Polymer JavaScript

Shabnam Shaikh[1], Lavina Jadhav[2]

[1]Student, Dept. of Institute of Computer Science, MET College, Maharashtra, India
[2]Professor, Dept. of Institute of Computer Science, MET College, Maharashtra, India

---***---

Abstract - *The library is designed to make it faster for developers to create reusable components for the modern web applications. It is used to develop web applications using HTML web components. Polymer library used for established a set of W3C standards and it is compatible with browser APIs which define our own custom HTML elements. Using polyfills (it is a piece of code that provides the technology that developer expects the browser to provide natively) and sugar, we can create our own custom elements and this features can be used in modern browsers today. The Polymer JS library is used by a number of Google services and websites, Gaming, faster web applications (depending on requirements) etc. This can be designed with different paper or core elements (Google material design, Vaadin elements etc.)*

Key Words*:* Features, Architecture, Custom Elements, Events, Data System, Comparisons

1. INTRODUCTION

Polymer is an open source JavaScript library which is developed by Google. It was released on May 27, 2015, and 1.7.0 is a stable release that was done on September 29, 2017.Polymer gives different web standards (E.g. web components, CSS variables) which don't have wide browser support yet but it's being implemented by every major browser in future which helps to run applications natively in the browsers very fast. Using Polymer we can create our own custom elements from scratch and reuse other elements to extend our custom ones. For this, we have to create our own template of the custom element. The template is a combination of HTML, CSS, and JavaScript which includes the functionality that will be available when we are going to use that element. The Polymer also provides Google material design for the UI so this can use in the building of hybrid mobile application.

1.1 Definition of Polymer.js

A Polymer is a library which provides a thin sugar layer to use the web-components specs called polyfills. And also polymer provides its own set of elements created using web-components specs which are helpful in creating your own customized, reusable elements.

Polymer, or the Polymer project, is the first library that gives developers the necessary tools and definitions to build an application via web components. Polymer benefits users and businesses. Due to the ease-of-use web components offer, developers can increase their productivity while building applications that boast better support and increased longevity.

1.2 Features of Polymer JS:

- It helps to create distributed custom elements across the network that can be used by users.
- The Polymer provides One-way and Two-way data binding.
- It has polyfills for creating own customized elements.
- It provides gesture events as well as computed properties.
- It is used to create custom HTML elements because this library is built on top of web standards API.
- Using Polymer, a creation of hybrid mobile application is easy.
- It distributes custom elements across network that can be used by users.
- It provides the polyfills by this we can create our own customized elements.
- It has command line interface (Polymer CLI) that helps to manage the project from simple to complicated application.
- It provides cross-browser compatibility for the application.

1.3 Use of Polymer JS:

- Polymer uses open web technologies and new web standards.
- Polymer supports shadow and shady Document Object Model (DOM).
- The Polymer DOM layer is closest to the Native JavaScript layer. (It's closest to native DOM APIs, meaning, there is no barrier for developers who already know them.)
- Onboarding is much faster.
- Connection with third party libraries is very easy.
- It's the best JavaScript library when it comes to progressive web apps and lazy loading.

1.4 Need of Polymer JS:

- Big single page applications.

- Completed projects where communication is required between parent – child component and child – parent component.
- Applications with many different widgets & components.
- Your application should support shadow DOM.
- Your application should contain different third party libraries.
- Keyboard accessibility.

2. Architecture of Polymer JS:

The architecture of Polymer.js is divided into four important layers:

Native Layer- It represents the current state of major browser support as well as implementation for the web component detail.

Foundation Layer- It consists polyfills libraries for the web component details. Polyfills is a code that implements the feature of web browser that does not support the feature.

