## Introduction

AngularJS was born in 2009 and was founded by Misko Hevery and later acquired by Google. It is an excellent front-end JS framework that has been used in Google's many products. AngularJS has many features, the most central of which are: MVC, modularization, automated two-way data binding, semantic labeling, dependent injection, and so on.

## Context View

We will discuss all external entities that Angular interacts with and the context in which they interact with each other in this section. Through this analysis, we want to get an overview of the ecosystem in which Angular resides through which we can identify dependencies and users of interest.

**System scope**

According to the introduction of [Angular at its website](https://angular.io/), Angular is defined as ‘**Angular is a development platform for building mobile and desktop web applications using TypeScript/JavaScript and other languages.**’ The scope of the software is clearly defined here. It is constrained to focus on one task, which is develop web applications, and on a specific platform.

The website also states some advantages and features of Angular.

**DEVELOP ACROSS ALL PLATFORMS**

Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target. For web, mobile web, native mobile and native desktop.

**SPEED&PERFORMANCE**

Achieve the maximum speed possible on the Web Platform today, and take it further, via Web Workers and server-side rendering.

Angular puts you in control over scalability. Meet huge data requirements by building data models on RxJS, Immutable.js or another push-model.

**INCREDIBLE TOOLING**

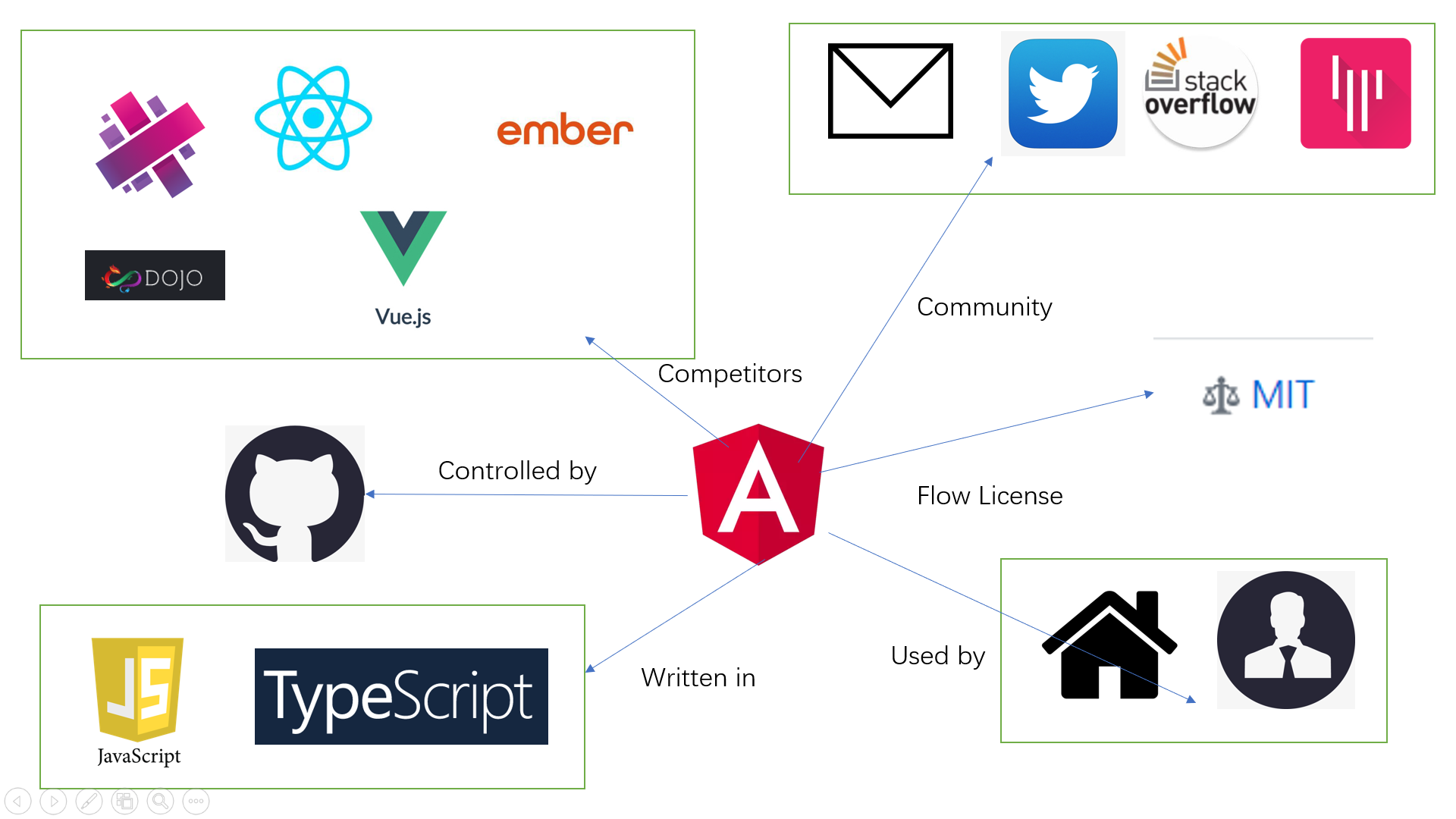
Build features quickly with simple, declarative templates. Extend the template language with your own components and use a wide array of existing components. Get immediate Angular-specific help and feedback with nearly every IDE and editor. All this comes together so you can focus on building amazing apps rather than trying to make the code work.

