ACADGILD

LEARN. DO. EARN

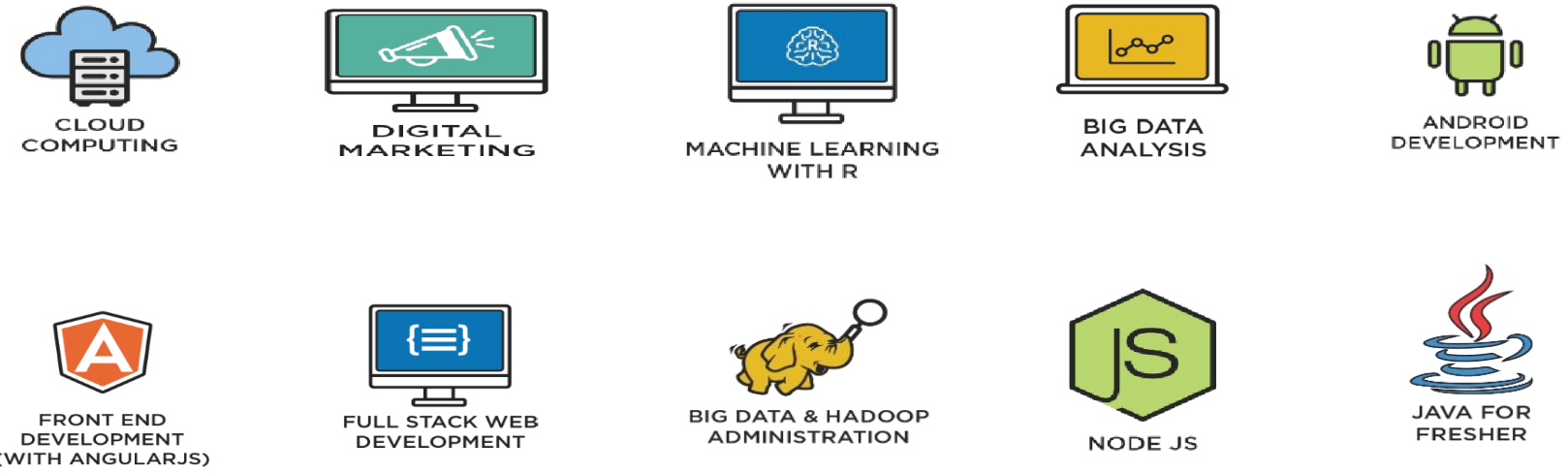
FRONTEND DEVELOPMENT

(WITH ANGULAR JS)

Brief Intro About AcadGild

ACADGILD

- **AcadGild** is a technology education start-up which provides online courses in latest technologies.



- **AcadGild** was started by IIT/IIM alumni.
- **Our aim** is to provide millions of high school graduates, college graduates and working professionals, skills to make them ready for jobs.

ACADGILD

Single Page Application & Angular 2

Session - 1

LEARN. DO. EARN

Agenda

ACADGILD

Sl No.	Agenda Title
1	Evolution of the Web
2	Introduction to Front-end Frameworks
3	What is Single Page Application?
4	How SPA Works?
5	Traditional websites
6	Misconception about SPA
7	What's new in SPA?

Sl No.	Agenda Title
8	Benefits of SPA
9	History of Angular
10	Problems with Angular 1.x
11	Introduction to Angular 2
12	Transpilers
13	Tech supporting ng2
14	Dev Environment Setup
15	Hello World Example – ng2

Evolution of the Web

ACADGILD

- **1993 - 1997:** Static HTML Websites, GIFs, Minimal Javascript and DHTML scripts, Macromedia Flash - Only CGI Scripts for backend
- **1995 - 2000:** Dynamic Webpages driven by backend PHP, JSP, Java & MySql, CSS being used for Styling, Tables used for Layouts, JS for Minimal scripting
- **2000 - 2008:** Content Management Systems like Wordpress, Drupal, Joomla etc., HTML4, Initial Ajax, Canvas, Web Fonts
- **2009-2013:** Rich Internet Applications, Client Server Architecture, CSS3, Animations, Ajax, Offline Web Apps, Responsive Design, Mobile Web
- **2014-15:** Single Page Apps, Pure Front End MVC Frameworks

Introduction to Front-end Frameworks

ACADGILD

- Today **Front-end Application** became more complex, with dozen of JS Files, plenty HTML files, images and huge CSS Files.