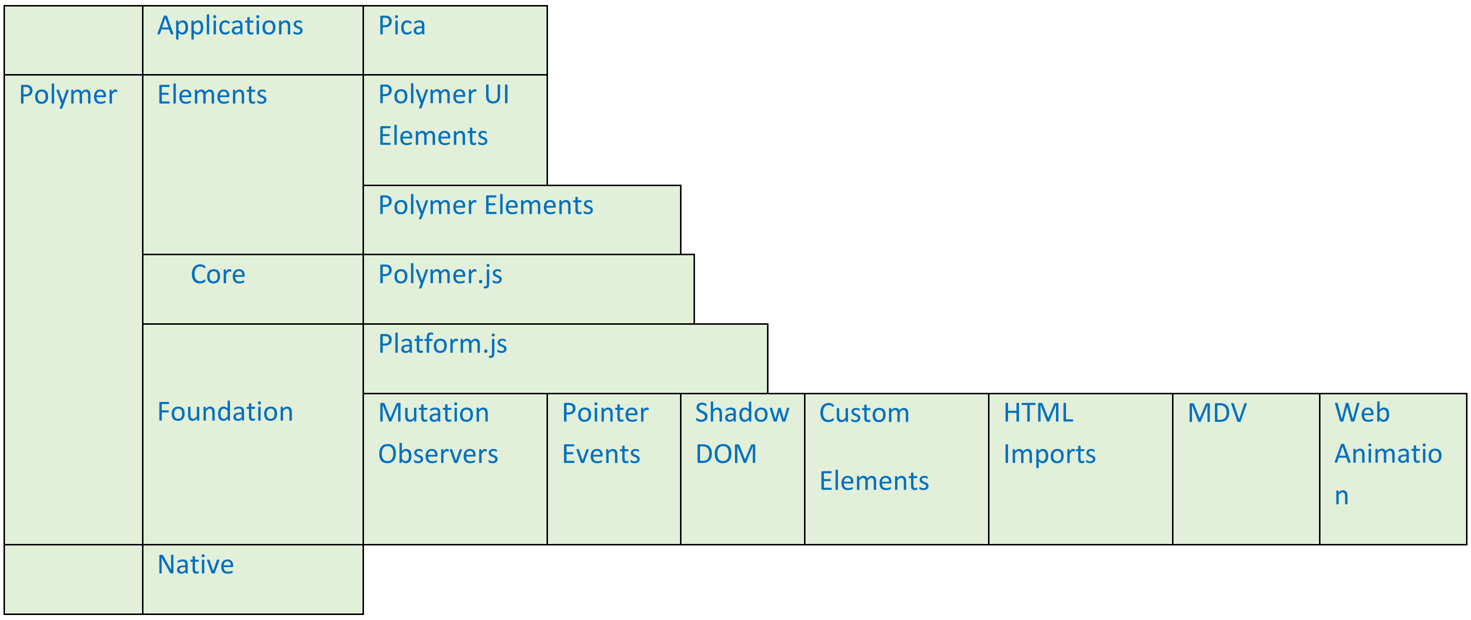

Fig -1: Architecture of Polymer JS

Core Layers- It contains necessary infrastructure for Polymer and the Polymer library code which is found in the polymer.html file.

Elements Layers- It consists of a basic set of elements which treated as building blocks that can help us to create our application.

2.1 Web-Component:

- Custom Elements – These are APIs enable developers to create their own HTML elements that are relevant to their design as part of the DOM structure with the ability to style/script them just like any other HTML tag.
- HTML Templates – We can define fragments of Mark-up within tag which stay consistent across web pages with the ability to inject dynamic content using JavaScript.
- Shadow DOM – This provides encapsulation of JavaScript, CSS, and templates so that your web component code remains modular and separated from the rest of the DOM.
- HTML Imports – Used for to import one CSS file into another which allows including and reusing HTML documents in other HTML documents.

2.2 Polymer JS Elements:

- **App elements**: it's used when we have to build entire application.
- **Iron elements:** It is used for create basic building blocks for application.
- **Paper elements:** It consists with set of UI (User Interface) elements that implement Google 'material Design guidelines.
- **Gold elements**: It is used to implement e-commerce-specific use cases.
- **Neon elements: it** is used to implement animated transitions which are related to Polymer elements.
- **Molecules elements:** It used to connect a group of plugins to the Polymer application.
- **Google web components:** These components are used for Google's API and Service collection.
- **Platinum elements:** This element is like application features which are used for push notifications, offline caching and Bluetooth.

2.2 Polymer JS Custom Elements:

Those elements provide a component model for the web documents and web applications.
Syntax:

```
class MyElement extends HTMLElement {
  // code
};
//link the new class with an our element name
window.customElements.define ('myElement', MyElement);
```

Life Cycle of Custom elements:
The set of custom element reactions that allows to changes in elements lifecycle.

Table -1:

Reaction	Description
Constructor	It is called when element is upgraded.
Connected Callback	It is called when element is added to a document.
Disconnected Callback	It is called when element is removed to a document.
AttributeChangedCallback	It is called when element is change, append, remove or replace.

2.3 Disadvantages of Polymer JS:

1. Lack of documentation and guidance:

Not all Polymer.js UI and non-UI Elements are documented. Some web components don't have good documentation and examples. There is only "guidance" on how to use them in demo code. In some cases we have to refer the source code of a Polymer Element to understand how it works and can use in our applications.

2. Browser supports:

All browser vendors are working on web components supports (Native support).If time is limited and your development team hasn't have experience with web components then there is no use of polymer.

