Python 与对话机器人

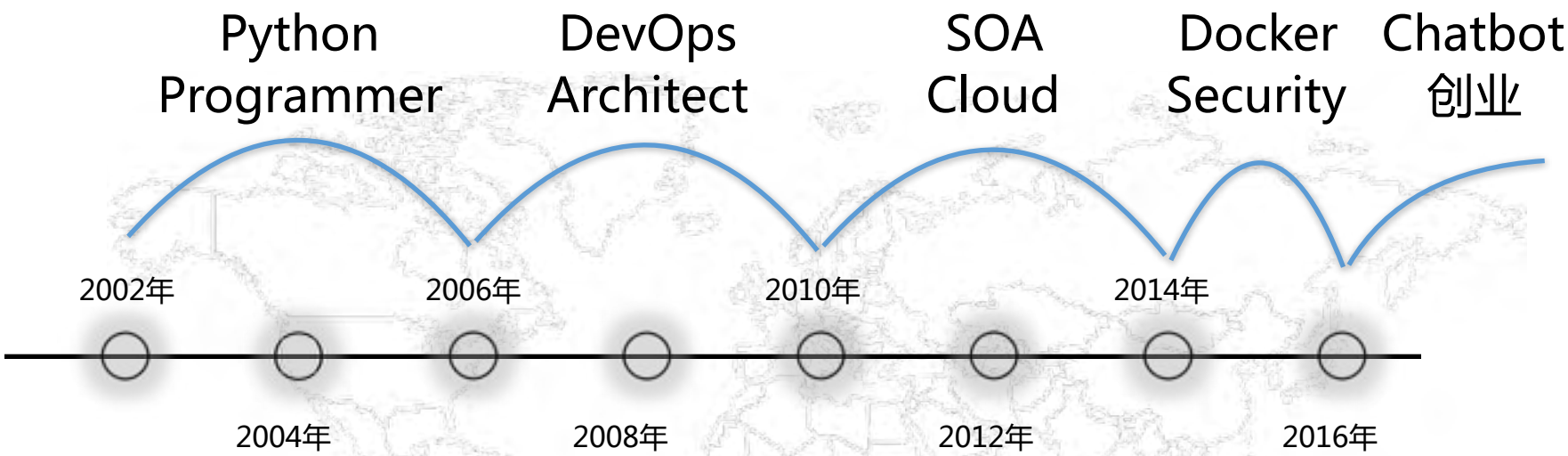
洪强宁 @ 爱因互动

自我介绍

洪强宁
@hongqn

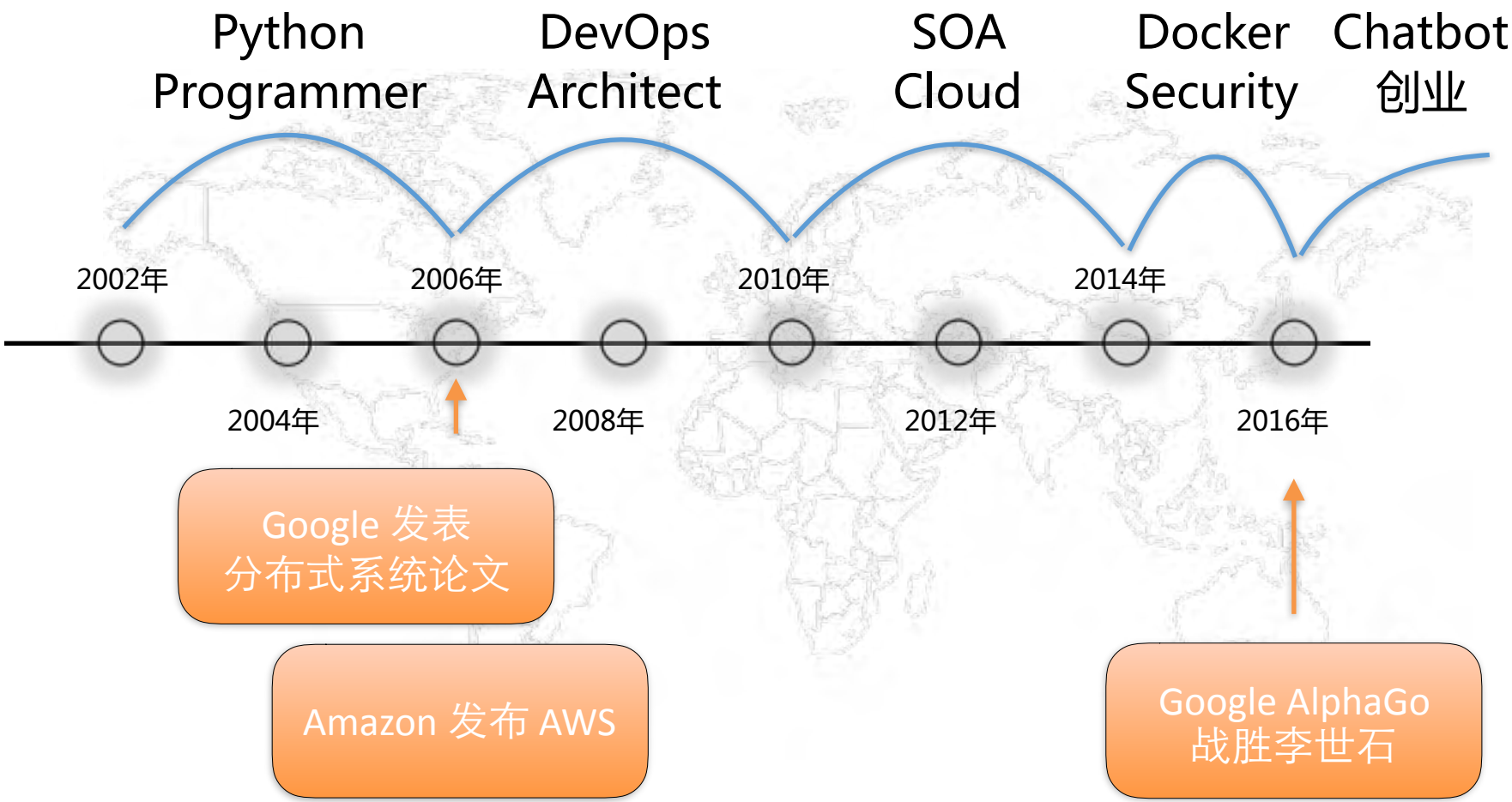
1995	清华大学 Tsinghua University	清华大学精密仪器系本硕
2002	python	国内Python早期用户，CPUG创立者之一
2006	豆瓣 douban	豆瓣首名全职员工，首席架构师
2014	宜信 CreditEase	宜信大数据创新中心首席架构师
2016	EIN+	爱因互动联合创始人兼CTO

技术历程


Python
Programmer
DevOps
Architect
SOA
Cloud
Docker
Security
Chatbot
创业
2002年
2004年
2006年
2008年
2010年
2012年
2014年
2016年

Why Chatbot?

