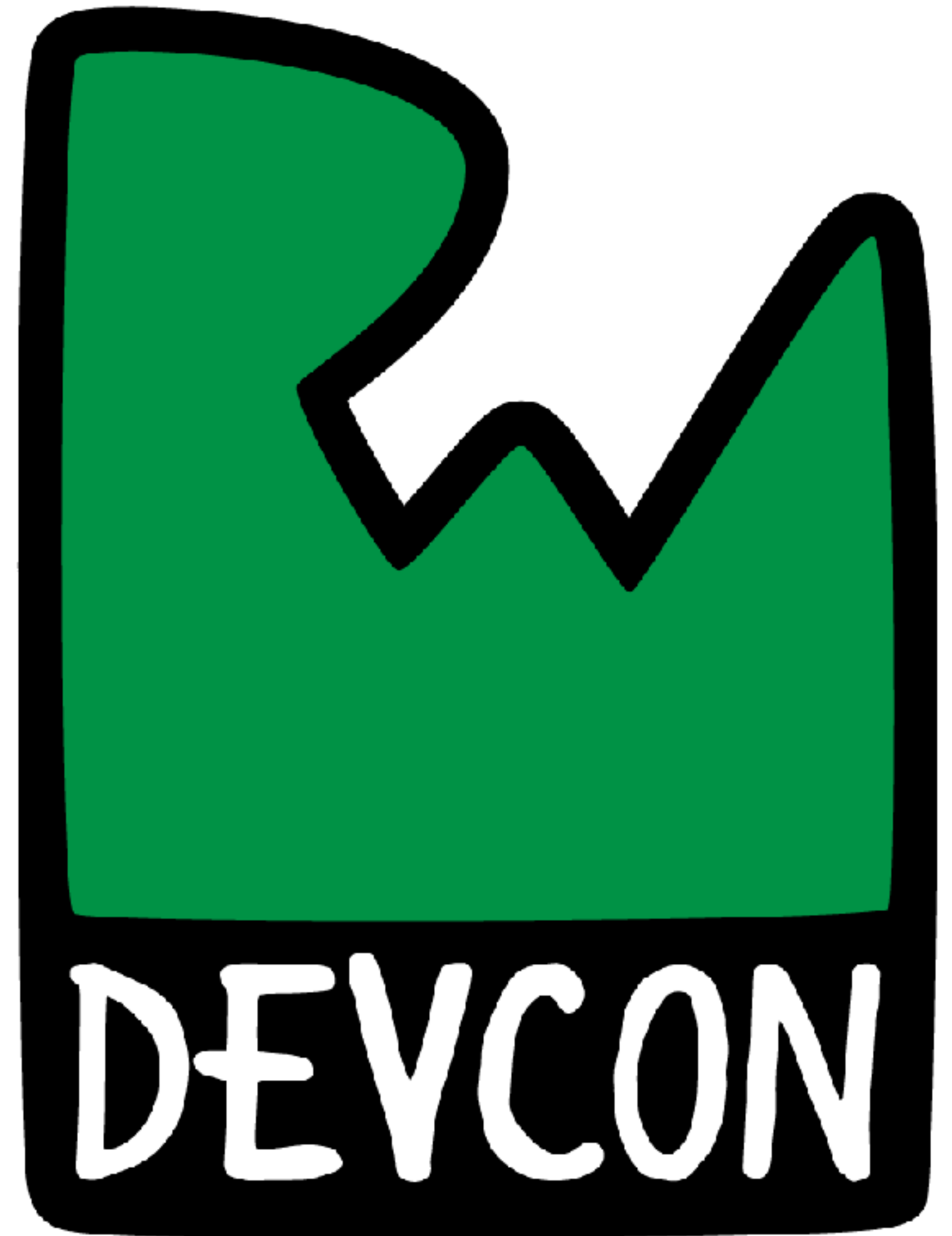


Session 18: Advanced Unidirectional Architecture