- **Front-end Frameworks** have following objectives:
  - Modularity
  - Built In features
  - Common Libs
  - Easier to maintain the code
  - Faster Development - Time to market is very essential and reiterations are required too quickly
  - Debugging is easier
  - Code Collaboration between multiple teams
  - Shorter learning curve for new developers
  - Supporting all concerns for Single Page Apps

Introduction to Front-end Frameworks (Contd.)

ACADGILD

- Popular **Front End Frameworks** :
  - AngularJs
  - ReactJs
  - NockoutJs
  - BackboneJs

What is Single Page Application?

ACADGILD

- **Single-Page Applications (SPAs)** are Web apps that load a single HTML page and dynamically update that page as the user interacts with the app.

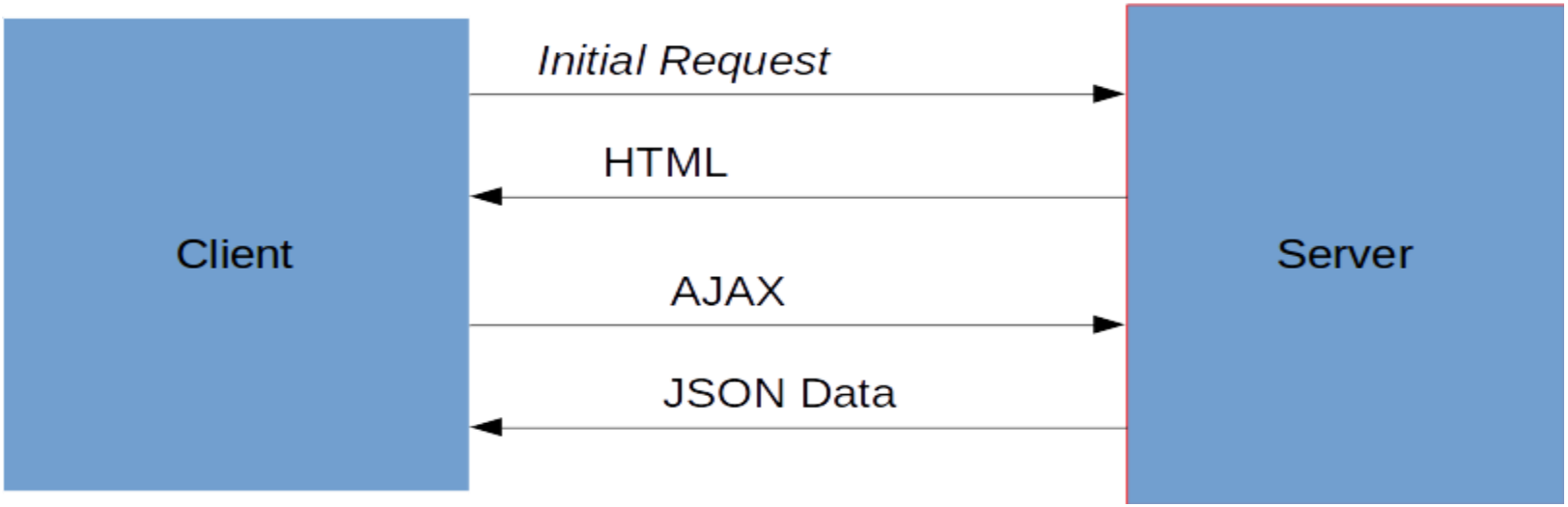
- A **single-page application (SPA)** is a web application or website that fits on a single web page with the goal of providing a more fluid user experience similar to a desktop application

How SPA Works?