十年一革命

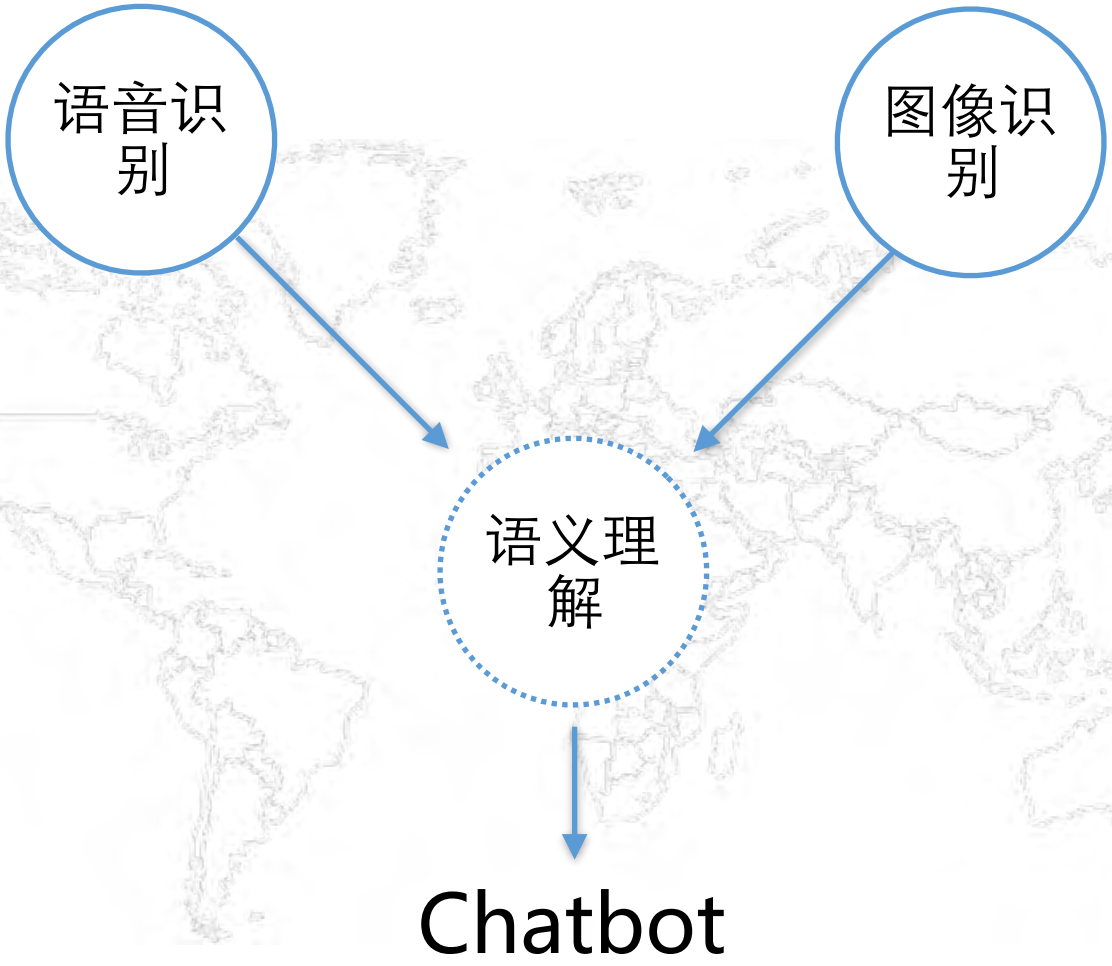


- Late 1980s: Internet 商用
- 1994: Mosaic web browser

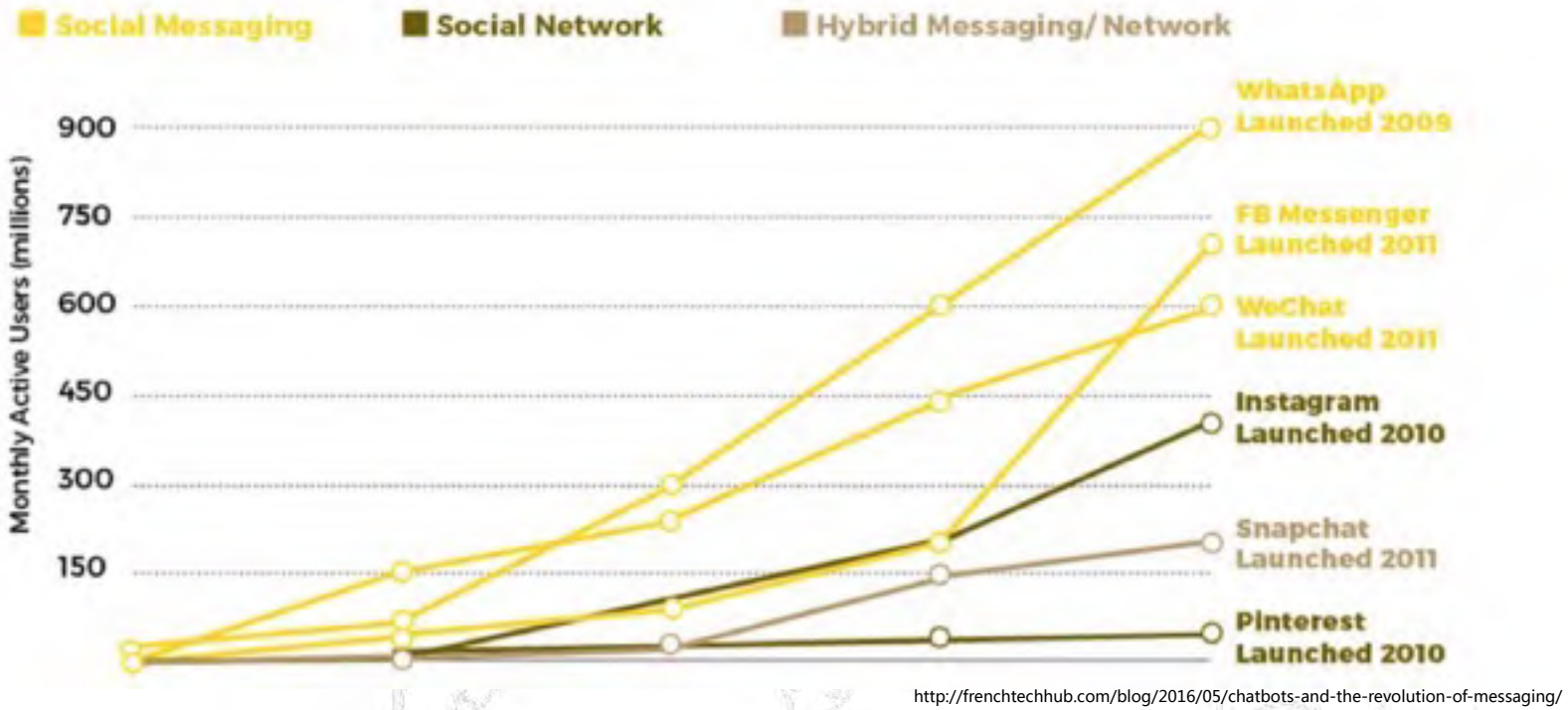
AI 是新时代的云计算

- 技术进步带来生产力的提升
- 商业边界的拓宽
- 开发者必须了解的技能

AI的应用领域



消息平台的增长



http://frenchtechhub.com/blog/2016/05/chatbots-and-the-revolution-of-messaging/

消息已经成为最大的应用

业界最近的一些事件

- *2016.3* **Microsoft** Bot Framework
- *2016.4* **Facebook** Messager Platform
- *2016.4* **Telegram** Bot Platform 2.0
- *2016.5* **Google** Assistant
- *2016.5* **Google** open sourced SyntaxNet
- *2016.6* **Apple** SiriKit and iMessage
- *2016.6* **Slack** Message Buttons
- *2016.8* **Amazon** Serverless Chatbot Hackathon
- *2016.8* **Facebook** open sourced FastText
- *2016.9* **Facebook**, **Amazon**, **Google**, **IBM** and **Microsoft** create Partnership on AI

