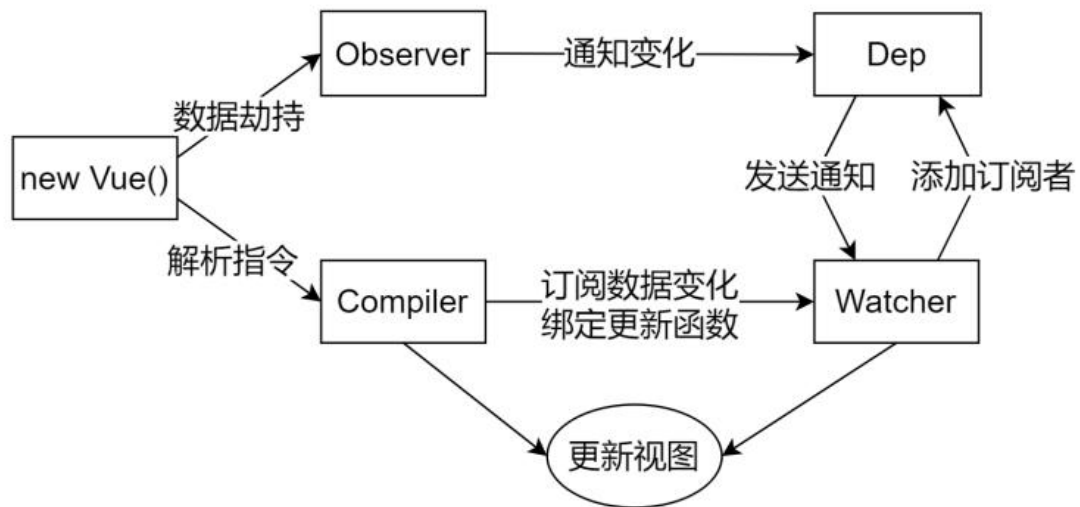


下图展示数据响应整体流程：



● Vue

1. 记录传入的选项，设置 `$data/$el`
2. 把 `data` 的成员注入到 `Vue` 实例
3. 负责调用 `Observer` 实现数据响应式处理（数据劫持）
4. 负责调用 `Compiler` 编译指令/插值表达式等

● Observer

1. 数据劫持
2. 负责把 `data` 中的成员转换成 `getter/setter`
3. 负责把多层属性转换成 `getter/setter`
4. 如果给属性赋值为新对象，把新对象的成员设置为 `getter/setter`
5. 添加 `Dep` 和 `Watcher` 的依赖关系
6. 数据变化发送通知

● Compiler

1. 负责编译模板，解析指令/插值表达式
2. 负责页面的首次渲染过程
3. 当数据变化后重新渲染

● Dep

1. 收集依赖，添加订阅者(watcher)
2. 通知所有订阅者

● Watcher

1. 自身实例化的时候往 `dep` 对象中添加自己
2. 当数据变化 `dep` 通知所有的 `Watcher` 实例更新视图