ACADGILD

- **SPA Flow:**

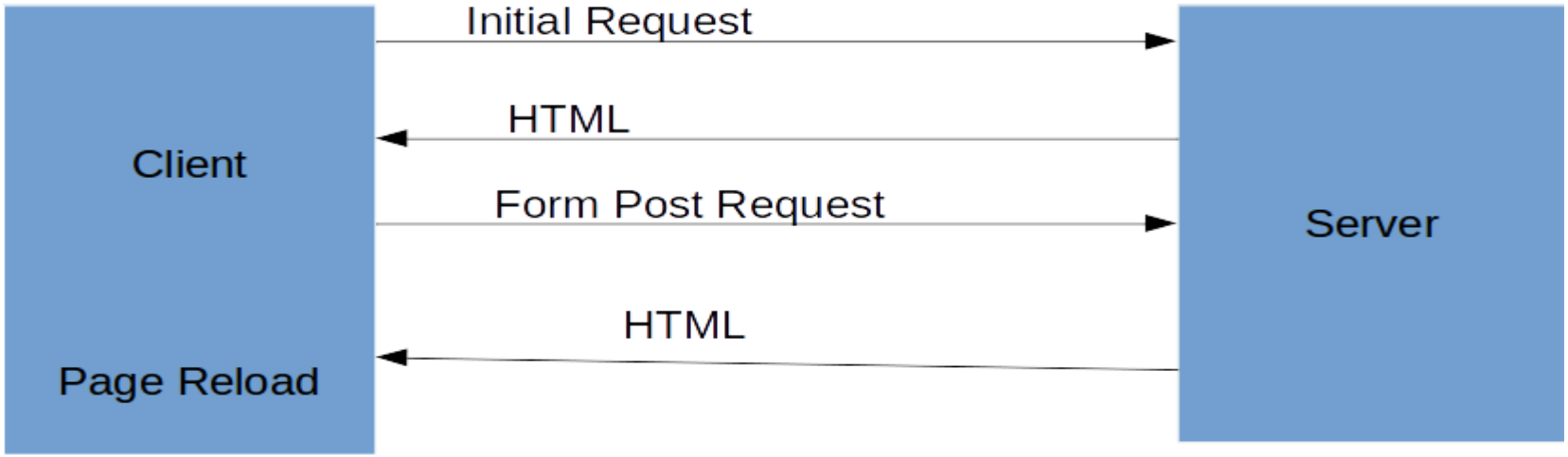
Figure#2: Single Page Application



Traditional Website

ACADGILD

- **How Traditional Website works:**



Figure#1: Traditional Web Application

Misconception about SPA

ACADGILD

- Lot of people have misconception about SPA, so let us know about SPA.
- The “page” in SPA is the single web page that the server sends to the browser when the application starts.
  - It’s the server rendered HTML that gets everything started. No more, no less.
  - After that initial page load, all of the presentation logic is on the client.
- Now point is: “what is single means”, so Single page is starting point for an application, where some specific content will display as a template, and then according to need, other template will load at specific place on same page. I mean here some part will rarely change like header(has common navigation), and footer(common for whole application), we can say those part as shell for application.
- And now “DATA”, which play main role for single page application. SPA requests data using javascript over HTTP from a server.

What's New in SPA?

ACADGILD

- Responsive design (support multiple platform)
- Load single page application
- More interactive
- Better user experience
- Use ajax and HTML5 to create responsible web apps without constant reloads
- Have the capability to perform DOM Manipulation
- Should be able to store Navigation History for views
- Capacity to manage Routing for views and nested views
- Capability to perform Ajax - fetch data asynchronously from servers
- Implement Templating for Views
- Capability for dynamic Data Binding
- Organising code into modules
- No Refresh of Page

Benefits of SPA

ACADGILD

- Faster Interaction
- SPA works and feels more like an application then a web page.
- It is easy to scale, and it is easy to cache resources.
- SPA can use caching and local storage effectively.
- SPA is fast, as most resources HTML + CSS + Scripts are only loaded once, throughout life span of application, only data is transmitted back and forth. (Reducing bandwidth usage is also a plus).

Principles of SPA:

- Renderer with javascript library
- Use css3
- Performance matters
- Front-end and back-end is decoupled

History of Angular

ACADGILD

- AngularJS was created, as a side project, in 2009 by two developers, **Misko Hevery** and **Adam Abrons**.
- Maintained by Misko Hevery & Brad Green it was used internally in Google for multiple projects like Double Click, Youtube for PS3,
- It was intended to be a **MVC** or **MVVM** framework.
- It is based on **Declarative** Programming with its directive constructs.
- Angular 1.x became very popular being used by large organisations like NBC, Intel, Sprint, Wolfram Alpha, Dell ABC News etc.
- Angular has a widespread developer community.
- Maximum amount of Jobs available in AngularJs after Javascript jobs.
- Widely available supporting Components/Libraries like Angular-UI Bootstrap, Supersonic, Ionic for Mobile, UI Grid.

Problems with Angular 1.x

ACADGILD

- AngularJS has become one of the most popular open source JavaScript frameworks in the world of web application development, but still it's having some problems:
  - Issues with Scope & child scopes
  - Issue with digest cycles: $scope.apply();
  - Two way data binding: Max of 2000 bindings
  - Modules: No built-in module loading
  - No inheritance supported for module
  - JqLite: No direct modification of dom?? dropped in 2.0
  - Different ways to create different things

- Why needs modification:
  - To knock off many old things(components)
    - $scope, $rootScope, services, controllers, modules, jqLite etc.
  - To leverage latest ECMA Script standards.
  - To leverage modern browser capabilities.
  - Improve many areas with different approaches with past expirence.

