Hablemos de Ruby

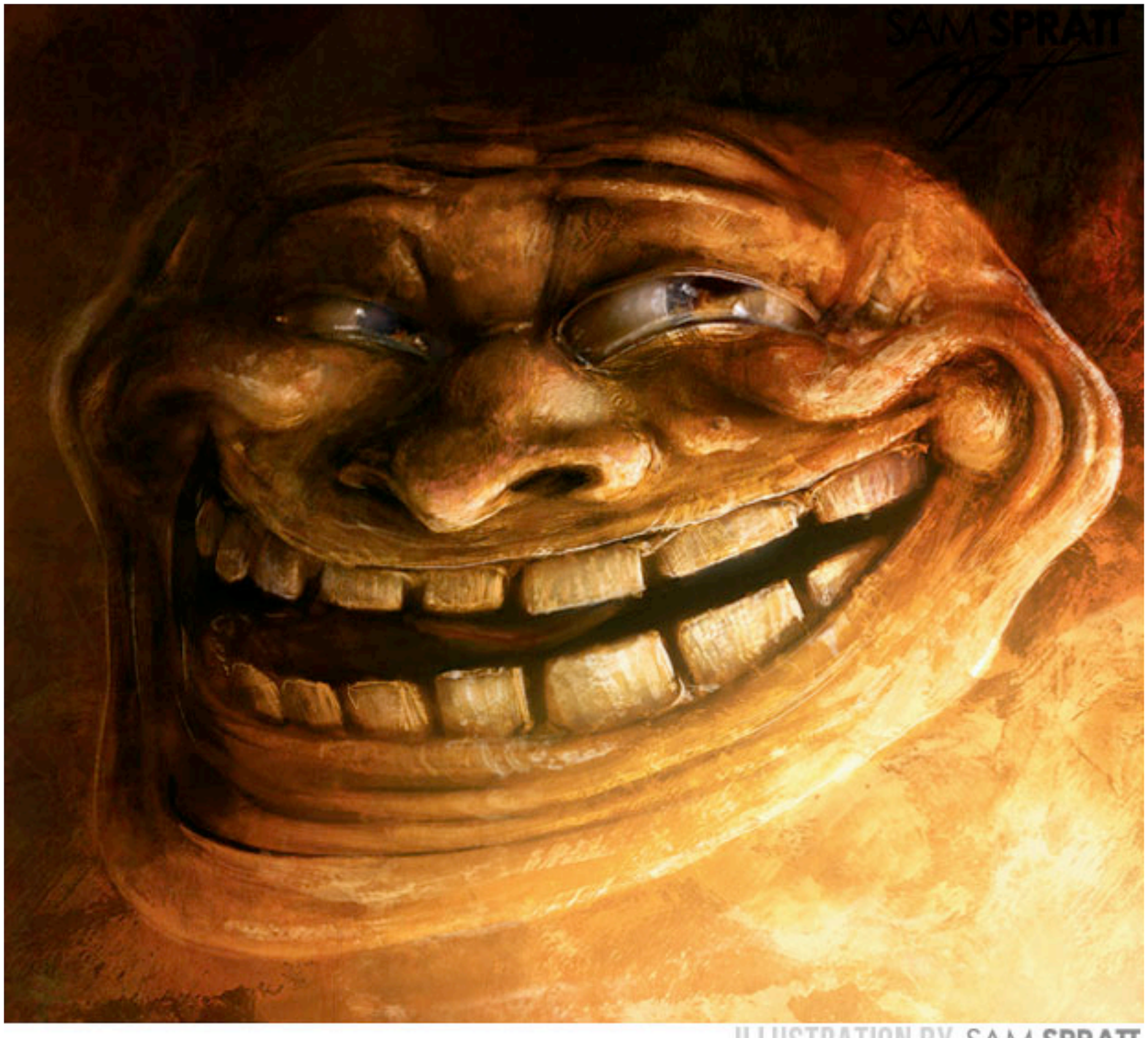


ILLUSTRATION BY SAM **SPRATT**

Pythomnic3k

Introducción a la computación distribuída.

Quién te conoce?

elCuervo

aka Bruno Aguirre

Qué esperar?

- Framework Web
- Comandos útiles
- Deploy a AppEngine

- ~~Framework Web~~
- ~~Comandos útiles~~
- ~~Deploy a AppEngine~~

No está enfocado en desarrollo web.

Frameworks web

- Flask
- Django
- Tornado
- Twisted
- Web2py
- etc, etc, etc...

Qué es entonces?

Pythomnic3k

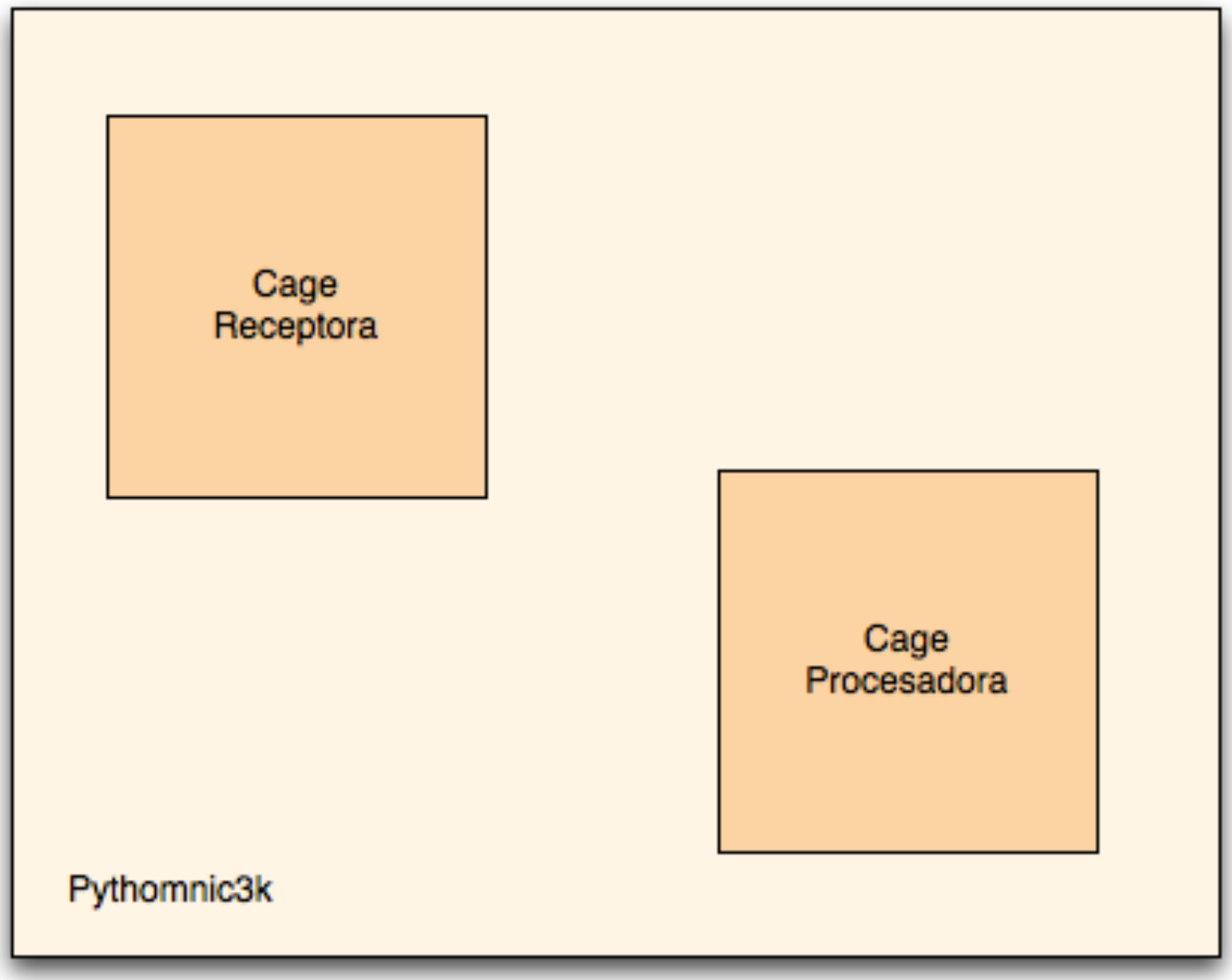
- Desarrollo de middlewares
- Multiprotocolo
- Recarga automática de código
- Multiplataforma
- Transaccional
- Sincrono / Asincrono
- Autodescubrimiento