巨头： 搭平台，提供基础设施

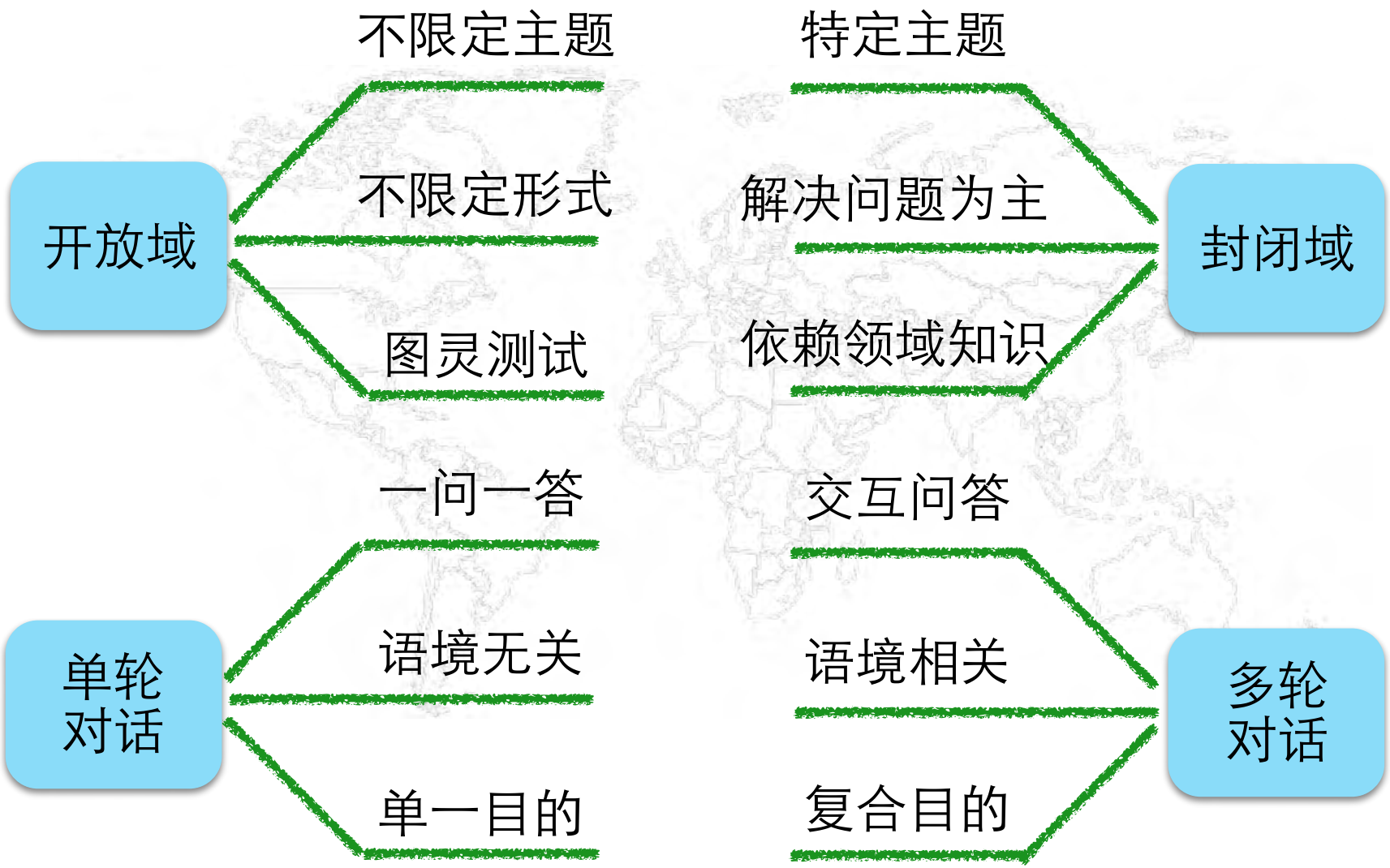
创业公司： 深入应用场景，创造价值

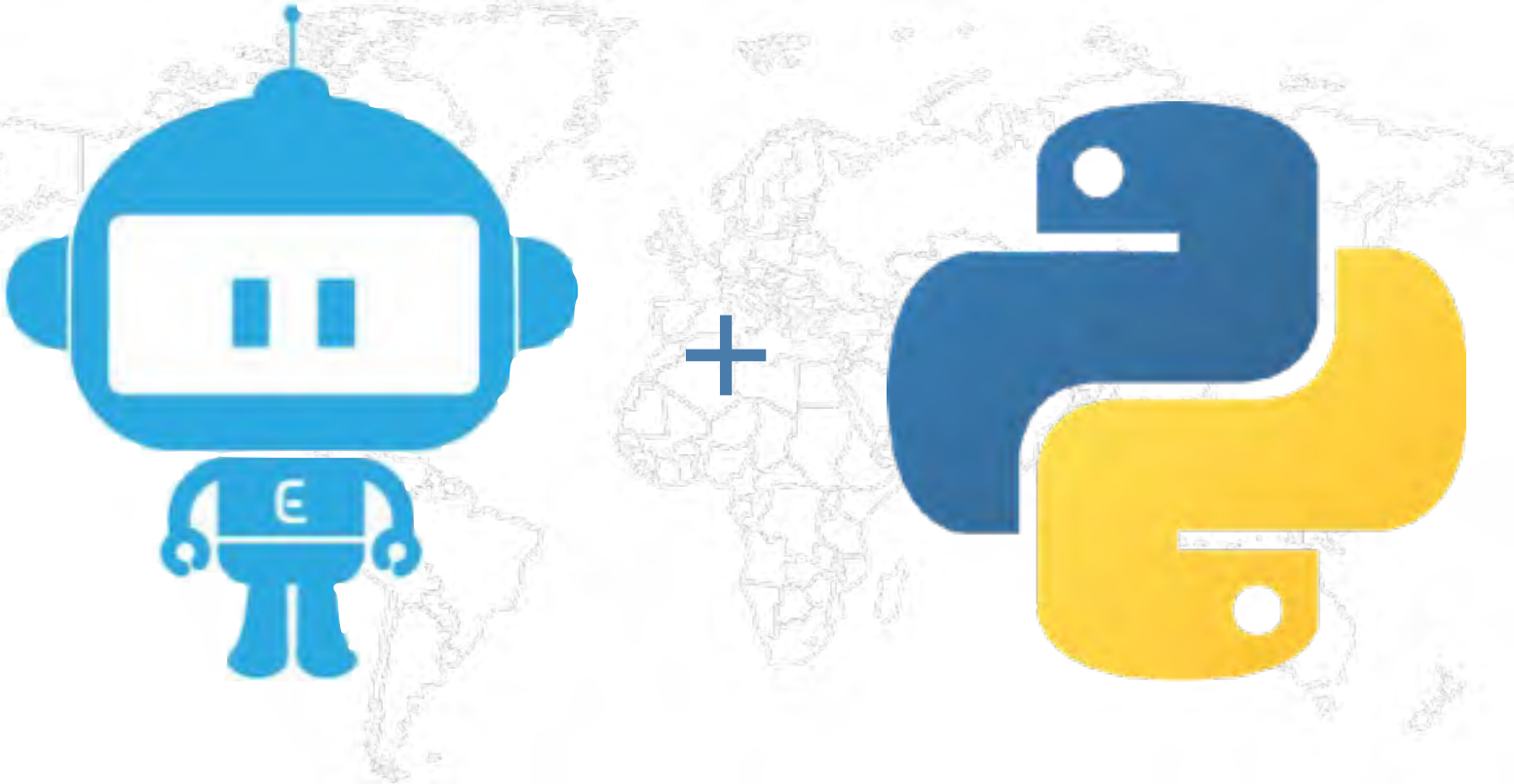
对话机器人


中国联通 下午12:28
微信(7)
小冰
你都能做什么啊
我会七十二变，你信么？
不信
不信是么..
是啊
什么是吗是吗是吗是吗？
不信你会七十二变
越变越挫啊!怎么破!
多学点聊天语料啊
口语？聊天还是论坛。


中国联通 下午6:18
微信(10)
爱因互动
请选择借款期限 1. 6个月 2. 9个月 3. 12个月
1
请输入信用卡号码，或者给您的信用卡拍照传给我也可以
12355
这张卡曾有过逾期记录吗 1. 有 2. 没有
没有
您从事什么职业 1. 上班族 2. 私营业主 3. 其他
其他

Chatbot 分类


开放域
不限定主题
不限定形式
图灵测试
封闭域
特定主题
解决问题为主
依赖领域知识
单轮对话
一问一答
语境无关
单一目的
多轮对话
交互问答
语境相关
复合目的



对话机器人是Python大显身手的机会

- Python 字符串处理功能强大
- Python 擅长服务端开发
- Python 擅长科学计算
- Python 拥有丰富的 NLP 算法库
- 大部分深度学习框架都支持 Python
- Python 有优秀的大数据处理能力
- 在 DevOps 领域 Python 是王者

- 做个能解数学题的对话机器人

```
0:54:38 [hongqn:~] 1m39s % python3
Python 3.5.2 (default, Oct  7 2016, 00:20:34)
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.38)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 3 * 14
42
```

对话机器人程序

```
In [1]: def bot():
