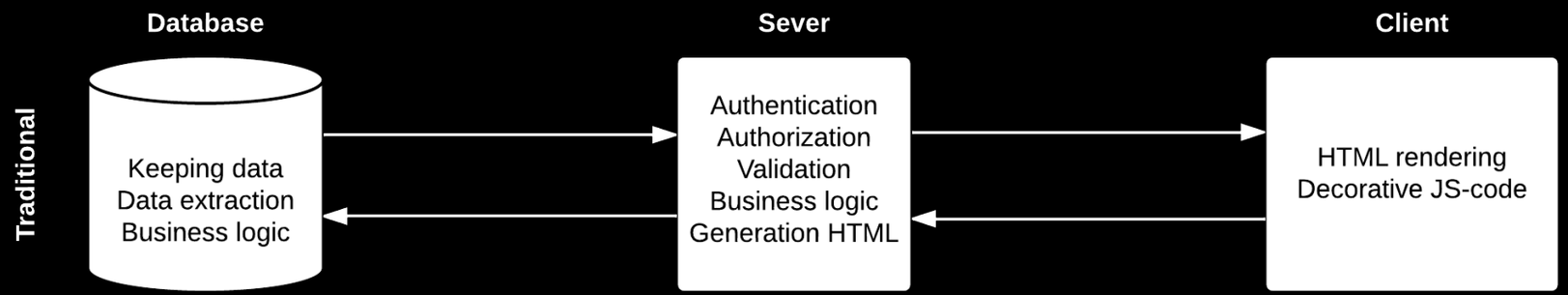
Single Page Application vs Traditional Website

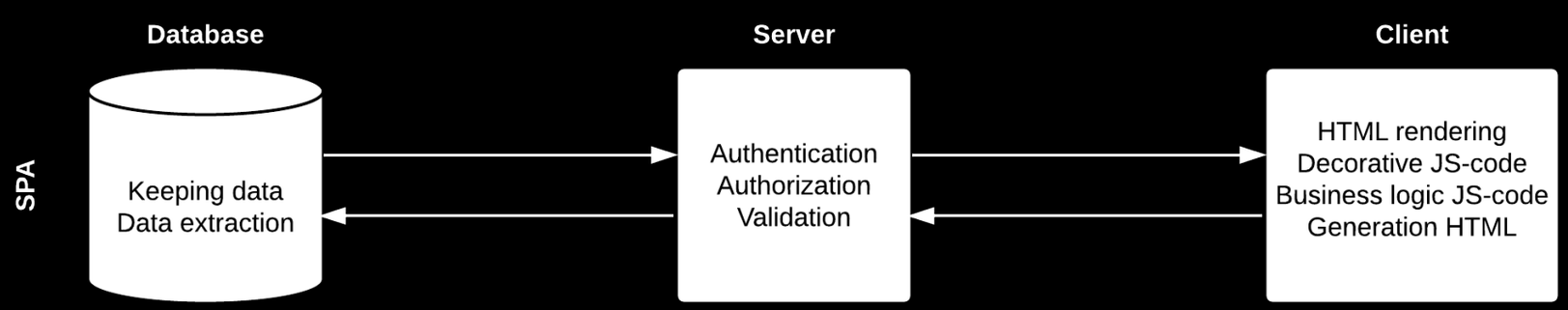
Eugene Nikanovich
zhenyanikon@gmail.com
+48 733-864-902

Single page application (SPA) - is a web application (website), which loads all resources what needed to work with the site, together with the first page. In the future, any desired content is downloaded dynamically. This app frequently updates the URL in the address bar of the browser, simulating transitions between pages.