ILLUSTRATION BY | SAM **SPRATT**

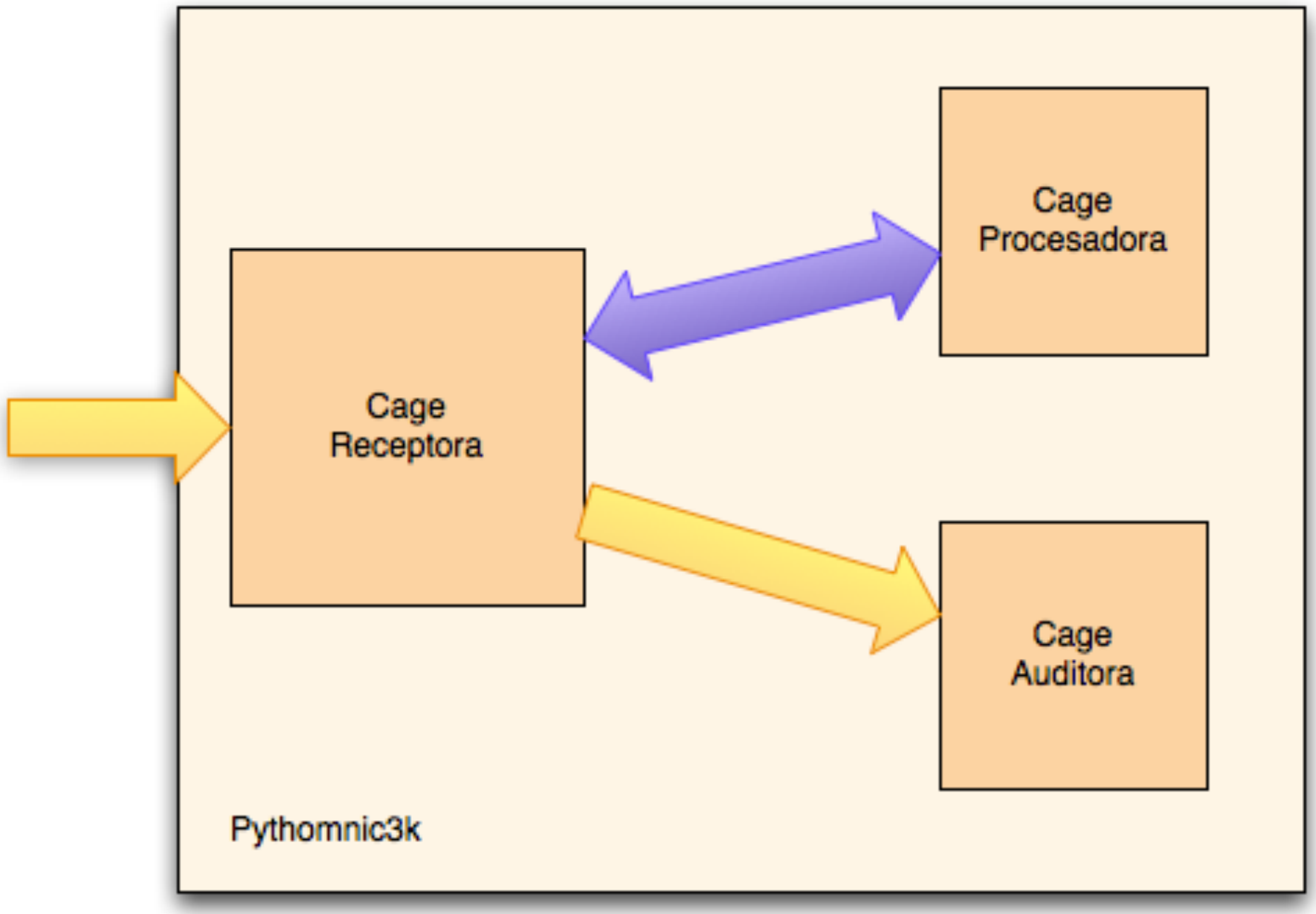
Sobre que hablaremos?

- Cages
- Distribución
- Redundancia
- Recarga
- Timeout
- Queues
- Tolerancia a fallas
- Recarga automatica
- Monitoreo
- Extensible