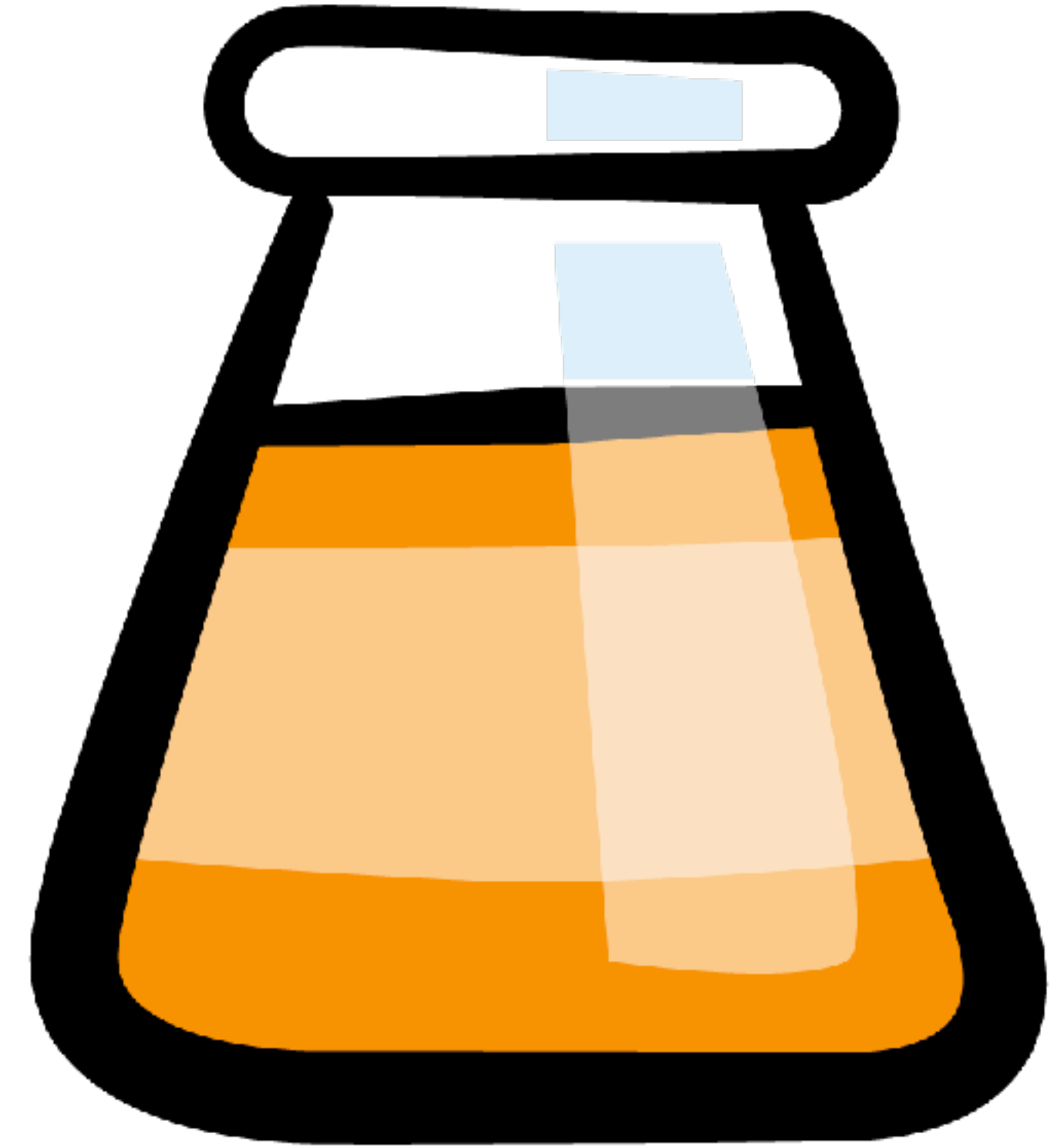
CODE THAT'S EASY TO CHANGE



Change is the only
constant in software
development.