Introduction to Angular 2

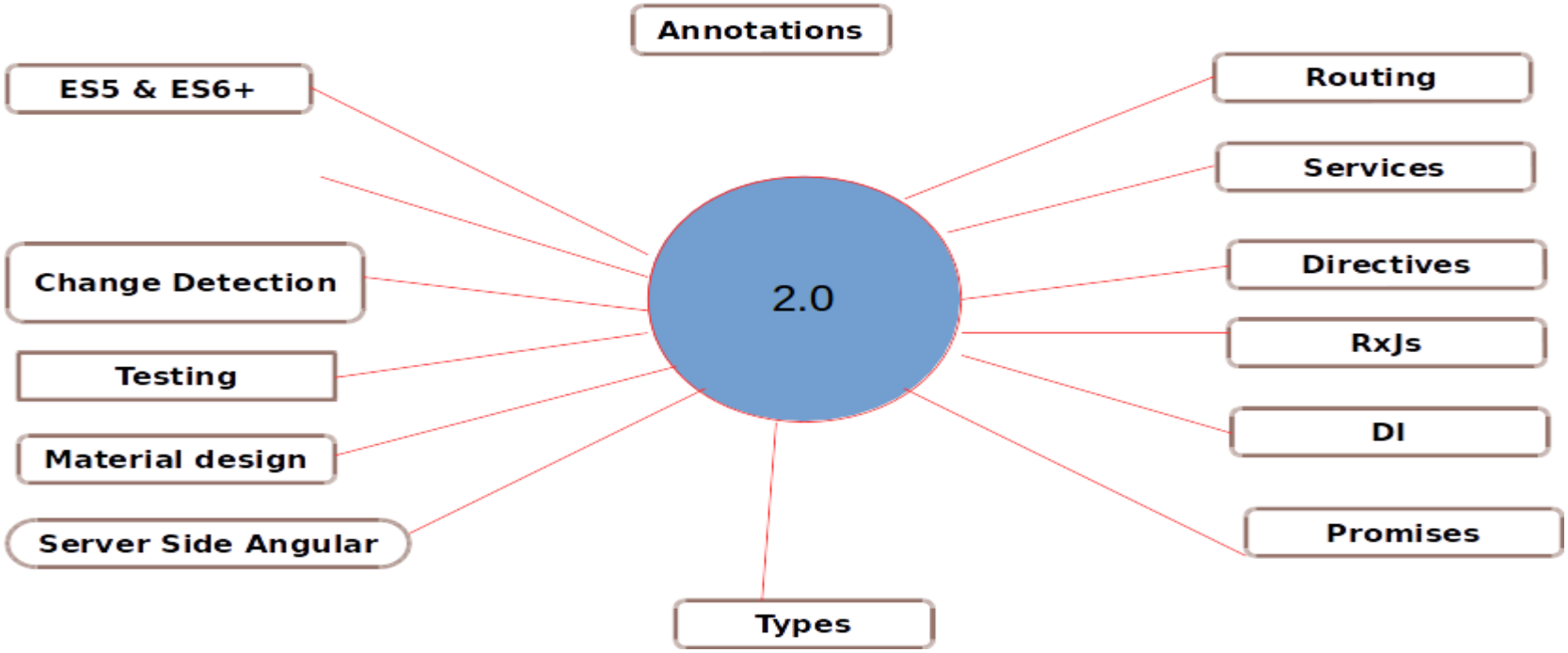
ACADGILD

- **Angular 1.5.x** is the latest stable release of Angular 1
- **Angular 2** is a complete overhaul of Angular framework with major core architectural changes to solve the problems of Angular 1 and align itself to the latest technology standards like Web Components.
- It supports ES6 at its core with Typescript
- Mobile First has been one of the core thoughts behind the framework.
- Speed & Performance has been given special focus using the new Change Detection system in Angular 2.
- Hierarchical Dependency Injection system.
- New directives and components with improved lifecycle hooks.

Introduction to Angular 2 (Contd.)