**LOVED BY MILLIONS**

From prototype through global deployment, Angular delivers the productivity and scalable infrastructure that supports Google's largest applications.

**External Entities**

The connections between Angular and its environment is summarized in the figure.



The external entities can be split into three groups.

1. Competitors

Angular is of course not the only platform that provides front-end functionalities. When Angular was initially created, products with the same function already exist on the market, but they have their own advantages and disadvantages. The following is a list of competitors that provide in some way the same functionalities that Angular provides.

* [React](https://github.com/facebook/react)
* [Vue](https://github.com/vuejs/vue)
* [Dojo2](https://github.com/dojo/core)
* [Ember](https://github.com/emberjs/ember.js)
* [Aurelia](https://github.com/aurelia/framework)

1. Software Platform & Dependencies

Programming languages: Angular is almost entirely written in TypeScript and JavaScript and some other languages.

Platforms: Angular is well tested for compatibility on all three major operating systems (Linux, Windows, OSX), and it’s also available on mobile.

Dependencies:

The packages listed in the [dependencies](https://angular.io/api/core/ResolvedReflectiveFactory#dependencies) section of package.json are essential to *running*  applications.

The [dependencies](https://angular.io/api/core/ResolvedReflectiveFactory#dependencies) section of package.json contains:

Angular packages: Angular core and optional modules; their package names begin @angular/.

Support packages: 3rd party libraries that must be present for Angular apps to run.

Polyfill packages: Polyfills plug gaps in a browser's JavaScript implementation.

More information is listed on their [own website](https://angular.io/guide/npm-packages) to introduce their npm-packages.

1. Development & Community

A platform that plays an important role during development is GitHub, which is used for code versioning, issue tracking and project management. All code gets tested and integrated via continuous tools.

The main communication channel used in the development of Angular is through their GitHub repository and their mailing list. More open discussion is done via Stack Overflow and the Angular Gitter channel. The Angular developers also provide support to users through their own website.

1. Users

The users of Angular can be divided into two subcategories. The individual community and enterprise.

Individual community

The individual community are the types of users that uses Angular as hobby for research. They do not intend to make money by using Angular. Such users are developers and universities.

Enterprise

Enterprise are the users who do use Angular as a tool in their company to help their product. Some major companies that use Angular commercially are:

* PayPal
* Netflix
* Lego
* Lambda Test

1. License

The Angular license closely follows the [MIT license](https://github.com/angular/angular/blob/master/LICENSE).

## Stakeholders

In a corporation, a stakeholder is a member of "groups without whose support the organization would cease to exist".[[1]](https://tool.lu/markdown/#ref_sta1)

As the definition above, stakeholders in Angular are interested in and influence Angular. More specifically, stakeholders can make a huge impact on Angular. Let's look at Angular's stakeholders, which include developers, maintainers, users, and other very important roles. After that we will have a deeper understanding and try to give the Power Interest Graph of Angular.