*This is an
advanced session.*



THE ARCHITECTURE - TECHNIQUES

- ⚙️ Unidirectional Dataflow, i.e. Redux
- ⚙️ Dependency Injection
- ⚙️ RxSwift Observables
- ⚙️ Use Cases





Point is to connect these techniques.



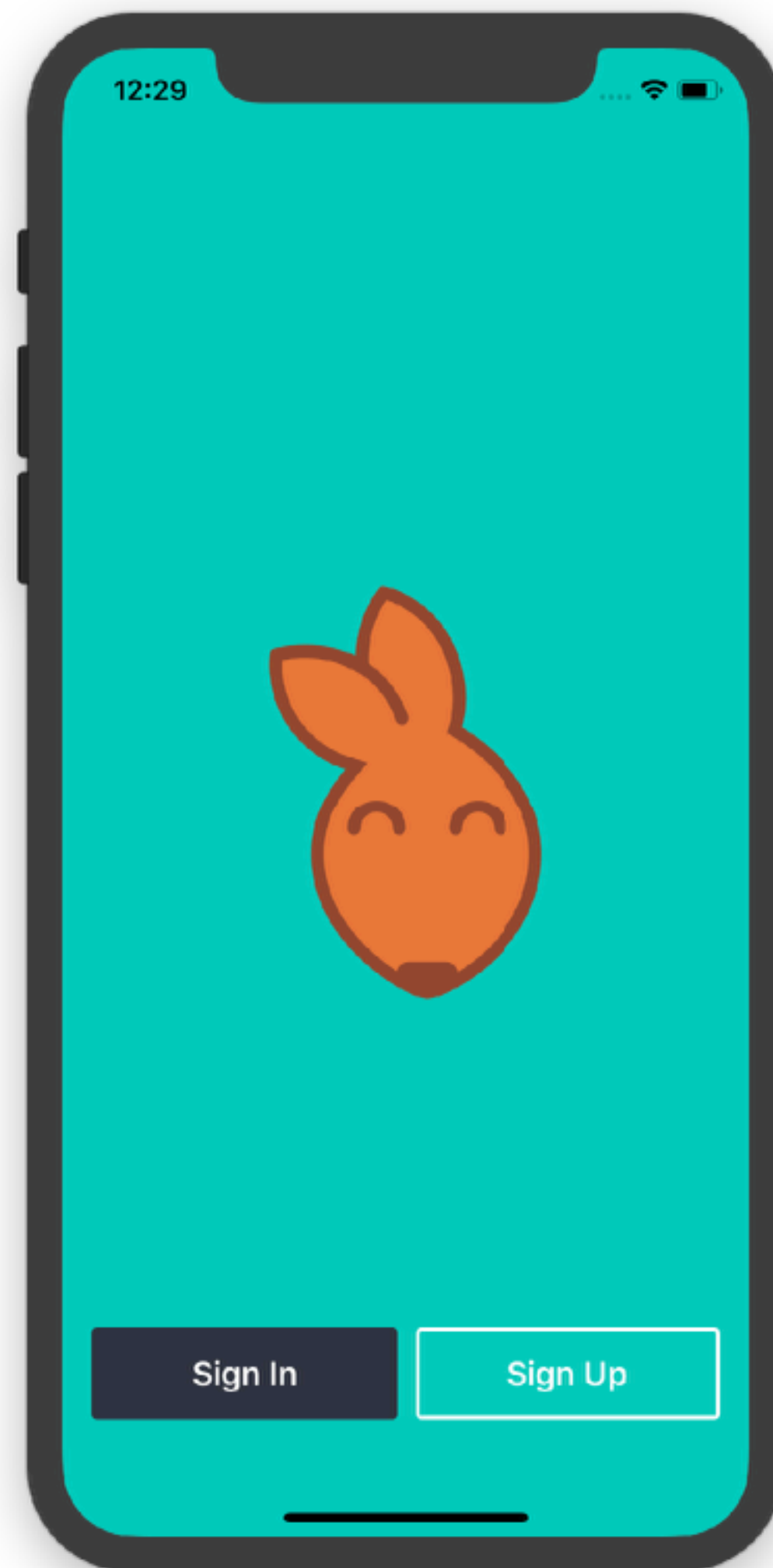