   ...:     while True:
   ...:         usermsg = input("请输入算式: ")
   ...:         try:
   ...:             # WARNING: Do not use eval() on user input in production!
   ...:             response = eval(usermsg)
   ...:         except Exception:
   ...:             response = "抱歉，不会算🤔…换一个算式？"
   ...:         print(response)
   ...:

In [2]: bot()
请输入算式: 3 * 14
42
请输入算式: Answer to the Ultimate Question of Life, The Universe, and Everything
抱歉，不会算🤔…换一个算式？
请输入算式:
```

接入消息平台 - 长连接式

Slack RTM API

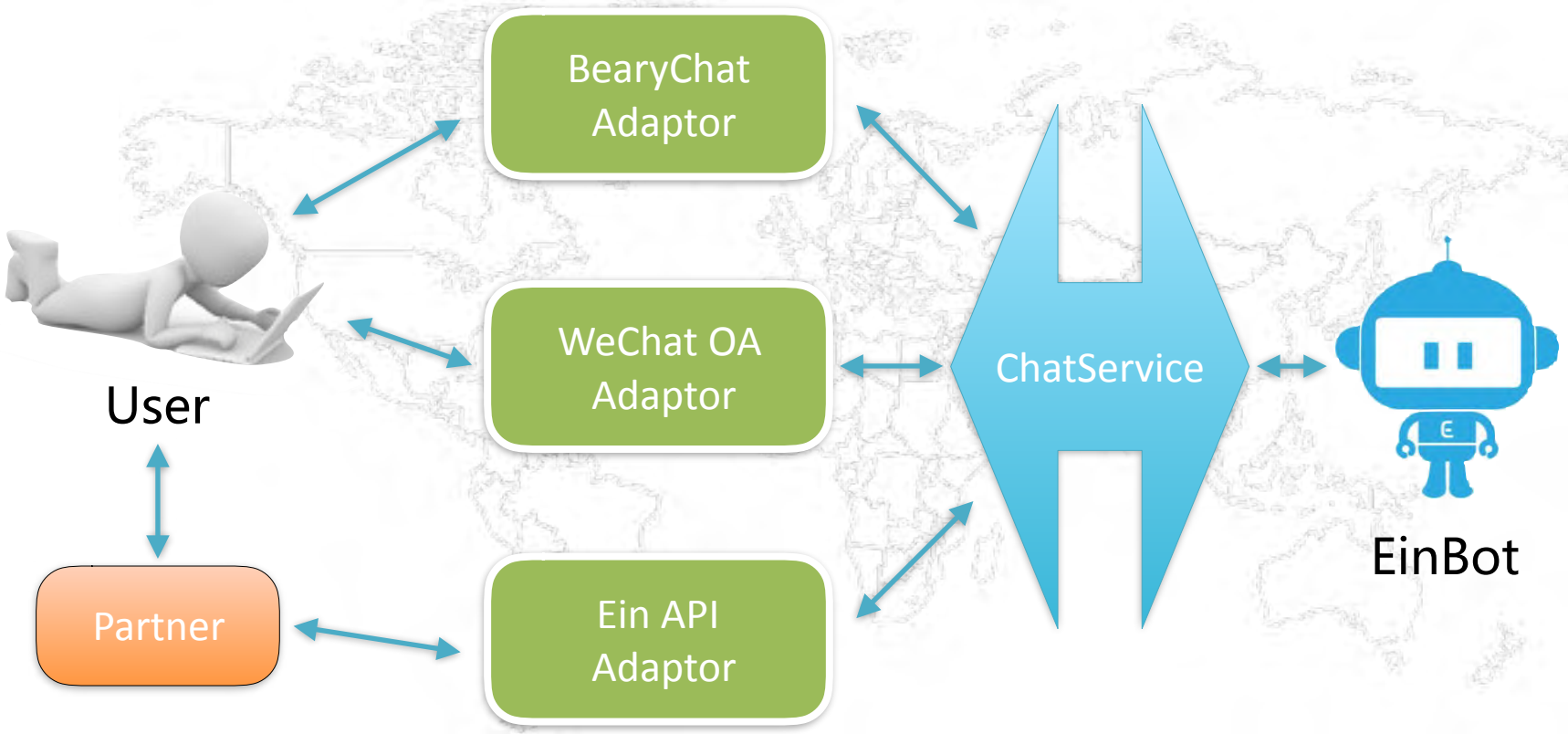
```
slack_client = SlackClient(token)
slack_client.rtm_connect()

while True:
    events = slack_client.rtm_read()
    if not events:
        time.sleep(1)
        continue

    for event in events:
        if event['type'] == 'message':
            usermsg = event['text']
            response = eval(usermsg)
            slack_client.rtm_send_message(event['channel'], response)
```

