

# App Engine

Cloud Computing платформа от Google



Петр Чардин, Михаил Дайчик  
Апрель 2009



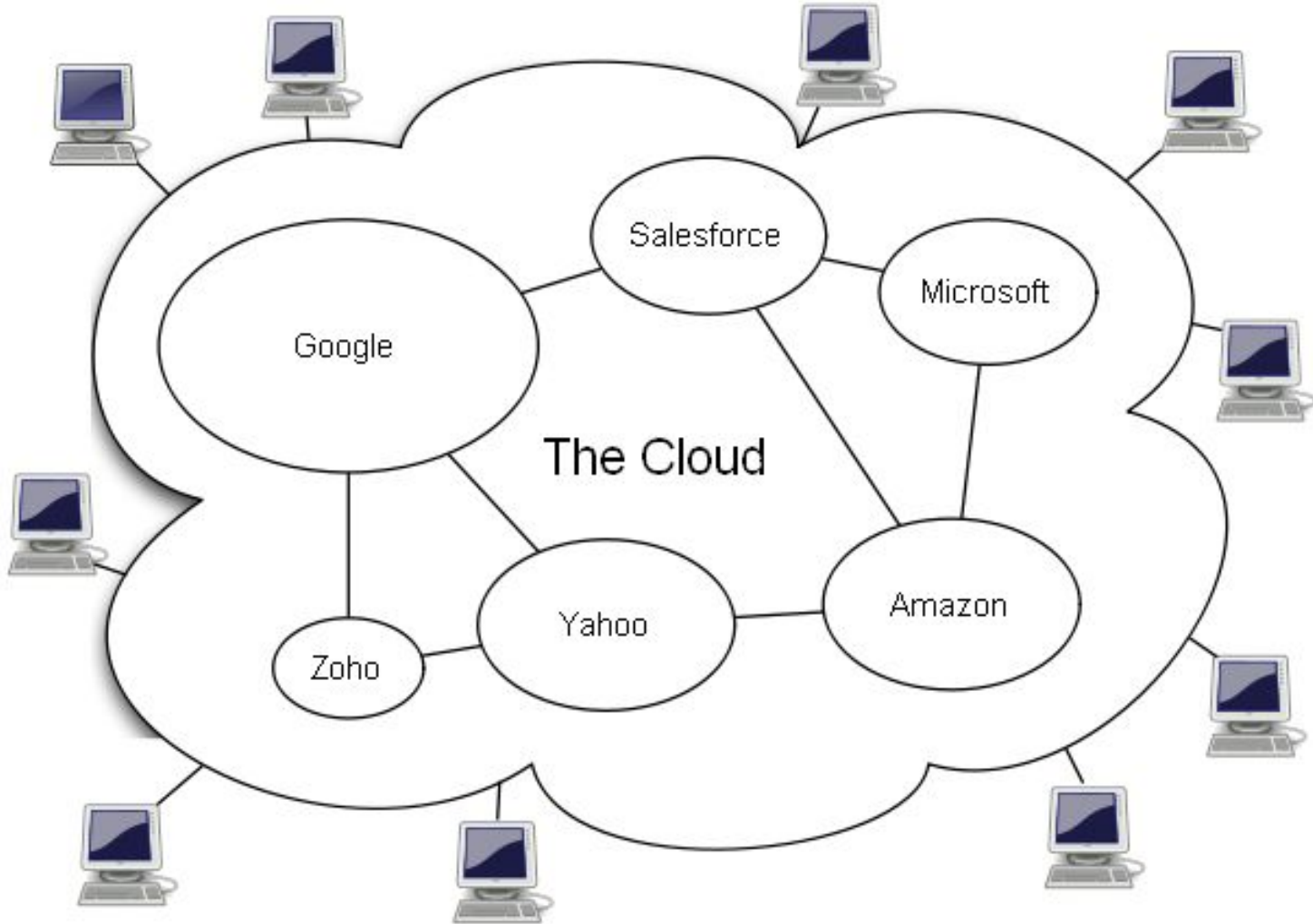
# Cloud Computing

---

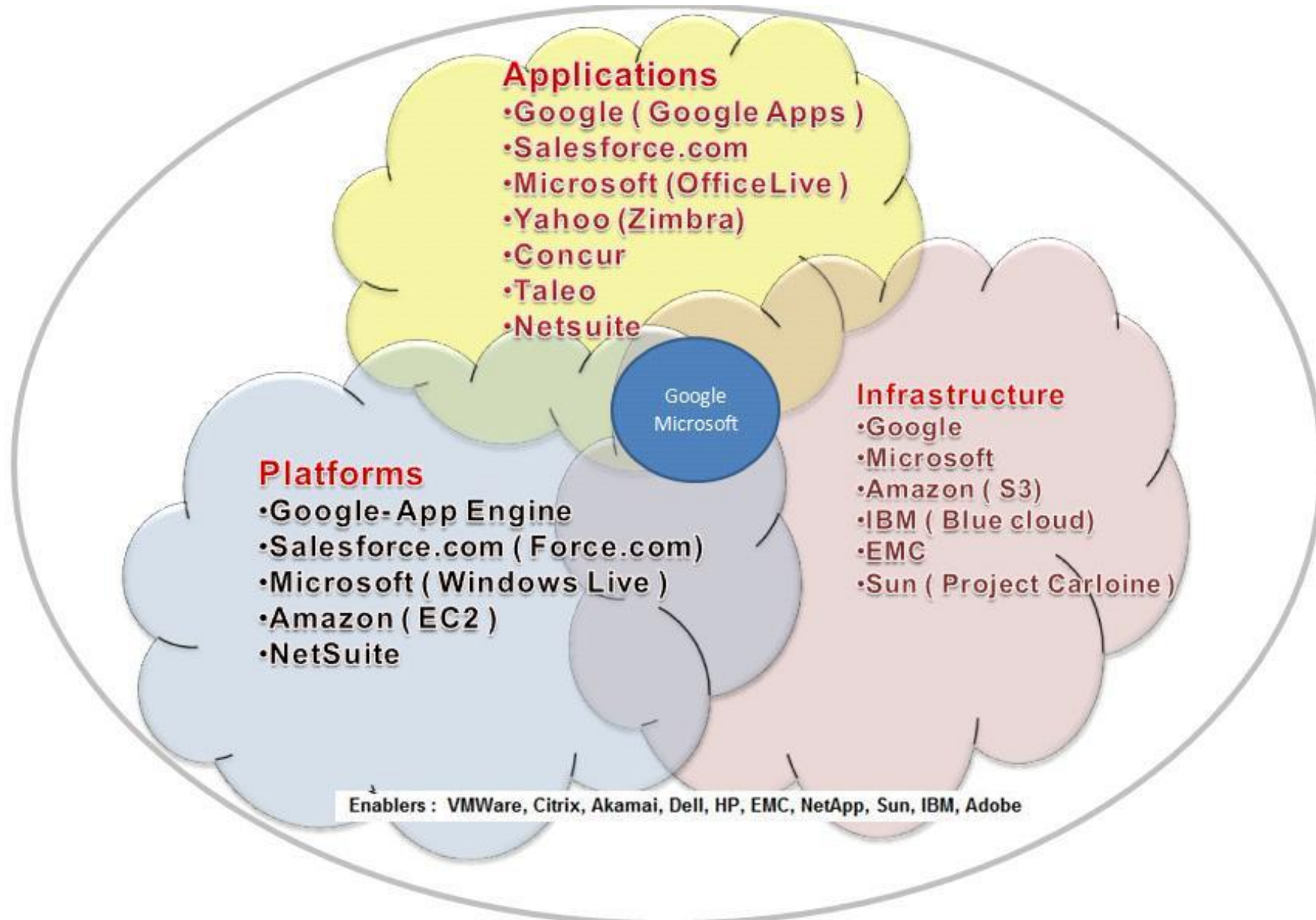
Cloud computing – технологический тренд, суть которого состоит в переносе программного обеспечения для личного и корпоративного пользования в интернет. Он объединяет такие концепции как software as a service, infrastructure as a service и platform as a service.



# Cloud Computing



# Cloud Computing



# Мотивация

- Создавать масштабируемые веб-приложения тяжело. Особенно для небольших групп разработчиков.
  - Типичный LAMP-стек требует нетривиальной конфигурации и последующего администрирования.
  - LAMP-приложения требуют дополнительной инфраструктуры для решения таких проблем как балансировка, репликация данных и мониторинг.
  - Такие приложения тяжело масштабировать.Нередко приходится менять архитектуру приложения с ростом нагрузки.
- Все эти проблемы решаются снова и снова.



# Мотивация

- У Google тоже имеется богатый опыт создания масштабируемых приложений. Наши приложения используют миллионы пользователей по всему миру.
- Мы потратили много сил для создания удобной и надежной инфраструктуры для разработки веб-приложений.
- Google App Engine это платформа которая позволяет использовать нашу инфраструктуру для создания и хостинга своих приложений.



