# Dlaczego warto izolować elementy?

## Po co izolacja?

Łatwo wymienić na lepszy pomysł

- cfengine + cfgen
- cfengine + logi z wykonania
- cfengine + SVN/git/whatever
- antyprzykład: Puppet + puppetmaster

Ogólnie, zamiatanie pod dywan.

# Dlaczego warto izolować elementy?

Po co izolacja?

# Dlaczego warto izolować elementy?

## Po co izolacja?

Łatwo zmodyfikować jeden element

# Dlaczego warto izolować elementy?

## Po co izolacja?

Łatwo zmodyfikować jeden element

- *XML-RPC* i rejestracja serwera



# Dlaczego warto izolować elementy?

## Po co izolacja?

Łatwo zmodyfikować jeden element

- *XML-RPC* i rejestracja serwera
- transport logów

# Dlaczego warto izolować elementy?

## Po co izolacja?

Łatwo zmodyfikować jeden element

- *XML-RPC* i rejestracja serwera
- transport logów
- *web dashboard* z raportami

# Dlaczego warto izolować elementy?

Po co izolacja?

# Dlaczego warto izolować elementy?

Po co izolacja?

Specjalne uprawnienia

# Dlaczego warto izolować elementy?

## Po co izolacja?

### Specjalne uprawnienia

- *web frontend* do firewalla

# Dlaczego warto izolować elementy?

## Po co izolacja?

### Specjalne uprawnienia

- *web frontend* do firewalla
- dynDNS

# Dlaczego warto izolować elementy?

## Po co izolacja?

### Specjalne uprawnienia

- *web frontend* do firewalla
- dynDNS
- rejestracja serwera w sieci

# Dlaczego warto izolować elementy?

## Po co izolacja?

### Specjalne uprawnienia

- *web frontend* do firewalla
- dynDNS
- rejestracja serwera w sieci
- pchnięcie konfiguracji



# Dlaczego warto izolować elementy?

## Po co izolacja?

### Specjalne uprawnienia

- *web frontend* do firewalla
- dynDNS
- rejestracja serwera w sieci
- pchnięcie konfiguracji
- restart daemona

# Przykład z życia

```
rsync --daemon
```

```
rsync --daemon
```

- UID?

```
rsync --daemon
```

- UID?
- katalog docelowy na serwerze?

```
rsync --daemon
```

- UID?
- katalog docelowy na serwerze?
- atomowość uploadu?

```
rsync --daemon
```

- UID?
- katalog docelowy na serwerze?
- atomowość uploadu?
- przygotowanie danych tuż przed uploadem?

*cfengine deployment*

## *cfengine deployment*

- SVN ze źródłami, SVN z wygenerowanymi konfigami



## *cfengine deployment*

- SVN ze źródłami, SVN z wygenerowanymi konfigami
  - adres

## *cfengine deployment*

- SVN ze źródłami, SVN z wygenerowanymi konfigami
  - adres
  - *credentials*

## *cfengine deployment*

- SVN ze źródłami, SVN z wygenerowanymi konfigami
  - adres
  - *credentials*
  - struktura katalogów

## *cfengine deployment*

- SVN ze źródłami, SVN z wygenerowanymi konfigami
  - adres
  - *credentials*
  - struktura katalogów
  - zezwolenia do zapisu

## *cfengine deployment*

- SVN ze źródłami, SVN z wygenerowanymi konfigami
  - adres
  - *credentials*
  - struktura katalogów
  - zezwolenia do zapisu
- grupy domyślne

## *cfengine deployment*

- SVN ze źródłami, SVN z wygenerowanymi konfigami
  - adres
  - *credentials*
  - struktura katalogów
  - zezwolenia do zapisu
- grupy domyślne
- zakładanie konta w SVN: httpswd? LDAP? PostgreSQL?

dynDNS

## dynDNS

- backend: BIND? PowerDNS? (nie daj Boże) djbdns?



## dynDNS

- backend: BIND? PowerDNS? (nie daj Boże) djbdns?
- *updates*: RFC 2136? PostgreSQL? LDAP?

## dynDNS

- backend: BIND? PowerDNS? (nie daj Boże) djbdns?
- *updates*: RFC 2136? PostgreSQL? LDAP?
- położenie serwera DNS/DB?

## dynDNS

- backend: BIND? PowerDNS? (nie daj Boże) djbdns?
- *updates*: RFC 2136? PostgreSQL? LDAP?
- położenie serwera DNS/DB?
- *credentials*?

```
collectd
```

## collectd

- plugin exec: co jest zbierane? skąd? jak?

## collectd

- plugin exec: co jest zbierane? skąd? jak?
- RRD

## collectd

- plugin `exec`: co jest zbierane? skąd? jak?
- RRD
  - funkcje agregujące dane?

## collectd

- plugin `exec`: co jest zbierane? skąd? jak?
- RRD
  - funkcje agregujące dane?
  - ograniczenia danych (przedział)?



## collectd

- plugin `exec`: co jest zbierane? skąd? jak?
- RRD
  - funkcje agregujące dane?
  - ograniczenia danych (przedział)?
  - *missing entries*?

## collectd

- plugin `exec`: co jest zbierane? skąd? jak?
- RRD
  - funkcje agregujące dane?
  - ograniczenia danych (przedział)?
  - *missing entries*?
  - retencja danych?

## collectd

- plugin `exec`: co jest zbierane? skąd? jak?
- RRD
  - funkcje agregujące dane?
  - ograniczenia danych (przedział)?
  - *missing entries*?
  - retencja danych?
  - ...

## collectd

- plugin *exec*: co jest zbierane? skąd? jak?
- RRD
  - funkcje agregujące dane?
  - ograniczenia danych (przedział)?
  - *missing entries*?
  - retencja danych?
  - ...
- *web frontend*: jak wyświetlać wykresy?



