

Technology Series

AngularJS

05

Presented by  
Subhash EP



Subhash



**ANGULARJS**  
by Google®

# Course Agenda – Introduction

- 01
  - AngularJS core concepts
  - Two-way binding
- 02
  - Routing basics
  - Templating basics
  - Validations introduction
  - Directives concepts
  - What is dependency injection
- 03
  - Development environment
  - Chrome
  - Chrome plugins for AngularJS – Batarang, ng-Inspector
  - Sublime Text
  - Angular seed project
  - Anatomy of Angular Project
  - MVW, MVC, MVVM

# Course Agenda – Structuring AngularJS

- 01
  - Modules
  - Built-in
  - Own modules
  - Best practices
- 02
  - Controllers
  - Scope
- 03
  - From a simple HTML page to a complex AngularJS app
  - Ng-prefix
  - Expressions
  - Adding Logic
  - Properties
  - Functions
  - Nesting
  - Dependency Injection and controllers

# Course Agenda – Client Side

- 04
  - Events
  - Services
- 05
  - Form element
  - Controls
  - Form Validations

# Course Agenda – Routing

- 06
  - Views
  - Routing
  - Factories
  - Providers

# Course Agenda – Let us go deeper

- 07
  - AJAX
  - \$http
  - Data binding
  - Two-way Data Binding
  - Simple Page Applications
- 08
  - Directives
  - More on DI

# Agenda

## Let us go deeper

# Agenda Routing

06

Views  
Routing

# Agenda Routing

07

AJAX

HTTP

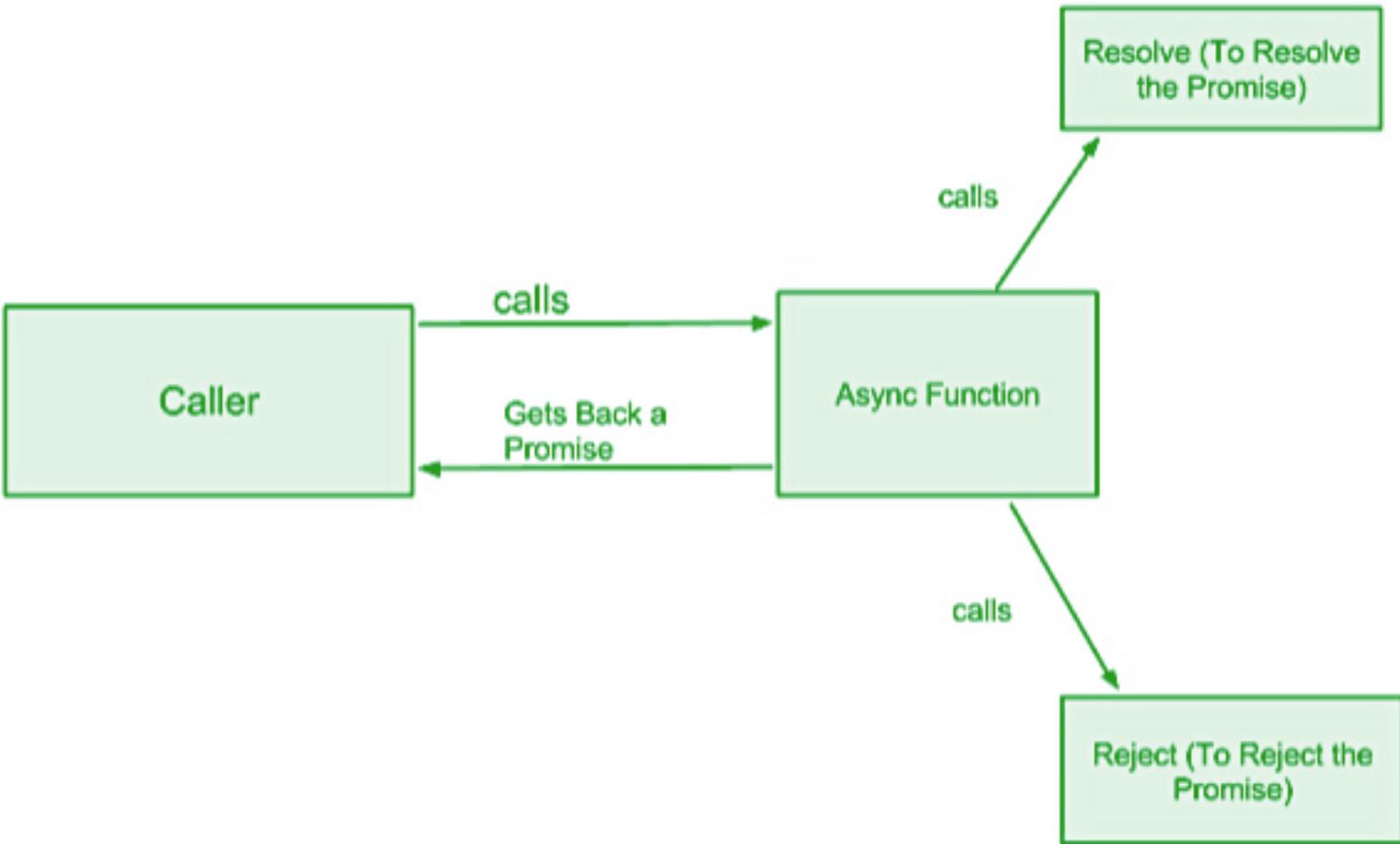
# \$http

- facilitates communication
  - with the remote HTTP servers
  - via the browser's XMLHttpRequest object
  - or via JSONP.

