

Web applications in the real world

Víctor López Ferrando

Who am I? Víctor López Ferrando

- Mathematics Degree + Informatics Engineering (UPC)
- 2 years experience as Software Engineer with heavy web work
- PhD Student in Bioinformatics (BSC)
- My first web application: 2004
- 3 “big” web applications developed

Contents of this presentation

- 1. Web technology evolution**
- 2. Web app development.**
- 3. Deployment.**
- 4. Conclusions**

1. Web technology evolution

Historical evolution of web technology.

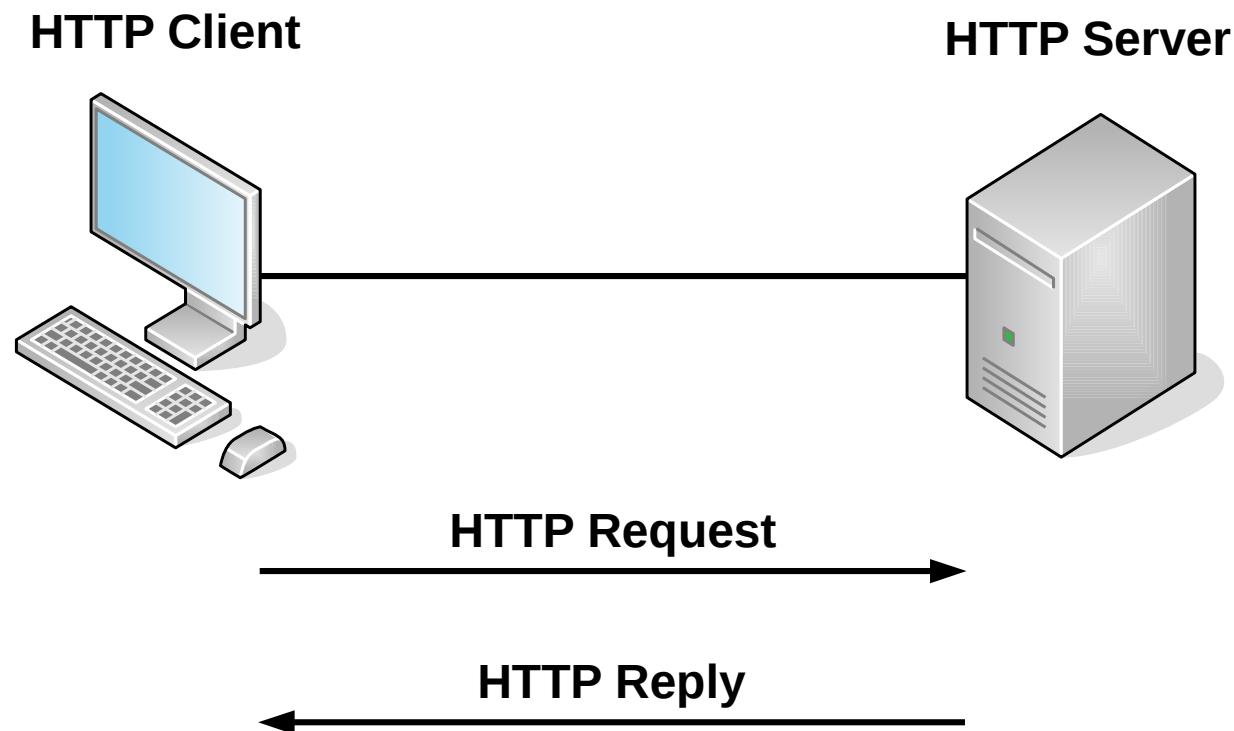
- 1980 – 1991: Birth of the web.
- 1992 – 1999: Static web.
- 2000 – 2006: Dynamic web.
- 2007 – 2011: Interactive web.
- 2012 – Present: current state of web technology.

1980 – 1991: Birth of the web

Invention and implementation of the basic building blocks.

HTTP (Hypertext Transfer Protocol)

How to transfer content: GET, POST...



HTML (HyperText Markup Language)

Which content: <h1>, <p>, , and above all... <a>



URI (Uniform Resource Identifier)

How to find resources:

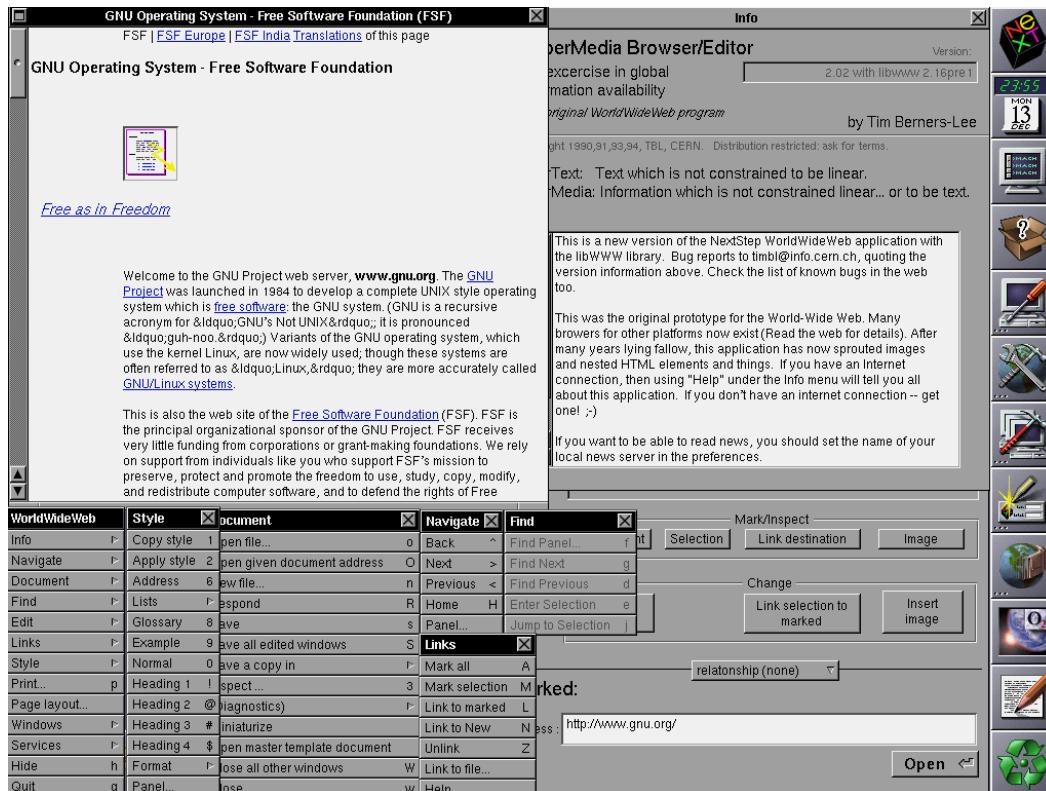
`http://dev.example.com:80/gallery/videos?id=217#comments`

The diagram illustrates the structure of a Uniform Resource Identifier (URI) by breaking it down into five labeled components. The URI itself is shown in green, purple, orange, and brown text. Below the URI, five red curly braces extend from left to right, each bracketing a specific part of the string. The labels 'Scheme', 'Net location', 'Path', 'Query', and 'Fragment' are positioned directly beneath their respective components. The 'Scheme' label is under 'http://'. The 'Net location' label is under 'dev.example.com:80'. The 'Path' label is under '/gallery/videos'. The 'Query' label is under '?id=217'. The 'Fragment' label is under '#comments'.

Scheme Net location Path Query Fragment

Web Browser: orchestrate everything

First browser: WorldWideWeb (released December 25, 1990)



First website (1991, CERN)

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

What's out there?

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

Help

on the browser you are using

Software Products

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,[X11 Viola](#) ,[NeXTStep](#) ,[Servers](#) ,[Tools](#) ,[Mail robot](#) ,[Library](#))

Technical

Details of protocols, formats, program internals etc

Bibliography

Paper documentation on W3 and references.

People

A list of some people involved in the project.

History

A summary of the history of the project.

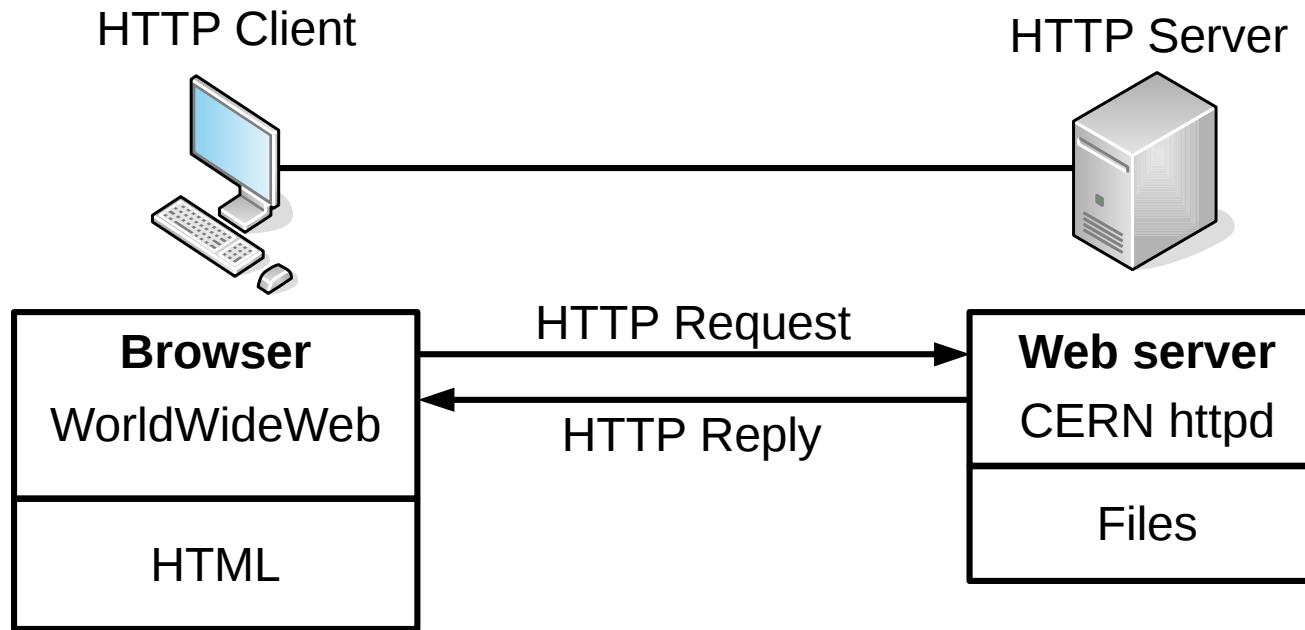
How can I help ?

If you would like to support the web..

Getting code

Getting the code by [anonymous FTP](#) , etc.

1980 – 1991: Summary



1992 – 1999: The static web



- Static, information-oriented websites.
- First million websites: 1997.

1992 – 1999: The static web

Static content in HTML with inline style.

```
<html>
  <head>
    <title>Web title</title>
  </head>
  <body>
    <font color="#A40317">
      <font size=+2>
        Header
      </font>
    </font>
    <p>
      <font color="#000000">
        Lorem ipsum...
      </font>
    </p>
    <p>
      <font color="#000000">
        Lorem ipsum...
      </font>
    </p>
  </body>
</html>
```

Universitat de Barcelona (1996)

The screenshot shows the homepage of the Universitat de Barcelona's website from 1996. The header features the university's name "UNIVERSITAT DE BARCELONA" and its motto "Libertas perfundet omnia luce". A decorative banner with a Santa Claus figure and a blue ribbon is present. The main navigation menu on the right includes links for "Presentació", "Estructura", "Ensenyaments", "Notícies", "Biblioteca", "Serveis", and "La Bústia del Rector". Below the menu, a counter displays "288309" visits since November 30, 1995. The footer contains the university's name, links to text versions in different languages, and an email address for the webmaster.

UNIVERSITAT DE BARCELONA
Libertas perfundet omnia luce

Bones Festes!

UBWeb

Presentació

Estructura

Ensenyaments

Notícies

Biblioteca

Serveis

La Bústia del Rector

Aquesta pàgina ha tingut
288309
visites des del 30 de novembre de 1995.

UNIVERSITAT DE BARCELONA
[[Versió text](#) / [Versión texto](#) / [Text version](#)]

webmaster@www.ub.es

Generalitat de Catalunya (1997)

The screenshot shows the homepage of the Generalitat de Catalunya website. At the top left is the Catalan coat of arms and the text "Generalitat de Catalunya". To the right is a decorative banner featuring a landscape with three stone lions. Below the banner are several menu items: "Catalunya", "Generalitat", "Serveis d'informació", "Escaec a escaec", "English", "Altres www", "Buscar" (with a magnifying glass icon), and "Bústies" (with a newspaper icon). A central news item reads: "El 5 de juny, Dia Mundial del Medi Ambient, el conseller Pere Macias inaugurarà els Xats de Medi Ambient". At the bottom left is a "Catalunya" link with a map icon. On the right side, there is a sidebar with a list of links: "Història de Catalunya", "Els símbols nacionals i l'idioma", "Estatut d'Autonomia", "L'organització del territori", "Catalunya poble a poble", and "Catalunya de ben a prop".

Catalunya

Generalitat

Serveis d'informació

Escaec a escaec

English

Altres www

Buscar

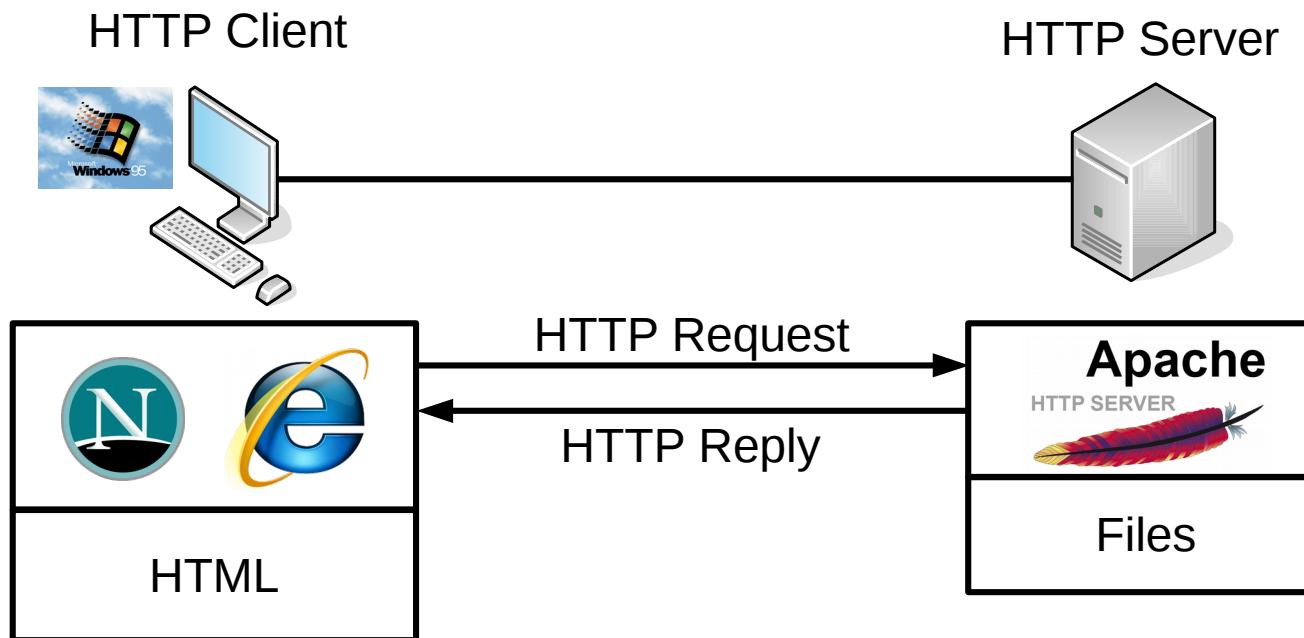
Bústies

El 5 de juny, Dia Mundial del Medi Ambient, el conseller Pere Macias inaugurarà els Xats de Medi Ambient.

Catalunya

- [Història de Catalunya](#)
- [Els símbols nacionals i l'idioma](#)
- [Estatut d'Autonomia](#)
- [L'organització del territori](#)
- [Catalunya poble a poble](#)
- [Catalunya de ben a prop](#)

1992 – 1999: Summary



2000 – 2006: The dynamic web

Blogs, forums, wikis...



ubuntu
linux for human beings



WIKIPEDIA
The Free Encyclopedia

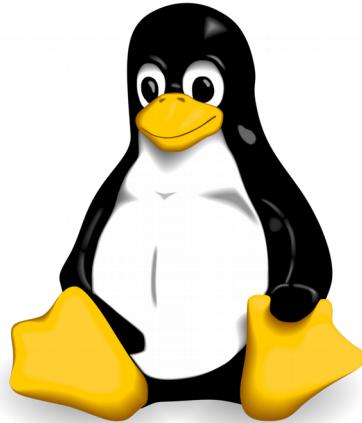
CSS (Cascading Style Sheets)

Style separated from HTML: fonts, colors, spacing...