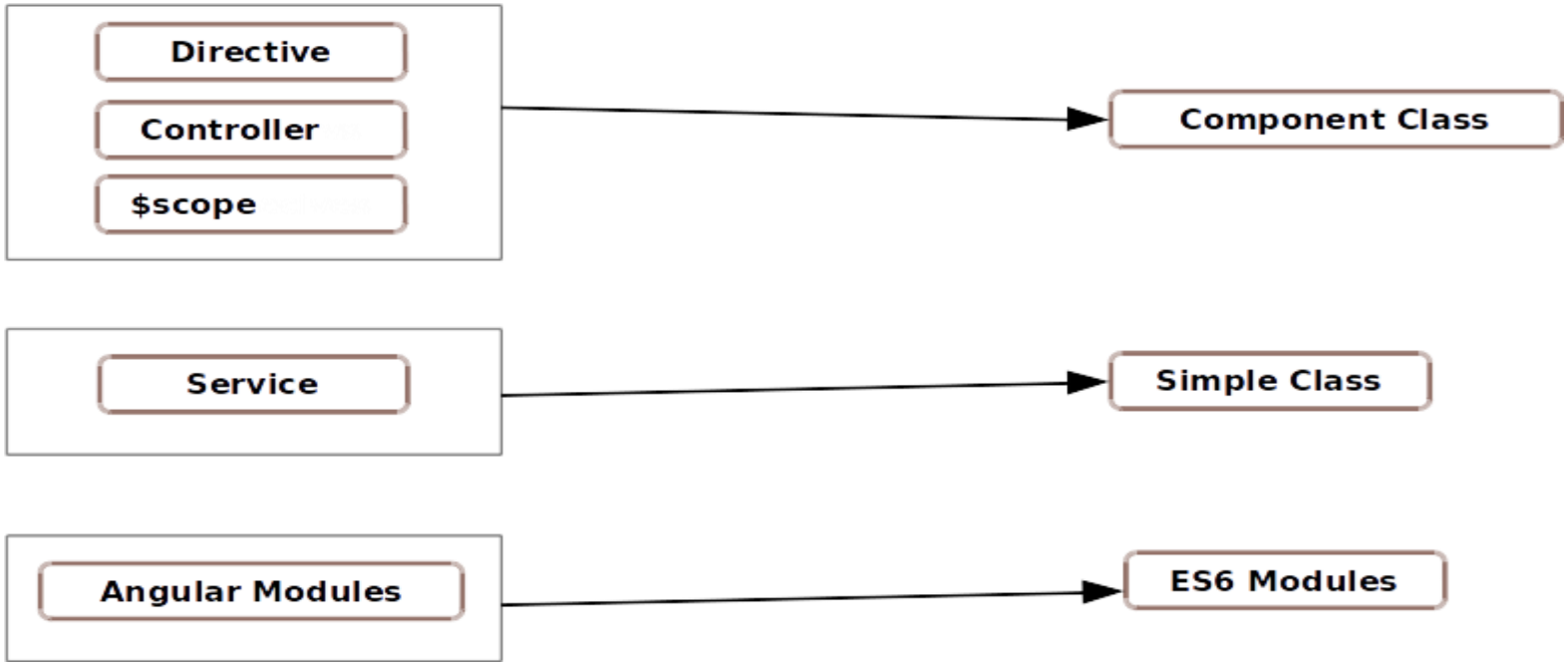
ACADGILD

- **NG2 Flow**



Introduction to Angular 2 (Contd.)

ACADGILD



Transpilers

ACADGILD

- Converts source code from one programming language to other.

- ES6 to ES5 Transpilers:
  - Traceur
  - Typescript
  - Flow
  - Babel
  - Dart

- Angular2 apps should be written in ES6. since the browsers doesn't support ES6 you can use any of the above transpiler to convert it to ES5.

- Google and Microsoft are working together to make type script as a standard for developing angular2.0 apps.

Tech Supporting ng2

ACADGILD

Dev Environment Setup

ACADGILD

- Step1: Install NodeJS & NPM for managing dependencies.
- Step2: Install npm dependencies using package.json or npm install commands - angular2, systemjs, es6-promise, es6-shim, rxjs, zone.js
- Step3: Configure Typescript configurations using a file tsconfig.json
- Step4: Load Typescript typings for ES6 in a file named typings.son
- Step5: Download IDEs supporting Typescript & ES6 Syntax, Recommended IDEs:
  - Visual Code
  - Atom
  - Sublime Text

Hello World Example - ng2

ACADGILD

```
import {bootstrap} from 'angular2/platform/browser';
import {Component} from 'angular2/core';
@Component({
selector: 'app',
template:`<input type='text' placeholder= "Enter first name
" [(ngModel)] = "firstName">
<br>Hello {{firstName}} `
})
class GreetingComponent {
firstName: string;
lastName: string;
}
bootstrap(GreetingComponent);
```


Questions?

ACADGILD

THANK YOU

For more details contact us at:

Support - +91 91 8884666874

Email us at - support@acadgild.com

LEARN. DO. EARN