# \$http

- The \$http API is based
  - on the deferred/promise APIs
  - exposed by the \$q service

# Promise



# Agenda Routing

# 08

Directives  
More on DI

# What is Angular directive? (1)

- Use them when you have to make DOM manipulations
- Ever wanted to create your own tag or attribute – this is how you do it!
- Takes you one step closer to creating domain-specific markup

# What is Angular directive? (2)

- Directives should not change a model or controller's logic
- All they handle is creating the page's content and structure of elements

# What is Angular filter? (1)

- Reusable operation using which you can modify the content that is shown on the page
- Examples: uppercase a value, filter search results, etc.

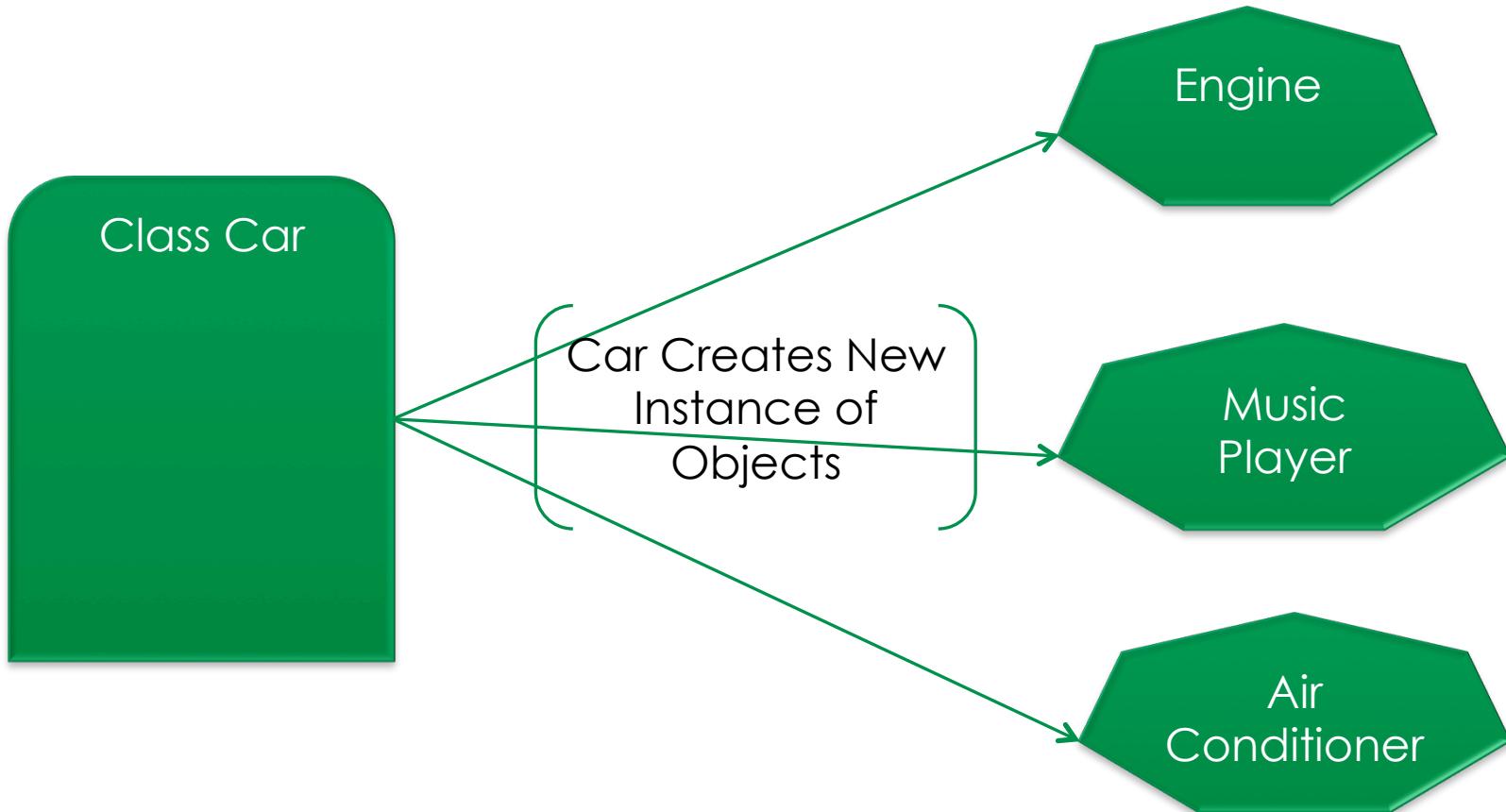
# Defining a new filter

```
module.filter("filter_name", function(data) {  
    // data is filtered or modified  
    // in a specific manner and  
    returned  
    return data;  
})
```