### Overview

* Acquirers

Acquirers, or called as investors, is the first part of the chain which decides the survival of tech companies. They are more concerned about the profit model of the product and the specific operation cost and market profit of the product. Angular's predecessor was Angular. AngularJS was born in 2009 and was created by **Misko Hevery** and others and later acquired by **Google**. Angular is built by a team of engineers who share a passion for making web development feel effortless, and now Angular is managed by them and their collaborators. Actually they are a large group of people, so we just list some of them.

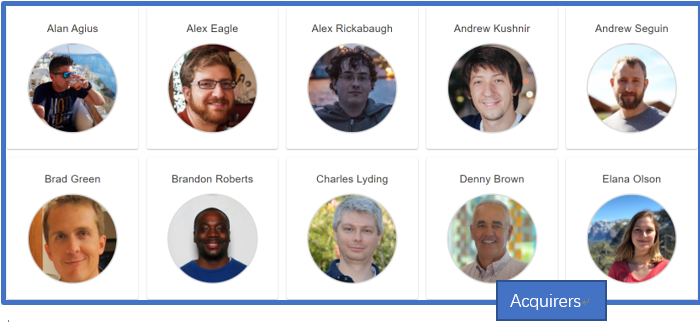


Figure 1: Acquirers of Angular

* Developers

As an open-source program on GitHub, dozens of developers would like to take part in this job. We can easily find them on GitHub. To make it clearer who contributed the most to Angular, here we list the contributors sorted by number of commits.

| **Developer** | **Commits** | **Additions** | **Deletions** |
| --- | --- | --- | --- |
| @[petebacondarwin](https://github.com/petebacondarwin) | 1120 | 406,538 | 247,475 |
| @[gkalpak](https://github.com/gkalpak) | 933 | 96,310 | 65,985 |
| @[vsavkin](https://github.com/vsavkin) | 952 | 165,431 | 117,393 |
| @[vicb](https://github.com/vicb) | 916 | 160,889 | 182,997 |
| @[IgorMinar](https://github.com/IgorMinar) | 856 | 244,201 | 221,320 |

Table 1: Most contributed developers in trems of commit number

* Maintainers

Maintainers are responsible for managing the development and evolution of Angular. They discuss the development direction of Angular. What’s more, they accept and review pull-requests. We can view the main contributors on GitHub. They are the same as the Developers above.

* Users

Angular is an MVVM framework used by **Google** and other front-end engineers in major projects.

* Suppliers

**GitHub** can be regarded as the supplier of Angular owing to it coordinated the process of Angular’s development. Besides, **JavaScript** is the language of Angular, which can be also considered as the supplier.

* Communicators

Angular has some large communities on **GitHub**, **Gitter**，**Stack Overflow** and so on. Every stakeholder of Angular can get help and find more about Angular here.

* Competitors

Obviously, there are a lot of companies which also engaged in front-end development, such as **Vue** and **React**.

### Power Interest Grid

The following figure shows the Power Interest Grid. Power Interest Grid contains the main stakeholder categories and more detailed explanation will be listed.