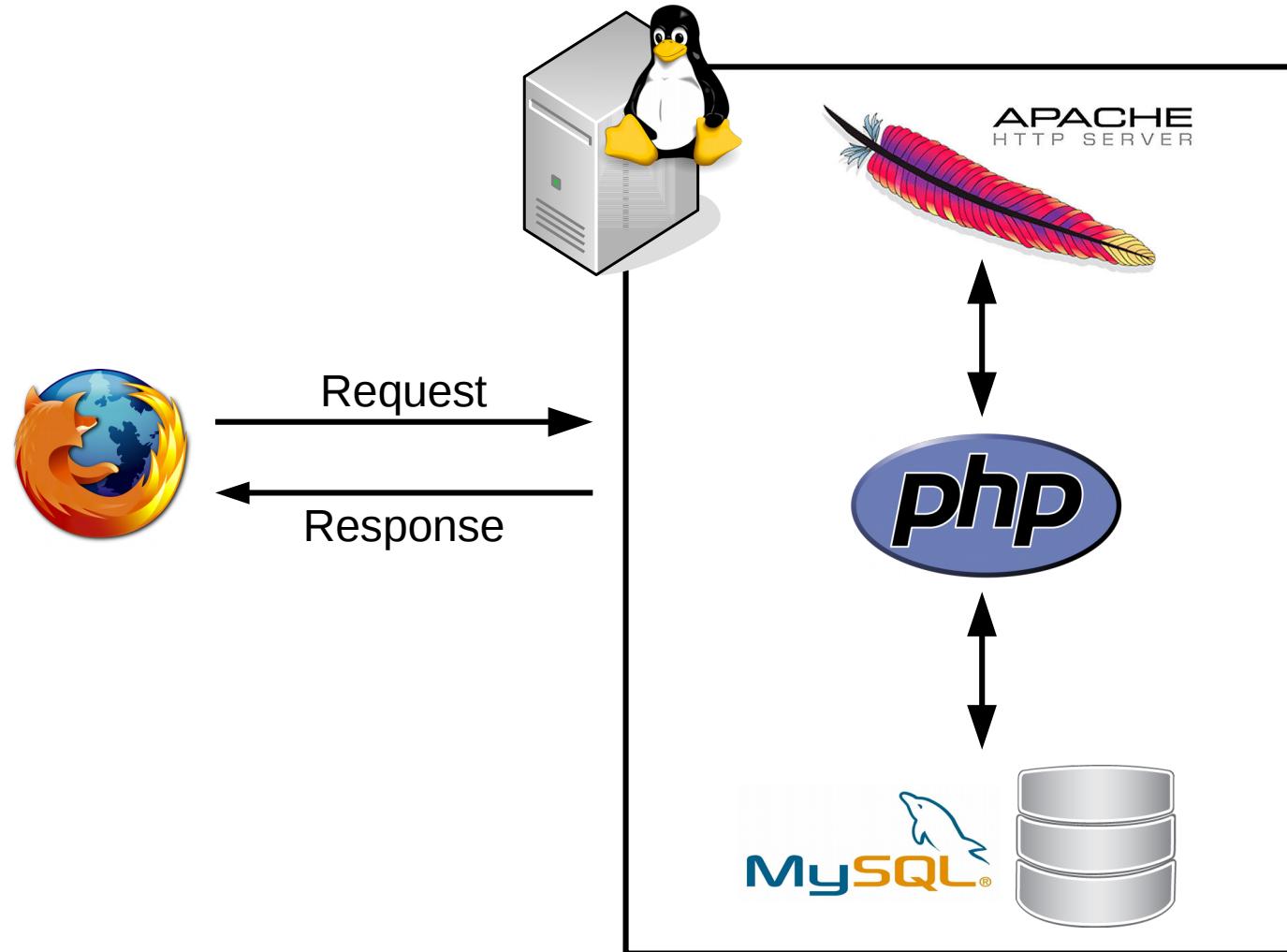


LAMP rules the server

LAMP: Linux + Apache + MySQL + PHP



LAMP architecture



Wikipedia (2003)



[Main Page](#)
[Recent changes](#)
[Random page](#)
[Current events](#)

[Protected page](#)
[Discuss this page](#)
[Page history](#)
[What links here](#)
[Related changes](#)

[Special pages](#)
[Bug reports](#)
[Donations](#)

[Main Page](#) | [Recent changes](#) | [Protected page](#) | [Page history](#)
[Printable version](#)

Not logged in
[Log in](#) | [Help](#)
 [Go](#) [Search](#)

Main Page

From Wikipedia, the free encyclopedia.
Find out how you can help [support](#) Wikipedia's phenomenal growth.

Wikipedia is a [multilingual](#) project to create a complete and accurate [free content](#) encyclopedia. We started in January 2001 and are currently working on 164918 [articles](#) in the English version. Visit the [help page](#) and experiment in the [sandbox](#) to learn how **you** can edit any article right now.

Selected Articles

In the news: [Shenzhou 5](#) - [Shirin Ebadi](#) - [Bolivia crisis](#) - [Japan election](#) - [Limbaugh](#)
Recent deaths: [Jim Cairns](#) - [Izzy Asper](#) - [Neil Postman](#) - [Sid McMath](#) - [William Steig](#)
New articles: [Devil's Brigade](#) - [Megachurch](#) - [Lagentium](#) - [Zinnia](#) - [Lindos](#) - [Prodigy](#)
Week in history: [October 15](#) - [Napoleon](#) - [CSS Hunley](#) - [Peace Corps](#) - [Gorbachev](#) -
[Cuban Missile Crisis](#) - [Eric Rudolph](#) - [Knights Templar](#) - [Columbus](#)

Encyclopedia

Mathematical and Natural Sciences

[Astronomy and astrophysics](#) - [Biology](#) - [Chemistry](#) -
[Computer science](#) - [Earth science](#) - [Health science](#) -
[Mathematics](#) - [Physics](#) - [Statistics](#)

Applied Arts and Sciences

Community

Writing Articles

[Welcome, newcomers](#) - [Policy](#) -
[Copyrights](#) - [How to edit](#) - [Guide to layout](#) and [style](#) - [Public domain](#) and [shared resources](#) - [Requested articles](#) and [pictures](#) - [Brilliant prose](#) and

Wordpress (2005)

“Web Application
of the Year”

HOME | ABOUT | Docs | DEV BLOG | SUPPORT | DONATE | DOWNLOAD

WordPress 1.5 is available!

WordPress is a state-of-the-art semantic personal publishing platform with a focus on aesthetics, web standards, and usability. What a mouthful. WordPress is both free and priceless at the same time.

More simply, WordPress is what you use when you want to work with your blogging software, not fight it.

If you are curious what WordPress looks like, we have [old screenshots up](#). New 1.5 screenshots coming soon.

For information about the WordPress project—its philosophy, current features, developers, and future direction—see the [About section](#). For updates on development and announcements, see the [WordPress Development Blog](#). If you'd like

A screenshot of the WordPress 1.5 dashboard. The main area shows a post editor with fields for Title, Text, and a sidebar for categories and tags. The sidebar includes links for Write, Edit, Categories, Links, Users, Options, Templates, Upload, My Profile, View site, Logout, and Comment. At the bottom, there are buttons for Save as Draft, Save as Private, Publish, and Advanced Editing.

Welcome to WordPress

WordPress [features](#):

- Instant publishing with no rebuilding
- Comment spam protection
- Integrated blogrolling features
- Manage Pages
- Password-protected posts
- Custom fields
- Dead simple upgrades and installation
- Easy to customize
- Multiple authors
- Multiple categories
- Sub-categories
- Comment registration

22

1. Web technology

Richer front end

Option #1: JavaScript

- Code that runs in the browser. Very slow!
- Examples: countdown, hiding and showing menus



Richer front end

Option #2: Flash

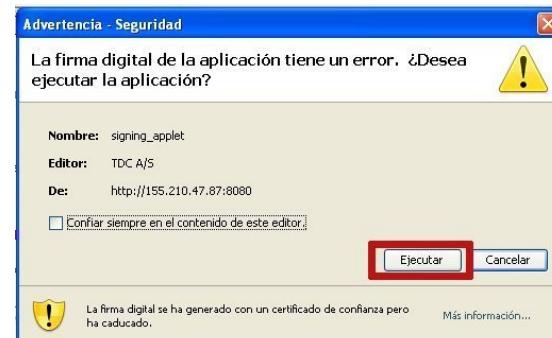
- Closed platform (Macromedia, Adobe)
- Examples: games, videos...



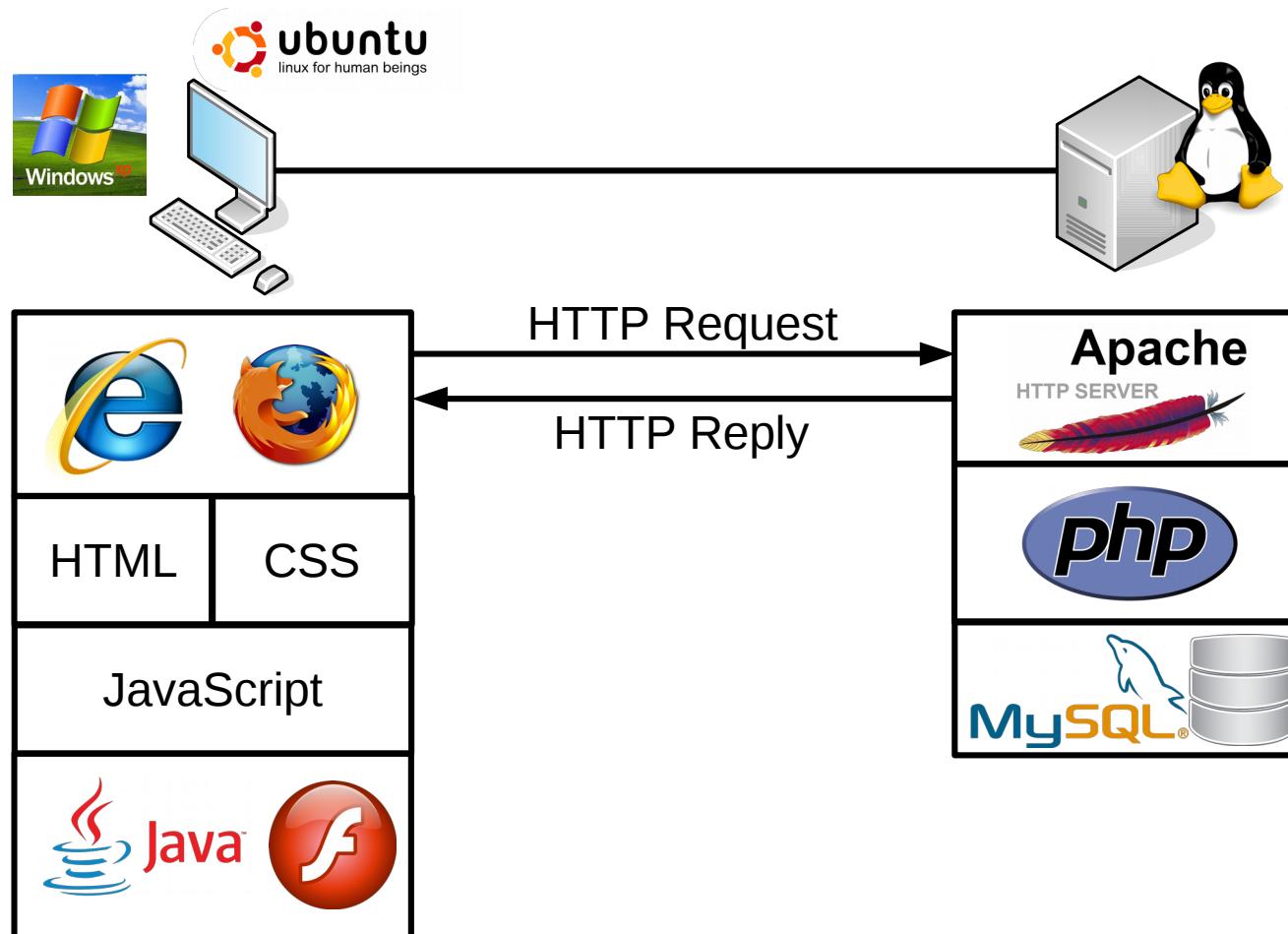
Richer front end

Option #3: Java Applets

- Require external installation
- Examples: games, “serious” applications...



2000 – 2006: Summary



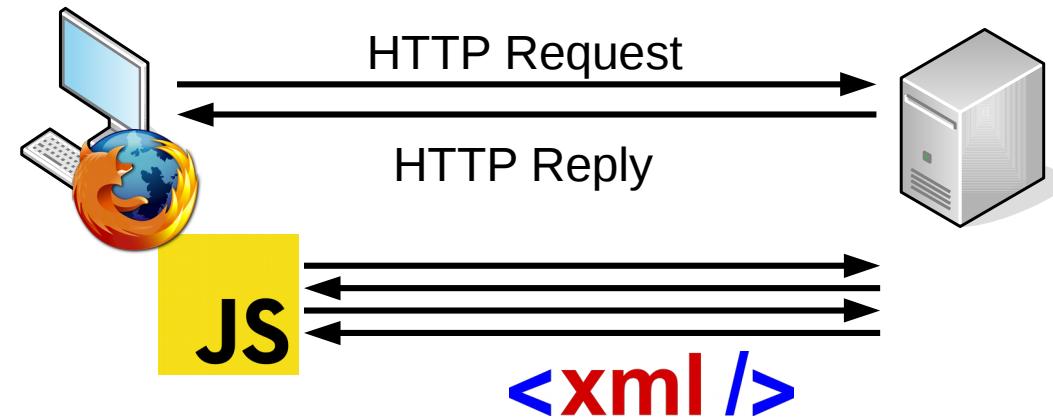
2007 – 2011: The interactive Web

Chat, rich applications, collaborative edition...