- 接入消息平台 - 回调式

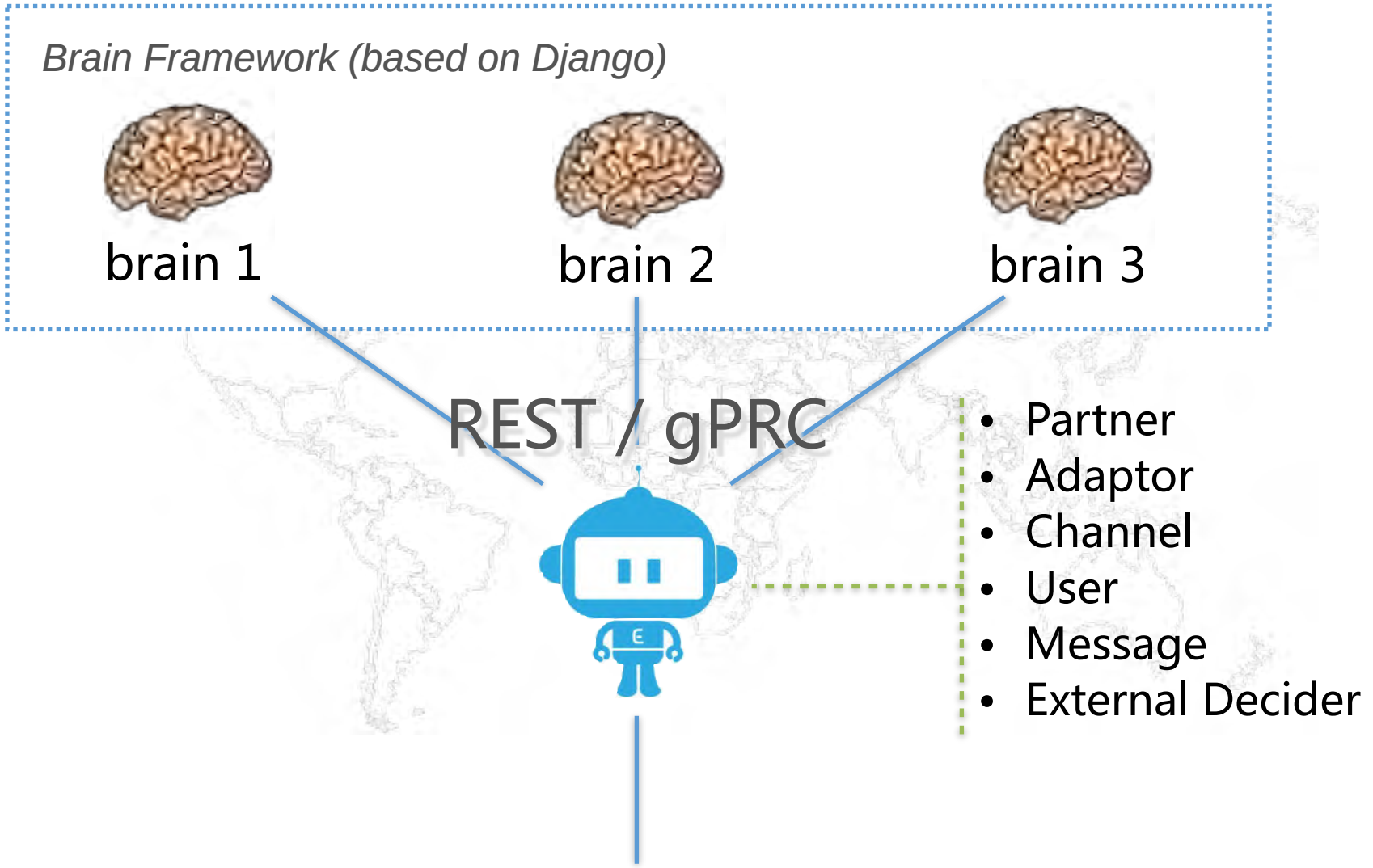
微信公众号

```
def wechat_callback(request):
    msg = parse_wechat_message_xml(request.body)
    if msg['type'] == 'text':
        response = str(eval(msg['data']))
        response_xml = make_wechat_response_xml(response)
        return HttpResponse(response_xml)
```

