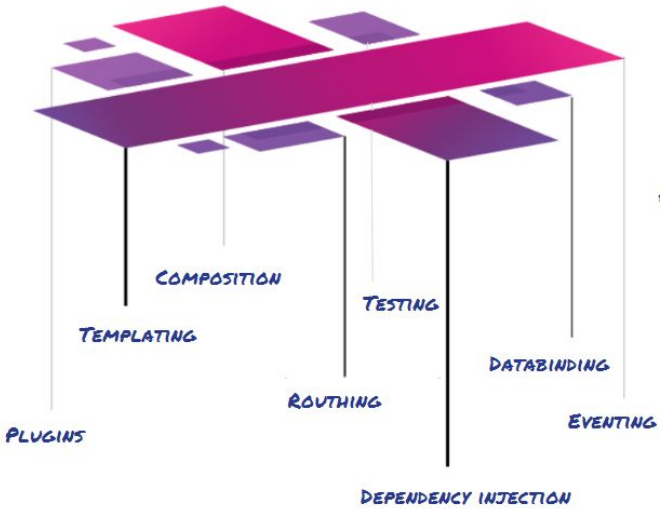
the next generation framework

Aurelia is a modern, front-end JavaScript framework that is designed for building browser, mobile, and desktop applications - all open source and built on open web standards. This framework focuses on aligning closely with web platform specifications, using convention over configuration, and having minimal framework intrusion. Aurelia allows you to write the code you want to write without the framework getting in your way.

As a framework, aurelia does not interfere with writing code in vanilla Javascript. It is only an add-on that introduces separate facilities.



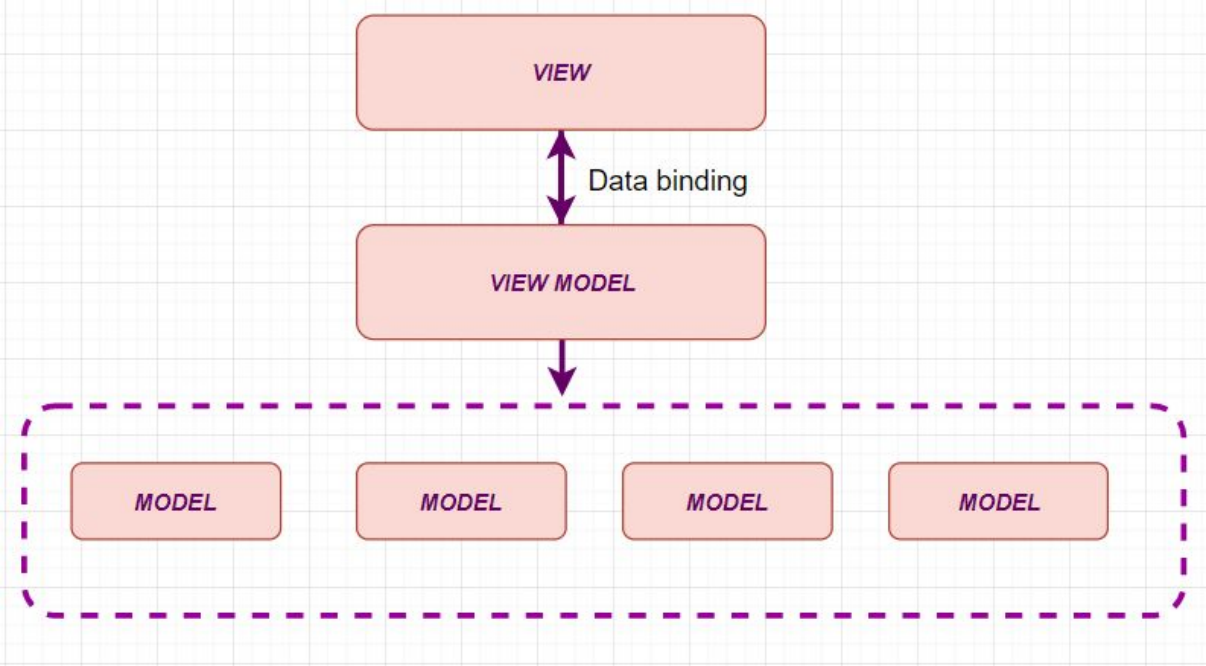
The architecture is not monolithic, it is a collection of more functionally oriented modules that help us build simple web applications. The Aurelia feature-oriented modules include plugins, templating, composition, routing, testing, dependency injection, data binding, and eventing.

On top of all this, Aurelia provides a number of additional plugins for internationalization, validation, modal dialogs and much more. You don't have to cobble together a bunch of different tools - Aurelia provides a great-working CLI for generating and building projects. And yet, you don't have to use it as Aurelia is structured to let you swap out literally any detail - even down to the templating engine, in order to guarantee maximum flexibility. Aurelia.js is using MVVM Architecture

MVVM (Model View ViewModel) Architecture

The three main elements in this Architecture, which is based mainly on the **MVC** pattern are View, ViewModel, and Model.

- **View** - what your client can see on their screen
- **ViewModel** - contains the data and information that will be displayed in the View
- **Model** - contains all declared parameters and their client-side data



What are the benefits of MVVM?

- Separation of Skills: This enables a separation of responsibilities on teams have a designers and programmers
- Views are agnostic from the code that runs behind them, enabling the same view-models to be reused across multiple views
- No duplicated code to update views - rely on databinding to keep view and view-model in sync.
- Since view-model is separated from view view-model classes can be tested independently

The main advantages of using aurelia.js are:

Modularity

It’s a collection modern javascript modules - each module is written using ECMAScript (aka JavaScript) or TypeScript (a strict superset of JavaScript that adds compile-time type checking). Many of these modules can be used individually in any type of JavaScript project, including Node.js.

Databinding

Self-adjusting data binding between DOM and JS. Depending on the situation, it can use Object.observe, Array.observe, getters, setters or so-called dirty checking.

Advanced Router

Ready-to-use implementation of the router, offering rich possibilities of configuration, path creation, definition of nested routers or so-called navigation steps, thanks to which we can check e.g. the user's authentication while navigating to the next path.

Simple testing

By combining modern JS modules with an unobtrusive approach, Aurelia makes unit testing as simple as testing vanilla JS. Need to write integration tests? A powerful Dependency Injection Container and testing library make it quick and easy. You benefit from highly maintainable and longer-lived apps.

Boundless extensibility

No framework in the industry may match Aurelia's extensibility. You are able to create fully custom elements, add custom attributes to elements that already exist, control the template generation, customize template syntax, create new reactive binding types, extend the DI, and more..

Sources:

https://www.iborn.net/blog/aurelia-next-gen-javascript-framework

https://aurelia.io/