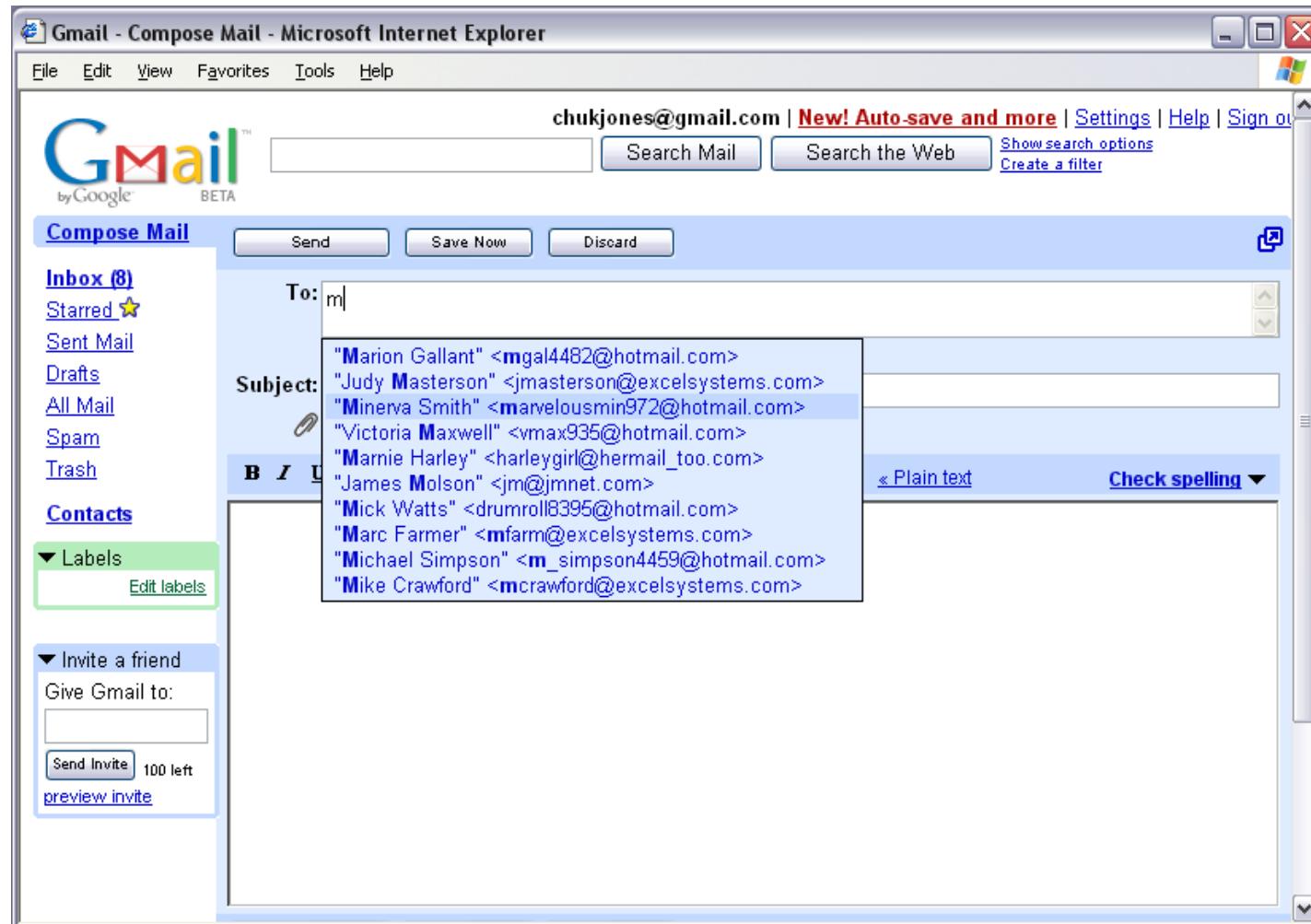


AJAX: Asynchronous Javascript And Xml

i. e., change the web content without refreshing



Gmail



In the server: more languages and frameworks

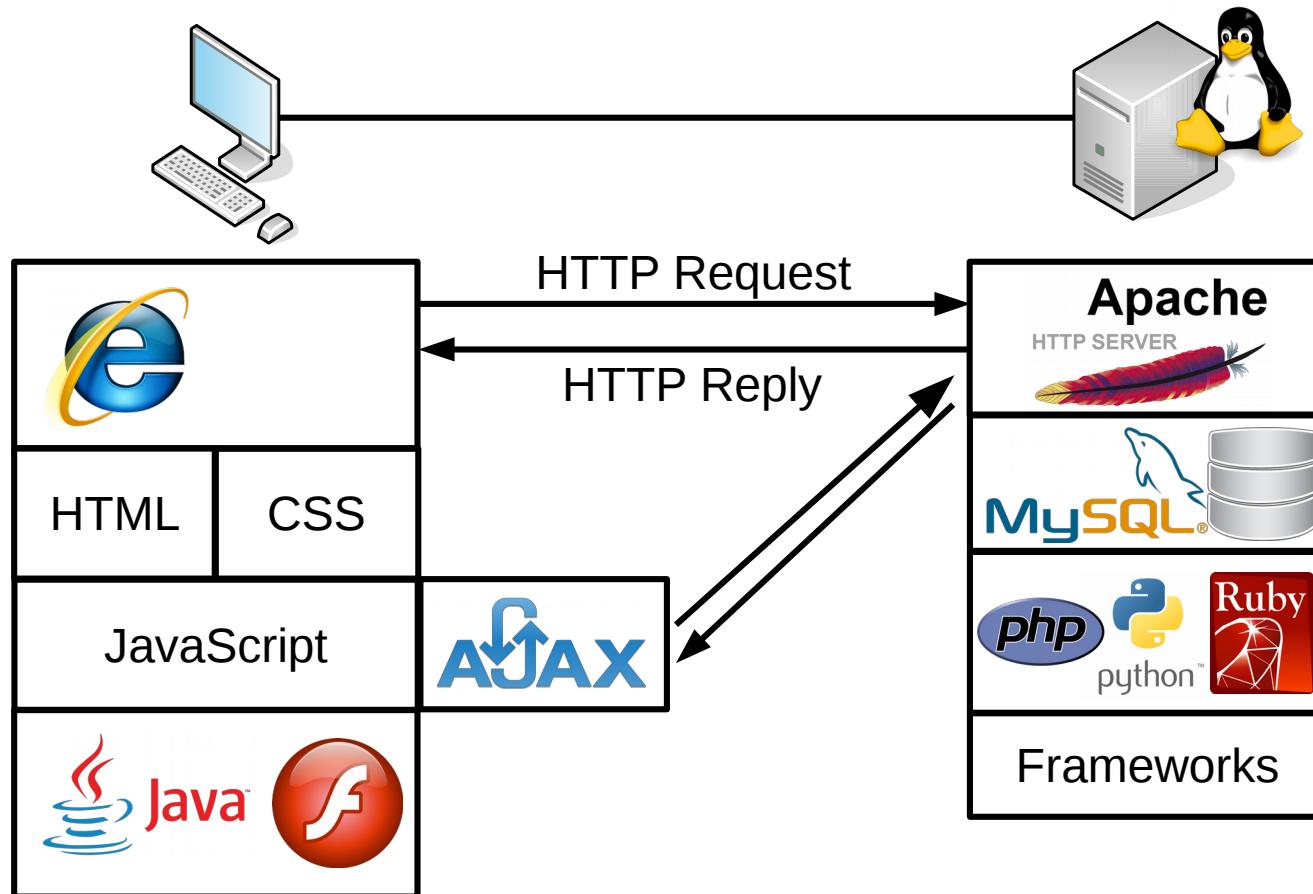
- More languages: Python, Ruby...
- Frameworks boost application development.
- Companies using Ruby on Rails / Django + AJAX:



Instagram



2007 – 2011: Summary

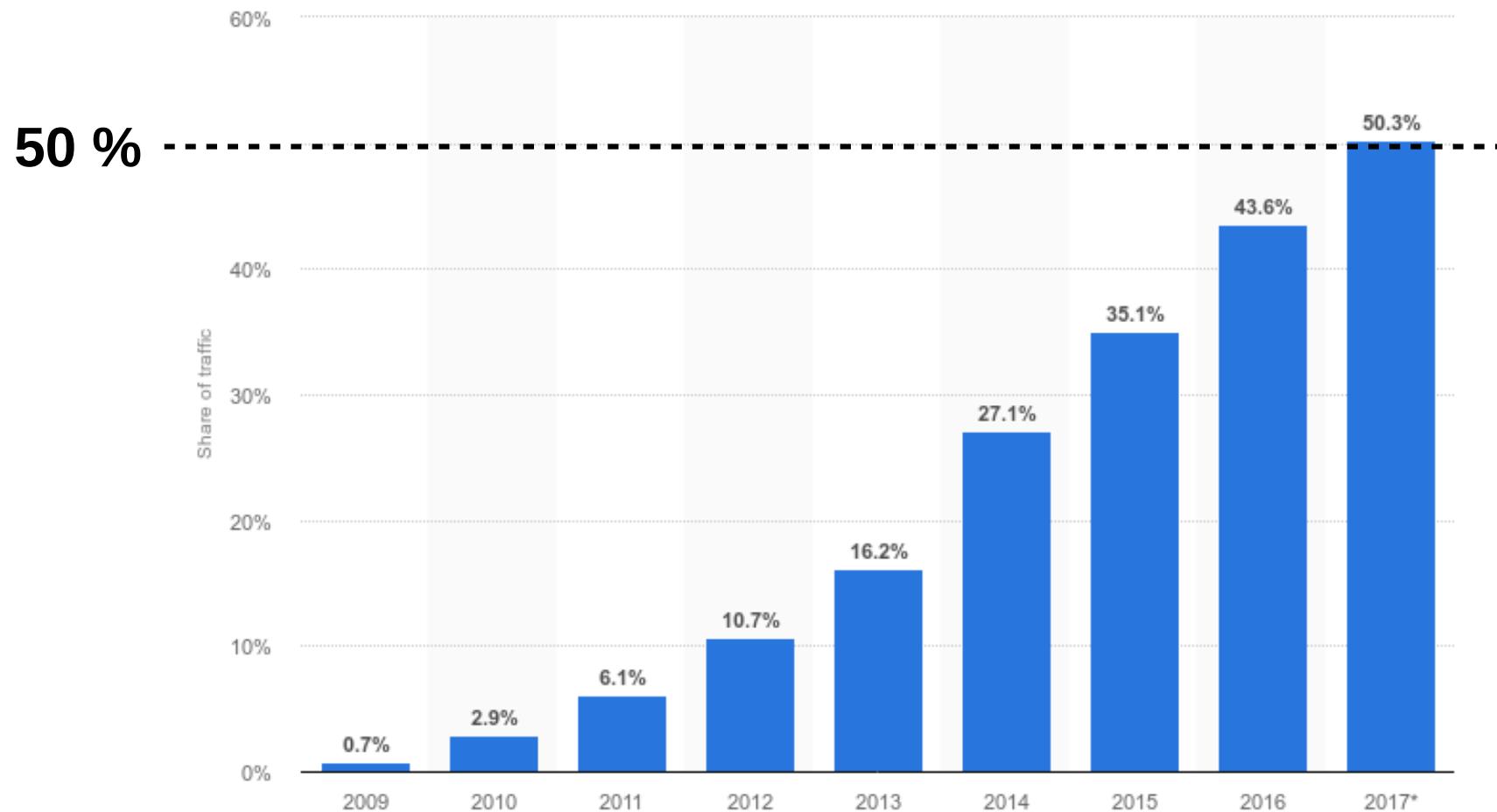


2012 - ...: Welcome to the present



Mobile browsers match Desktop

% of web pages served to mobile phones (2009 - 2017)



JavaScript reborn

Faster execution & richer front-end applications



Propuesta del Proyecto Everest

Archivo Editar Ver Insertar Formato Herramientas Tabla Ayuda Comentarios Compartir

Propuesta del Proyecto Everest

andrés

eduardo marta andrés carla antonio

Propuesta del Proyecto Everest

adipiscing elit. Pellentesque quis accu nulla. Donec volutpat molestie volutpat Morbi quis mi orci, eget scelerisque ell. Etiam ullamcorper iaculis pharetra. Vestibulum vitae velit sit amet diam adipiscing faucibus. Nullam vel velit et lacus elementum mollis eu vel lorem. Suspendisse a dolor in est aliquet vestibulum. Duis non quam vel quam rhoncus porttitor ac eget augue. Quisque pulvinar volutpat suscipit. Mauris augue risus, pretium in hendrerit vel, adipiscing ut sapien. Maecenas vestibulum tellus feugiat risus gravida pellentesque. Sed rutrum commodo tristique. Maecenas interdum mollis magna non mattis. Mauris et ligula sed nisl bibendum iaculis et in risus. Fusce id enim mauris. Nullam sit amet vehicula tortor. I metus, gravida eu elementum at, fermentum id leo. Donec sodales lorem et eros cursus placerat. Maecenas quis diam orci. Nunc eget augue eget dui gravida molestie. Nam dignissim risus sed nulla aliquam interdum.

eduardo

Mauris laoreet lectus vitae quam porttitor vestibulum. Quisque pharetra turpis eget augue

Standard heaven

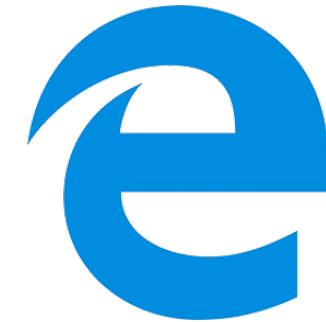
HTML



CSS



JS

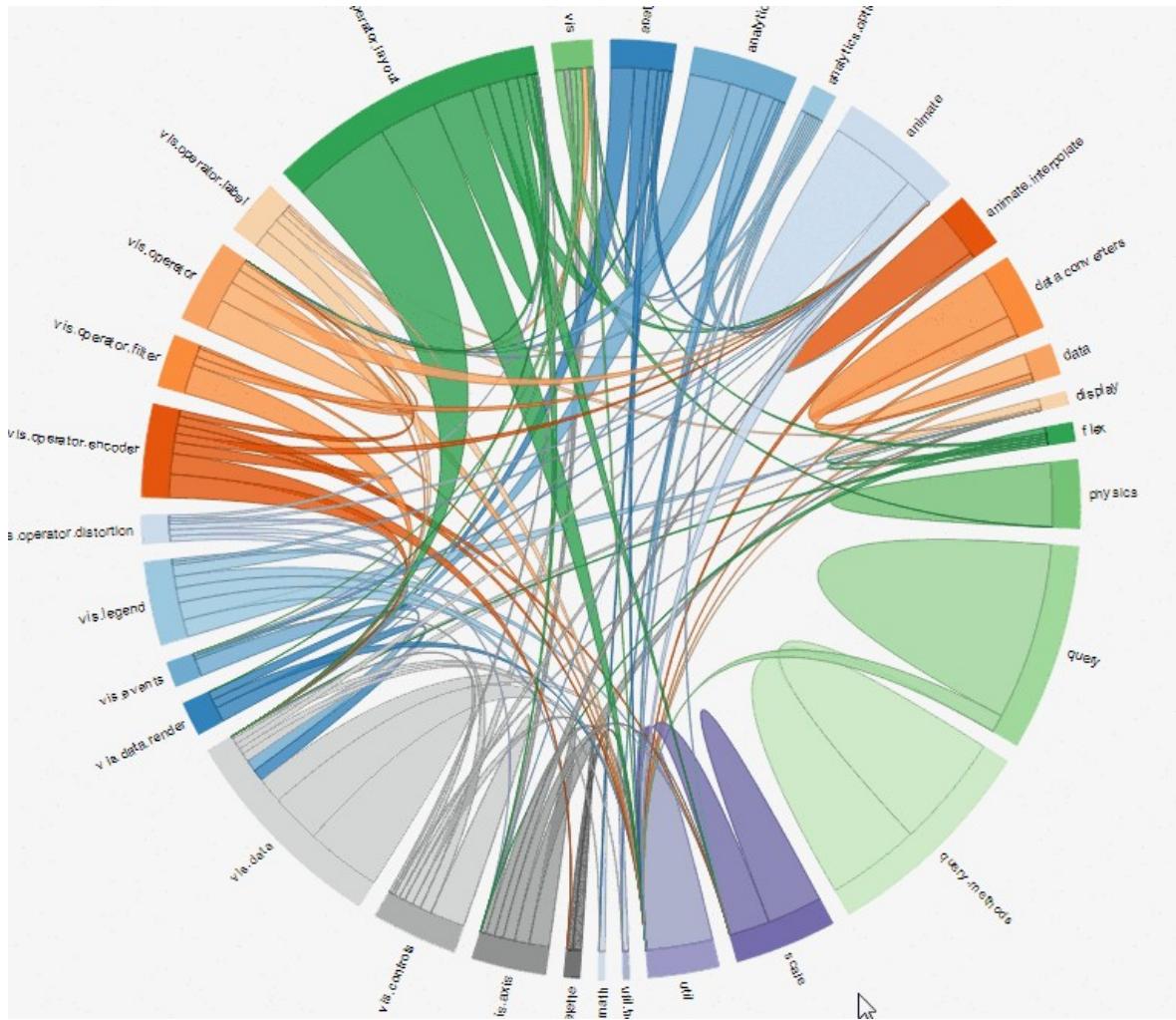


HTML5: Video tag

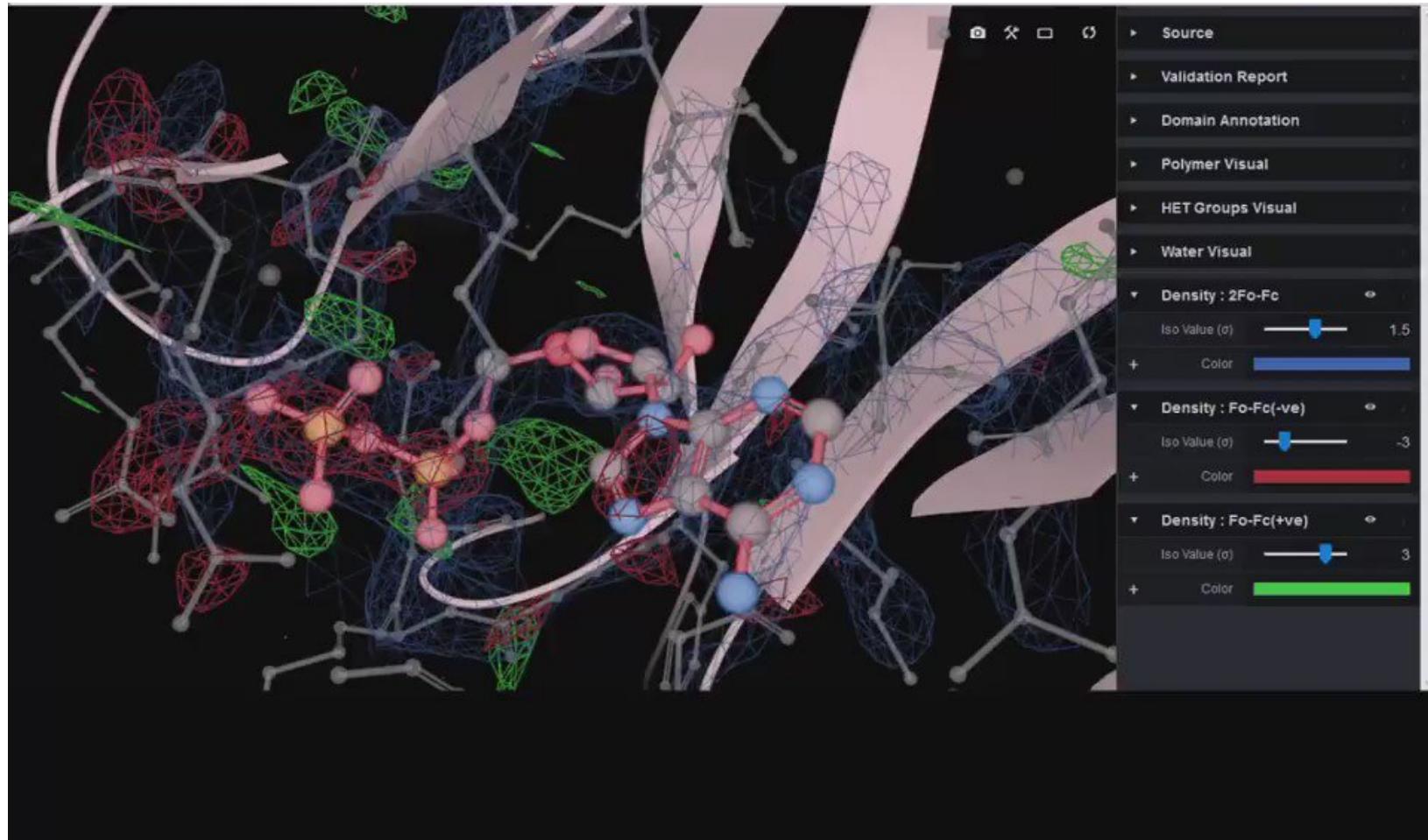
```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
</video>
```