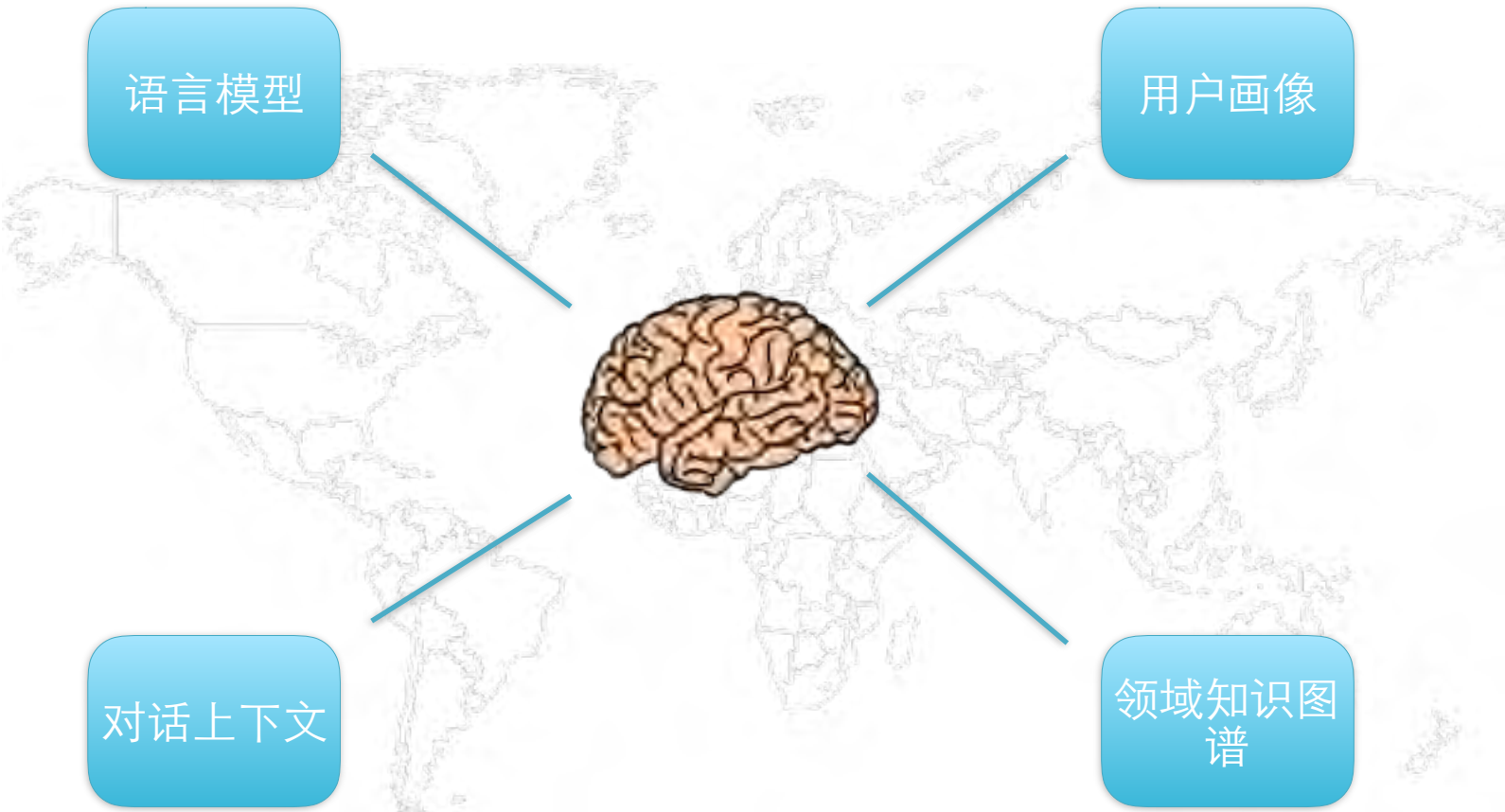
多通道机器人


BearyChat
Adaptor
WeChat OA
Adaptor
Ein API
Adaptor
ChatService
User
Partner
EinBot