WHAT YOU'LL LEARN

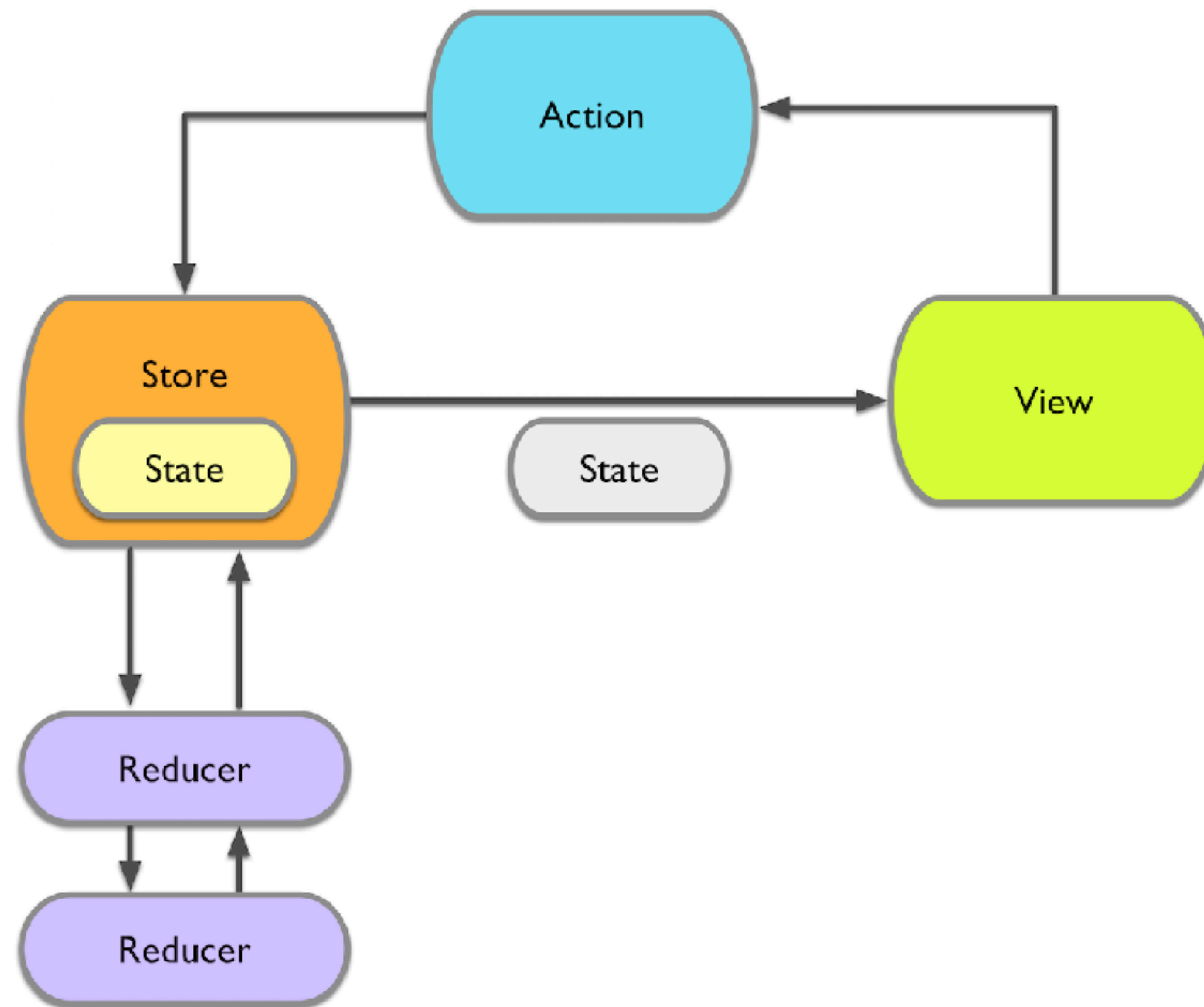
- ⚙ Use RxSwift to render UI from Redux store
- ⚙ Incorporate persistence into Redux
- ⚙ Abstract ReSwift away from view controllers
- ⚙ Maximize testability and playground-ability