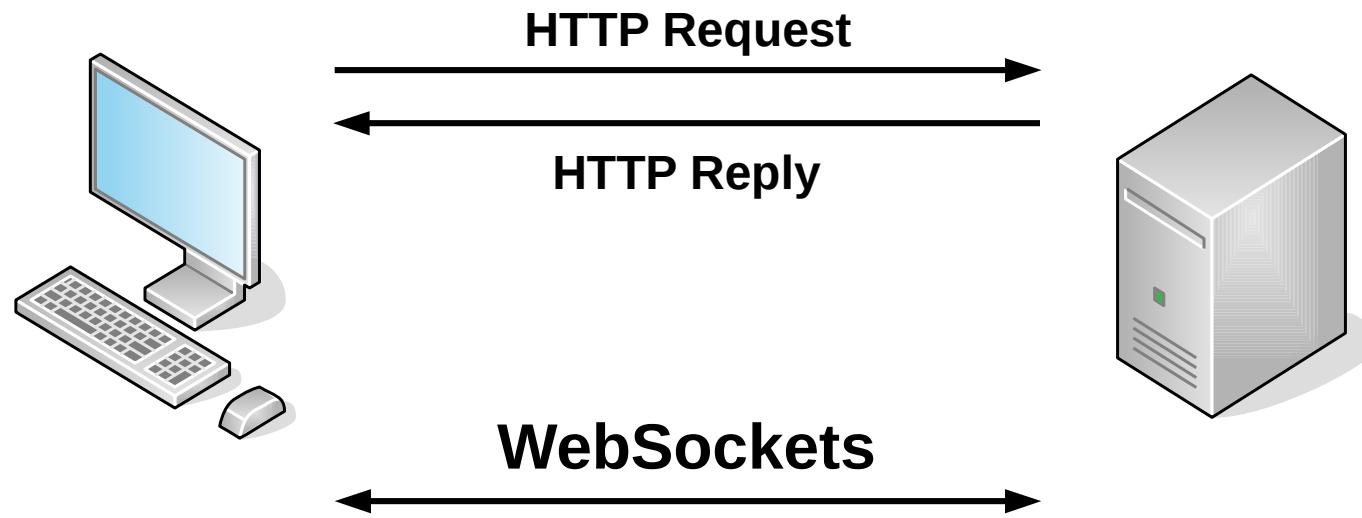
HTML5: Scalable Vector Graphics (SVG)



HTML5: Canvas + WebGL



HTML5: WebSockets



Example use case: Chat application

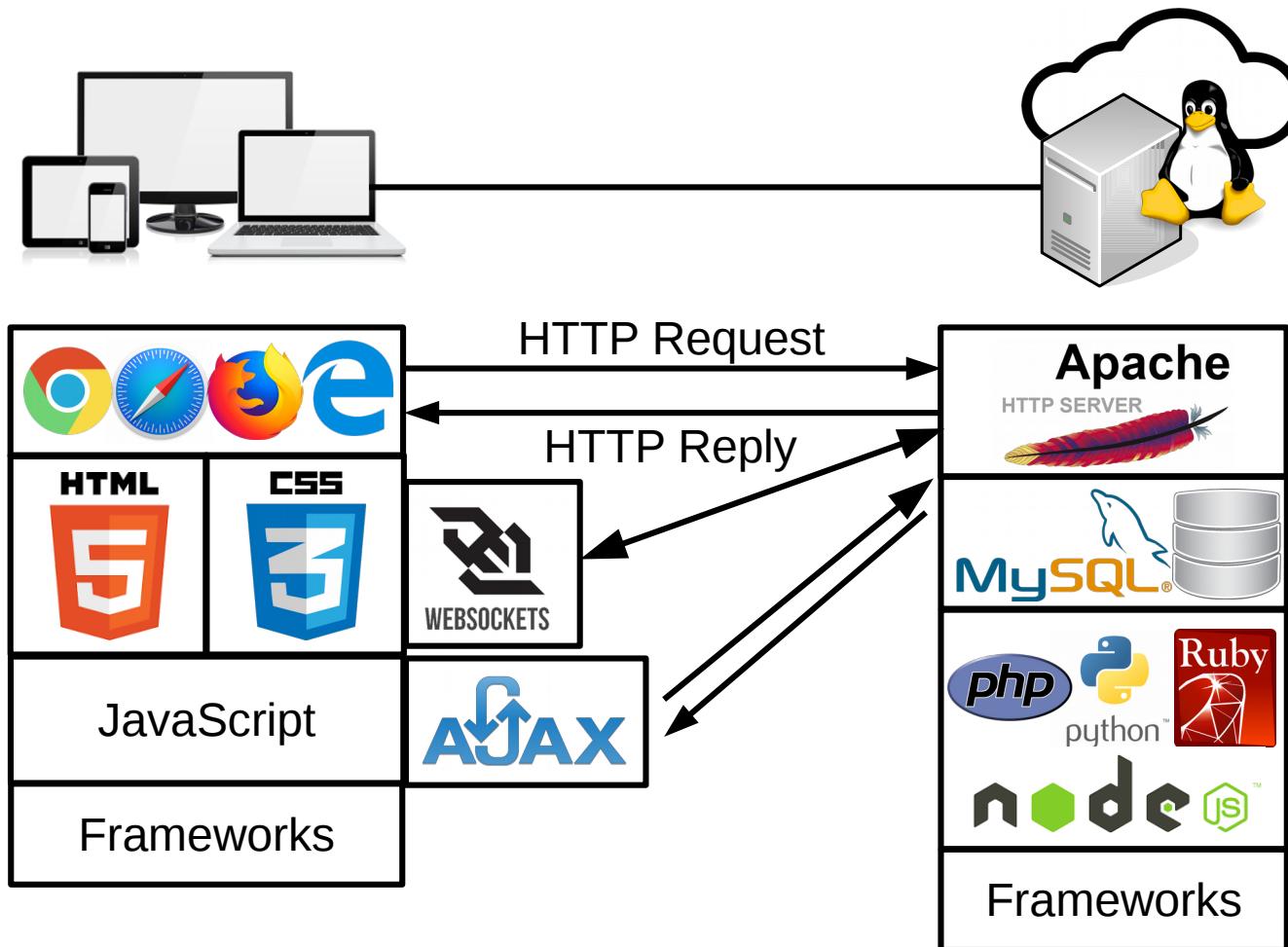
HTML5: WebAssembly

Fast bytecode that runs in the browser.

Can compile other languages (e.g. C++) to Webassembly.

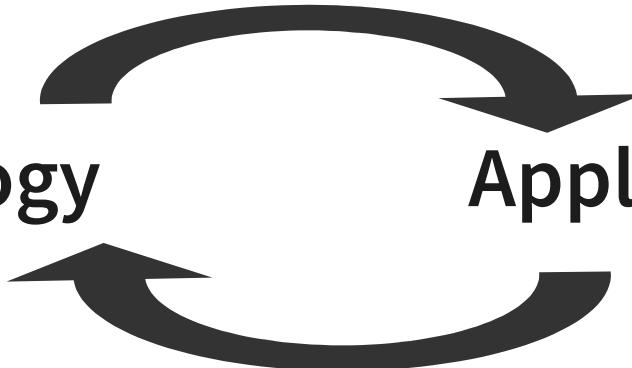


2012 - ...: Summary



Conclusions

1. Technology → Applications



2. Standards vs. Implementations

3. Constant evolution

2. Development of web applications

- Back end development:**

- Frameworks

- Front end development:**

- Components
 - Frameworks

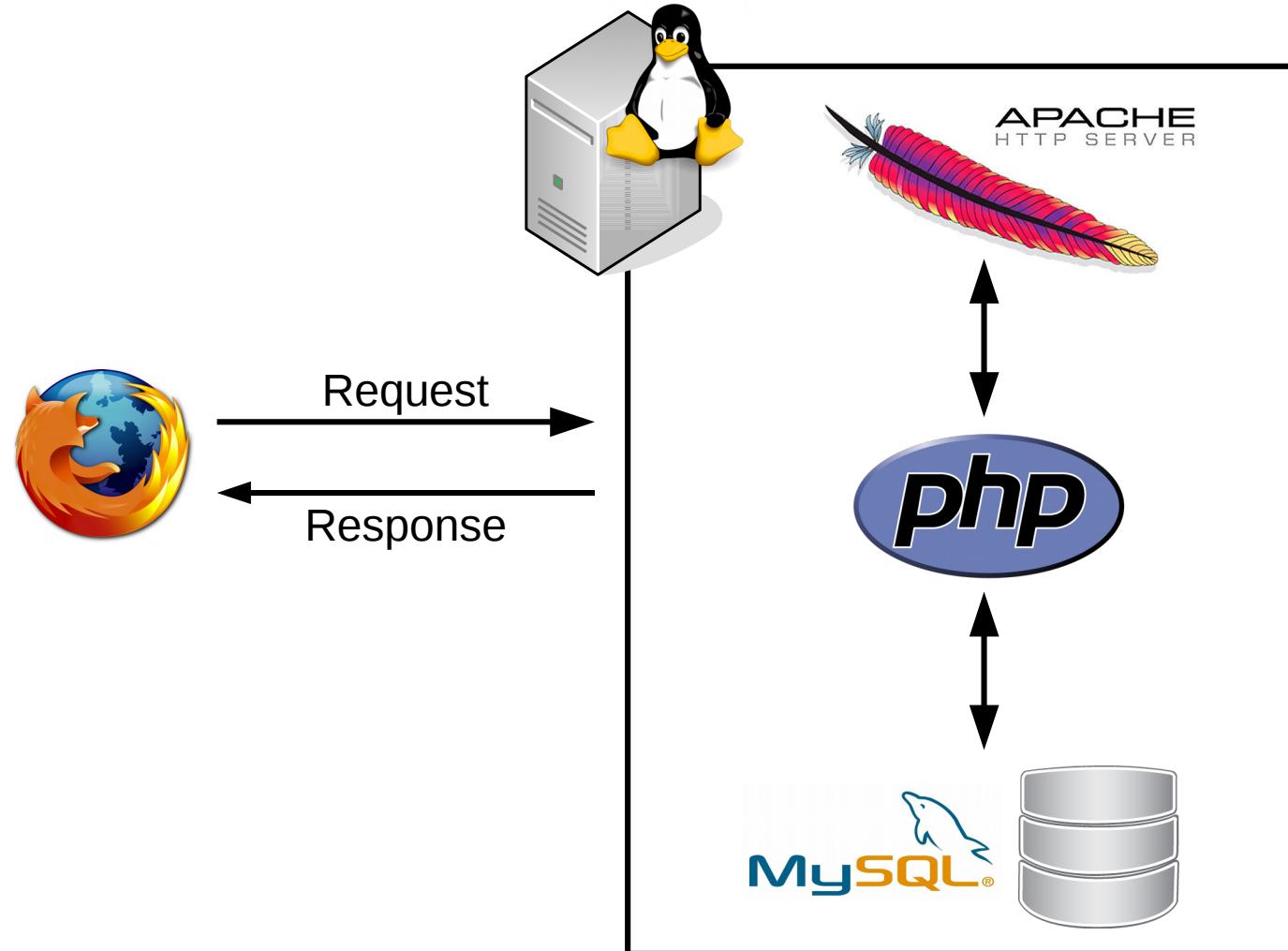
What do we want when we develop a web app?

- Correct
- Understandable
- Extensible
- Maintainable
- Reusable
- Secure

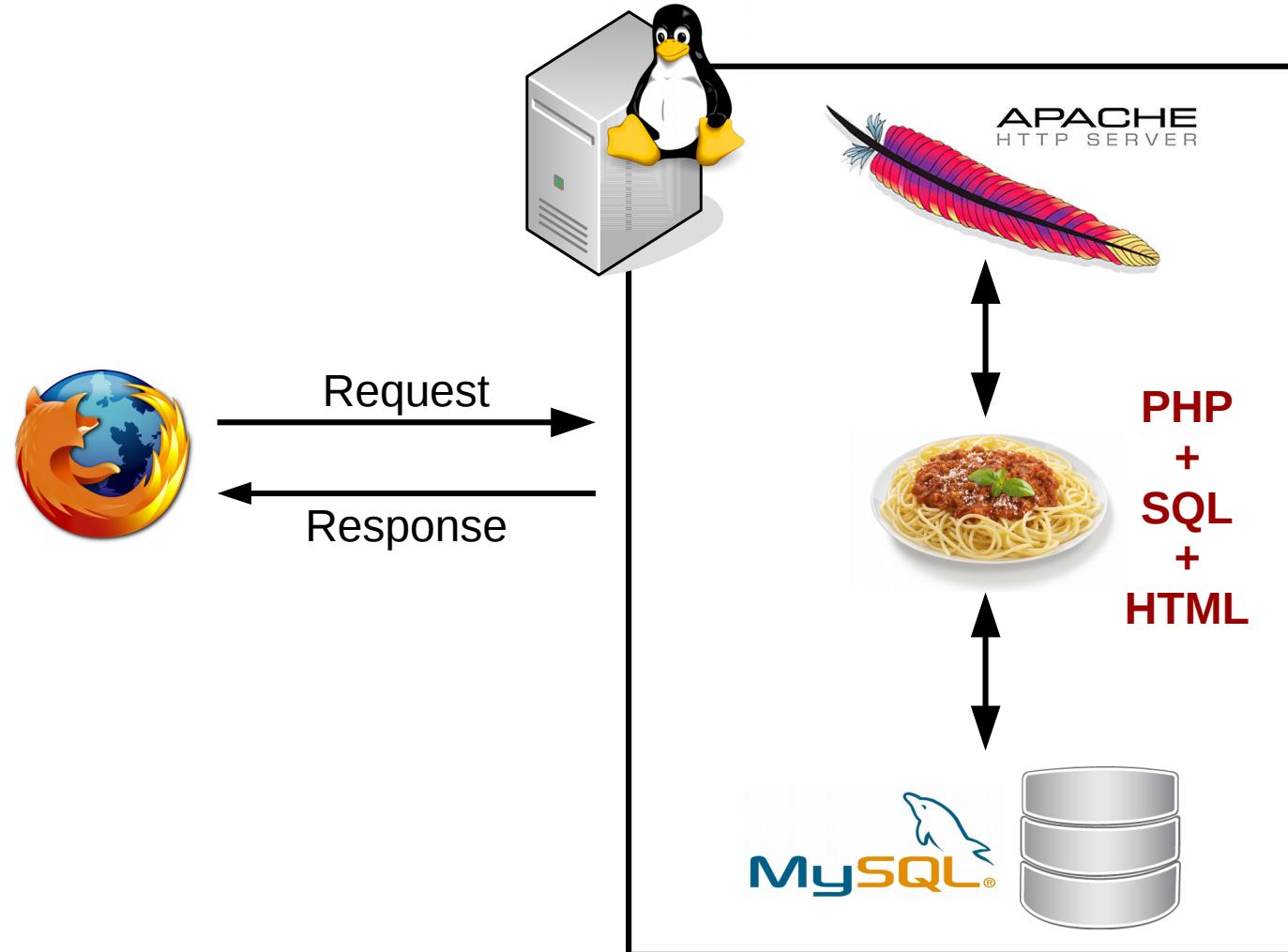
Focus on our logic.

Get things done!