# App Engine

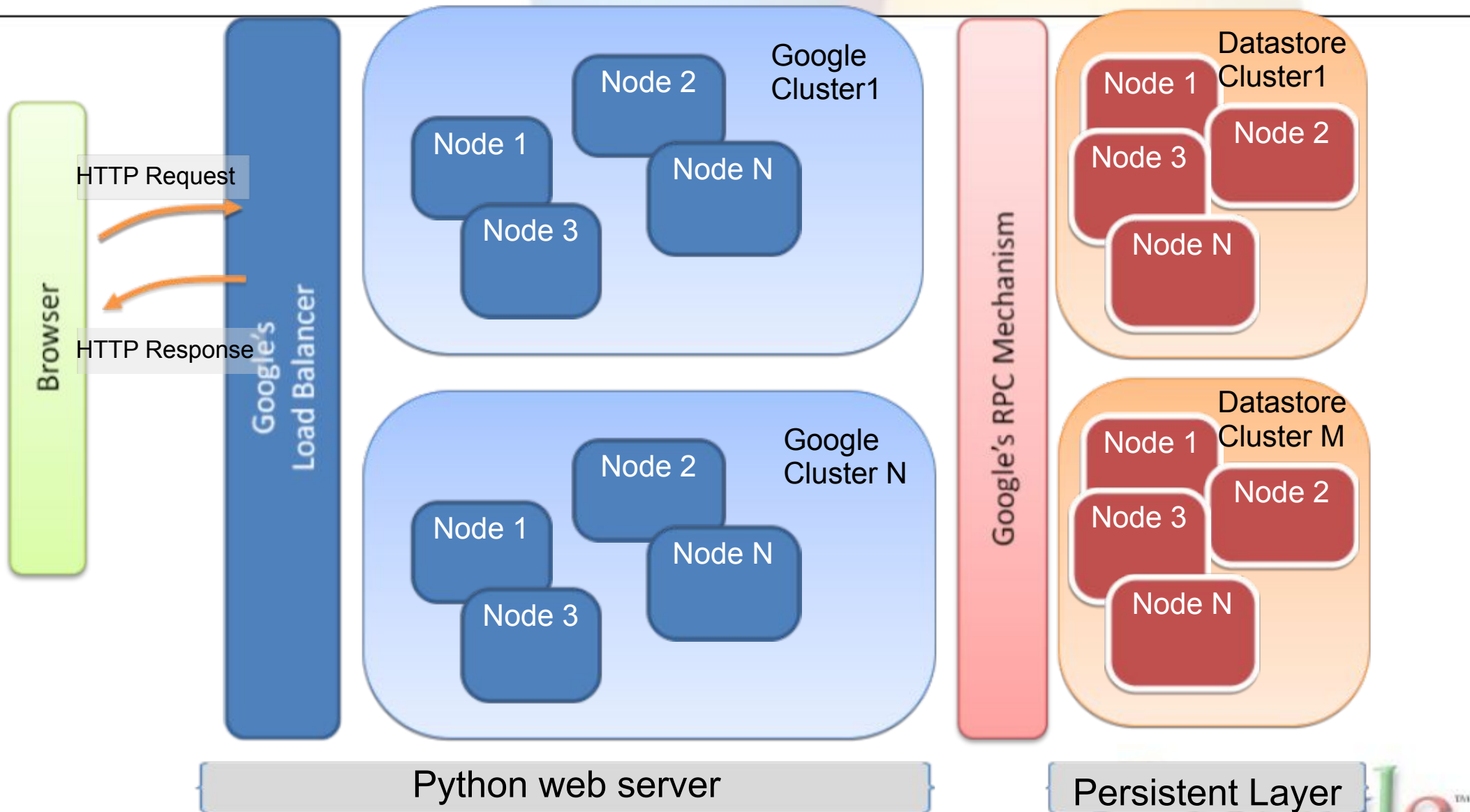
## Cloud Computing платформа от Google

---

- Платформа отвечает за автоматическую масштабируемость и балансировку нагрузки
- Платформа предоставляет динамический веб-сервер с поддержкой многих распространенных веб-технологий
- Хранилище данных на основе Big Table
- Простая интеграция с аккаунтами Google через API.