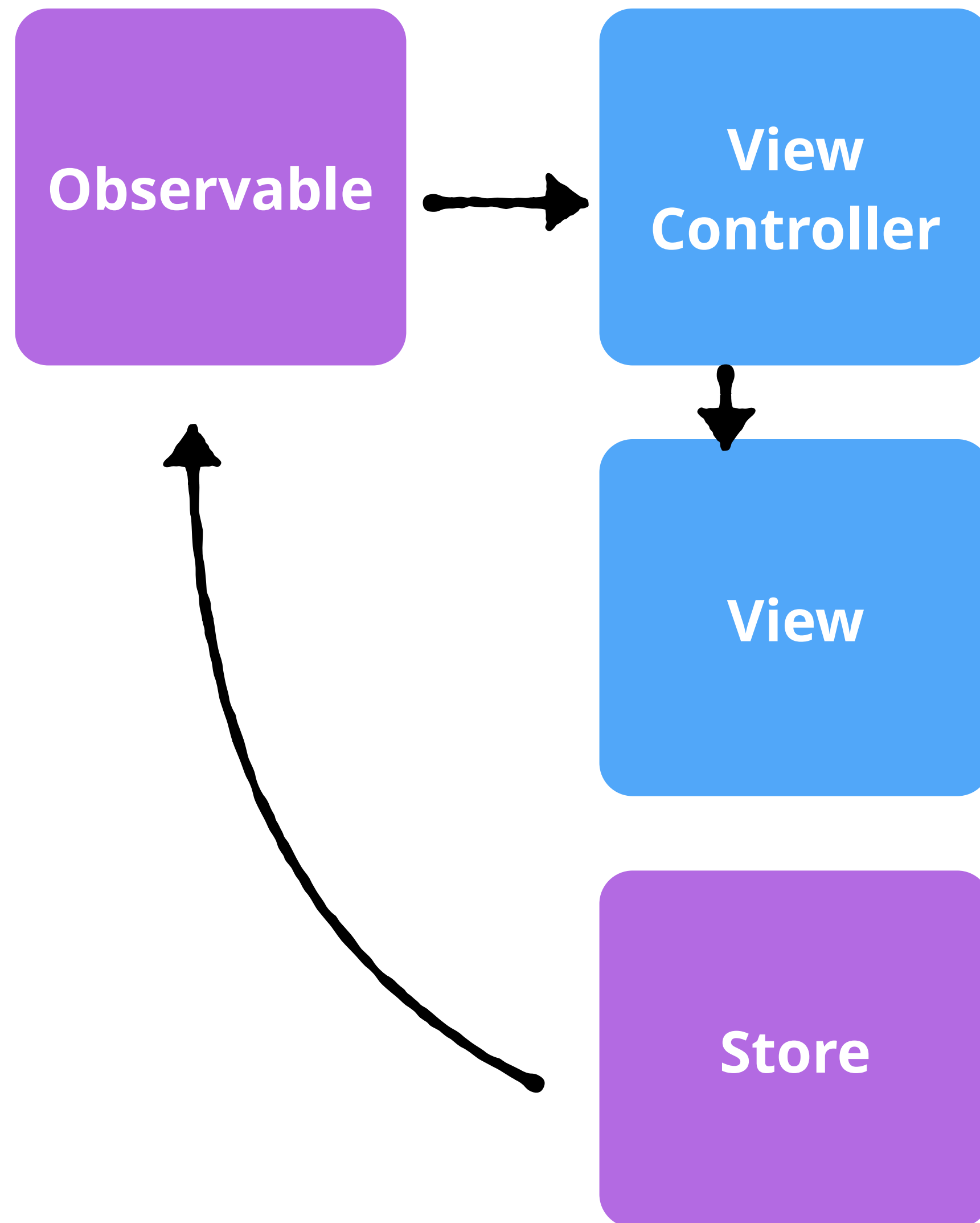
KOOBER



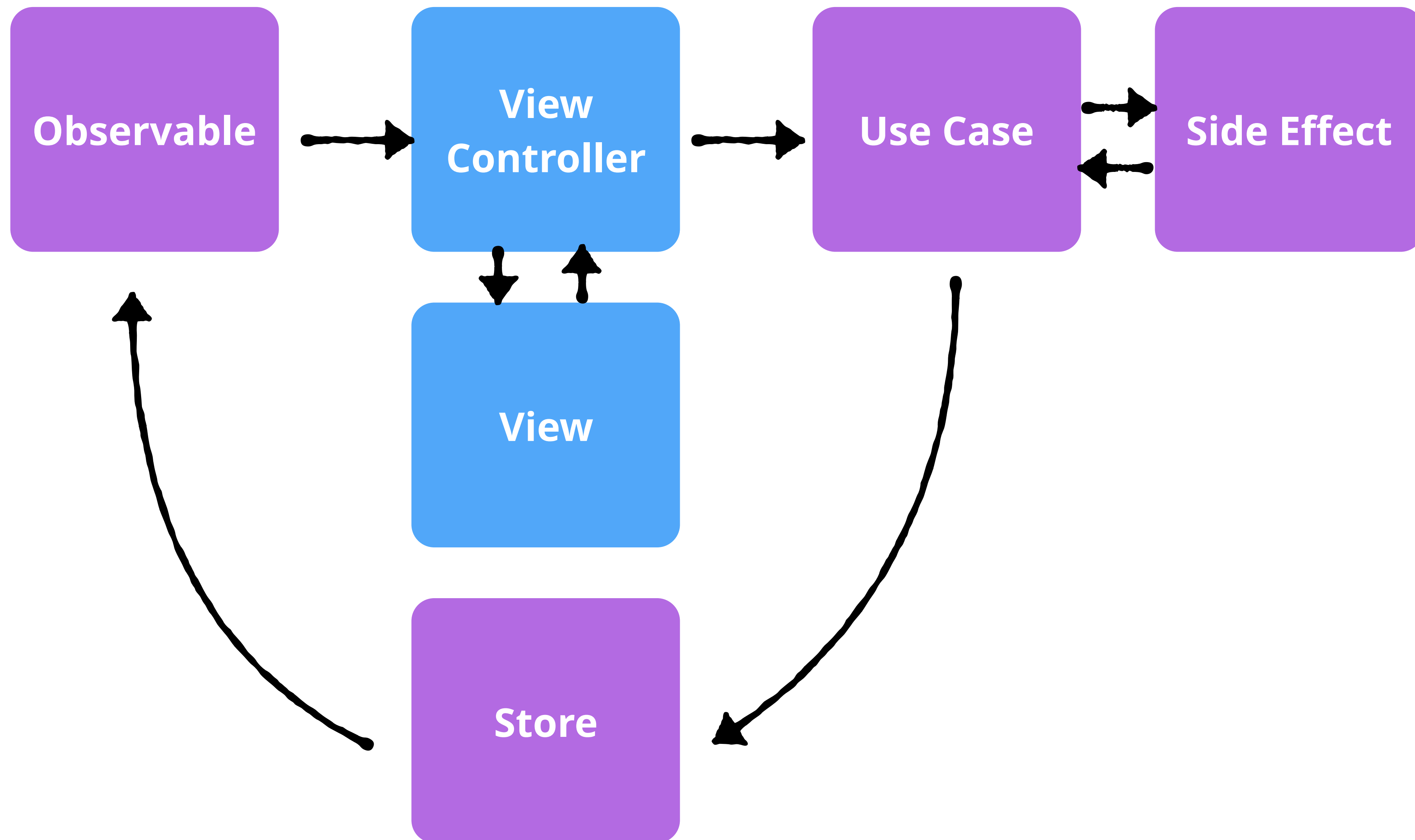