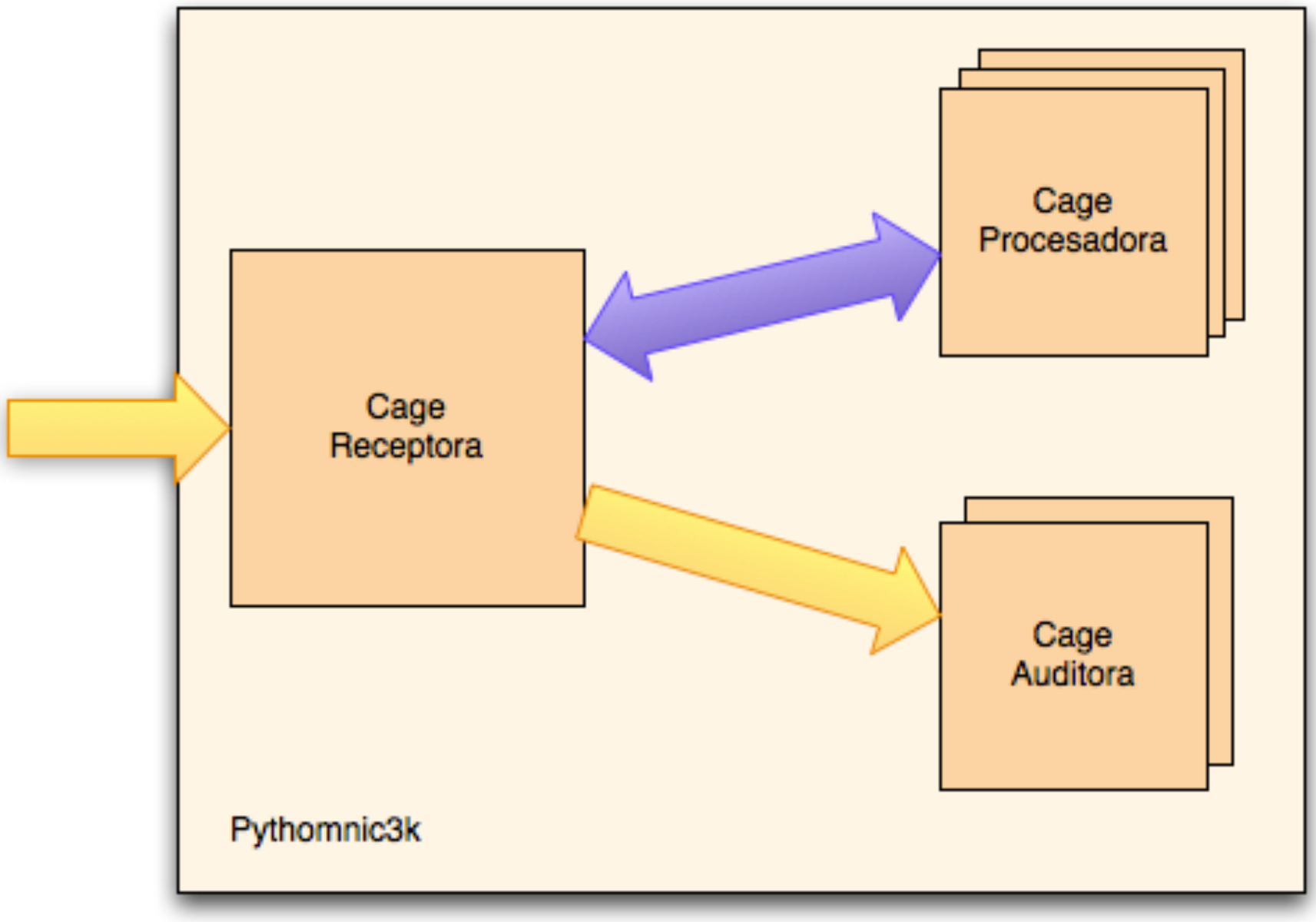
Cages



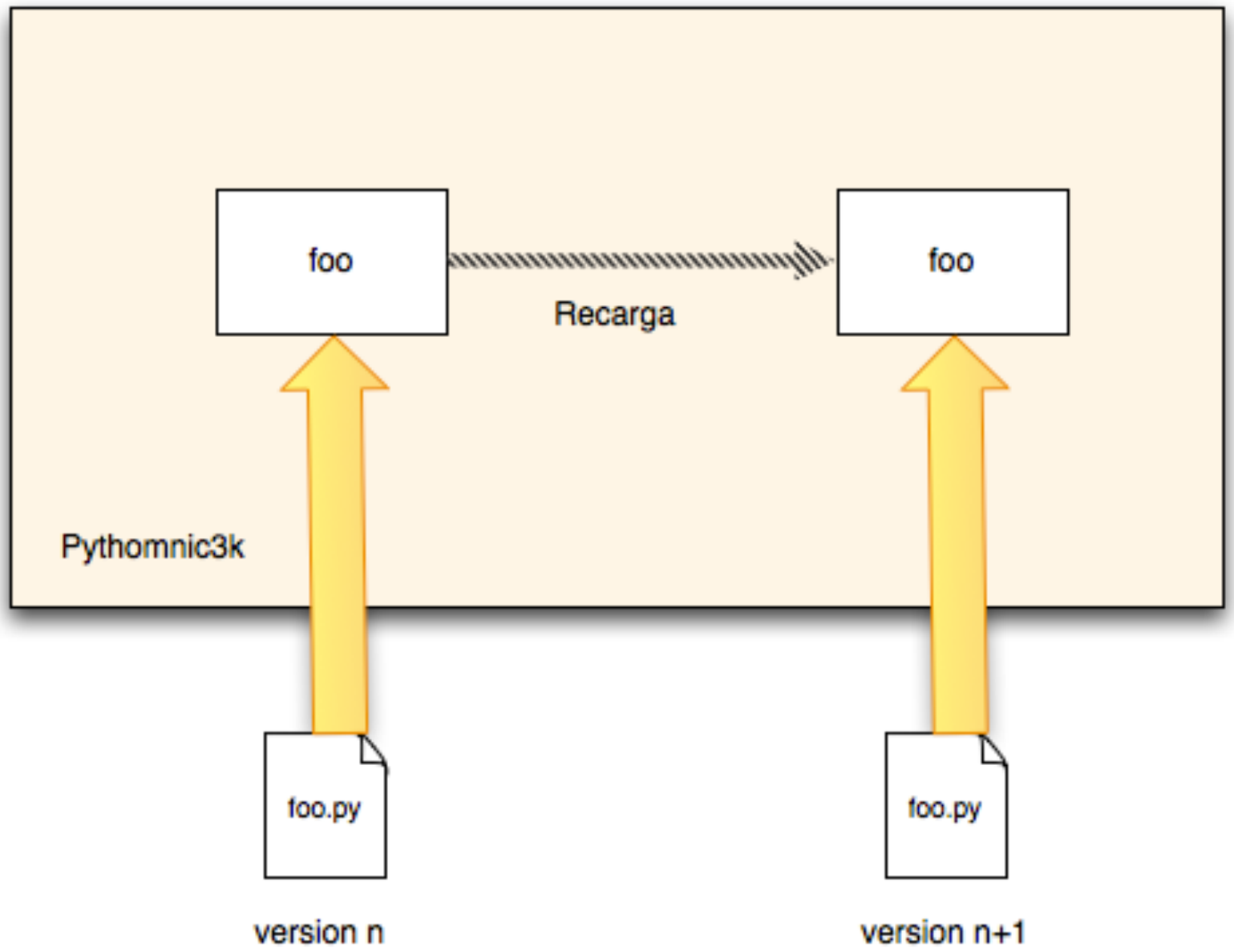
Distribución



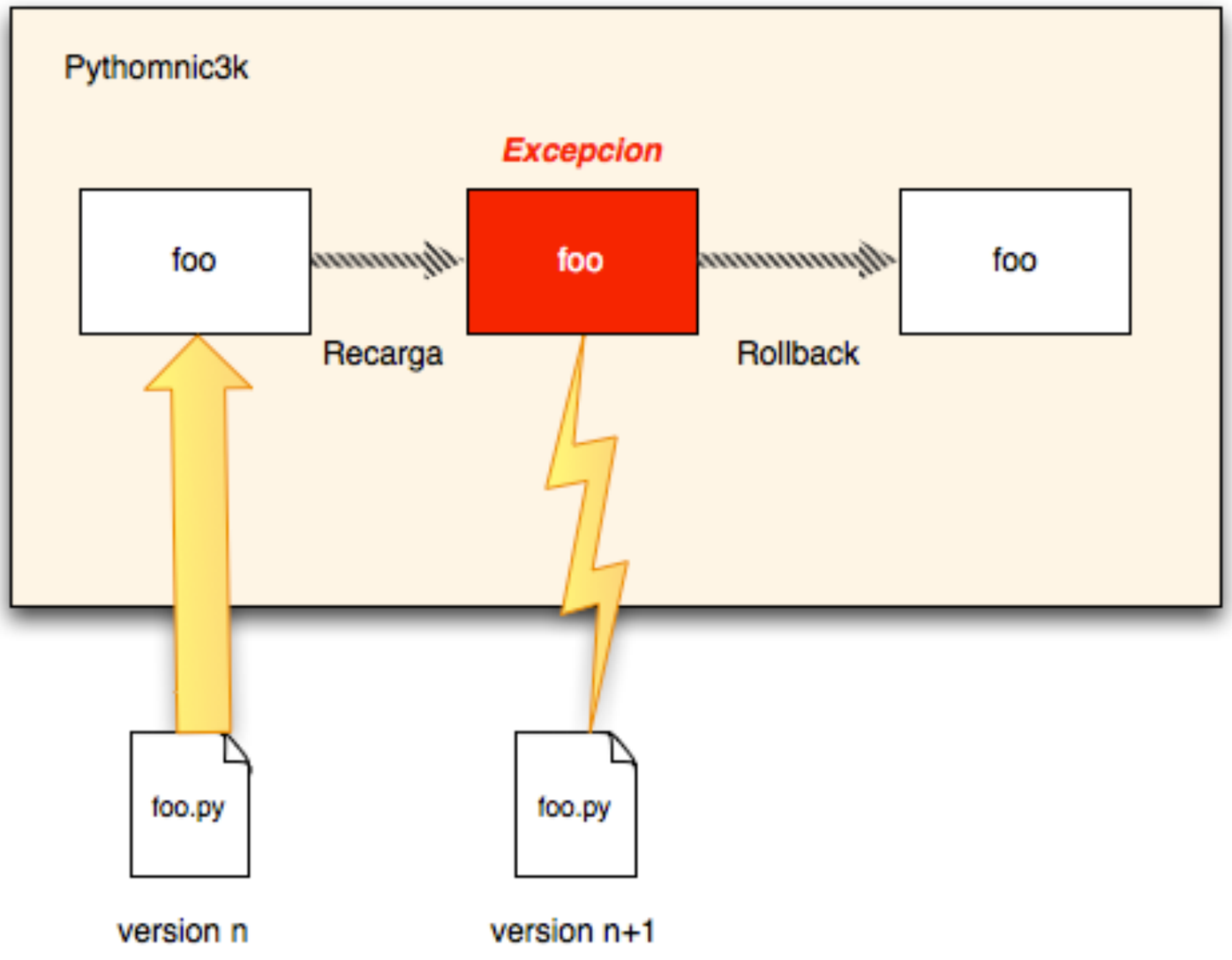
Redundancia



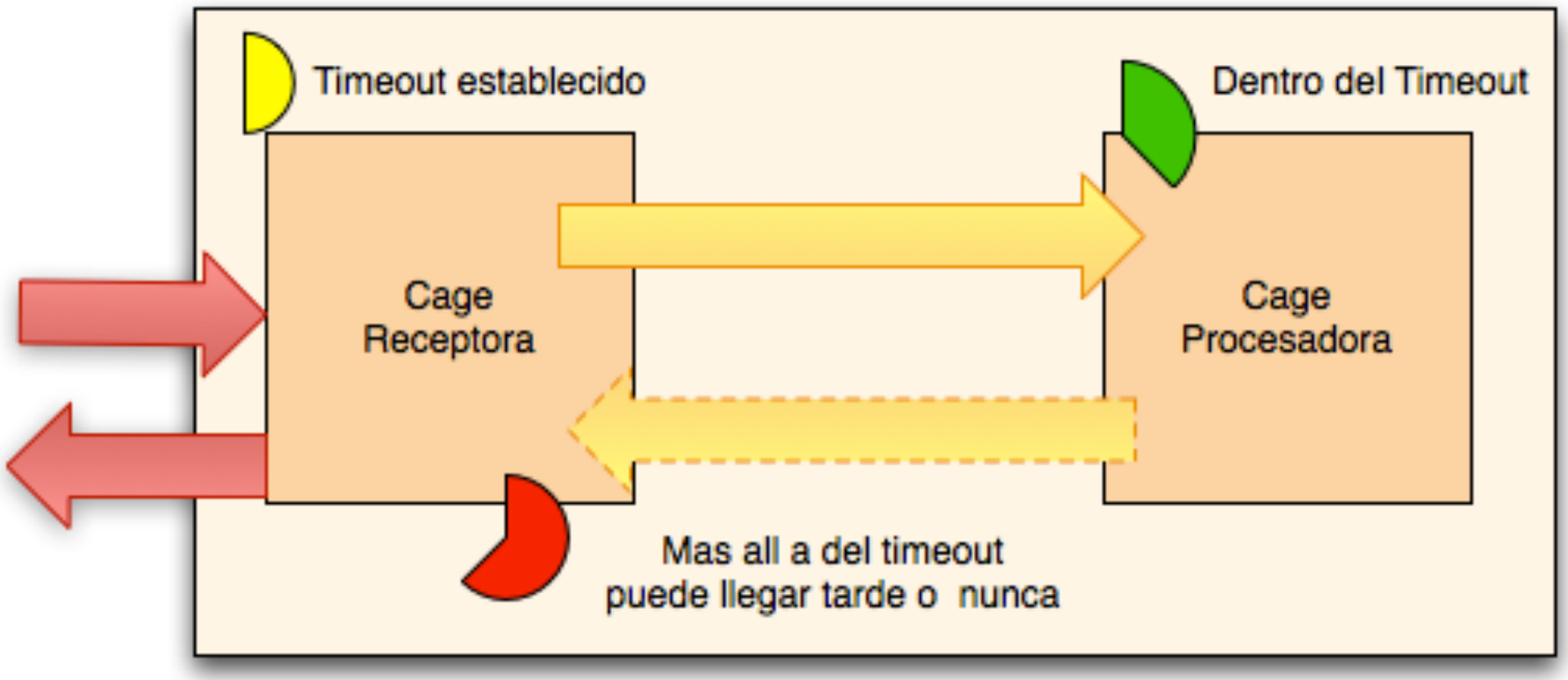
Recarga


foo
Recarga
foo
Pythomnic3k
