

# 芯雲流程引擎

使用Python协程解决阻塞问题

taowen@gmail.com

# Service Orchestration

DNS  
系统

接入层  
系统

告警  
系统

配置管理  
系统

Game  
Server

SA  
系统

IaaS  
系统

官网  
系统

# Python with @task

```
1  #-*-coding: utf-8 -*-  
2  from bpm.kernel import *  
3  from bk import http  
4  from bk import cc  
5  from bk import gslb  
6  from bk import res  
7  from bk import ijobs  
8  
9  @task  
10 def expand_capacity():  
11     ip = res.request_new_server(model='B6')  
12     cc.update_module(ip, module='GameSvr')  
13     ijobs.exec_job(ip, job_name='deploy')  
14     gslb.add_ip_to_domain(domain='abc.com', ip=ip)
```

# 阻塞问题

```
result = call_func()
```

result 无法立即获得, 所谓“阻塞”

# I/O 阻塞

```
bytes = tcp_socket.recv(4096)
```

网络I/O操作需要时间, 所谓“I/O 阻塞”

# 例子



# 回调

```
def on_received(bytes):
```

```
    .....
```

```
def on_sent(sock):
```

```
    sock.recv(on_received)
```

```
def on_connect(sock):
```

```
    sock.send('xxx', on_sent)
```

```
connect('a.b.c.d', on_connect)
```

上下文

# CALLBACK HELL

```
Selection Find View Tools Window Callback Hell.js  
callback-hell.js  
1 var amount=req.param("amount");  
2 db.select("++ from sessions where session_id=?", req.param("session_id"), function(err, sessions) {  
3   if (err) throw err;  
4   db.select("++ from accounts where user_id=?", sessions[0].user_id, function(err, accounts) {  
5     if (err) throw err;  
6     if (accounts[0].balance < amount) throw new Error("insufficient funds");  
7     db.execute("withdrawal(?, ?)", accounts[0].id, req.param("amount"), function(err, res) {  
8       if (err) throw err;  
9       res.write("withdrawal OK, amount: "+ req.param("amount"));  
10      db.select("balance from accounts where account_id=?", accounts[0].id, function(err, balance) {  
11        if (err) throw err;  
12        res.end("your current balance is " + balance[0].balance);  
13      }  
14    }  
15  }  
16 }  
17 }
```





# Logic Locality

最小化开发人员阅读流程代码的眼球移动距离



# 协程可以表达I/O阻塞

```
def some_process():  
    sock = connect('a.b.c.d', on_connect)  
    sock.send('xxx')  
    bytes = sock.recv(on_received)  
    ....
```

locals()

上下文

# 流程阻塞

```
ip = res.request_new_server(model='B6')
```

调用“外部系统”需要时间，所谓“流程阻塞”

# 流程阻塞

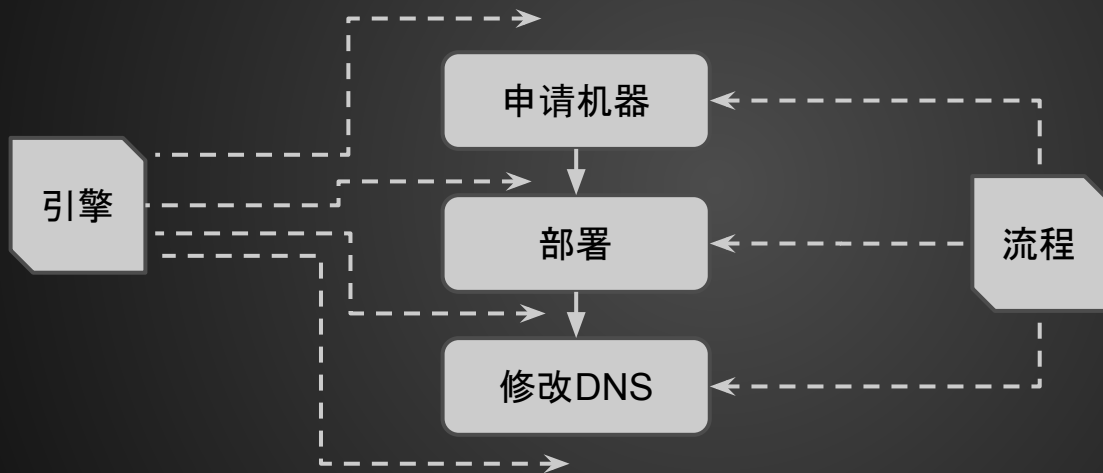
```
yes_no = smcs.wechat_approve('reboot?')
```

人工审批需要时间，所谓“流程阻塞”