Back end: the P in the LAMP



Back end: Spaghetti in the LAMP



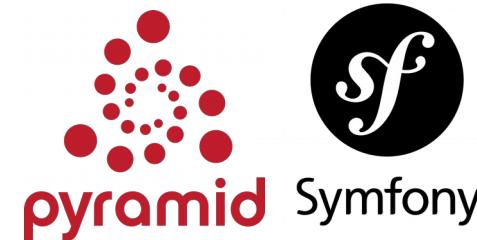
Frameworks



CodeIgniter



Struts™



Choosing a framework

1. Choose a programming language.
2. Frameworks vs. microframeworks.
3. Personal taste: Stable vs. Experimental...

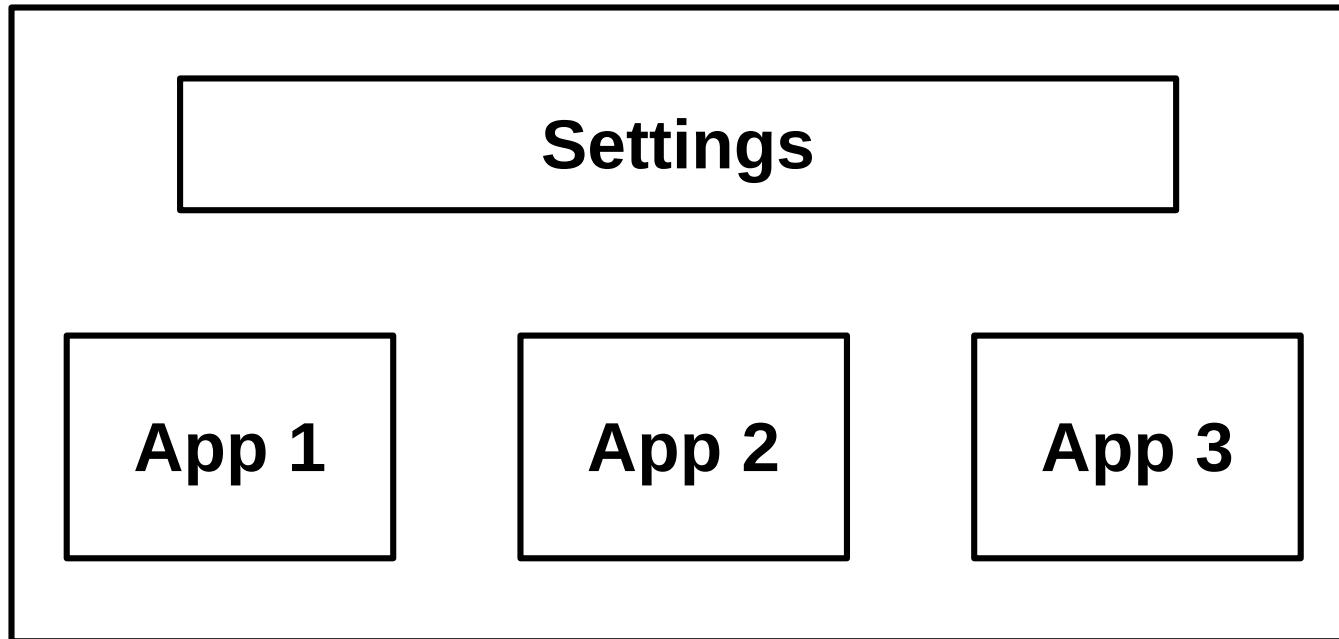
My choice: Django

The logo for the Django web framework, featuring the word "django" in a lowercase, bold, white sans-serif font against a dark green background.

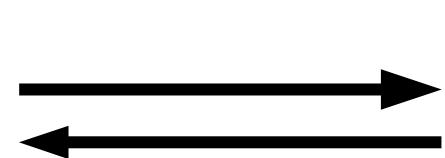
The web framework for
perfectionists with deadlines.

- **Python**
 - Numpy, SciPy, Scikit-learn, Pandas, Jupyter, BioPython...
- **Full-fledged framework**
 - Batteries included
- **Learn web development by using a framework.**

Django project organization



App design pattern: Model – View – Controller



Controller

URLs, GET, POST

Model

```
class BlogPost:  
    author = User()  
    title = String()  
    body = Text()
```

View

```
<h1>  
    {{ post.title }}  
</h1>  
<p>  
    by {{ post.author }}  
</p>  
{{ post.body }}
```

Models: Python classes mapped to the Database

ORM: Object-relational mapping.

```
class BlogPost(models.Model):
    title = models.CharField(max_length=255)
    body = models.TextField()
    author = models.ForeignKey(User)
```

```
user = User.objects.get(username='victor')
post = BlogPost(title='Hello world', body='...', author=user)
post.save()
my_posts = BlogPost.objects.filter(author=user)
```

Controller

URL dispatcher

```
urlpatterns = [
    path('posts/', views.all_posts),
    path('posts/page/<int:page_number>/', views.posts_page),
    path('posts/<author>/', views.posts_author),
]
```

Views: functions that handle requests

```
def posts_author(request, author):
    posts = BlogPost.objects.filter(author=author)
    return render(request, 'posts/list.html', {'posts': posts})
```

Templates (View in MVC)

- Basic template language with almost no logic.
- Composable ⇒ Consistent
- Don't Repeat Yourself (DRY)

base.html

```
<html>
  <head>...</head>
  <body>
    {% block content %}{% endblock %}
  </body>
</html>
```

posts/list.html

```
{% extends 'base.html' %}

{% block content %}

  {% for post in posts %}
    {{ post.title }}

    {% if post.author %}
      {{ author.username }}
    {% else %}
      Anonymous
    {% endif %}
  {% endfor %}

  {% endblock %}
```

Features (I)

- **Local development.**

- Debug server.
- Development database in repository: SQLite.

- **Testing**

- Easy to develop Unit Tests on models.
- Test client: dummy Web browser.

- **Caching**

- Many backends (Database, Files, Memcached, Redis...)
- Per site, per view, template fragments, low-level...

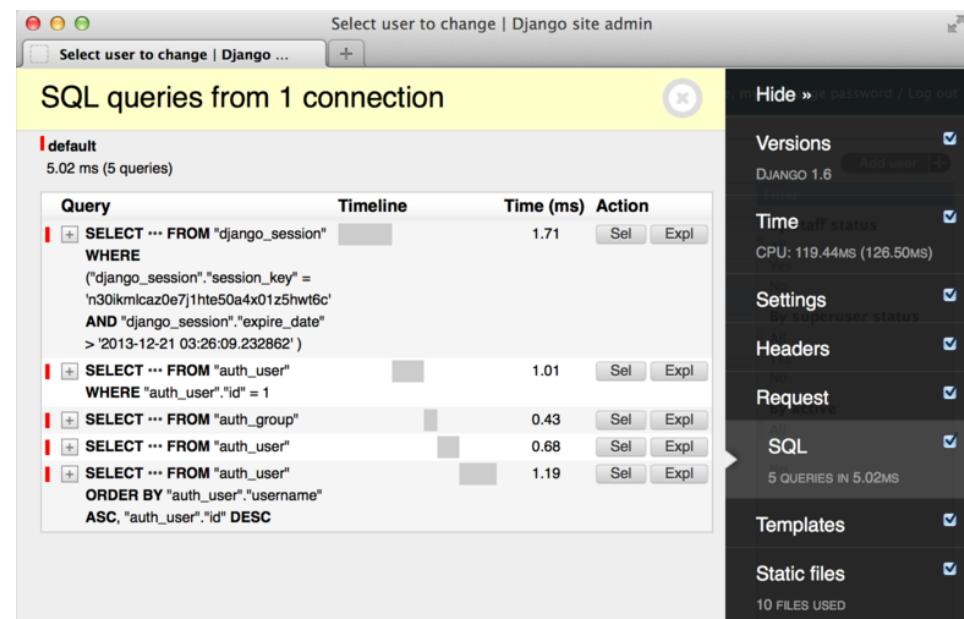
Features (II)

- **Internationalization and localization**

- Translate strings in templates and Python code.
- Localize formats (dates, numbers...).
- Timezone management.

- **Debugging**

- SQL queries
- Template generation



Features (III)

- Admin interface to Create / Edit / Delete objects.

Select user to change

Action:	-----	Go	0 of 3 selected		
<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	adrian	adrian@example.com	Adrian	Holovaty	✗
<input type="checkbox"/>	jacob	jacob@example.com	Jacob	Kaplan-Moss	✓
<input type="checkbox"/>	simon	simon@example.com	Simon	Willison	✗

3 users

FILTER

By staff status

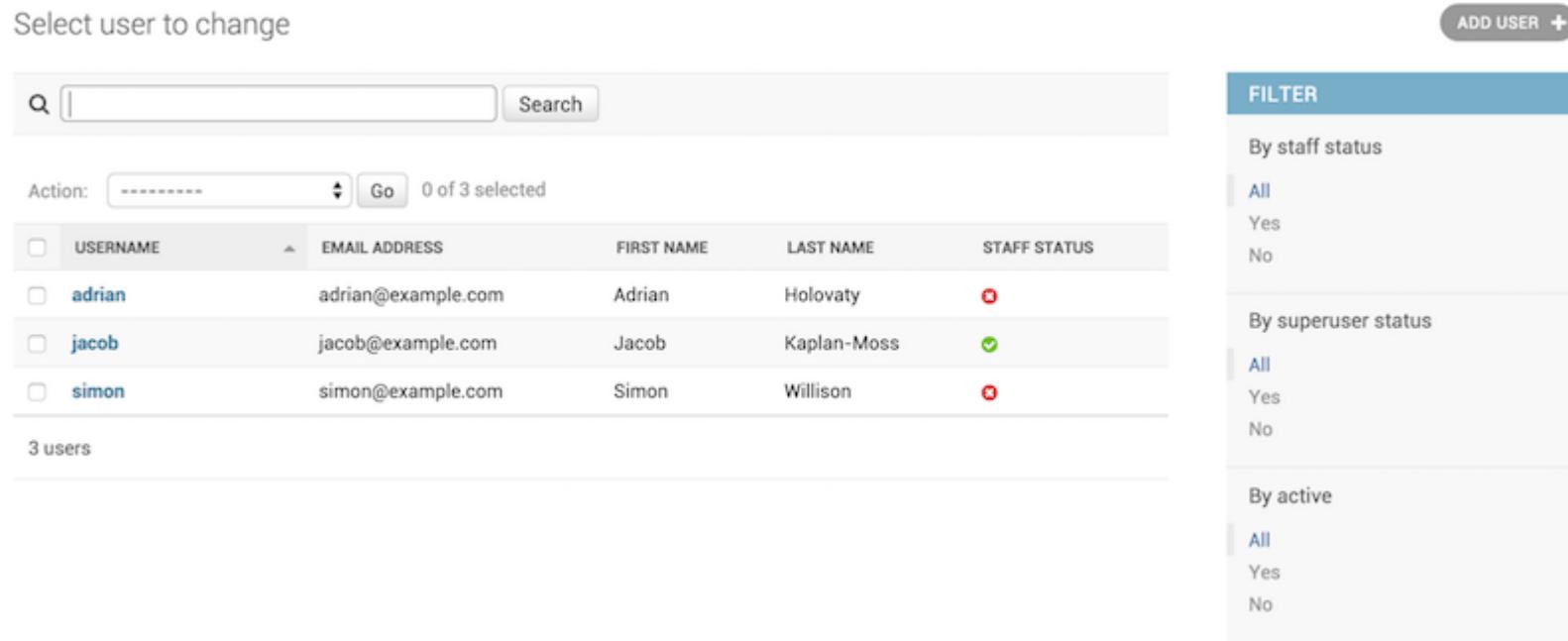
All
Yes
No

By superuser status

All
Yes
No

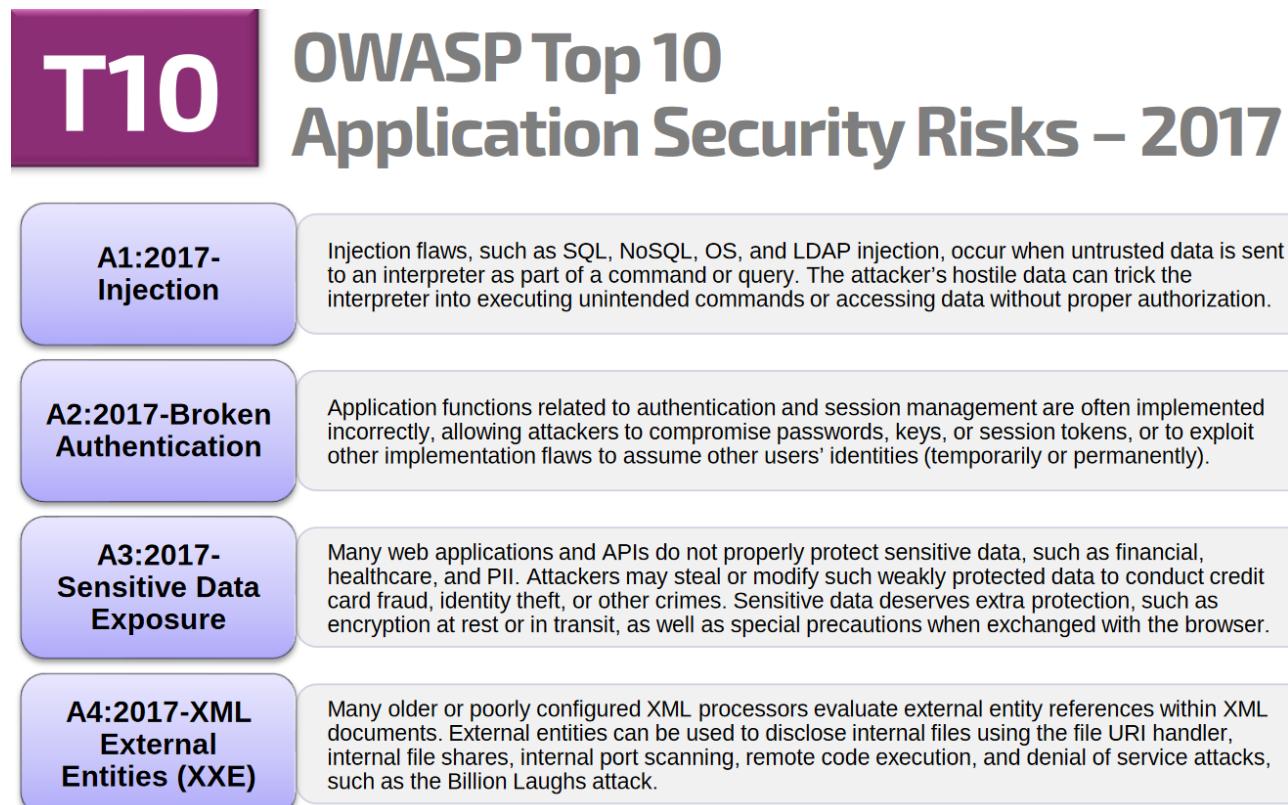
By active

All
Yes
No



Features (IV)

- Security



The slide displays the OWASP Top 10 Application Security Risks for 2017. It features a purple header bar with the text "T10" and "OWASP Top 10 Application Security Risks – 2017". Below this, four risk categories are listed in blue boxes, each with a brief description:

- A1:2017-
Injection**
Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
- A2:2017-Broken
Authentication**
Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities (temporarily or permanently).
- A3:2017-
Sensitive Data
Exposure**
Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection, such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.
- A4:2017-XML
External
Entities (XXE)**
Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks, such as the Billion Laughs attack.

OWASP security flaw #1: Injection

Login

```
SELECT * FROM Users WHERE Name = "username" AND Pass ="password"
```

Login

```
SELECT * FROM Users WHERE Name ="" or ""="" AND Pass ="" or ""=""
```

- ❖ SQL injection can't happen if there is no SQL.

OWASP security flaw #2: Broken Authentication

Authentication should:

- Store hashed passwords (strong hash, many iterations).
 - Sign session cookies.
 - Confirm e-mail addresses, but not send passwords.
 - ...
- ✓ All included in the Django built-in Authentication system.

OWASP security flaw #7: Cross-Site Scripting (XSS)

Inject JavaScript in pages seen by other users.

Comment

```
Hello!  
<script>alert("XSS");</script>
```

- ✓ Django escapes all variables in templates.

```
{{ comment }}
```

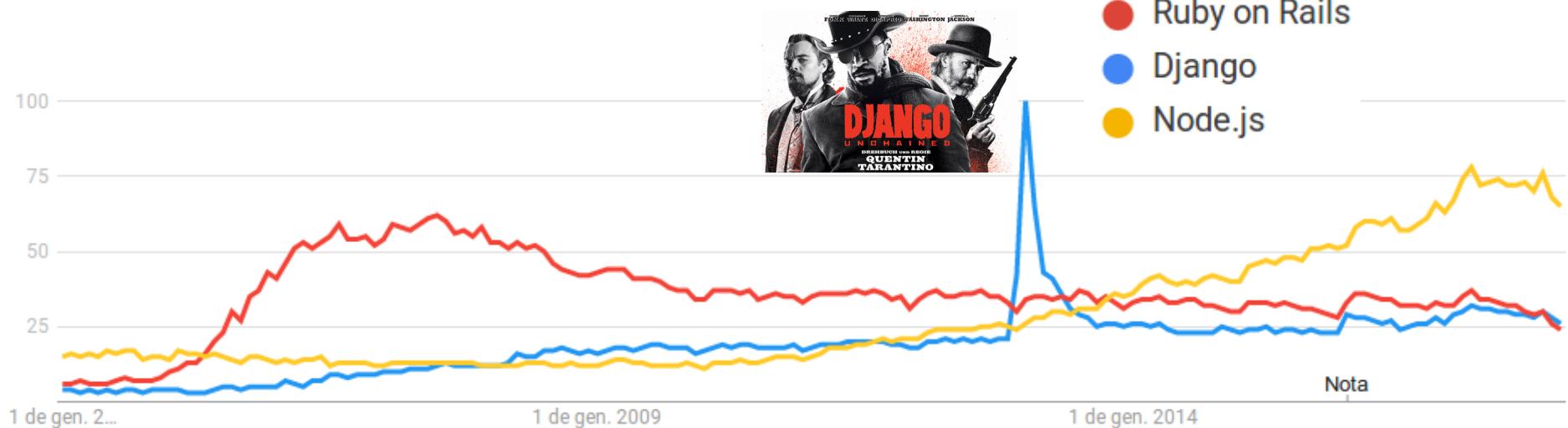
What did we want for our web app?