foo.py
foo.py
version n
version n+1

Recarga cuando cosas malas pasan

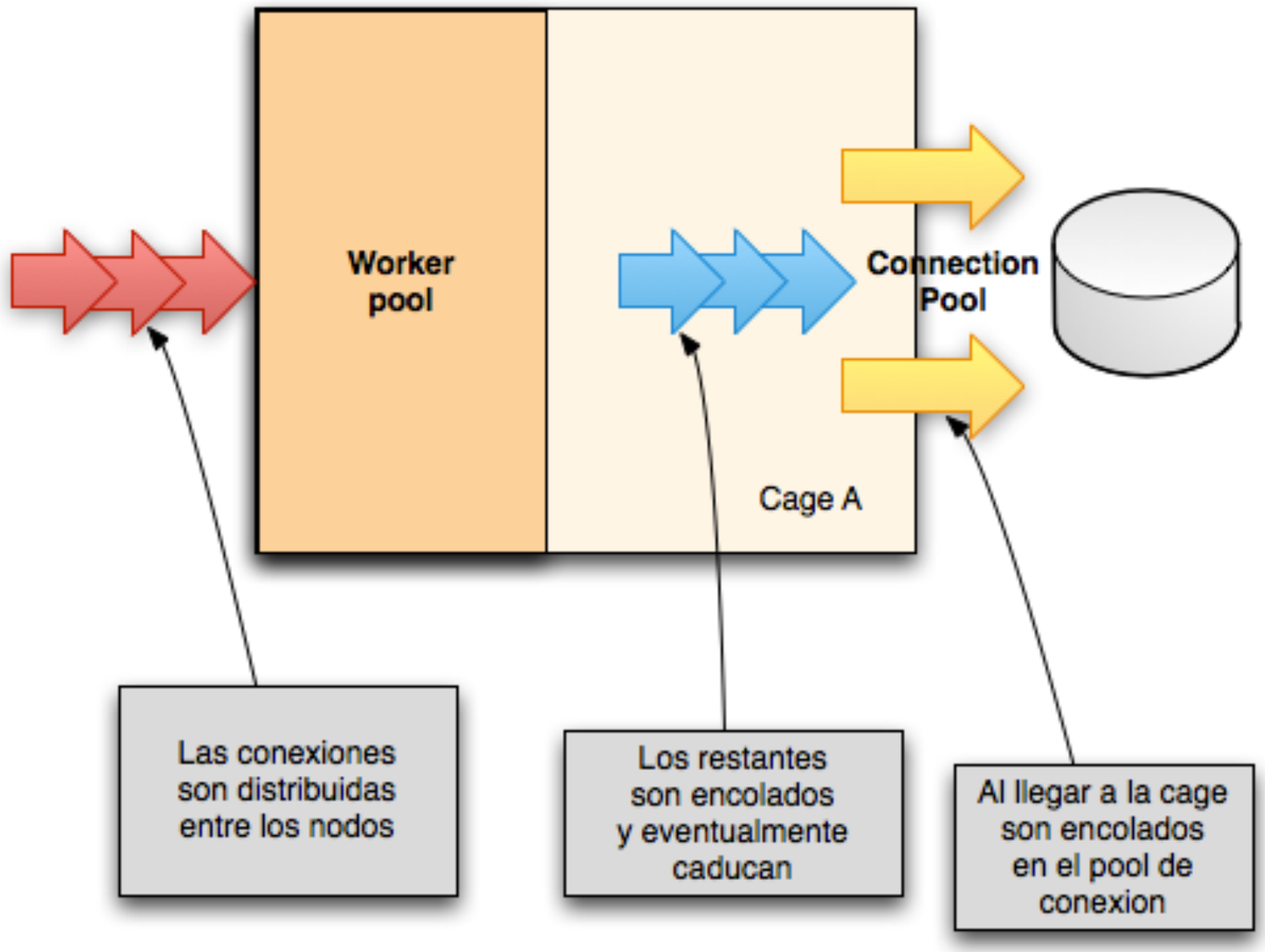


Timeout

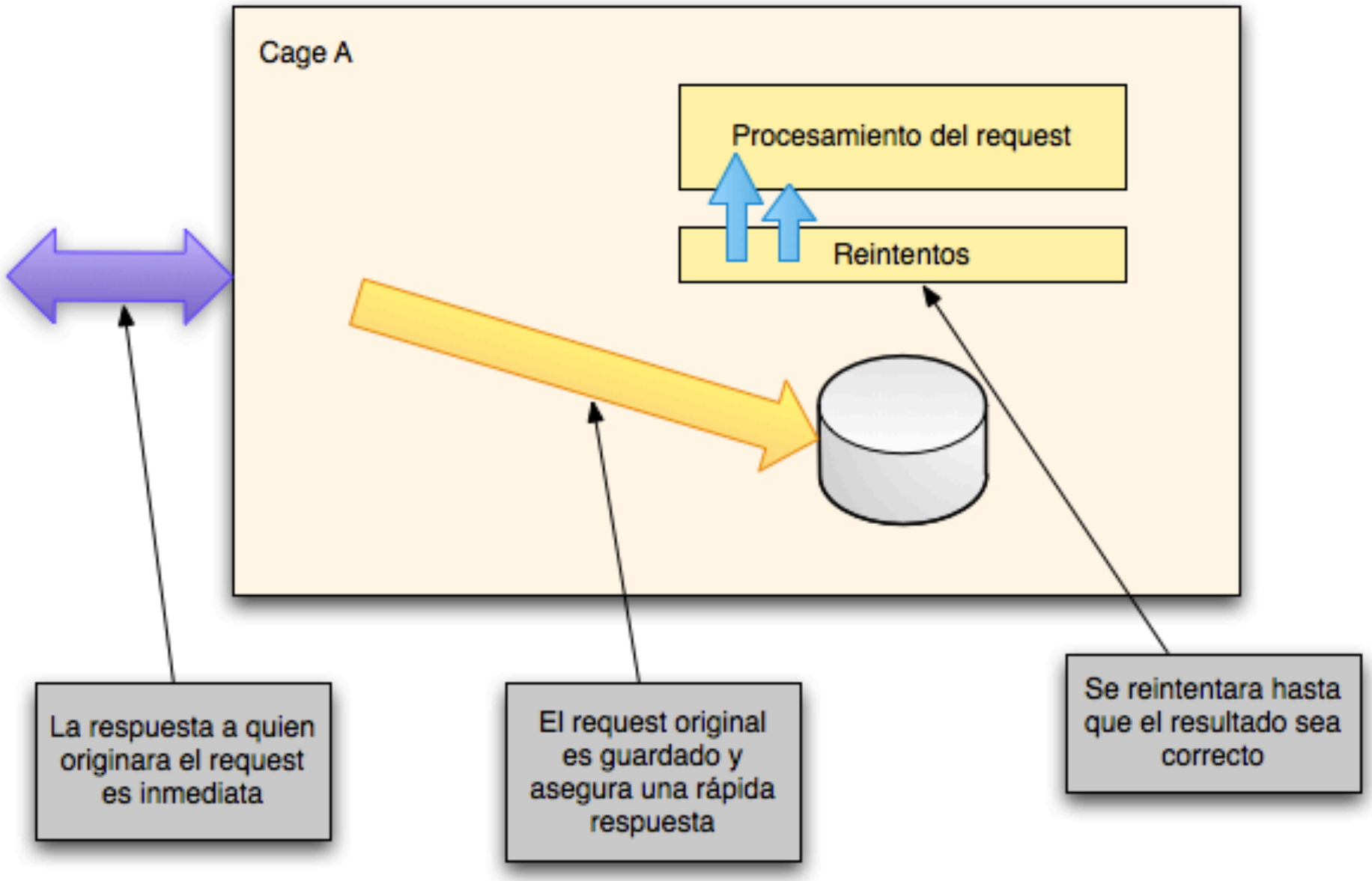


Se envía un mensaje de error por defecto cuando el deadline es alcanzado