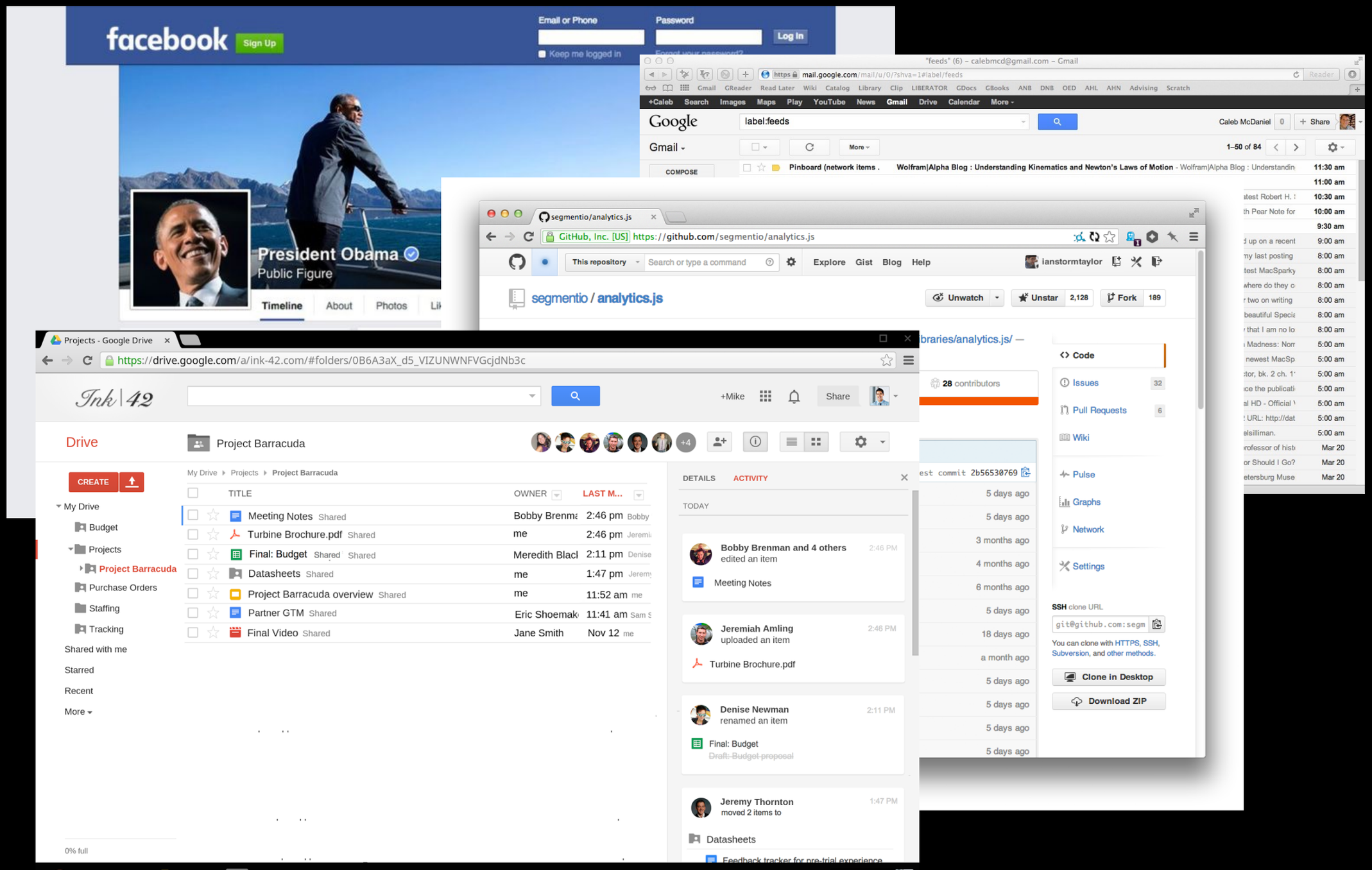
In **SPA** can be used any server side technologies cause big part of web application moves to browser.

Requirements to the server less than in **traditional** website





Give me an example!



Why well-written SPA convenient for user?

- SPA is rendered as personal application
- SPA can react like a personal application
- SPA can notify user about his state like a personal application
- SPA like a site, almost always available
- SPA like a site, working on different platforms

- SPA is rendered as personal application

SPA re-render just those parts of the interface which were changed and only when it necessary.

Traditional site re-render all page in response to users actions which result is delay and ‘blinking’. Because browser should get a page in response from server and render this page
If page is huge, server is busy or connection to internet is slow then it may take a couple of seconds.

- SPA can respond like a personal application

SPA is minimizing time to reaction due to the fact what move working (temporary) data and part of processing from server to browser.
SPA has a data and business-logic what necessary for accepting most of solutions locally - what means quickly
Only user authentication, validation and permanent data keeping should be on the server.

In case of **Traditional** site large part of logic is located at the server. so to get a response for actions - user should waiting when server will finish request-response-render loop. It can take a few seconds.

- SPA can notify user about his state like a personal application

If **SPA** still has to wait for a response from the server, it can be dynamically update the progress bar to communicate with user.

When working with a **Traditional** site the user is forced to guess when the page loading is finished and it will be possible to continue

- SPA like a site, almost always available

In difference of many personal applications, user can use **SPA**, having just internet connection and modern browser. Now all of these we have in smartphones, tablets, TV's, laptops and desktop computers.

- SPA like a site, working on different platforms

Well-written **SPA** can working in all operating systems where have a modern browser with supporting HTML5. Usually this feature considered the advantage for developer and it is no less important to the users which use more devices.
e.g. with Windows at work, with Mac at home, with Android at smartphone and tablet with Amazon OS.

There are many choices when it come to build SPA

let's consider tools for one case with

using AngularJS (1.x)

angularjs.org

Back-end Tools

Express Server Running on Node.js

Express is the most common choice for application framework when it comes to Node.js development. We can use it because it is very easy to get started and there are tons of support resources on the web.

nodejs.org

expressjs.com

Build Tool/Task-Runner

gulpjs.com

gulpjs.com

CSS

We can use **Sass** for writing the CSS. It is far more powerful than writing vanilla CSS as it provides useful abstractions to avoid repetition and save time.

sass-lang.com

Linting Tools

We can use it like a plugin in your IDE / text redactor

jshint.com

jslint.com

Testing Tools

Jasmine & Mocha

Jasmine and Mocha are the two most famous frameworks for writing unit tests in the Angular world.

jasmine.github.io

mochajs.org

Thanks for attention!

Prepared for Vistula Web Warsaw #1

Slides:

https://github.com/Yauheni-blr/spa-vs-traditional