- ✓ Correct: testing.
- ✓ Understandable: standard code organization.
- ✓ Extensible: separated apps, MVC.
- ✓ Maintainable: admin interface.
- ✓ Reusable: separated apps.
- ✓ Secure: built-in authentication and security measures.

Compromise:

Get things done vs. Learning the framework

NodeJS: what's all the fuzz about?



NodeJS: Pros and Cons



Pros:

JavaScript: same language for front end and back end.

Fast execution thanks to V8.

Convenient execution model: event loop.

Fast adoption of new technologies, e. g. WebSockets.

Cons:

Not as mature ecosystem: lots of packages of dubious quality.

Front end development

- Web app development doesn't end in the back end.
- Components.
- Front end frameworks.

Front end components



Bootstrap

Before Bootstrap: most of the web was ugly.

After Bootstrap: most of the web is Bootstrap.



Heading

Etiam porta sem malesuada magna mollis euismod. Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Aenean eu leo quam. Pellentesque ornare sem lacinia quam venenatis vestibulum. Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit.

[View details »](#)

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

[View details »](#)

© Company 2011

Bootstrap templates



Font Awesome

In 2016, got \$1M in Kickstarter to develop version 5.



Google Fonts

Google Fonts

Viewing 804 of 804 font families

Open Sans
Steve Matteson (10 styles)

I think you're a
rockstar and
you have what
it takes.

Slabo 27px
John Hudson (1 style)

I think you're a
rockstar and
you have what it
takes.

Hind Guntur
Indian Type Foundry (5 styles)

I think you're a
rockstar and you
have what it
takes.

Lato
Łukasz Dziedzic (10 styles)

I think you're a
rockstar and
you have what it
takes.

Denk One
Sorkin Type (1 style)

Cust... ▾ | Regular 400 | 40px
**I think you're a
rockstar and
you have what it
takes.**

Taviraj
Cadson Demak (18 styles)

I think you're a
rockstar and
you have what
it takes.

DIRECTORY

FEATURED

ABOUT



Search

Categories

- Serif
- Sans Serif
- Display
- Handwriting
- Monospace

Sorting

Trending ▾

Languages

All Languages ▾

Number of styles



Thickness



Slant



Width



Roboto
Christian Robertson (12 styles)

I think you're a
rockstar and
you have what it
takes.

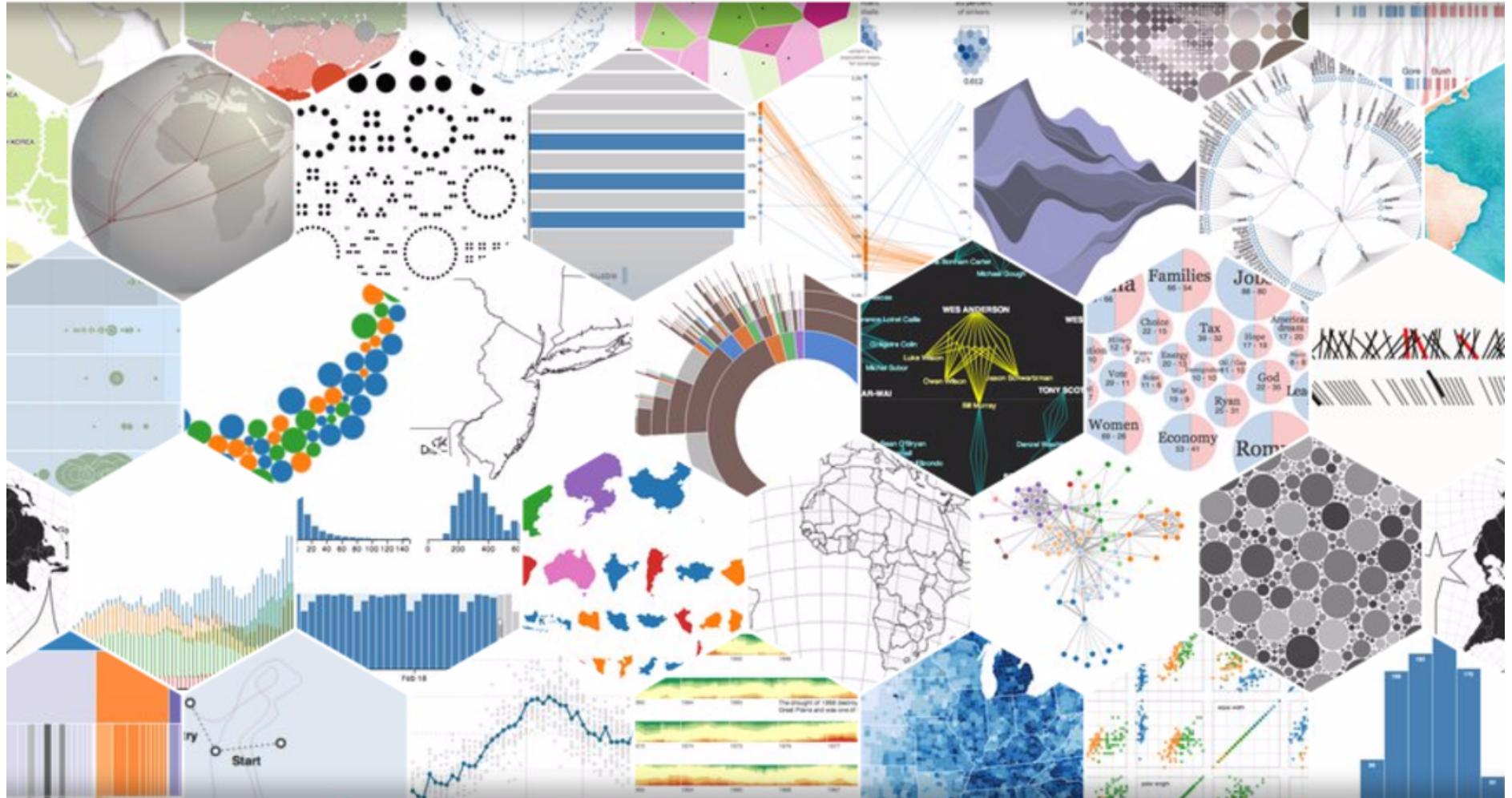
Oswald
Vernon Adams (3 styles)

**I think you're a
rockstar and you
have what it
takes.**

Chart.js

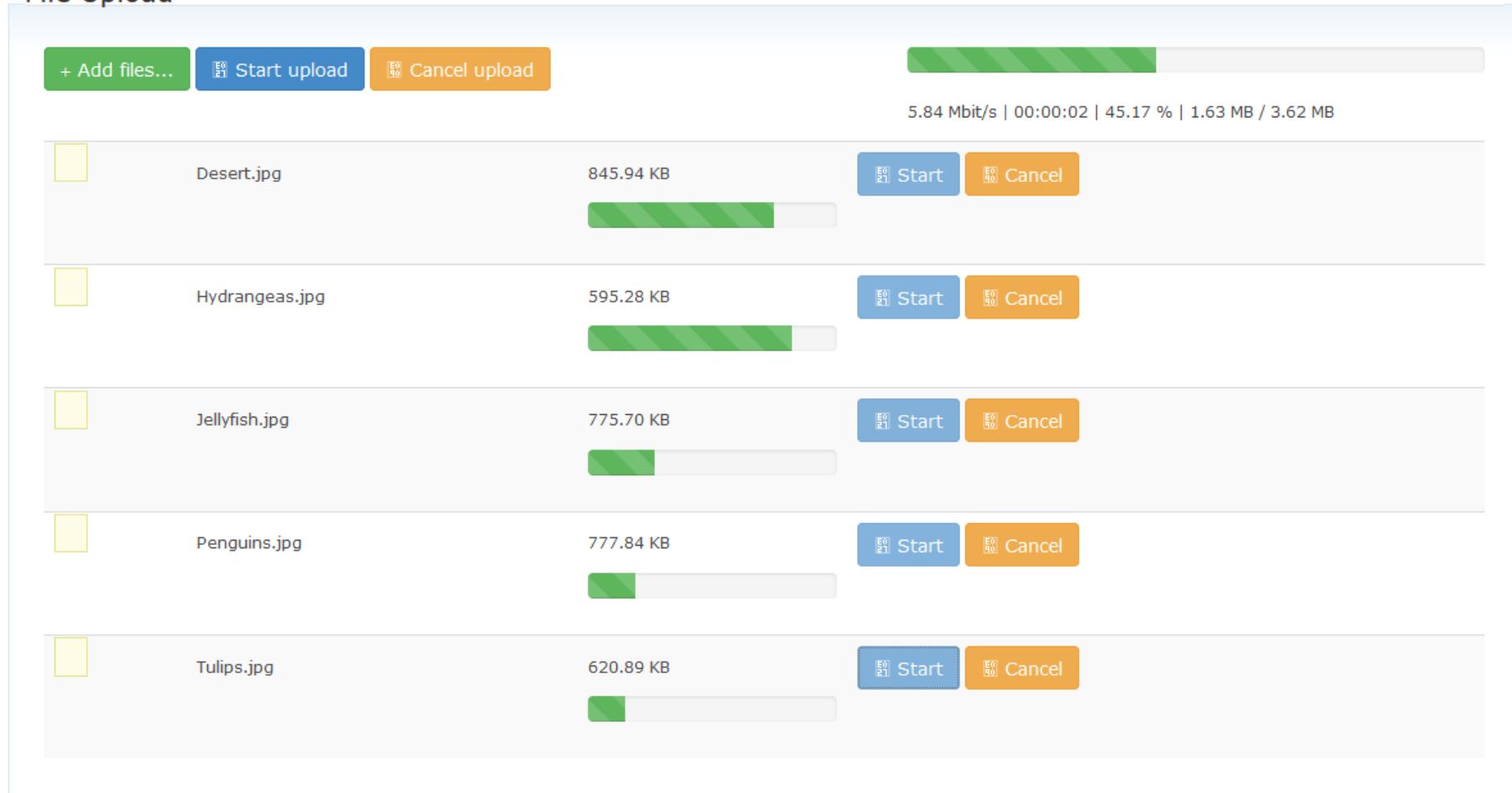


D3.js

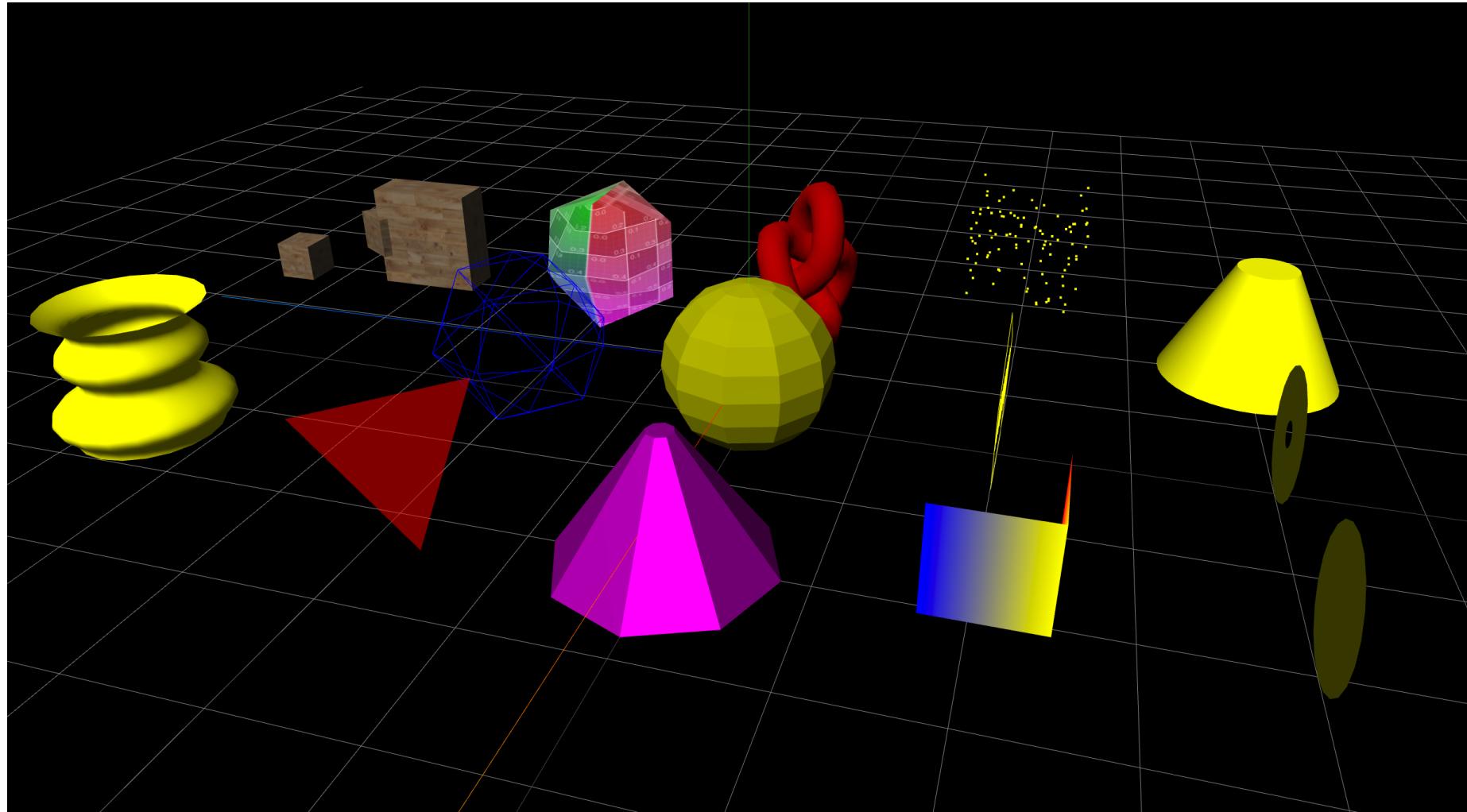


Jquery file upload

File Upload



Three.js



Animate.css

Animate.css

Just-add-water CSS animations

tada

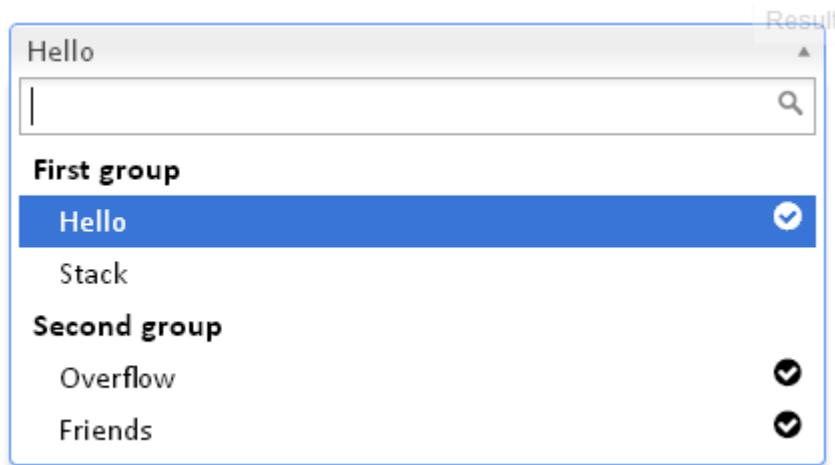
Animate it

[Download Animate.css](#) or [View on GitHub](#)

Another thing from [Daniel Eden](#).

Select2

Customizable select box: searching, tagging...