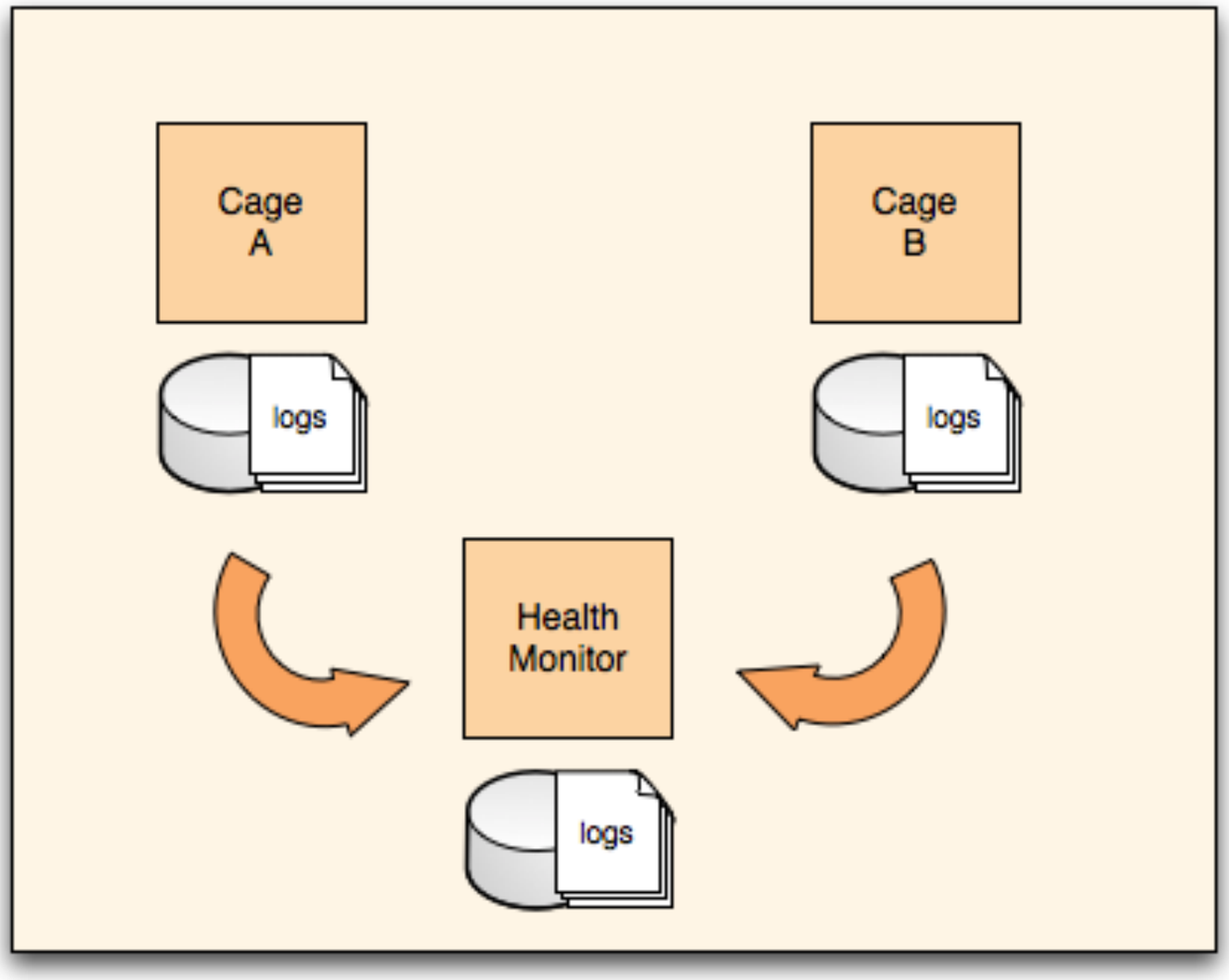
Queues


Worker
pool
Connection
Pool
Cage A
Las conexiones
son distribuidas
entre los nodos
Los restantes
son encolados
y eventualmente
caducan
Al llegar a la cage
son encolados
en el pool de
conexion