3. Dependency errors and version:

Even if we install any bower component (Polymer.js elements) as required in project, we might find dependency error, pointing to different version dependencies in the same element. These kinds of problems can make development quite challenging.

4. Inadequate or confusing error reporting:

Sometimes when you misspell an attribute name, or just break something related to the core layer itself, we receive a strange error message on console with error stacks that we need to investigate and try determine where the problem is. Sometimes it's easy to solve this, but sometimes you end up needing to try a completely different strategy just to avoid the error since you can't track down its source.

5. Downloading entire library and polyfills:

There is an overhead of loading both polymer and webcomponents.js and then having to go through the upgrade process of custom elements. This means user has to wait for the app to load fully before interacting with it.

6. Lack of server side rendering:

As time goes on and browsers implement the standard, using Polymer will provide speed enhancements as the Shadow DOM allows for efficient rendering (knowing there are no CSS leaks etc.). And with HTTP 2.0 with service workers, a Polymer page will have no speed penalty. It's not clear how to organize in larger applications.

2.4 Events of Polymer JS:

Event is something that happens at specific time and lead to some changes. Polymer.js provides own methods to perform events like: on-click, on-tap, on-mouse over etc. we can also create our own custom events.

It also provides listener objects that can be used to bind events to execute functions.
For E.g.:

```
<dom-module id="my-event">
  <template>
```

```
    <button on-click="submit">Click Me</button>
  </template>
  <script>
    class XCustom extends Polymer.Element {
static get is() {return ' my-event '}
submit() {
            console.log('Hello !!!');
    }
   }
 customElements.define(my-event.is, my-event);
  </script>
</dom-module>
```

2.5 Data System:

In Data system, polymer.js allows us to observe changes on an element's properties by taking different actions.

Data system actions include following properties:

Observers: It invokes specified method whenever data changes.
Computed properties: It computes the virtual properties which is based on other properties and recomputed when the input data change.
Data binding: By using of annotation we can update the properties, attributes or text content whenever data changes.
A data binding connects data from a custom element (the host element) to a property or attribute of an element in its local DOM (the child or target element). The host element data can be a property or sub-property represented by a data path, or data generated based on one or more paths.

For e.g.:

```
//import statements
<dom-module id="myhost-element">
<template>
      <style>
       /* CSS rules for your element */
      </style>
<div>{{textspan}}</div> <!-- data bindings in local DOM -->
<paper-button id="btn" on-click="submit"></paper-button>
 </template>
     <script>
      class MyElement extends Polymer.Element {

        static get is() { return ' myhost-element '; }

        // Declare properties for the element's public API
        static get properties() {
         return {
           textspan: {
             type: String,
             value: "Hello!"
           }
         }
        } // Add methods to the element's public API
        submit() {
          console.log(this. textspan);
        } }
// Register the x-custom element with the browser
 customElements.define(MyElement.is, MyElement)
  </script>
</dom-module>
```

3. Comparisons:

Table -2

Polymer JS	Angular JS
It is a library which is used for creating Web Components to build web apps.	It is a complete framework for building web apps.
It doesn’t provide services, routing etc. except as separate web components from their core library. Polymer focuses on allowing creating rich, powerful, reusable web components, which used to build web apps.	Angular has high-level APIs like services,routing,server communication etc.
Polymer (more specifically Shadow DOM) provides the ability to compose encapsulated JS, CSS, and HTML as custom elements.	Angular element has directives which implemented without the use of the Web Components APIs.Angular directives doesn’t have such encapsulation functionality. Angular compose only html and JavaScript to create reusable directive without use of web component API.
Polymer Web Components specification are the standards-based.	Angular directives are a way to build custom elements.
Polymer elements have templates and bi-directional (one-way and two-way) data binding.	Angular don’t have templates.
Definition Polymer is a webcomponents.js library for creating Web Components, which are a set of W3C APIs and upcoming browser APIs for defining your own custom HTML elements.	Angular is a robust application level JS framework based on MVC pattern, facilitate you to extend HTML’s

Syntax :
`<dom-module id="my-element">`
`<template>`
`</template>`
`<script>`
`Polymer({`
`is: "my-element"`
`});`
`</script></dom-module>`
	syntax to express application's components clearly.
	Syntax:
	`<div ng-app="myApp">...</div>`
	