MathJax / Katex

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

$$f(a) = \frac{1}{2\pi i} \oint \frac{f(z)}{z-a} dz$$

$$\cos(\theta + \varphi) = \cos(\theta)\cos(\varphi) - \sin(\theta)\sin(\varphi)$$

$$\int_D (\nabla \cdot F) dV = \int_{\partial D} F \cdot n dS$$

$$\vec{\nabla} \times \vec{F} = \left(\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z} \right) \mathbf{i} + \left(\frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x} \right) \mathbf{j} + \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \right) \mathbf{k}$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

$$(\nabla_X Y)^k = X^i (\nabla_i Y)^k = X^i \left(\frac{\partial Y^k}{\partial x^i} + \Gamma_{im}^k Y^m \right)$$

Leaflet



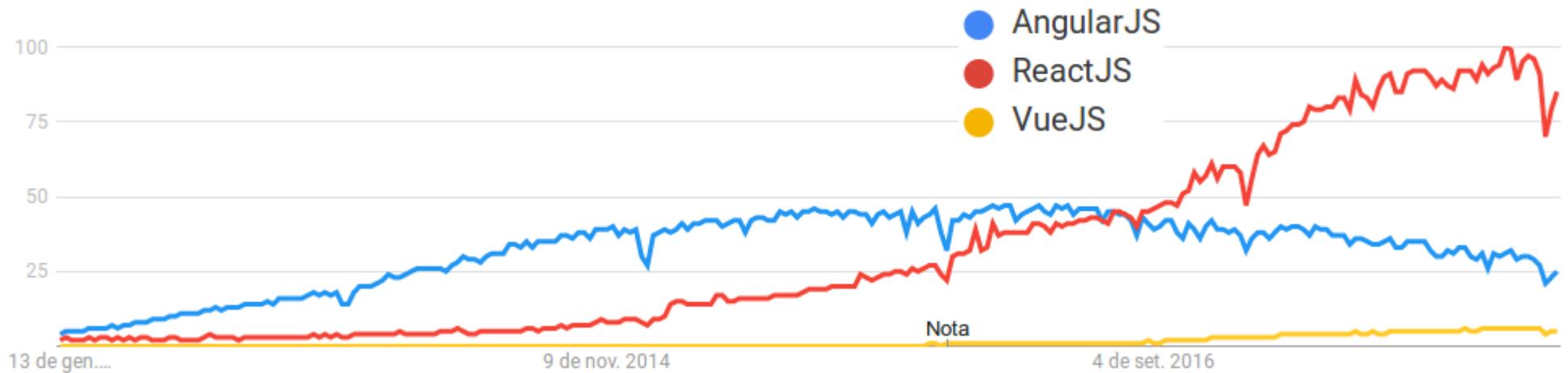
Front end frameworks

- Big Single Page Applications.
- Follow good practices in the Front end.

Front end frameworks

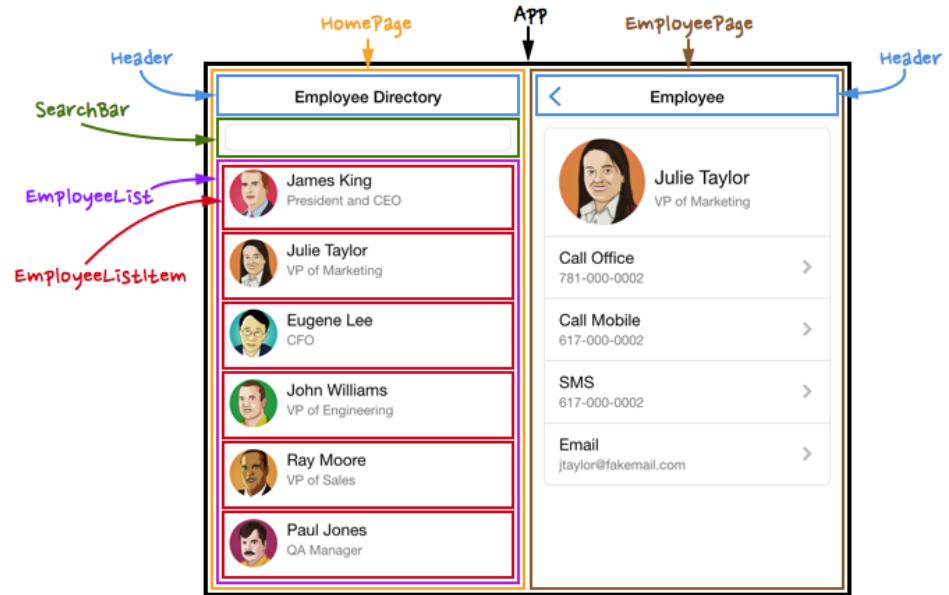
- **Big players:**

- AngularJS (Google)
- ReactJS (Facebook)
- VueJS (crowdsourcing)

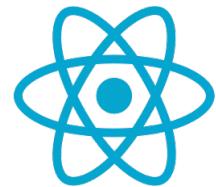


ReactJS

- Composable components
- Virtual DOM
- Redux: State ⇒ View



- React Native: build iOS and Android native apps.



React Native

Conclusions

- In the back end: frameworks helps us.
- In the front end: lots of components available, also frameworks.

Bottom line:

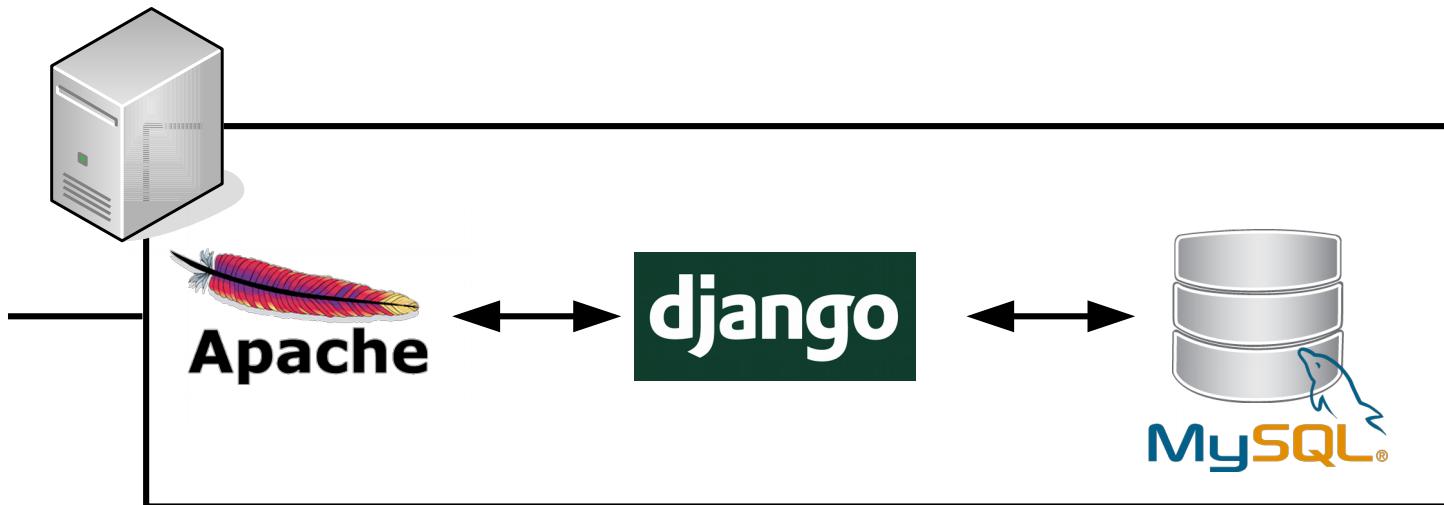
- Tools allow us to develop great apps with less effort.
- Compromise between learning new tools and getting things done.

3. Deploying web applications

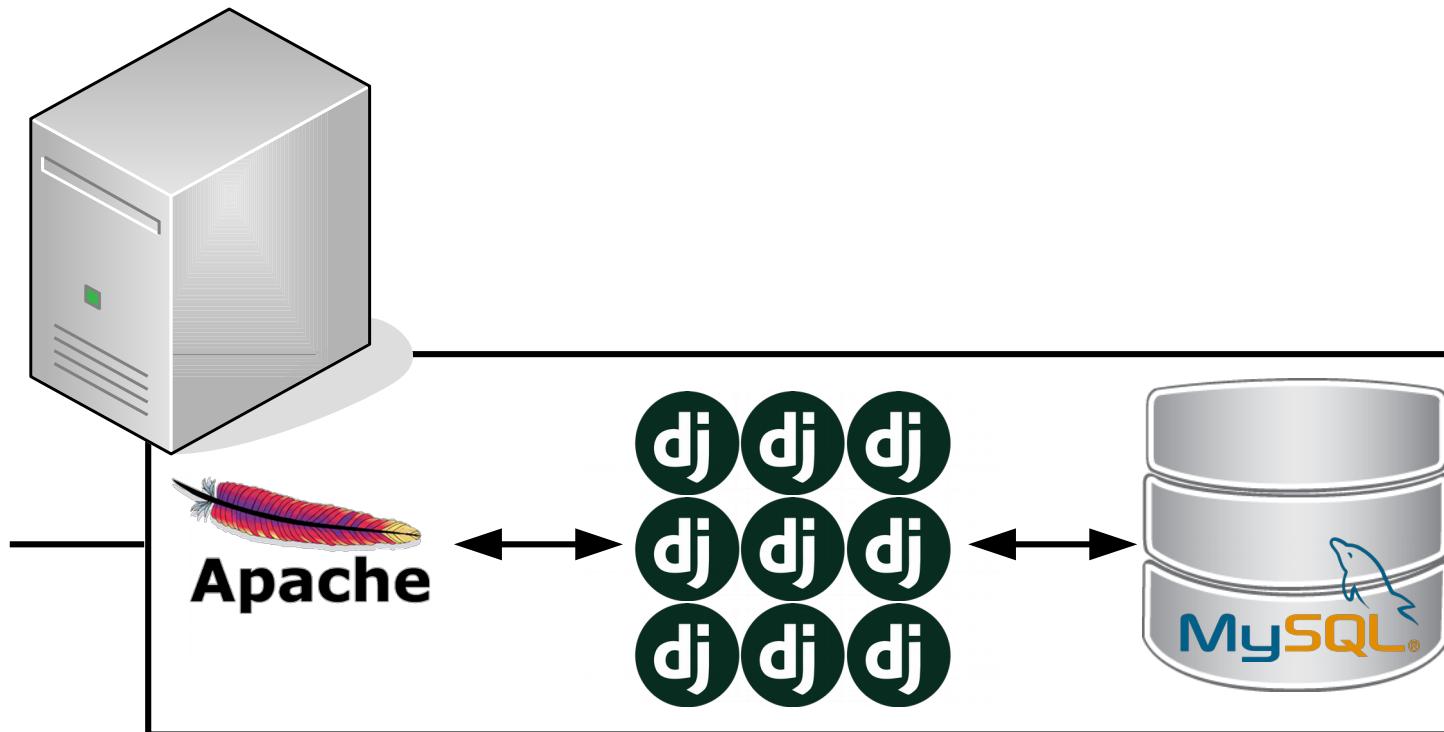
After developing, we need to deploy!

- Servers
- The cloud
- Some useful services

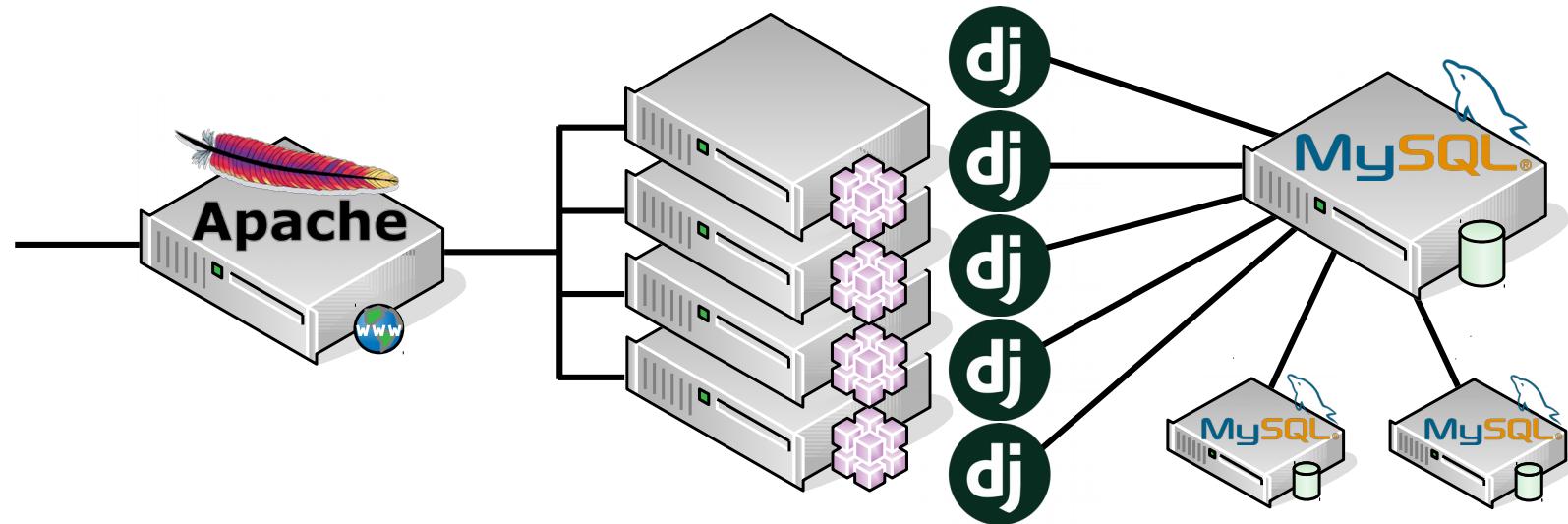
Server: how can we scale?



Bigger server



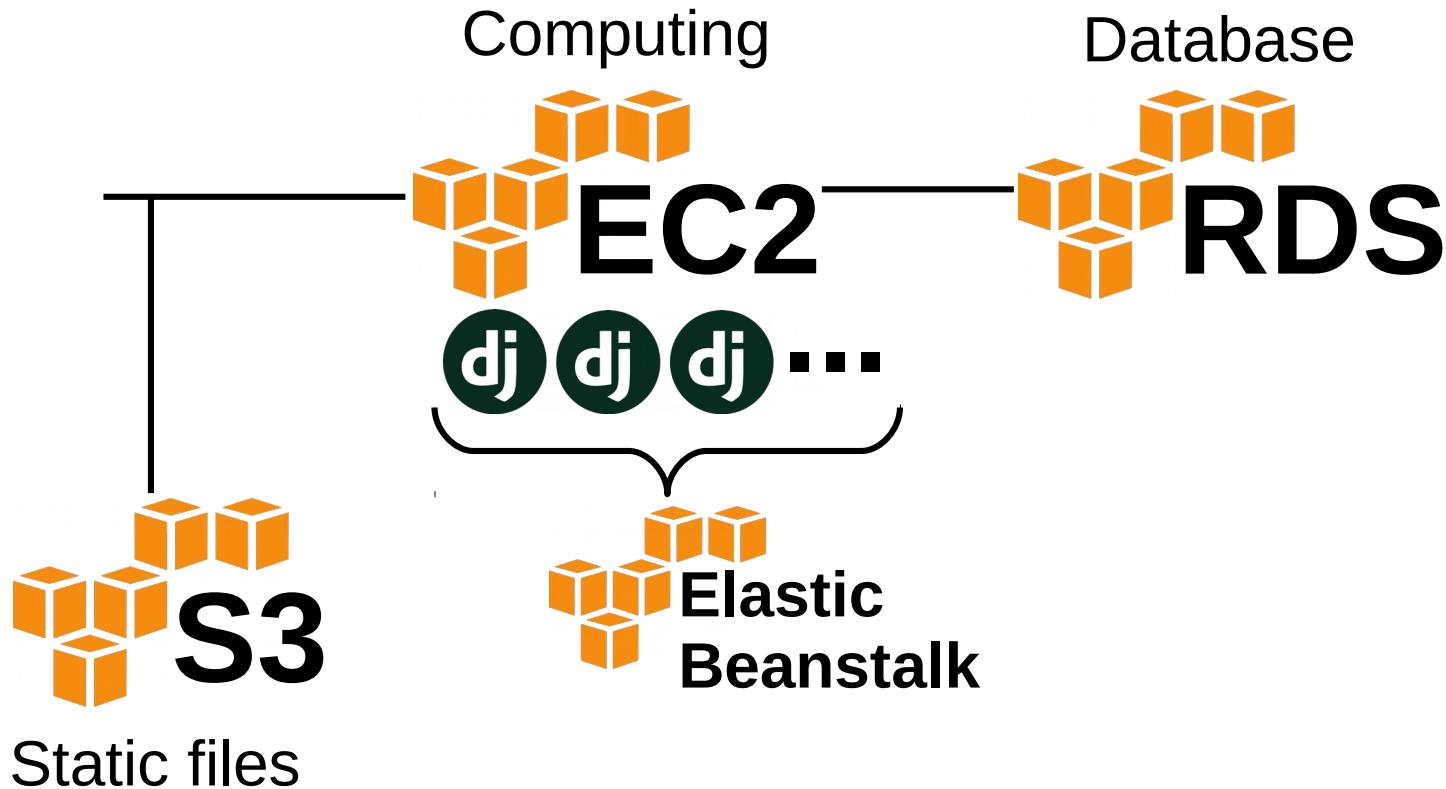
More servers!