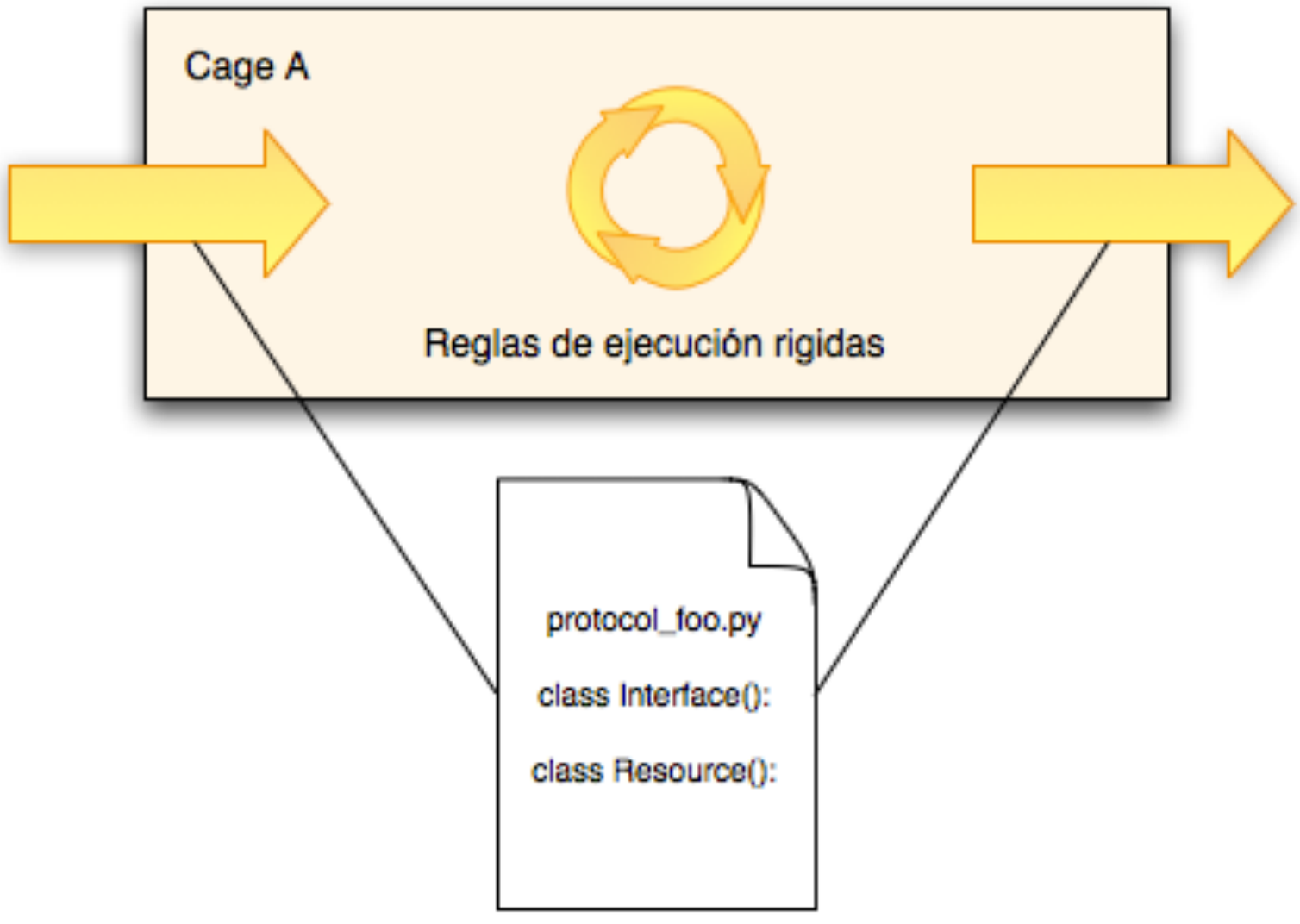
Tolerancia a Fallas



Monitoreo



Extensible


Cage A
Reglas de ejecución rigidas
protocol_foo.py
class Interface():
class Resource():

Protocolos por defecto

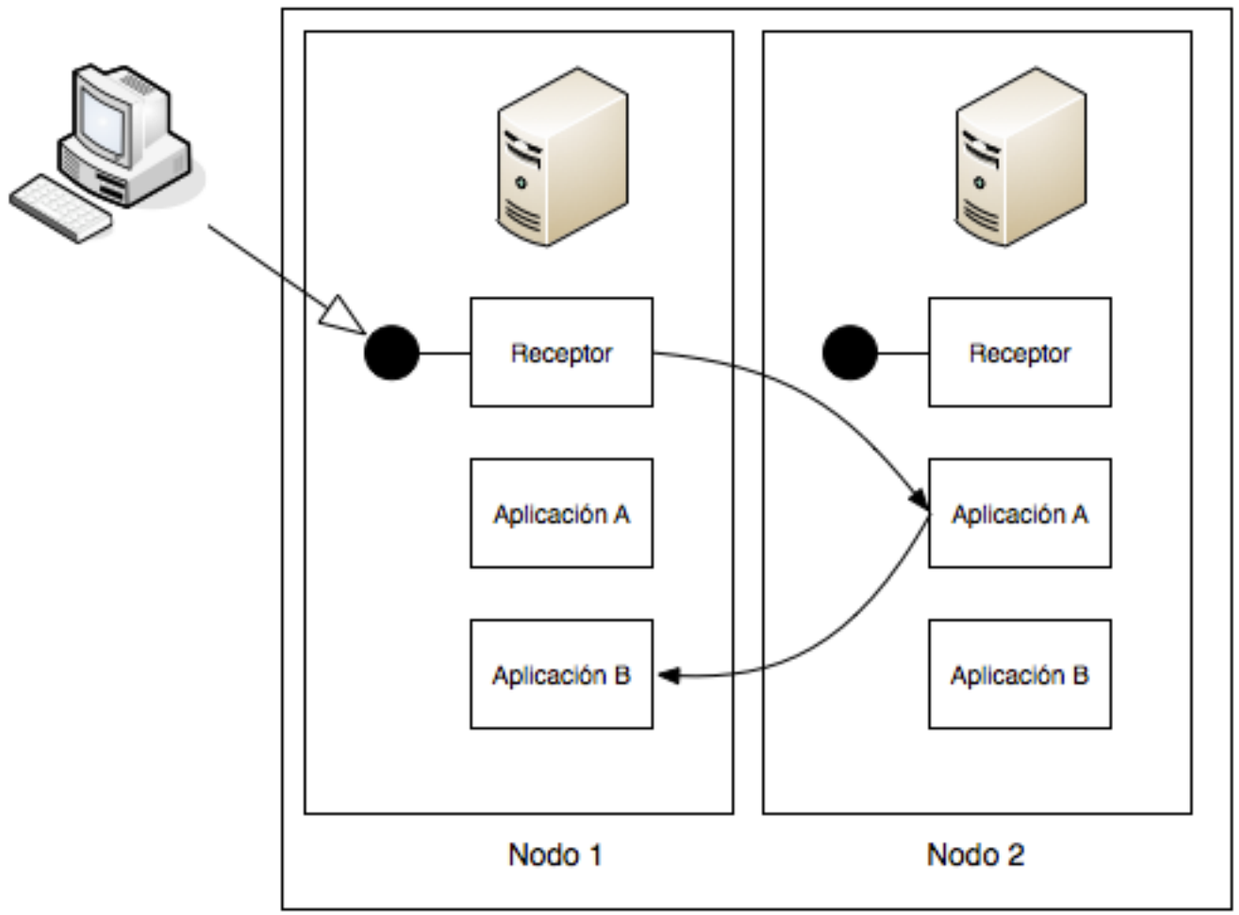
- HTTP(S)
- XMLRPC
- JMS
- EMAIL
- PostgreSQL
- Oracle
- MongoDB

SMPP

v3.4

Para qué lo usaría?

