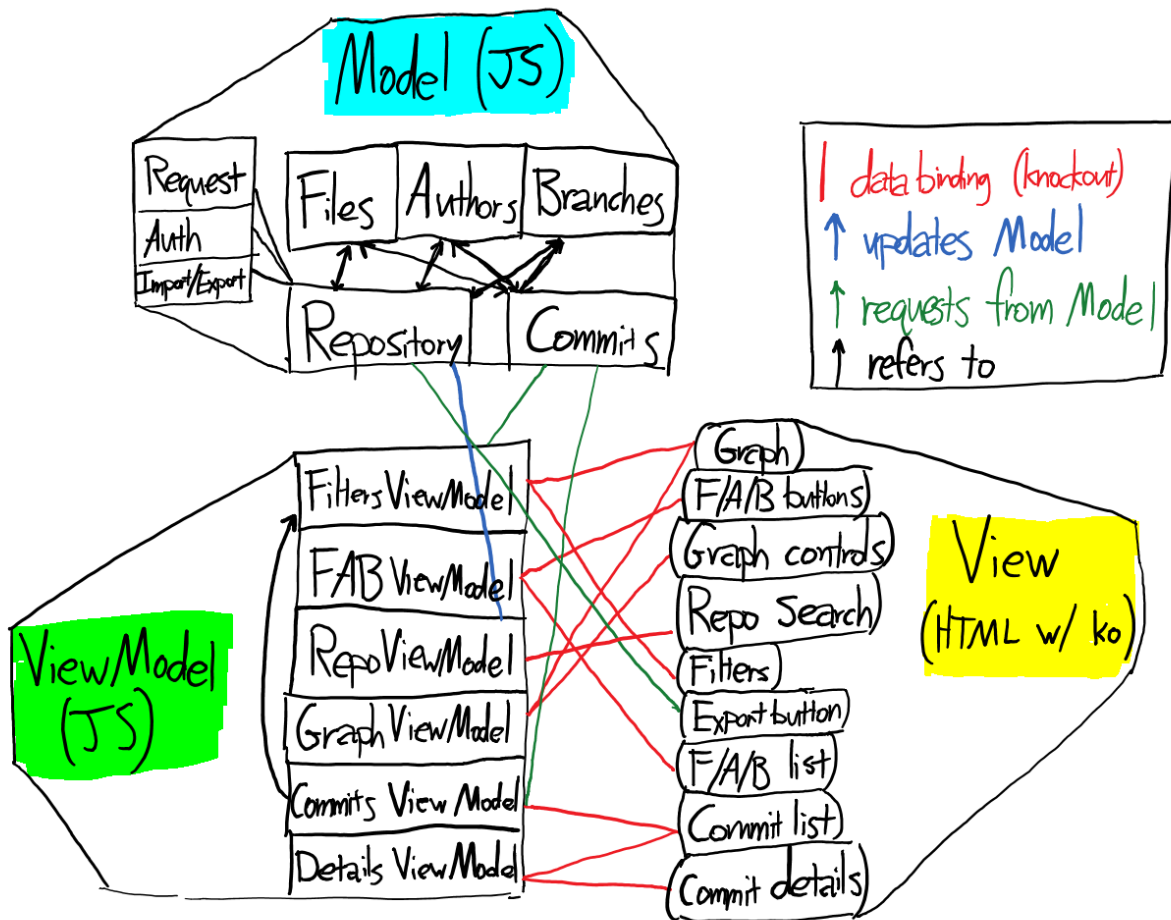# Assignment 2- High-Level Design

Artitecture Diagram

By: John McDouall, Somn Kafley, Jordan Kuhn and Steffen Moseley

Model (JS)

Request
Auth
Import/Export

Files | Authors | Branches

Repository | Commits

data binding (knockout)
↑ updates Model
↑ requests from Model
↑ refers to

ViewModel (JS)

Filters ViewModel
FAB ViewModel
Repo ViewModel
Graph ViewModel
Commits View Model
Details ViewModel

View (HTML w/ ko)

Graph
F/A/B buttons
Graph controls
Repo Search
Filters
Export button
F/A/B list
Commit list
Commit details

This project follows the Model-View-ViewModel (MVVM) architectural style, which bears some resemblance to Model-View-Controller (MVC), with the addition of a ViewModel that facilitates transparent two-way communication between View and Model through the use of data bindings. The role of Controller is still present, but transparent: the knockout.js library is used to connect the Views and ViewModels via these bindings, and so perhaps you could think of knockout as the Controller.

Although opinions on MVC are highly varied, the consensus seems to be that MVC is good for large server-side architectures, or for thin clients where the focus is on generating a mostly-static view, presenting it, collecting results, and regenerating, like with older webpages, while MVVM is better for richer clients, such as modern webapps powered by frameworks like knockout. This is the reasoning behind our choice of MVVM as our architectural style.