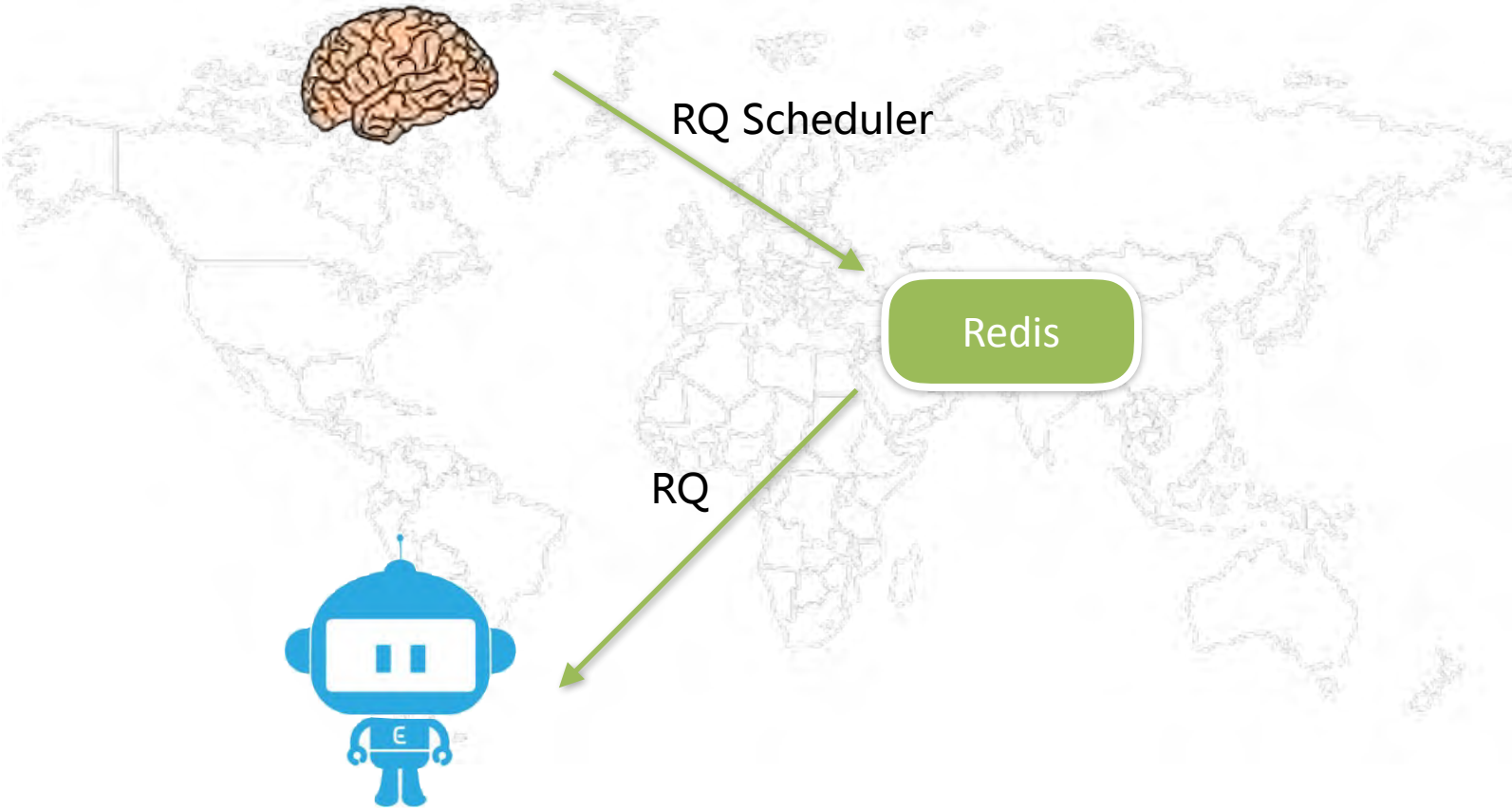
Multiple Brains



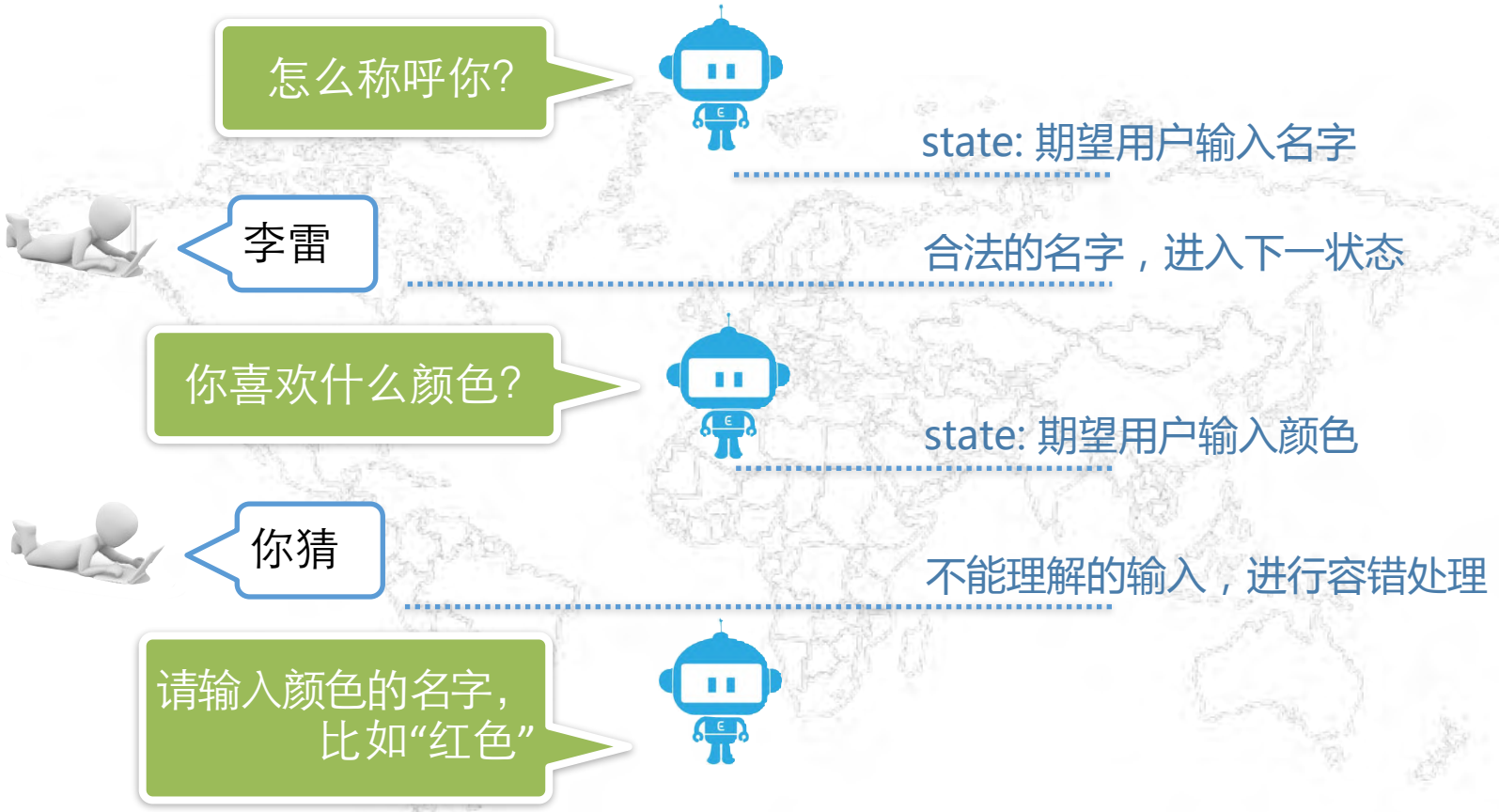
- brain 的“外存”



延时消息

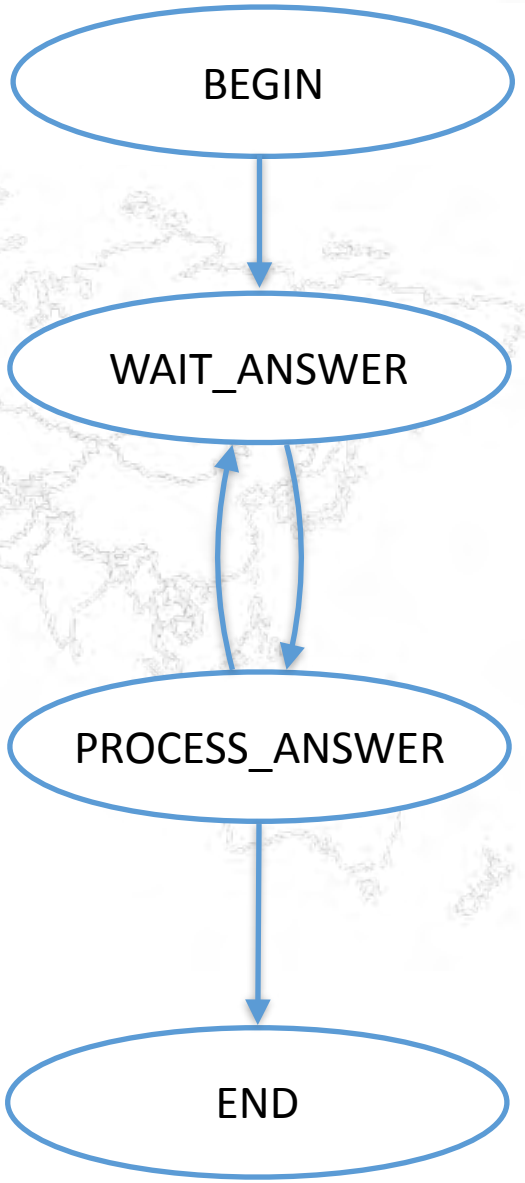


状态切换


怎么称呼你?
state: 期望用户输入名字
李雷
合法的名字，进入下一状态
你喜欢什么颜色?
state: 期望用户输入颜色
你猜
不能理解的输入，进行容错处理
请输入颜色的名字，
比如“红色”

用状态机实现状态切换

```
def __init__(self):
    self.questions = [NameQuestion(), ColorQuestion()]
    self.q_idx = 0
    self.state = STATE_BEGIN

def run(self):
    while self.state != STATE_END
        action, msg = self.process()
        if action == ACTION_WRITE:
            self.write(arg)
        elif action == ACTION_READ:
            self.usermsg = self.read()

def process(self):
    if self.state == STATE_BEGIN:
        question = self.questions[self.q_idx]
        self.state = STATE_WAIT_ANSWER
        return ACTION_WRITE, question.question

    elif self.state == STATE_WAIT_ANSWER:
        self.state = STATE_PROCESS_ANSWER
        return ACTION_READ, None

    elif self.state == STATE_PROCESS_ANSWER:
        question = self.questions[self.q_idx]
        valid = question.process_answer(self.usermsg)
        if not valid:
            self.state = STATE_WAIT_ANSWER
            return ACTION_WRITE, question.instruction

        self.q_idx += 1
        if self.q_idx >= len(self.questions):
            self.state = STATE_END
            return None, None
        question = self.questions[self.q_idx]
        self.state = STATE_WAIT_ANSWER
        return ACTION_WRITE, question.question
```