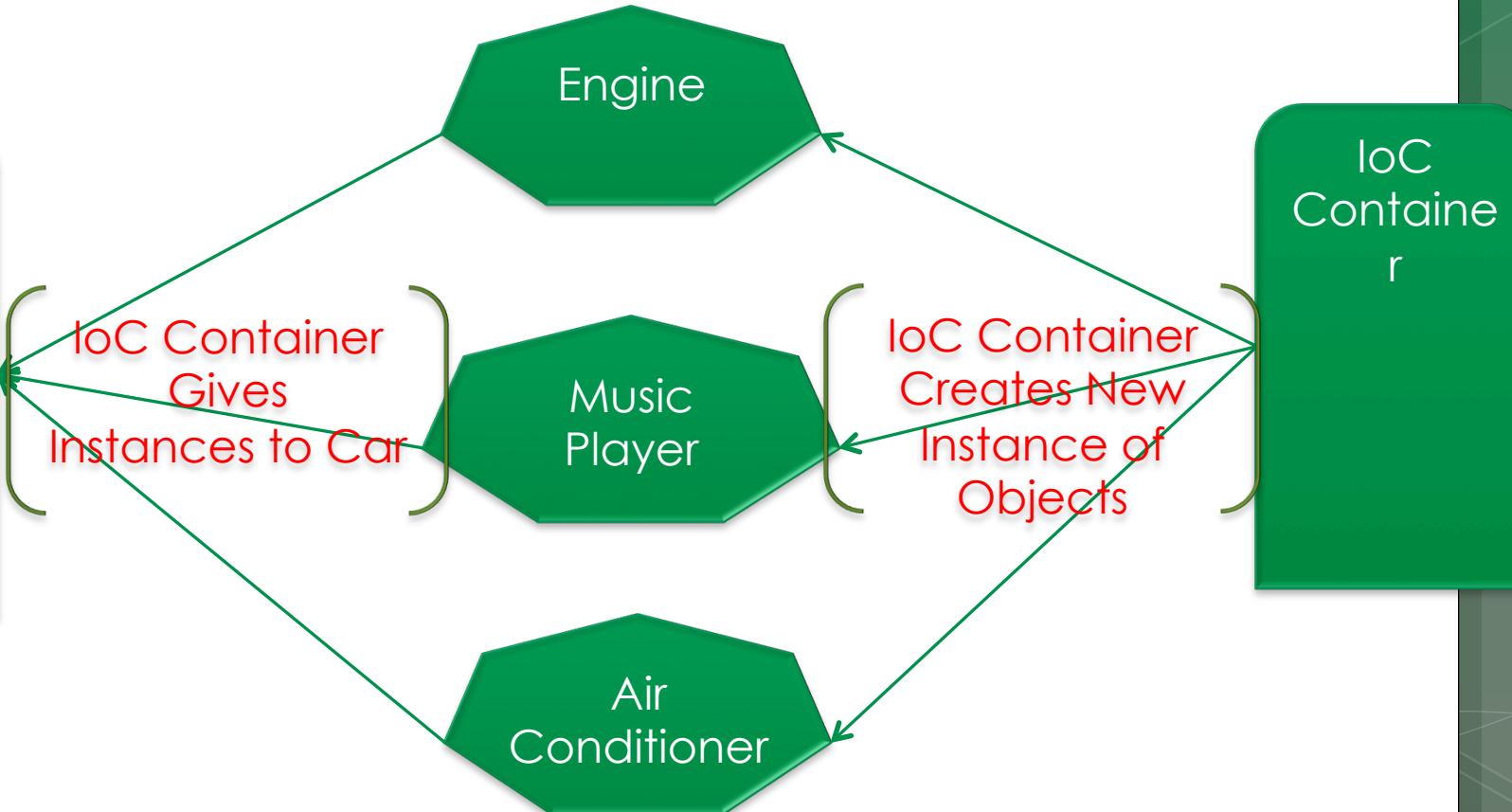
# Using the filter we just defined

```
<div  
    data-ng-repeat=  
        “record in records | filter:  
        filter_name”  
>  
</div>
```

# Without IoC



# With IoC



# No IoC – Love Marriage



# IoC – Arranged Marriage



# More on DI

- Injectable Types
- how are these injectable components registered?
- The answer is the \$provide service
- \$Injector is responsible for instantiating the injectable types
- \$provide service registers the injectable types with \$Injector.

# That's all



## End of Session

©

[subhash.ep@gmail.com](mailto:subhash.ep@gmail.com)  
[linkedin.com/in/subhashep](https://linkedin.com/in/subhashep)