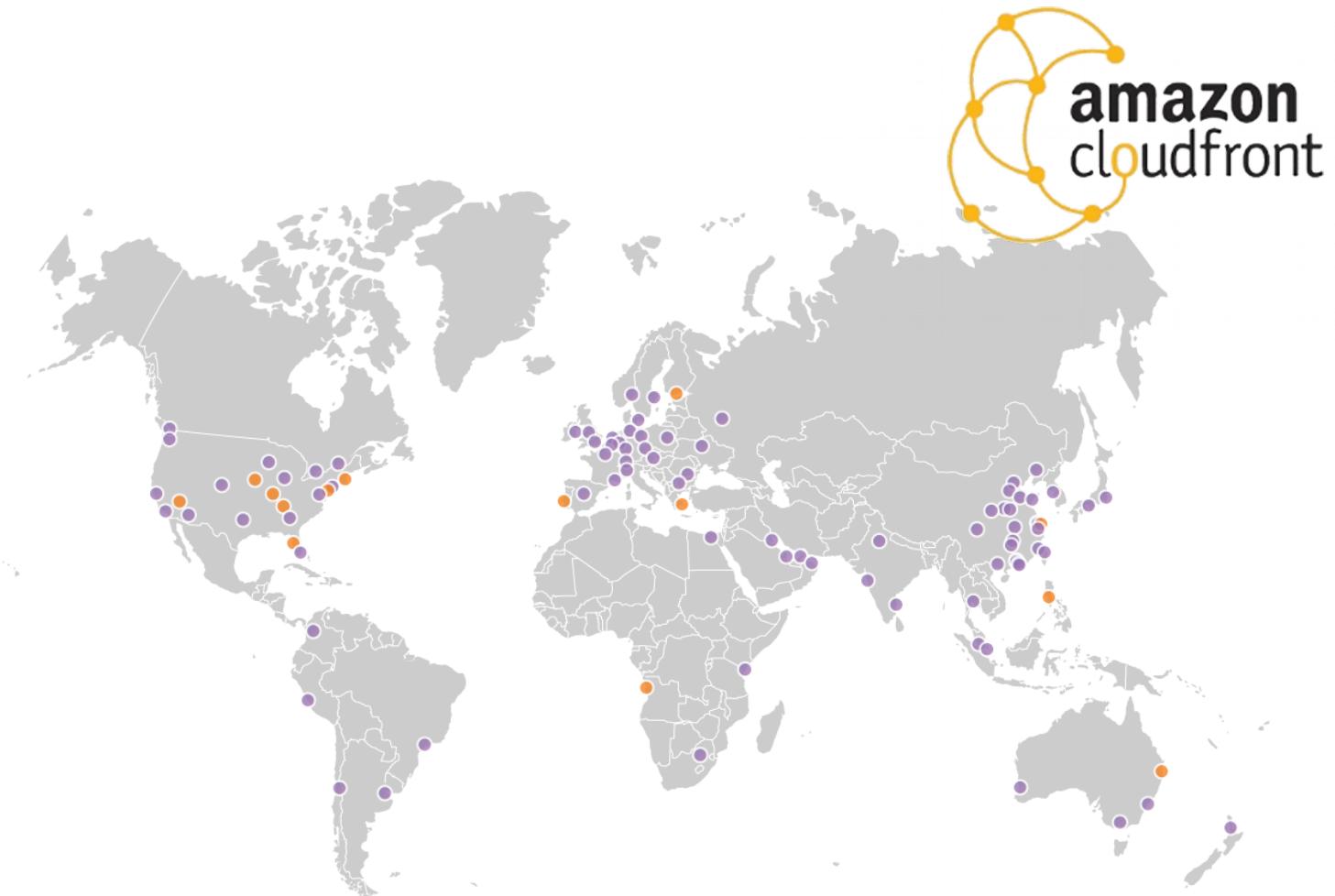
The cloud



Amazon Web Services (AWS)

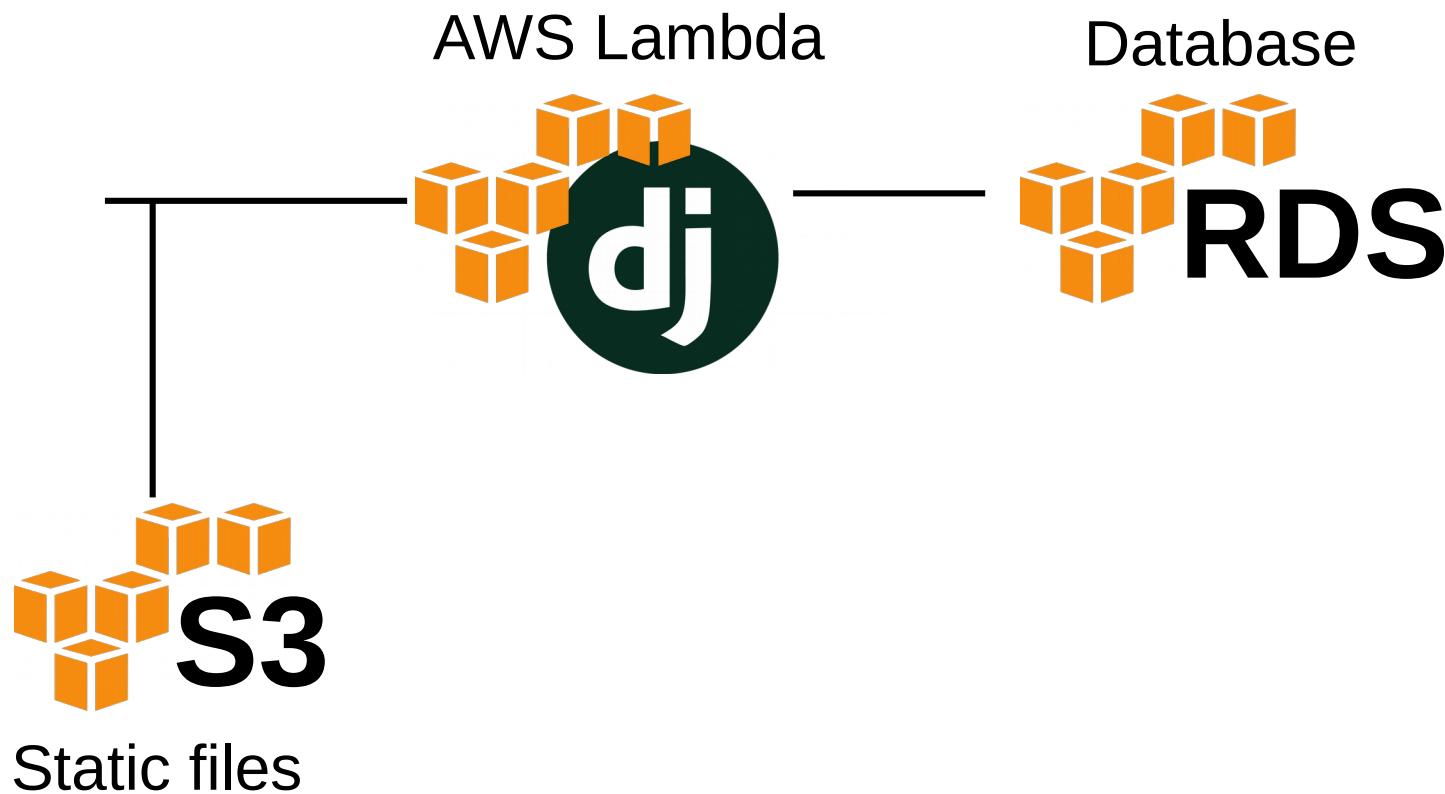


Content Delivery Networks



Towards Serverless architectures

Code in the Cloud: No servers.

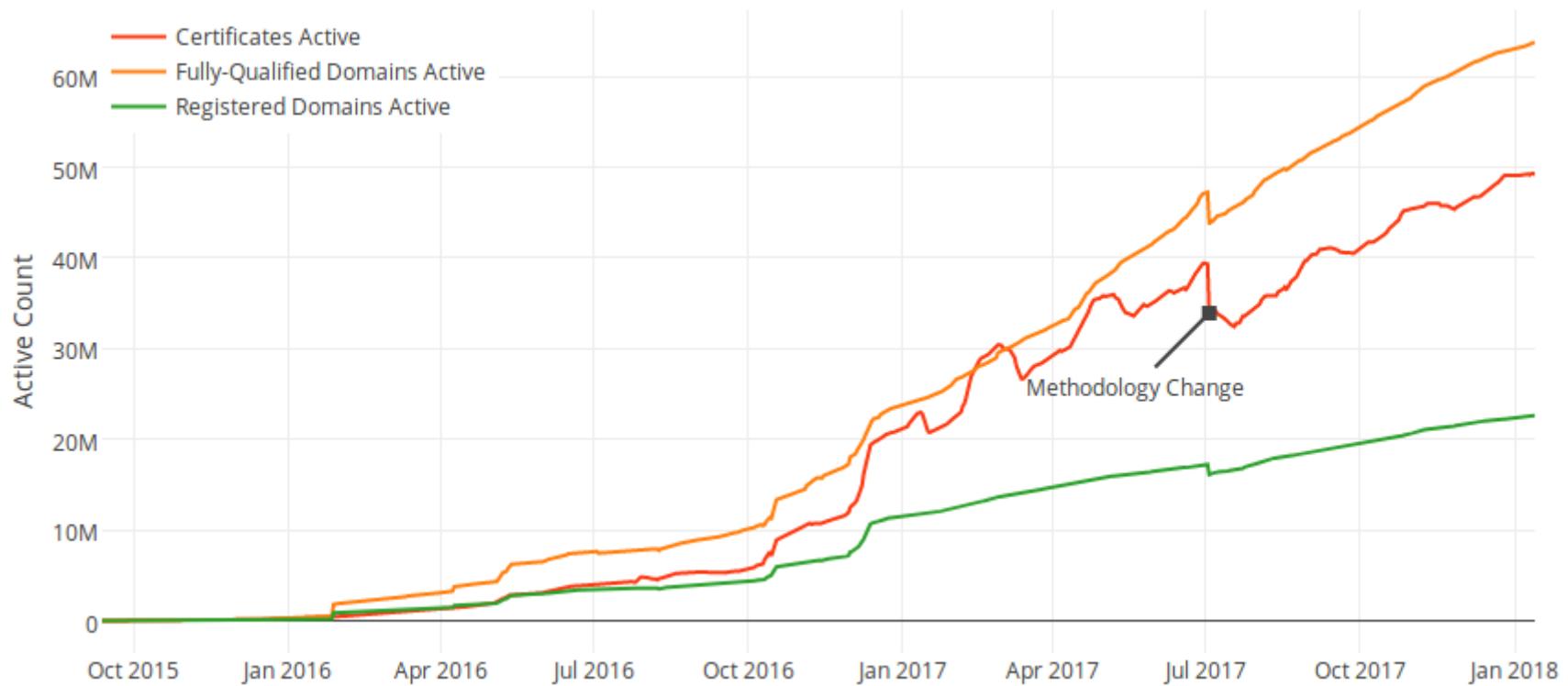


Useful services

Some services that help us in the deployment phase.

Let's Encrypt

Free SSL/TLS Certificates



Google PageSpeed Insights

PageSpeed Insights [G+1](#)

[ANALYZE](#)

! Mobile ! Desktop

63 / 100 Suggestions Summary

! Should Fix:

- Leverage browser caching
 - › [Show how to fix](#)
- Eliminate render-blocking JavaScript and CSS in above-the-fold content
 - › [Show how to fix](#)
- Enable compression
 - › [Show how to fix](#)
- Minify JavaScript
 - › [Show how to fix](#)

✓ 5 Passed Rules

- › [Show details](#)

! Mobile ✓ Desktop

88 / 100 Suggestions Summary

! Consider Fixing:

- Eliminate render-blocking JavaScript and CSS in above-the-fold content
 - › [Show how to fix](#)
- Leverage browser caching
 - › [Show how to fix](#)

✓ 8 Passed Rules

- › [Show details](#)

Visit Analytics (Google Analytics, Matomo)

PIWIK

Dashboard | All Websites | Widgets | API | Email & SMS Reports | Give us Feedback! | English ▾ Hello, Piwik team! | Settings | Sign out

Website PIWIK FORUMS ▾

Dashboard Visitors Actions Referrers Goals

Dashboard of team Live Overview Dashboard empty

DATE RANGE: 2013-12-17 ALL VISITS WIDGETS & DASHBOARD

YOU ARE CURRENTLY VIEWING THE DEMO OF PIWIK

Visits Over Time

Keywords

KEYWORD	VISITS	UNIQUE VISITORS	ACTION
g Keyword not defined - Goo...	1099	1047	2200
g .htaccess restrict access...	1	1	1
g /forum - Google	1	1	4
g alle download anzeigen - ...	1	1	3
g canada birth certificate ...	1	1	2
g can direct downloads be t...	1	1	1
r child pid 29168 exit sign...	1	1	1
g class 'piwik\log' not fou...	1	1	10
g cron.daily send email err...	1	1	1
g cron is real time - Googl...	1	1	1
g email tracking open sourc...	1	1	1
g fake drivers license swed...	1	1	2
g fake id card - Google	1	1	1
g fake south korea passport...	1	1	1
g htaccess authentication a...	1	1	1
v! imagevenue peesian - Yah...	1	1	1
g impossible d'afficher la ...	1	1	1
g list of arialunicid0 char...	1	1	1
g phpinfo segmentation faul...	1	1	1

Visitor Map

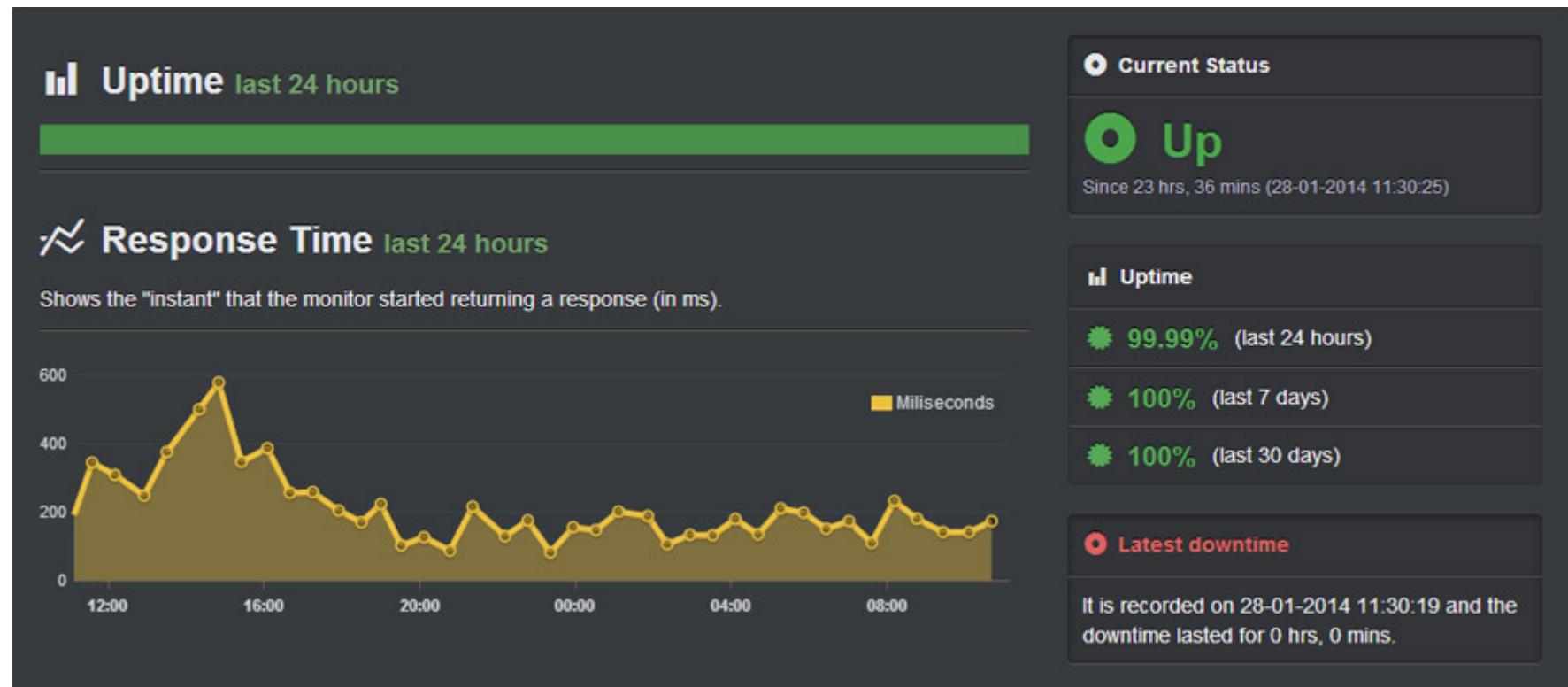
1.6k visits

World-Wide Countries Visits

Real-time Map

15:52:34 Geo-located visits of last 48 minutes

Uptime Robot



Conclusions

- Deployment is important.
- If your budget allows it, you can scale as much as you want.

Example:

Instagram, when sold to Facebook for \$1B in 2012,
only had 6 engineers, and it was built with Django.

4. Conclusions

- 1. The Web is very powerful and an evolving technology.**
- 2. Frameworks help us follow development good practices and get more done with less effort.**
- 3. We can even scale our apps!**