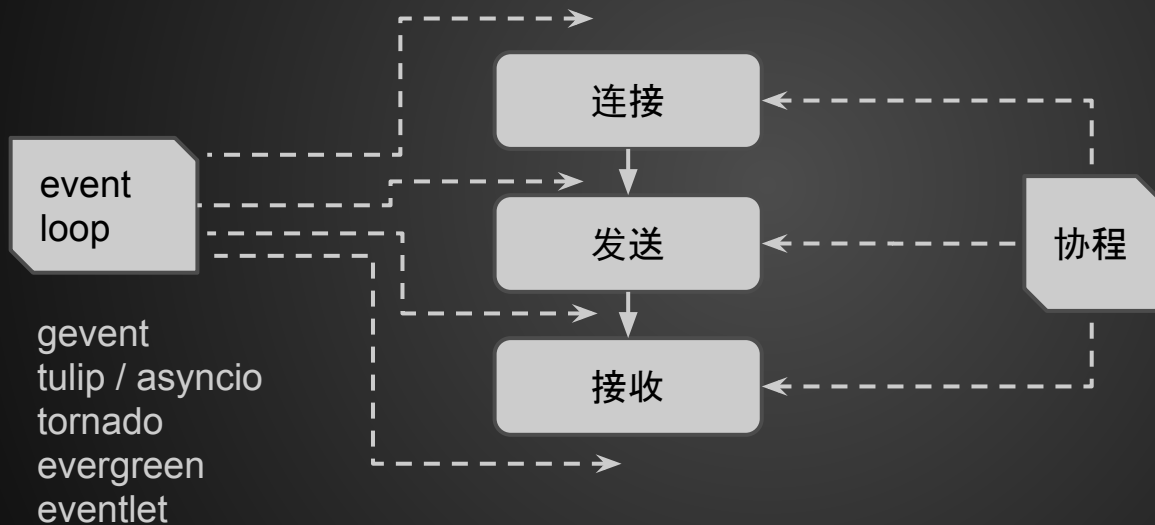
# 协程也可以表达流程阻塞

```
def some_process():  
    ip = res.request_new_server(model='B6')  
    yes_no = smcs.wechat_approve('deploy %s?' % ip)  
    if yes_no:  
        ijobs.exec_job('deploy', ip)
```