	`<script>`
	
	`var app1=angular.module("myApp", []);`
	
	`</script>`

Size: The size of minified Polymer is 127KB. It also requires a polyfill called webcomponents.js for browsers where web components are not natively supported. Size of webcomponents-lite.js is 41 KB	Size : The size of minified Angular package is 566K.It consists with observable pattern provided by Rxjs library. Size of Angular 2 with Rxjs library is 766K
Supports components and reuse: Polymer components can ideally be re-used in any web application. Non-polymer application would need to import polymer library to be able to reuse polymer components.	Supports components and reuse : Angular components can be used only in Angular applications
Structure of Code: Polymer facilitates only component creation. There are few components which can be used for routing.	Structure of code : It is a framework. It provides the way to structure the application. It comes with built-in application routing, state management and data communication.
Automated Tests: Polymer's Web Component Tester is an end-to-end testing environment built by the Polymer team.	Automated Tests: Both unit and e2e test cases can be written and executed using jasmine and protractor.
Learning: Polymer components usually written in ES5 / ES6 JavaScript. Developers will have to get used to the concept of components. Polymer provides minimal syntactical sugar over web components REST API, which doesn't impose a steep learning curve.	Learning: Typescript is a new language and Angular supports 'decorator' which used to writing code is which well-known JavaScript developers.Although,upcoming versions of JavaScript have concept of decorators. Developer will have to learn the framework as well as the language
Security: Polymer team suggests usage of Crisper tool for content security.	Security: Angular has in-built features for content security and XSS protection. Angular has in-built route guards, which can help in implementing application security.
Native script Support: There is no known support for Polymer components in Nativescript framework.	Native script Support: It is a framework for building native iOS and Android app using javascript. Components can be used with Nativescript framework.

Table -3

Polymer JS	**React JS**
Polymer has a better template engine than React.	React has a better template engine than Angular but not polymer.
Polymer has a large set of helpful behaviors.This sets of the behaviors very helpful for developers to reduce the development time.	React has complexities not beneficial for the developers or product owners. For example, it doesn't have cleared a definition between Factory and Service. When the concepts are not clear, this is a painful for the

	development team and critical for the application and deadlines.
Styling Polymer supports: • Mixin declaration (Custom CSS mixin) • CSS variables	Mixin declaration and CSS variables these features are not supported from React.
Working with Native JavaScript Objects Is Much Easier The polymer DOM layer is closest to the DOM API layer. Example with querySelectorAll() : It Returns a list of the elements within the document (using DFS pre-order traversal of the document's nodes) that match the specified group of selectors. The object returned is a NodeList. var mynodes = Polymer.dom(element).querySelectorAll('my-element');	Data binding is not a problem for React, but the story is different for DOM manipulation and element selection. Syntax: var mynodes = Polymer.dom(element).getEffectiveChildNodes();
Onboarding Polymer documentation has a better structure and examples than React. Polycasts Web Shows – Google Developers is a great developer show.	React don't have a developer cast.
Connecting with Third Party Libraries Is Very Easy : Polymer doesn't have an extra layer and sharing data between components.	In react, There is middleware to communicate to third party libraries.
The Best JavaScript Library When It Comes to Progressive Web Apps and Lazy Loading.	If application has a lot of dynamic content changing within the view, like the gallery of photos on your Instagram feed, React library is most preferable.
Polymer provides Very Large Set of Custom Elements for Free.	React have a lot of third-party libraries like widgets but when we talk about the data grids, custom input elements, uploaders, icons, etc. in many cases these widgets are not free or don't have a commercial license.
Polymer follows web component architecture.	React has own component architecture.
To reuse the component we have to do HTML import.	By Loading the JavaScript module we can reused the component.
No compilation and pre-processing are required in Polymer JS.	For compilation code should be transpiled since it uses JSP syntax.
Polymer JS is not fully supported to server-side rendering and limited to prerendering.	It is completely supported to server-side rendering.
Polymer JS library is set of basic components and material design elements involved in standard distribution.	For standard distribution react consists with only library not components.
Polymer catalog is central registry of components.	NPM is central registry of components.

4. CONCLUSIONS

Polymer technology which is not well suited to development of a large, enterprise level, production-ready application. Additionally, there isn't many guidance or tutorials available specific to Polymer.js web development.

As far as whether the JavaScript-centric or DOM-centric approach is really absolutely better. Choosing between one over the other technologies has a lot to do with the unique requirement of your project. If our application has a lot of dynamic content changing within the view or pages, like the gallery of photos on your Instagram then react is the most prefer. If we don't have time to learn React, those JSX files have feeling uncomfortable, or want to be ready for the WC3's latest web standards, it might be easier and faster to quickly setup simple website using Polymer.

REFERENCES

[1] "Hashnode" Jounals

[2] Tutorials Example of Polymer

[3] "Dotjs file" blogs

[4] "Polymer Project" org website

[5] "binpress" blog

[6] https://mentormate.com/blog/polymer-vs-angular-future-web-apps/