# Как работает App Engine

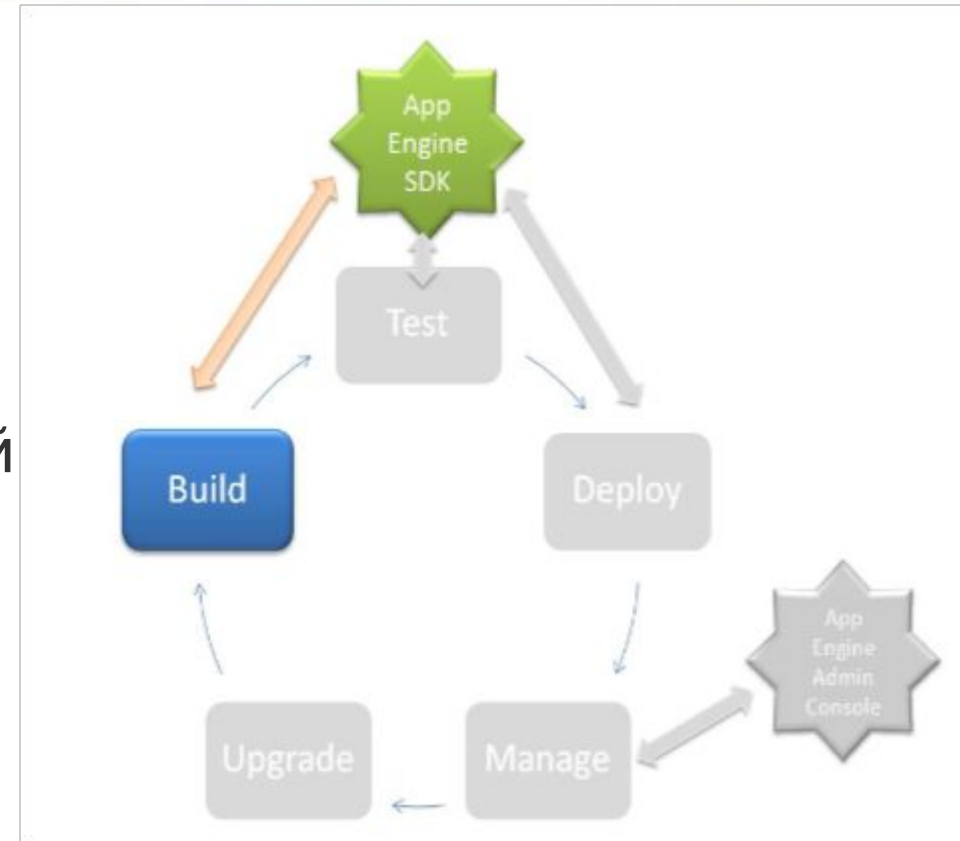




# Разработка приложений с помощью App Engine

## App Engine SDK

- Веб-сервер
  - [dev\\_appserver.py](http://dev_appserver.py)
- Утилита для загрузки приложений
  - [appcfg.py](http://appcfg.py)
- Локальная реализация DataStore
- App Engine APIs





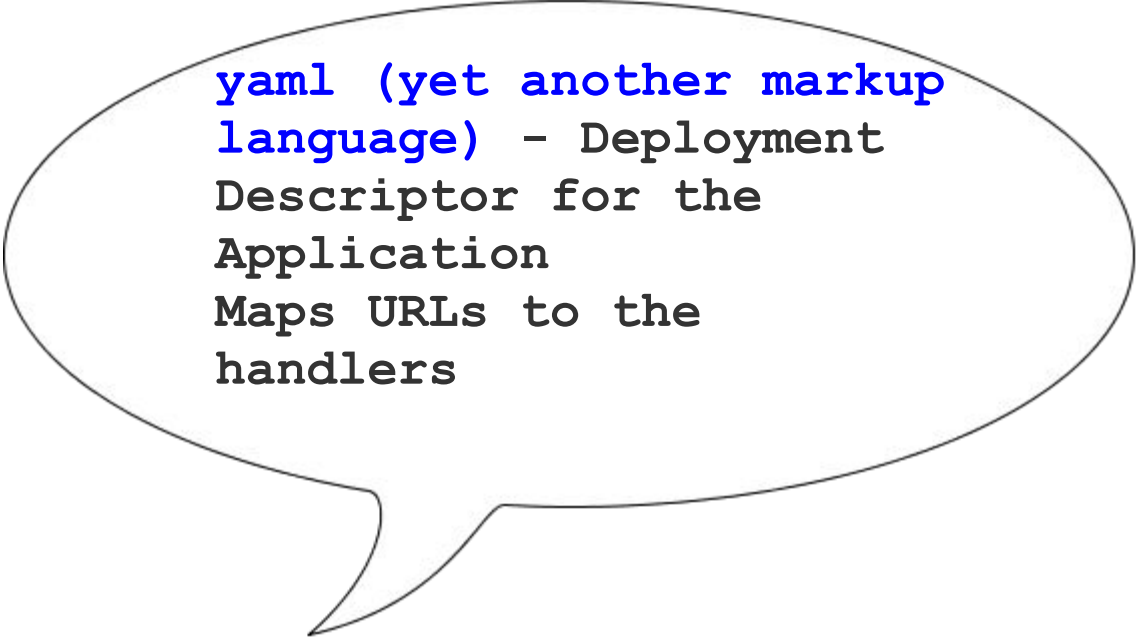
# Разработка приложений с помощью AppEngine