用 coroutine 实现状态切换

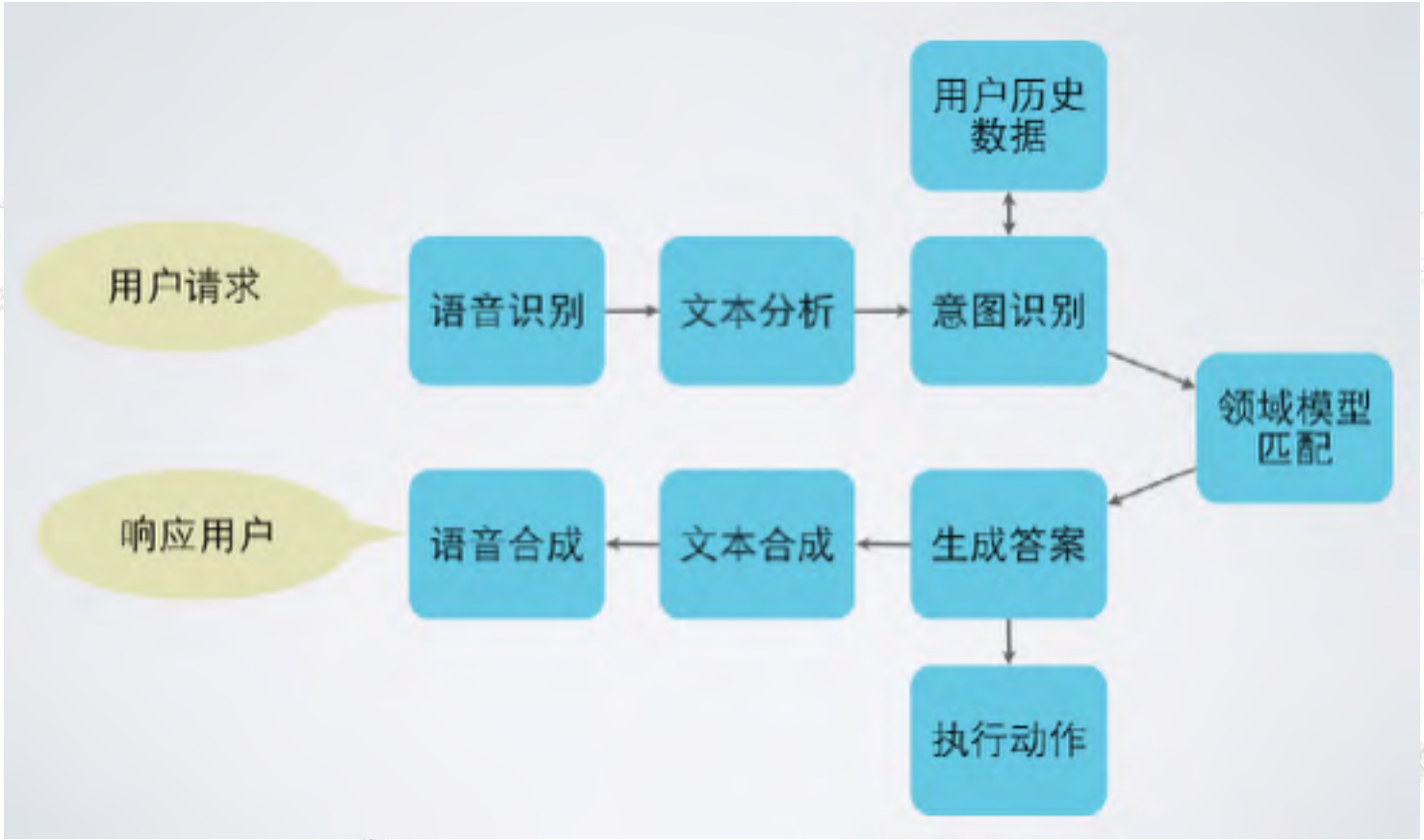
```
def run(self):
    co = self.process()
    usermsg = None
    while True:
        try:
            msg = co.send(usermsg)
        except StopIteration:
            break
        self.write(msg)
        usermsg = self.read()


def process(self):
    for q in self.questions:
        usermsg = yield q.question
        while True:
            valid = q.process_answer(usermsg)
            if valid:
                break
            else:
                usermsg = yield q.instruction
```

状态持久化？

- Stackless Python 3.4 可以 pickle coroutine 对象
- 使用 Docker 简化 Stackless 部署

```
FROM registry.cn-hangzhou.aliyuncs.com/einplus/centos-base:stackless-3.4.1
```

系统框架



- Chatbot 流派

```
<category>
    <pattern>谢 谢 </pattern>
    <template>
        <random>
            <li>不客气 .</li>
            <li>你太客气了 .</li>
        </random>
    </template>
</category>
```

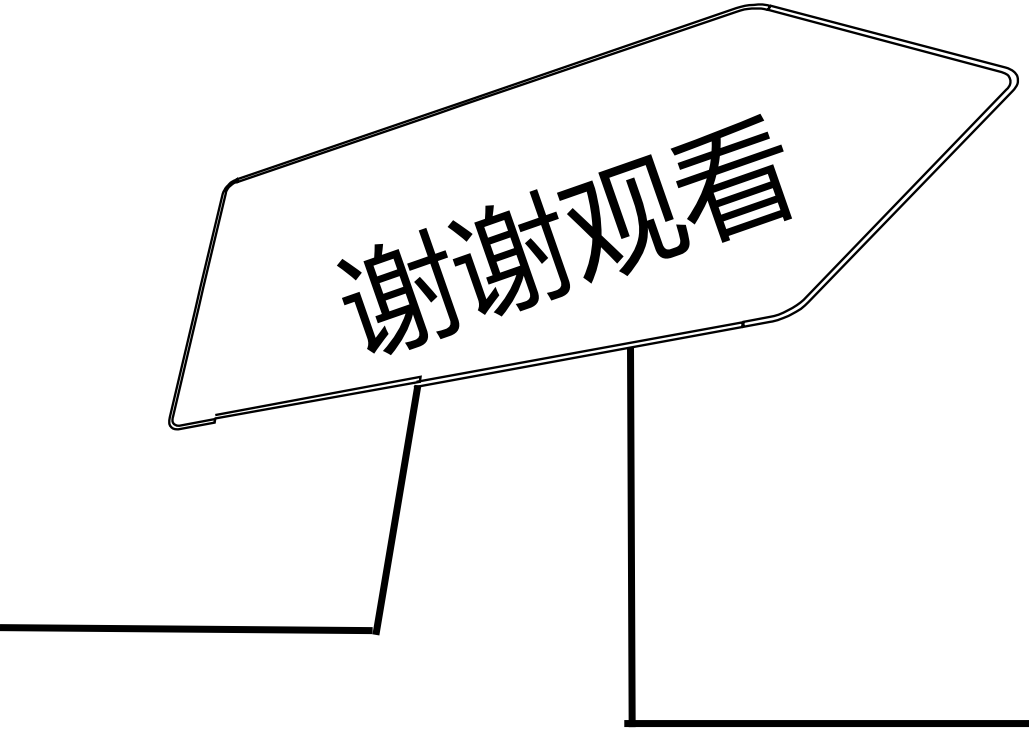
- 基于规则
- 基于搜索
- 基于统计
- end-to-end 生成模型
- 混合模型

爱因用到的一些Python工具

- NLP相关
  - NLTK
  - gensim
  - sklearn
  - jieba
  - fasttext
  - TensorFlow
  - numpy/pandas
- 爬虫
  - scrapy
- 服务
  - Django
  - django-rq
  - gunicorn
- 集群管理
  - Lain
  - Sentry
  - Graphite
  - Logster
  - Ansible

一个激动人心的新时代正在来临

Pythonista 们, 准备好了吗？



- 微信号：hongqiangning
- 爱因互动官网：einplus.cn
- 简历请投：jobs@ein.plus