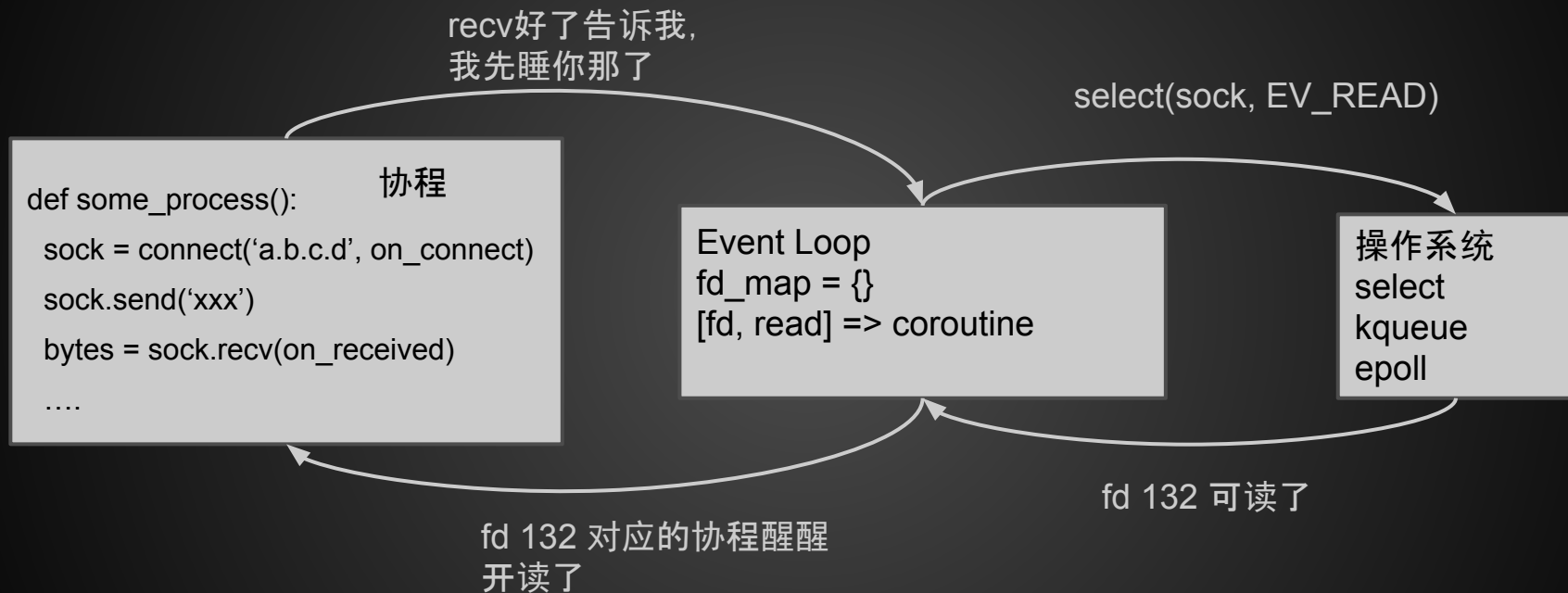
# 流程引擎



# 各式Event Loop

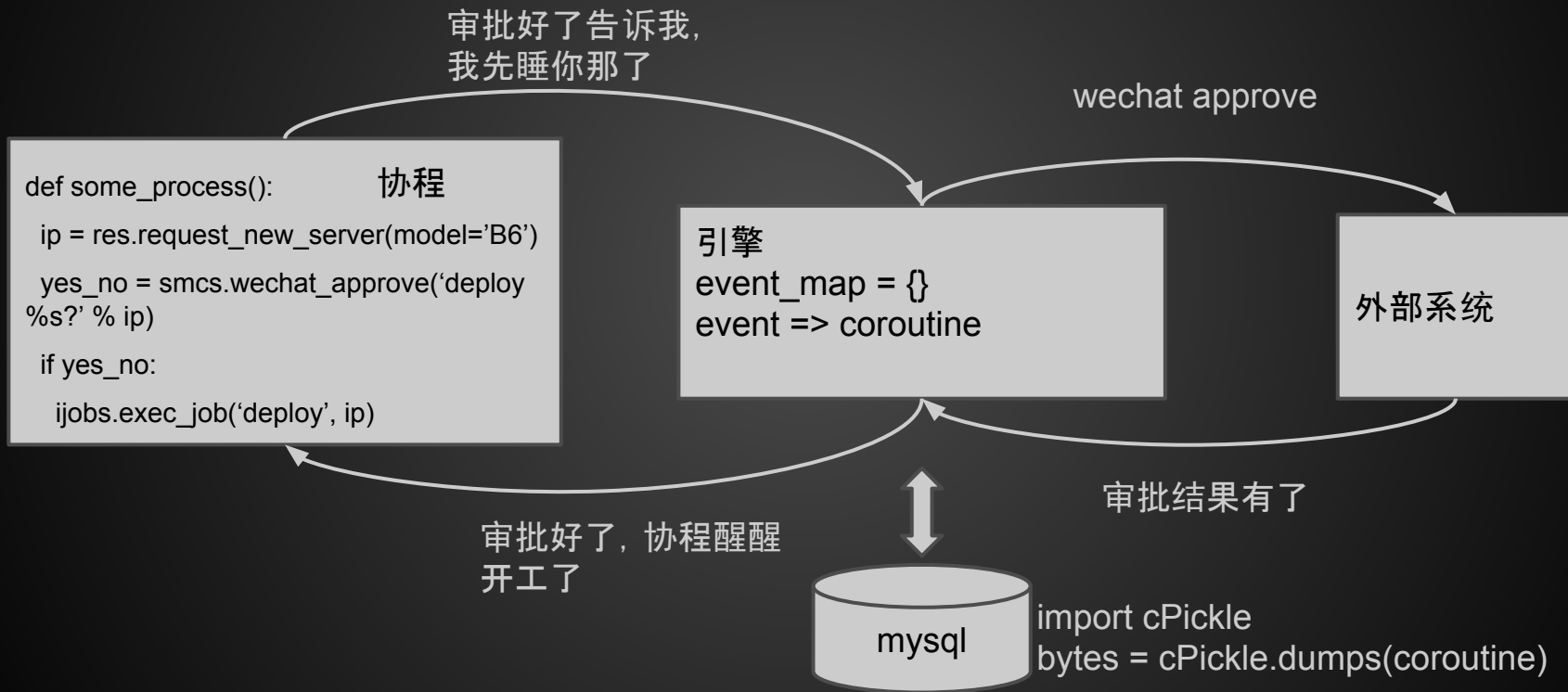


# 所谓Event Loop





# 协程可以存数据库的



# Python协程实现

- yield
- stackless python / pypy
- greenlet / python-fibers / stacklet / libgevent

# Ultimate Coroutine Scheduler

- 语言: Java for true parallelism
- 协程: co-routine with bytecode weaver
- I/O阻塞: 包装NIO
- 流程阻塞: event map, 持久化可选

<https://github.com/taowen/daili>

# 协程大法好



招聘: [blueking.jobs@gmail.com](mailto:blueking.jobs@gmail.com)