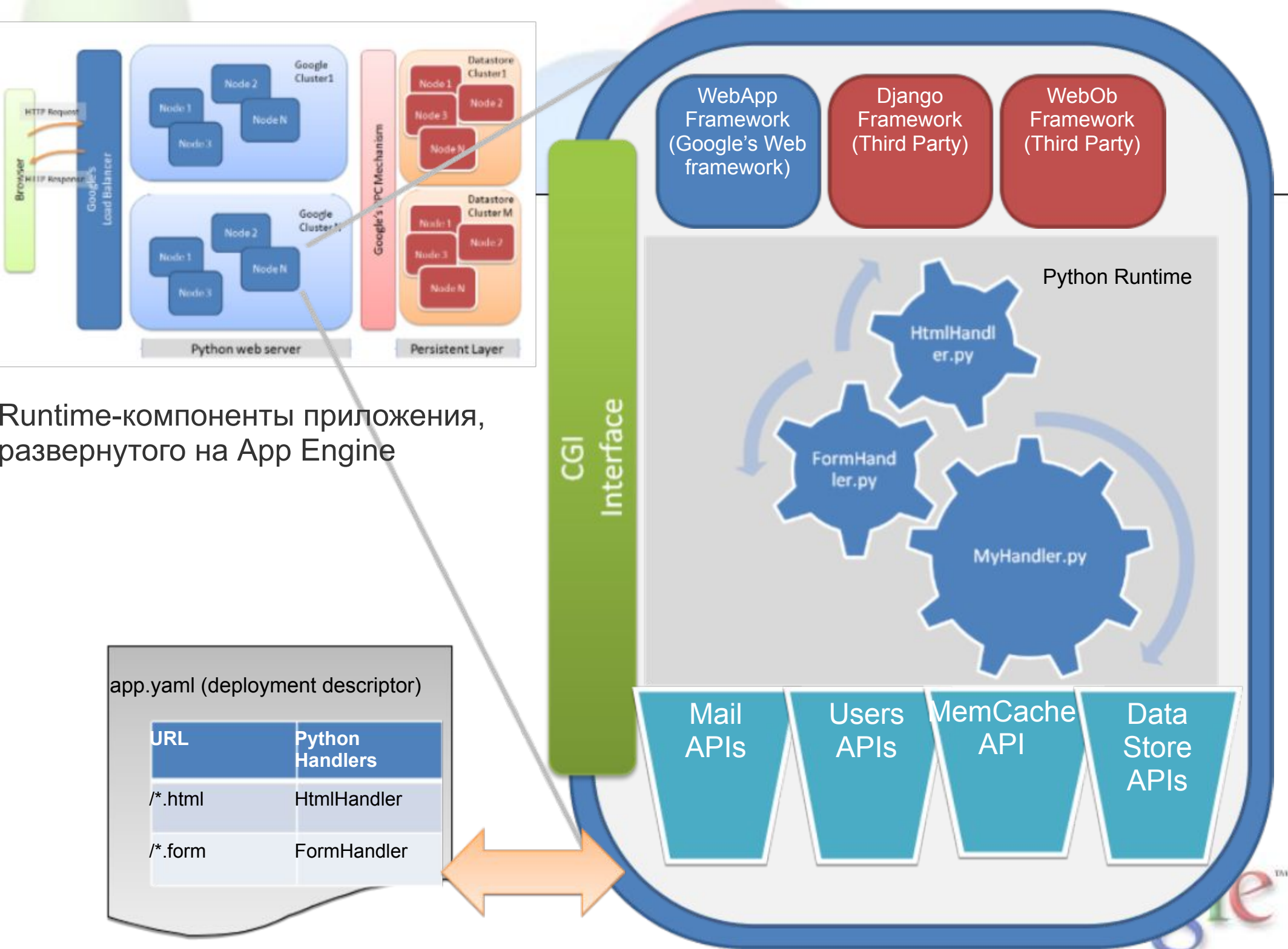
# Конфигурация приложения - app.yaml

```
application: helloworld  
version: 1  
runtime: python  
api_version: 1
```

```
handlers:  
- url: /*  
  script: helloworld.py
```



**yml (yet another markup language)** - Deployment Descriptor for the Application  
Maps URLs to the handlers



# Request Handler - helloworld.py

```
from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app

class MainPage(webapp.RequestHandler):
    def get(self):
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.out.write('Hello, webapp World!')

application = webapp.WSGIApplication([('/', MainPage)],
                                     debug=True)

def main():
    run_wsgi_app(application)

if __name__ == "__main__":
    main()
```

Extends webapp.RequestHandler

WSGI CGI Adapter

# App Engine DataStore

---

- Мощное не реляционное распределенное хранилище данных
- Поддержка SQL-образного языка GQL
- DataStore API



# Объекты и их свойства



- В хранилище данных хранятся сущности (Entities)
- У каждой сущности есть набор свойств, заданного типа (например User, IM, Link, Rating, GeoPt, PhoneNumber и тд)
- Сущности имеют возможность ссылаться на другие сущности (many to one relationships)

# Ключи

- У каждой сущности (Entity) есть ключ, являющийся ее уникальным идентификатором
- Ключ состоит из трех частей (1) типа сущности, (2) пути определяющего его положение относительно предка и (3) уникального имени или числового идентификатора.