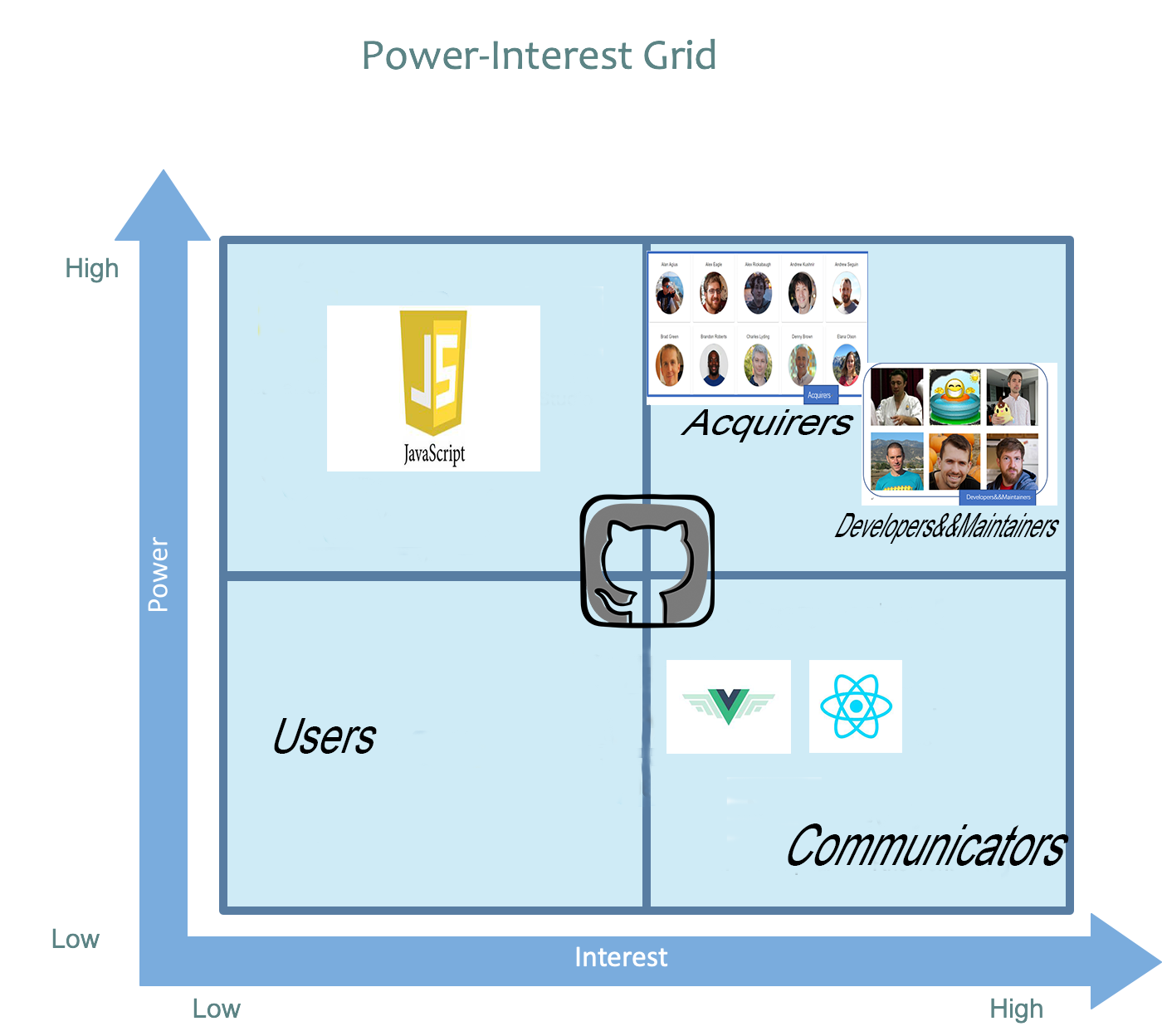


Figure 2: Power Interest Grid

* Low power and low interest

Users, GitHub are stakeholders who do not has any control over Angular, and do not have significant roles in the development of Angular.

* Low power and high interest

GitHub is the supplier of Angular and follow the latest development of Angular. Besides, the communities of Angular on GitHub, Gitter and Stake Overflow are the communicators. They are active in the discussion of Angular, but they do not have significant power to directly change the Angular. What’s more, the Competitors of Angular will be greatly affected by the changes in Angular, but they have nearly no power to influence it.

* High power and low interest

JavaScript are stakeholders who directly affect the development of Angular because JavaScript is the language Angular uses. But JavaScript won’t pay attention on Angular.

* High power and high interest

Core developers, maintainers and acquirers have significant interest and power in developing Angular.

## Development view

Angular JS is not just a class library, but a complete framework. Angular is an MVC framework that provides you with many API functions. You can achieve more complex effects without writing many native js, such as animation, $. animate. It avoids you interacting with multiple class libraries and requires familiarity with the tedious work of multiple interfaces. It is designed by Google Chrome developers and leads the next generation of Web application development. Maybe we won't use Angular JS in five or 10 years, but its design essence will continue to be used.

