ABC Portfolio Management

Solution and Proposal

Non Functional Requirement(1 of 2)

- Light weighted and faster to load on all devices & network bandwidths
- Scalable enough for future extension
- Reusable components should comply to Web Components standards
- Follow OWASP Top 10 security standards
- Cloud based web solution
- Source code should follow framework provided best practices, coding standards and naming conventions
- Progressive web application
- Support desktop browsers IE9+, Chrome23+, Safari6+ , Firefox 21+
- Safari and Chrome browsers running on iPhone 6 and above and Android Marshmallow and above
- Supports internationalization (i18n) and Accessibility WCAG 2.1(AAA conformance) guidelines
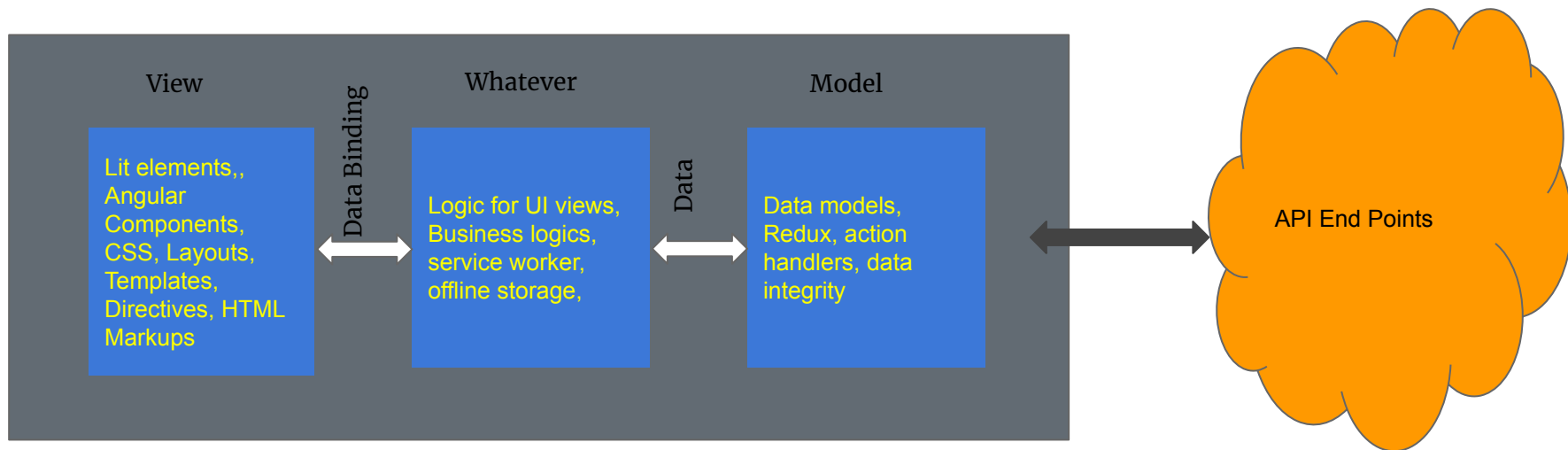
Non Functional Requirement(2 of 2)

- Under 3 second load time
- Responsive screens that covers all the popular screen types
- MVW architecture compliant in app
- SPA architecture for web app
- Location agnostic
- Provide access to their contents such as live stock market ticker, share price, top 5 performing stocks of the day through this digital channel
- Lighthouse report

Innovative Suggestion

- Infinite scrolling is not a good choice in this case. As a user I would want to have finite places to go and search for my stocks, if this were about news/social media posts/search results infinite scroll is suitable in those places
- Notifications on the cell phone, push notifications
- TWA/PWA from the same code base as we are including the service worker

Architecture Diagram(High Level)

MVW inside SPA

Framework of choice

- Angular has 100% support for webcomponent
- Angular has quick way to implement service worker support
- Angular also supports modularisation to a greater extent by providing ngModule
- Tooling for Angular is great as Angular Material is fairly mature
- Angular also has support for Redux and I have used the same for state management

Webcomponent

- For now I have created a WC which is to plot a graph for the top 5 best performing assets
- As project moves on we would create multiple webcomponent for each of the reusable component such as sort tables, profile, side nav bar content etc
- As we do not have the private npm so I have used .js files and distributing them manually
- Later the ideal situation has to be the npm based distribution
- Lit-Element has been used for building the web components

Below is the repository for the same

https://github.com/deepakjha14/HCLCaseStudyWebComponent