Escalabilidad Horizontal



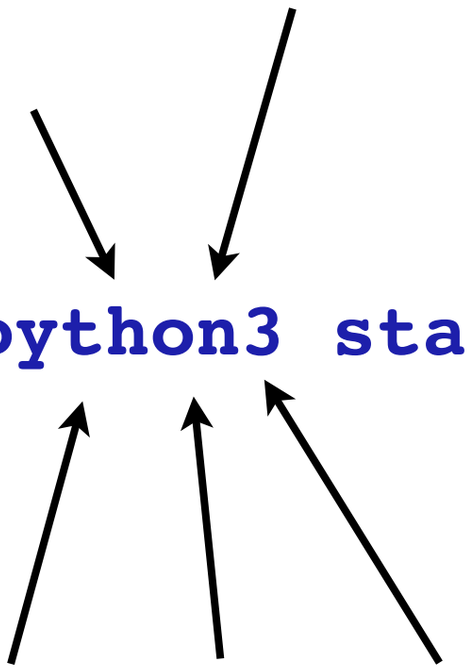
Ejemplo

```
$ git clone https://github.com/elcuervo/pythomnic3k.git
Cloning into pythomnic3k...
remote: Counting objects: 72, done.
remote: Compressing objects: 100% (63/63), done.
remote: Total 72 (delta 7), reused 72 (delta 7)
Unpacking objects: 100% (72/72), done.
```

```
$ cd pythomnic3k
pythomnic3k $ ls -la
README      cages       lib         startup.py
pythomnic3k $ cd cages
cages $ ls -a
.           ..          .shared
```

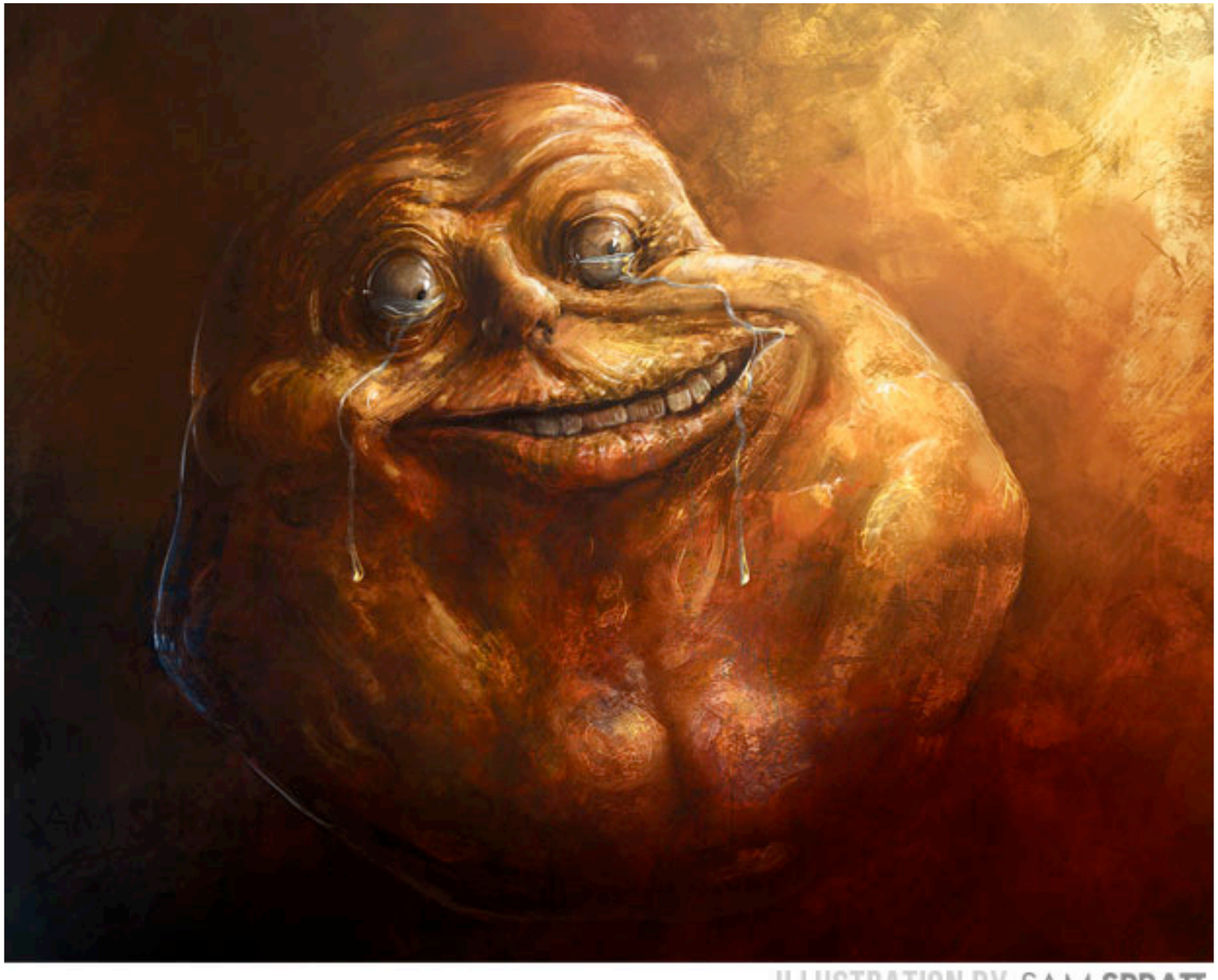

Recursos compartidos
entre las cages

```
cages $ git clone git://github.com/elcuervo/pythomnic3k-sample-webserver.git sample
Cloning into sample...
remote: Counting objects: 18, done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 18 (delta 3), reused 17 (delta 2)
Receiving objects: 100% (18/18), 4.49 KiB, done.
Resolving deltas: 100% (3/3), done.
cages $ ls
sample
```