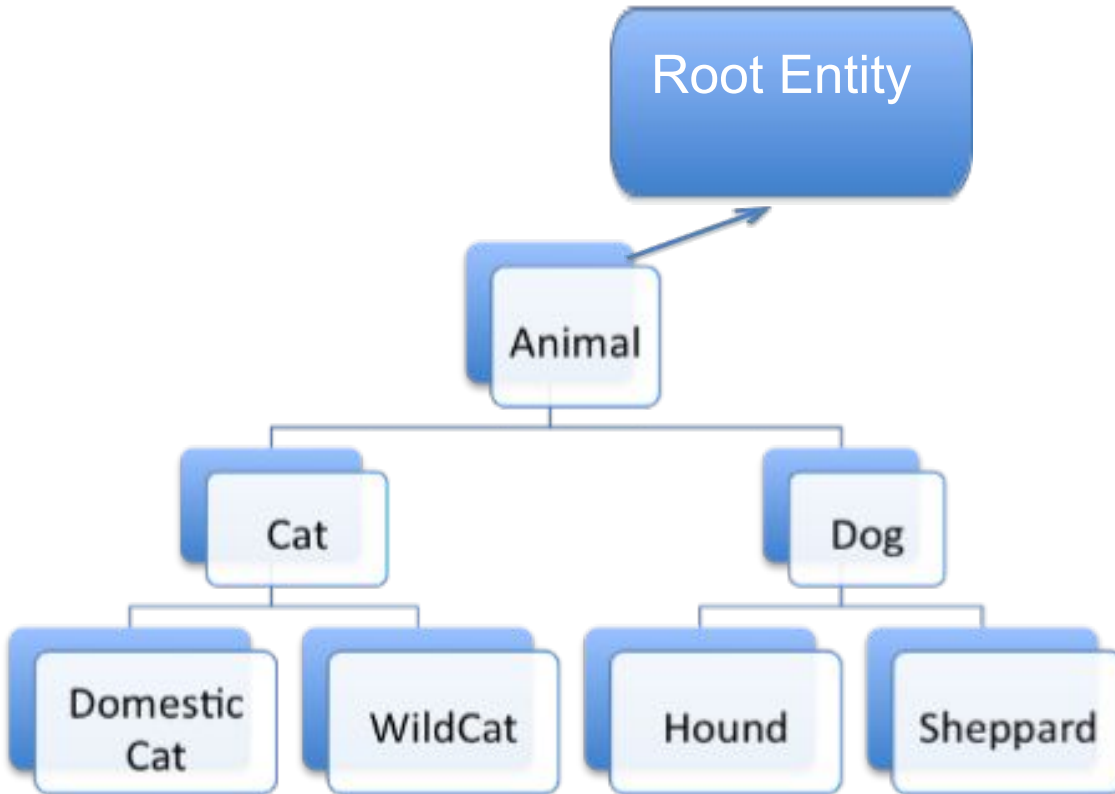
```
class Story(db.Model):  
    title = db.StringProperty()  
    author = db.StringProperty()
```

```
s = Story(key_name="xzy123")
```



