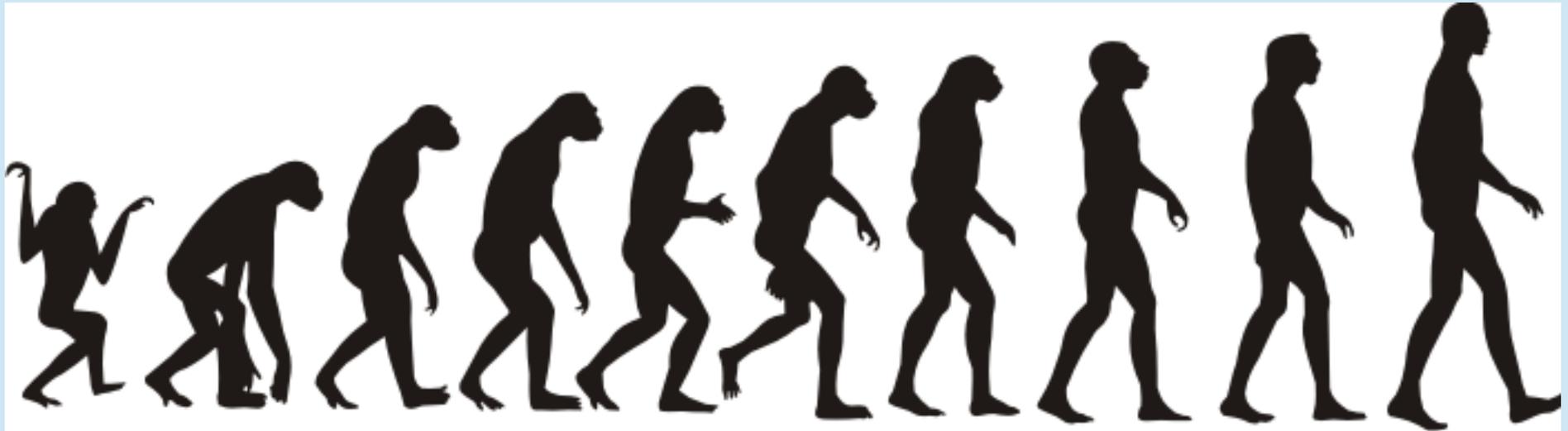


# Разработка веб-приложений без наркотиков

PostgreSQL + Nginx = C<sub>2</sub>H<sub>5</sub>OH

# Немного истории



HTML → CGI →

Шаблонизаторы → Фреймворки / ORM

# Слова, слова, слова

CGI YII Prolog IIS  
ASP **ORM** Apache C++  
MongoDB **PHP** Fortran Perl  
Framework Lisp FastCGI  
**J2EE** GoLang **.NET** Redis  
mod\_perl2 Symphony2 UWSGI  
Django

# Идея

- Модель данных
- Хранимые процедуры
- Представление
  
- Почему Nginx?
- Почему PostgreSQL?
  
- Конкурент — ngx\_postgres

# Пример приложения

Три схемы: `usr`, `posts`, `web`

- `web.index(cookies json, params json)`
- `web.user_login(cookies json, params json)`
- `web.user_join(cookies json, params json)`
  
- `web.comments(cookies json, params json)`
- `web.comment_save(cookies json, params json)`

return: `status`, `content`, `headers`

# Пример конфигурации

```
server {  
    listen 80; server_name localhost;  
  
    location /json {  
        c2h5oh_pass "<conn_string>" <pool_size>;  
        c2h5oh_timeout 100ms;  
    }  
}
```

# Проблемы

- Загрузка файлов
- Отправка почты
- Создание скриншотов сайтов
- Асинхронные HTTP-запросы
- Масштабируемость
- Интеграция с другими СУБД

# Плюсы C<sub>2</sub>H<sub>5</sub>OH

- Быстродействие
- Простота и скорость разработки
- Выше квалификация инженеров
- За данные и их обработку отвечает один
- Легче конфигурировать
- Безопасность
- Возможность расширения высокопроизводительными модулями
- Огромный инструментарий PostgreSQL

# Минусы C<sub>2</sub>H<sub>5</sub>OH

- Сложность разработки
- Труднее найти инженеров
- Ниже безопасность
- Скриптовые языки-таки всё одно нужны
- Непривычный подход
- Опасность изготовления неподдерживаемого кода
- Сложность в тестировании и отладке

# Задачи на будущее

- Воспринять критику
- Проверка боем
- Раздать продукт всем желающим (сырое отдавать стыдно)
- Доработка, отладка и тестирование

Благодарю за внимание, жду ваши  
вопросы, буду благодарен за критику

**[misha@genosse.org](mailto:misha@genosse.org)**