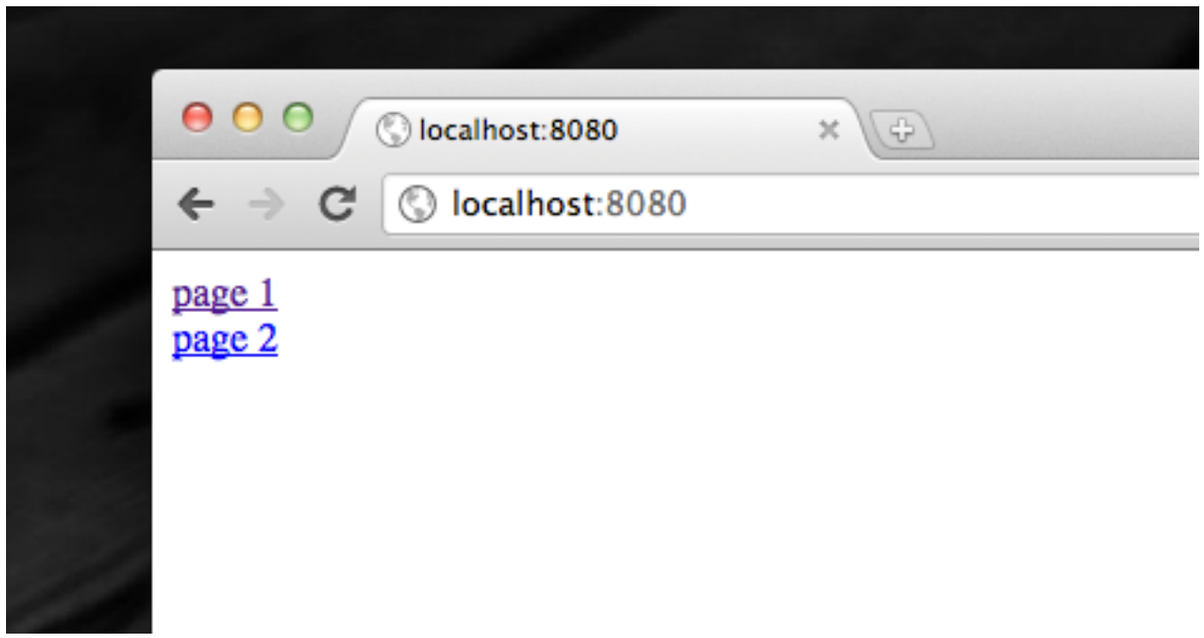


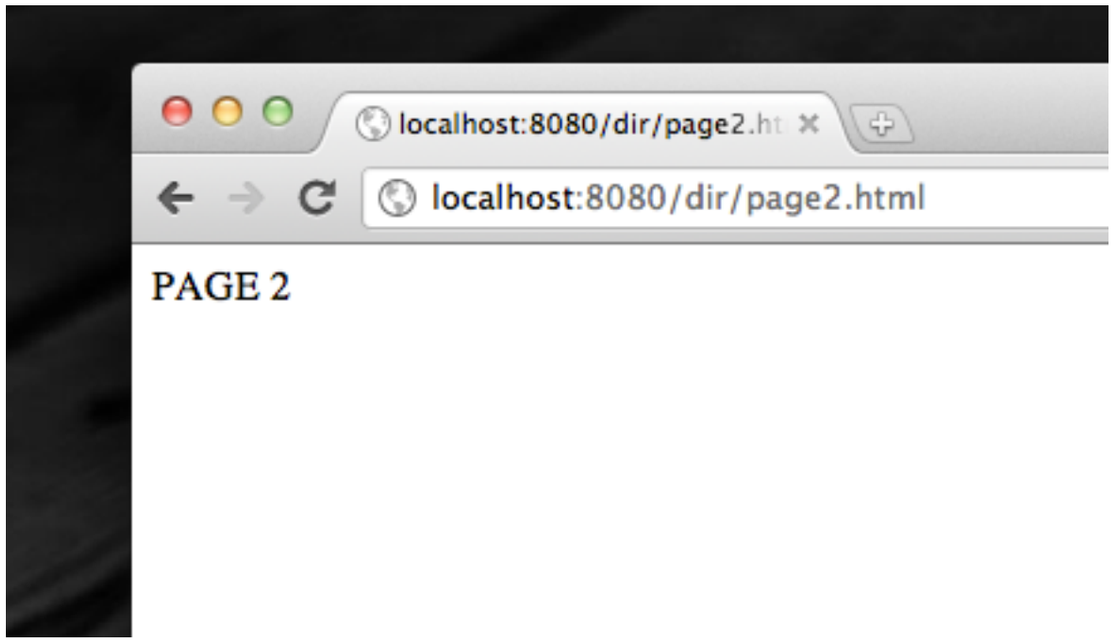
```
cages $ cd ..
pythomnic3k $ python3 startup.py sample
```

Python3 desde RC

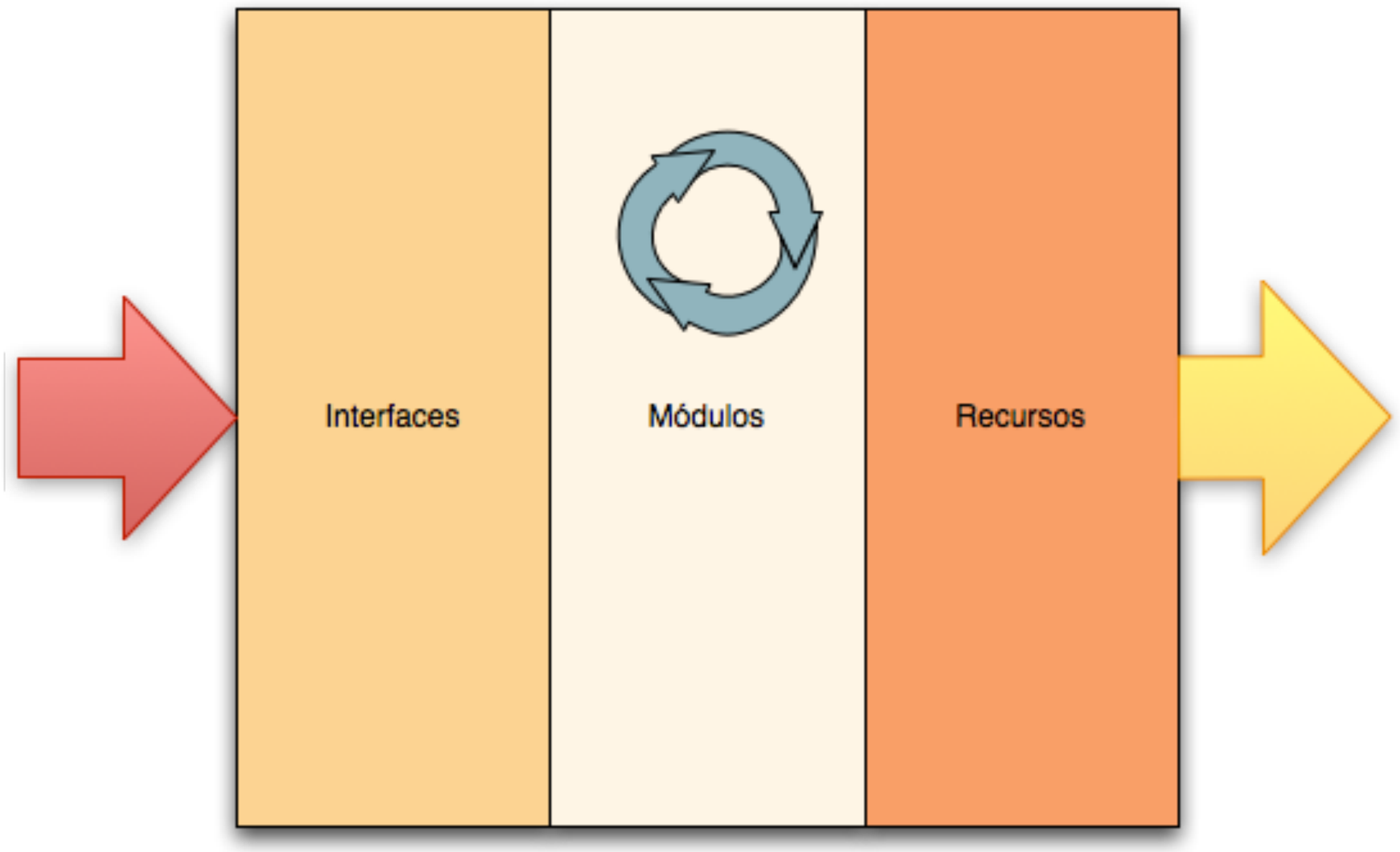


```
cages $ cd ..
pythomnic3k $ python3 startup.py sample
```


localhost:8080
localhost:8080
page 1
page 2


localhost:8080/dir/page2.ht
localhost:8080/dir/page2.html
PAGE 2

Composición de una cage



Composición de los módulos

```
__all__ = [ "method" ] # methods accessible with pmnc calls

def method(...):
    ...

# EOF
```

Comunicación entre módulos

bar.py

```
result = pmnc.foo.make_everything_ok()
```

foo.py

```
__all__ = ["make_everything_ok"]

def make_everything_ok():
    # something

# EOF
```

Recursos transaccionales

```
xa = pmnc.transaction.create()
xa.resource1.execute(...)
xa.resource2.execute(...)
result1, result2 = xa.execute()
```

Sigue?

- github.com/elcuervo
- twitter.com/elcuervocorax
- elcuervo.co