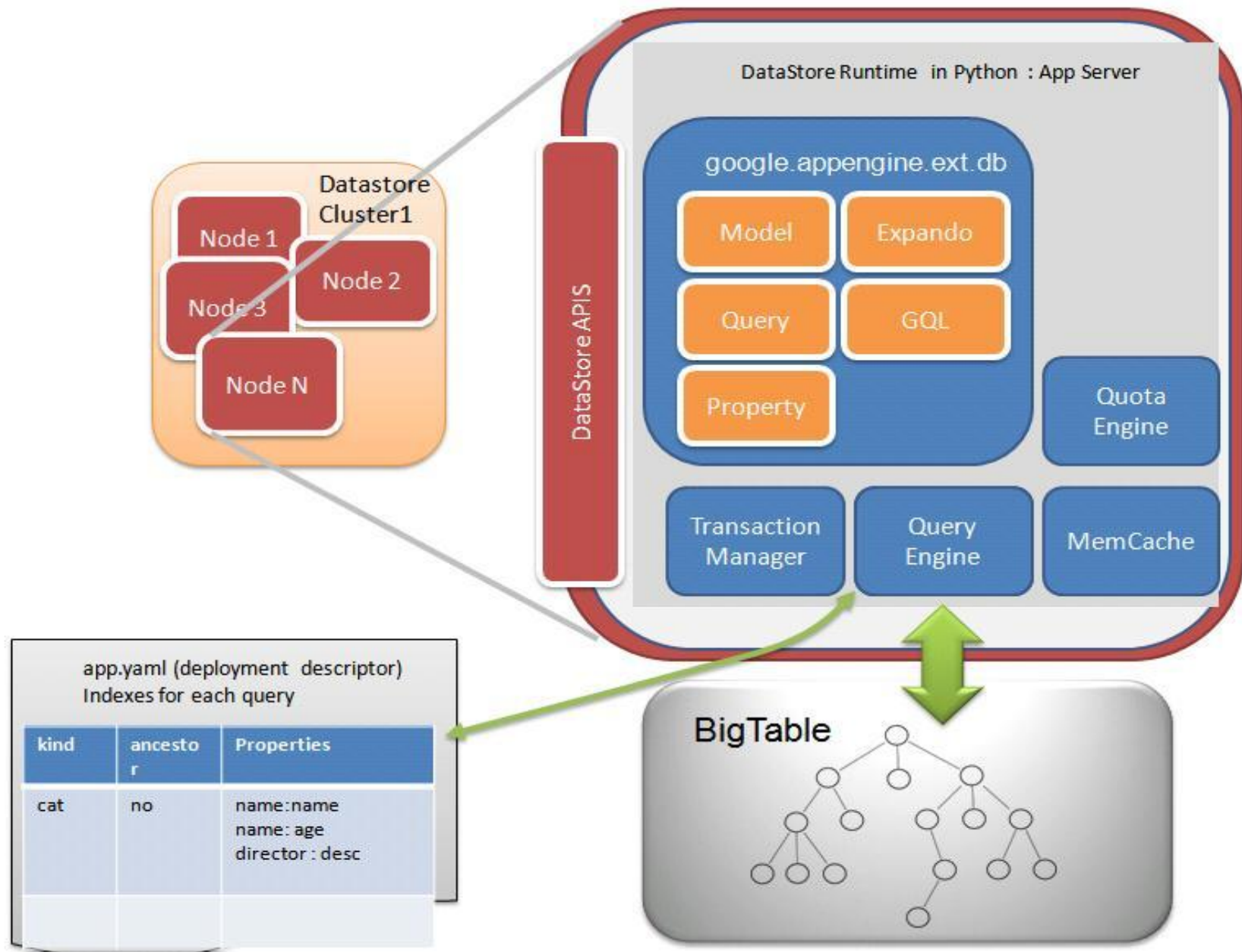


# Группы сущностей



- Группы сущностей хранятся в одном узле распределенного хранилища
- Группа сущностей определяет рамки одной транзакции

# DataStore - Runtime КОМПОНЕНТЫ



# App Engine DataStore

```
class Greeting(db.Model):  
    author = db.UserProperty()  
    content = db.StringProperty(multiline=True)  
    date = db.DateTimeProperty(auto_now_add=True)
```

Entity  
from db.Model

```
class Guestbook(webapp.RequestHandler):  
    def post(self):  
        greeting = Greeting()  
  
        if users.get_current_user():  
            greeting.author = users.get_current_user()  
  
        greeting.content = self.request.get('content')  
        greeting.put()  
        self.redirect('/')
```