Knowing the developers of Angular JS, you're sure to be excited about the ability of Angular JS custom instructions (which function like custom controls on the. NET platform). Custom instructions allow you to extend HTML tags and features. Instructions can be reused and used across projects. Large Web projects can be created using Angular JS. You can split the project into multiple modules and one module into multiple module files. At the same time, you can organize these documents according to your usage habits.

**Main Components of Angular**

Modules

Templetes

Directives

Components

Data Binding

Dependency Injection

Metadata

Services

**The characteristics of Angular**

**Two-way binding of data**

This is probably the most exciting feature. The data of view layer and model layer are two-way binding. One of them changes and the other changes accordingly. You don't need to write any code.

**Code modularization**

Each module has its own scope, model, controller and so on.

**Powerful directive**

It can encapsulate many functions into tags, attributes or annotations of HTML, which greatly beautifies the structure of HTML and enhances readability.

**Dependency injection**

It gives the design pattern of the back-end language to the front-end code, which means that the front-end code can improve reusability and flexibility. The future model may put a large number of operations on the client side, and the service side only provides data sources and other operations that the client can not complete.

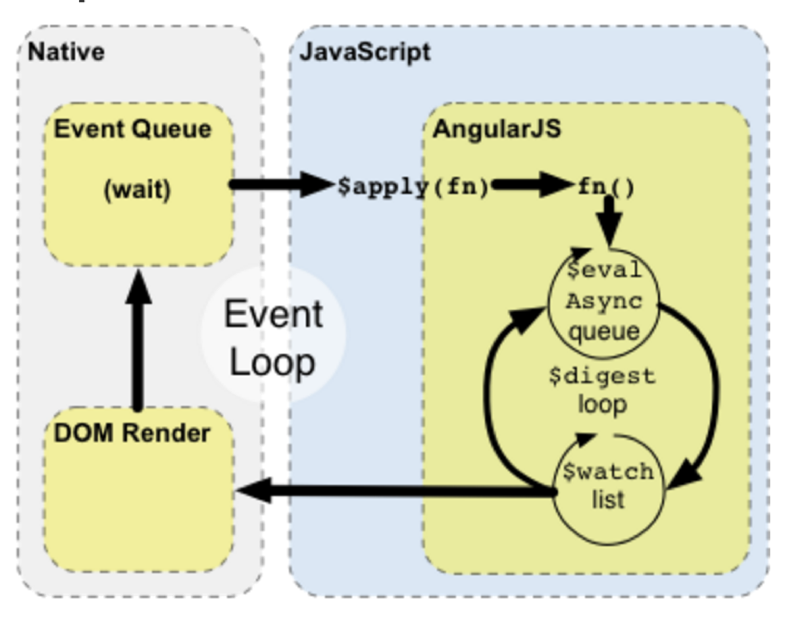
**Test-driven development**

This is the goal of angular JS from the beginning. The application developed by angular can easily carry out unit testing and end-to-end testing, which solves the shortcomings of traditional JS code that is difficult to test and maintain.

**Operating process**

The operation of angular is to implement a set of environments in JavaScript context, called angular context, and the non-angular environment is called classic context.

There is also a queue in the angular context, which contains the watch list, which contains the monitored variables, including those that bind to the data (that is, those that bind to the view). If the user changes a view that binds the data, then an angular function, $apply, is triggered (that is, putting the event queue in the event queue, and then triggering it when the training turns to this), and then updating the value of the change to the bound variable, and then starting to call a digest function, which is used to rotate the watch list to see the column. Table refers to whether or not to change, if there are changes to rewrite the corresponding DOM (do not need angular to write this part of the code, if you have 100 variables, you have to write 100 such changes, and if there are any changes in the future, you have to reconstruct it yourself).



## References

[1]Wikipedia. Stakeholder (corporate)[EB/OL］.Stakeholder (corporate) - Wikipedia，<http://wikipedia.moesalih.com/Stakeholder_(corporate)>

<https://www.imooc.com/article/29750>

<https://www.jianshu.com/p/aa2446510c97>

<https://www.cnblogs.com/airen123/p/9479399.html>

<https://blog.csdn.net/Mr_xiaoshulin/article/details/86029436>