Persisting  
An  
Entity

# GQL

```
class MainPage(webapp.RequestHandler):
    def get(self):
        self.response.out.write('<html><body>')

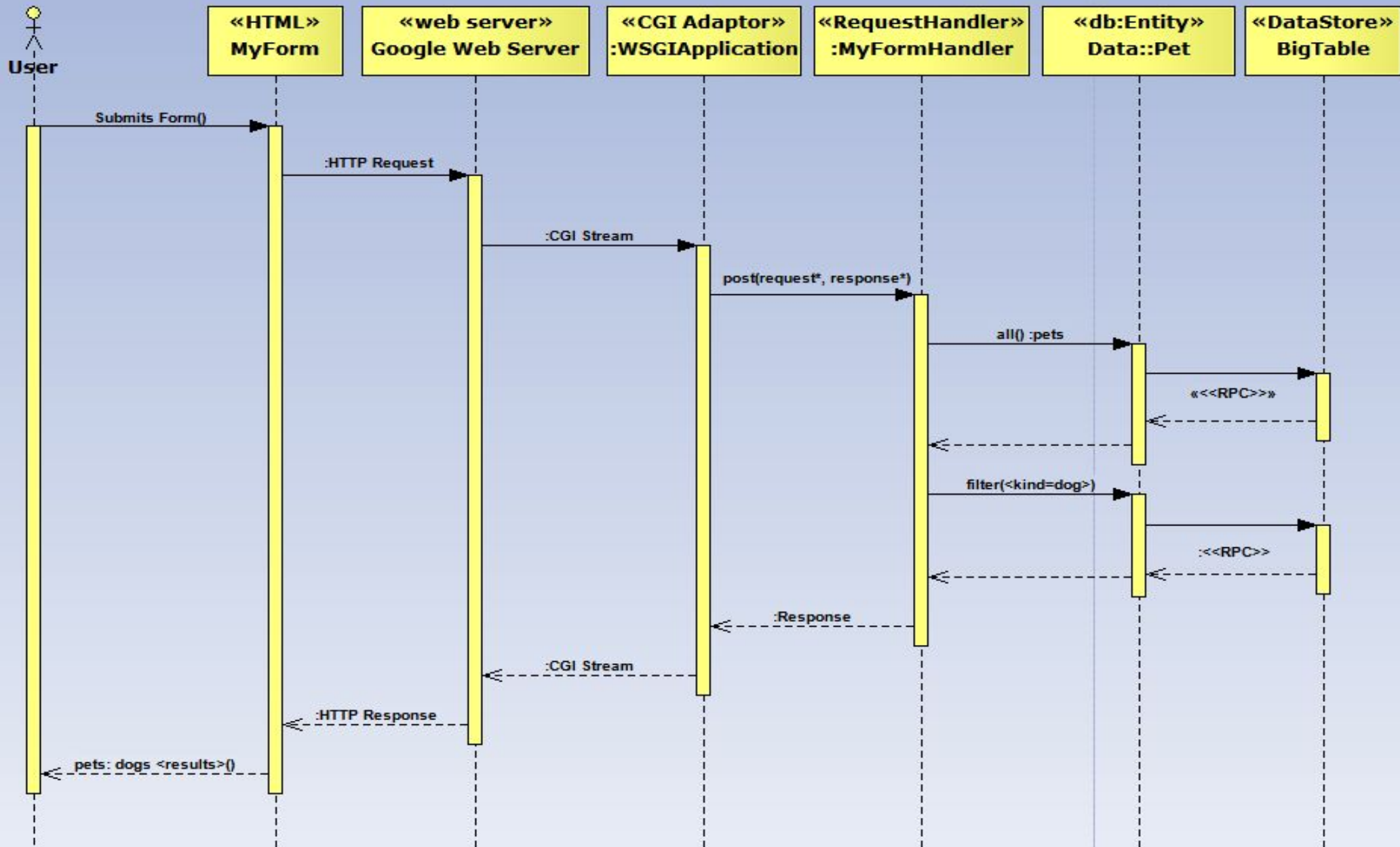
        greetings = db.GqlQuery("SELECT * FROM Greeting ORDER BY date DESC LIMIT 10")

        for greeting in greetings:
            if greeting.author:
                self.response.out.write('<b>%s</b> wrote:' % greeting.author.nickname())
            else:
                self.response.out.write('An anonymous person wrote:')
            self.response.out.write('<blockquote>%s</blockquote>' %
                                    cgi.escape(greeting.content))

        # Write the submission form and the footer of the page
        self.response.out.write("""
            <form action="/sign" method="post">
                <div><textarea name="content" rows="3" cols="60"></textarea></div>
                <div><input type="submit" value="Sign Guestbook"></div>
            </form>
            </body>
            </html>""")
```

GQL

# Обработка запроса к App Engine



# Тестирование и развертывание сервиса на App Engine

---

- Локальное тестирование
  - `dev_appserver.py helloworld/`
- Развертывание
  - `appcfg.py update helloworld/`
- Приложение «в облаке»
  - `http://<application-id>.appspot.com`



# Некоторые API для AppEngine

DataStore API

\* интерфейс к хранилищу данных

Images API

\* обработка изображений с помощью Images API

Memcache API

\* доступ к высоко-производительному кешу

Mail API

\* отправка почты из вашего приложения

Users API

\* интеграция с аккаунтами Google



# Обеспечение безопасности: Sandbox

создание и запись файлов запрещены

запрещены создание сетевых сокетов

приложения не могут порождать новые нити или процессы

процессы работающие более 10 секунд уничтожаются

системные вызовы (напр. посылка сигналов) запрещены

Расширения для python на языке C запрещены



# Полезные ресурсы

---

- Документация и SDK:  
<http://code.google.com/appengine/>
- Список рассылки:  
<http://groups.google.com/group/google-appengine/>