THE ARCHITECTURE - REDUX



THE ARCHITECTURE - DATAFLOW



THE ARCHITECTURE - DATAFLOW

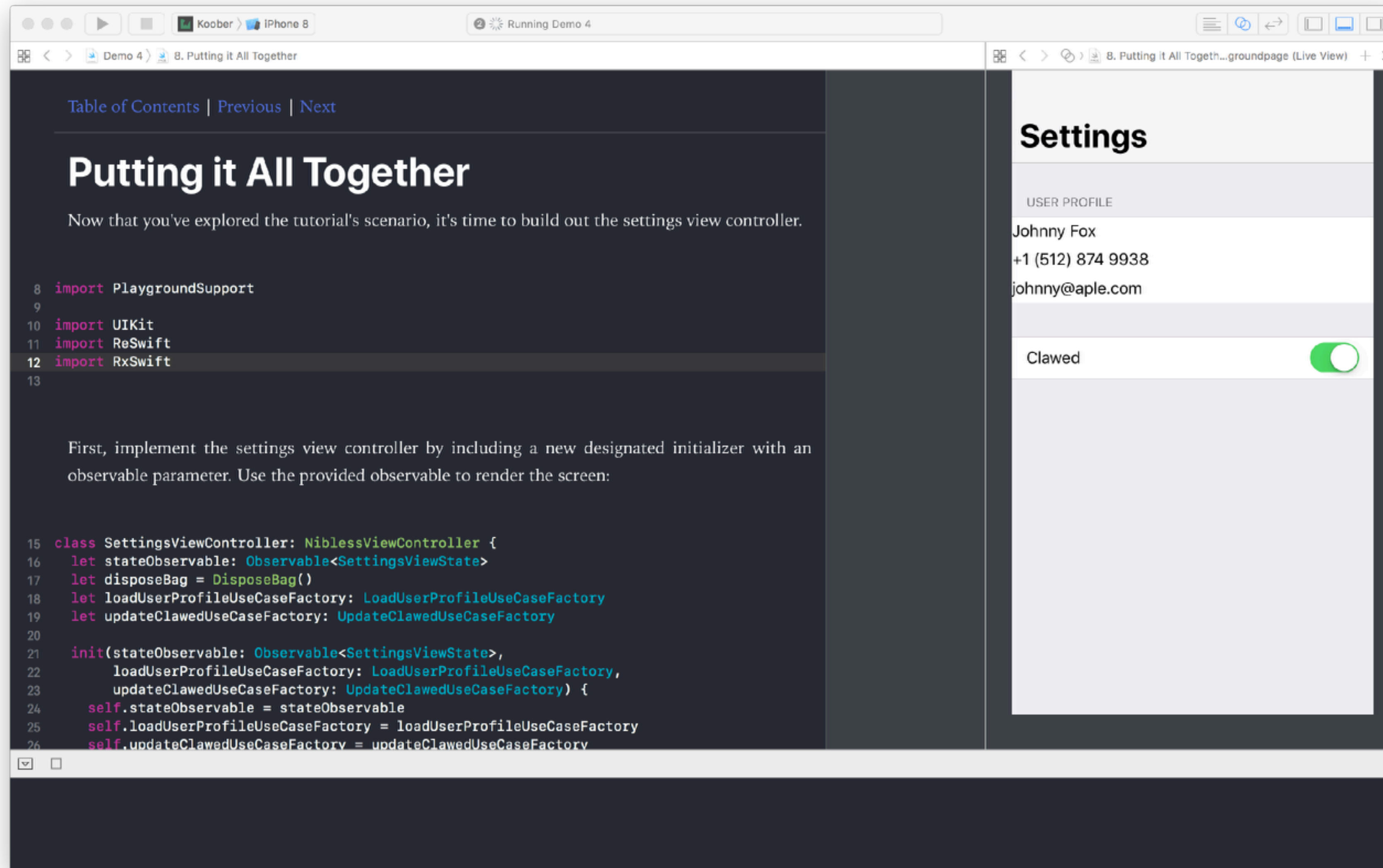


DEMOS

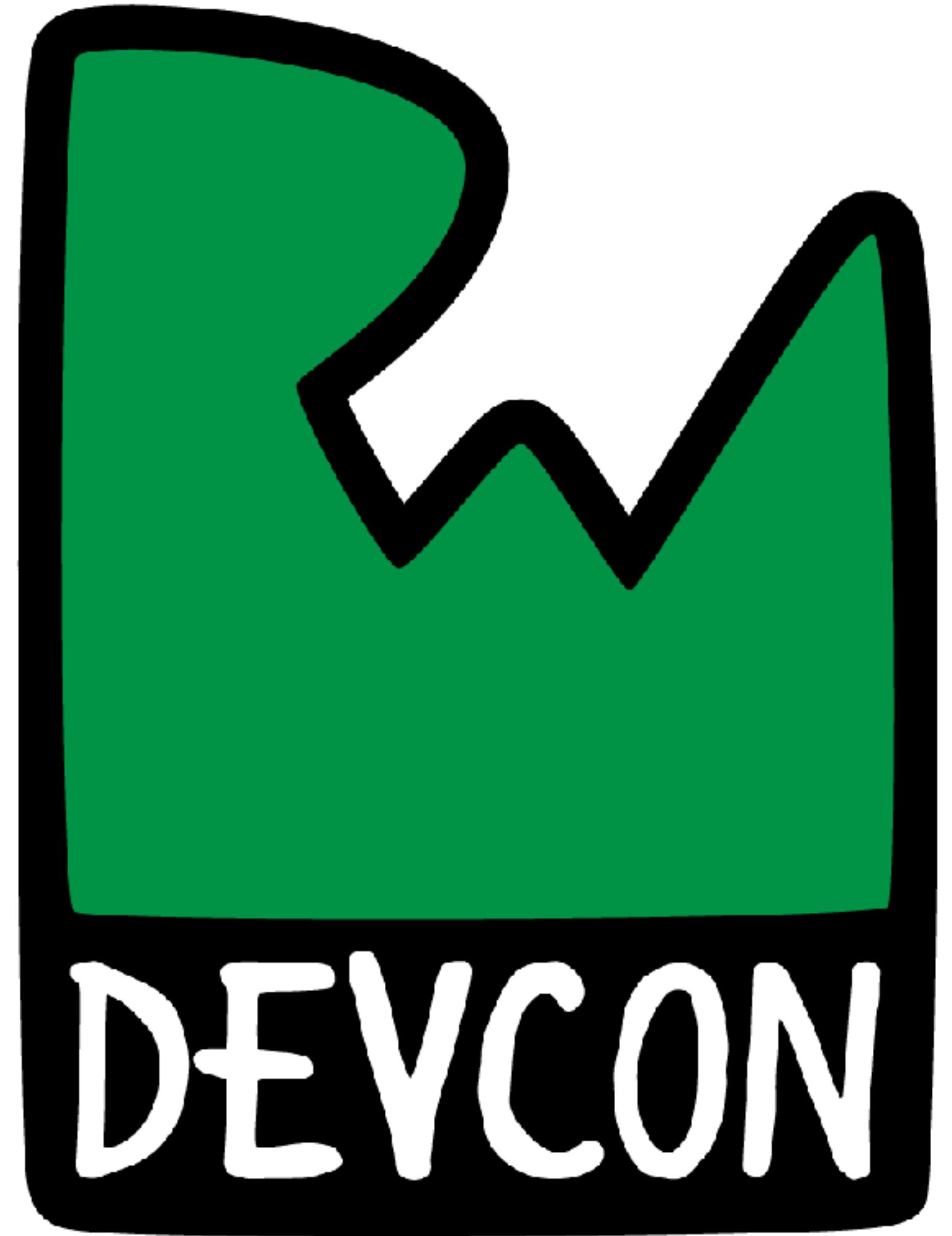
- ⚙ **Demo 1:** Injecting Observables
- ⚙ **Demo 2:** Persistence
- ⚙ **Demo 3:** Networking & User IX



SETTINGS SCREEN

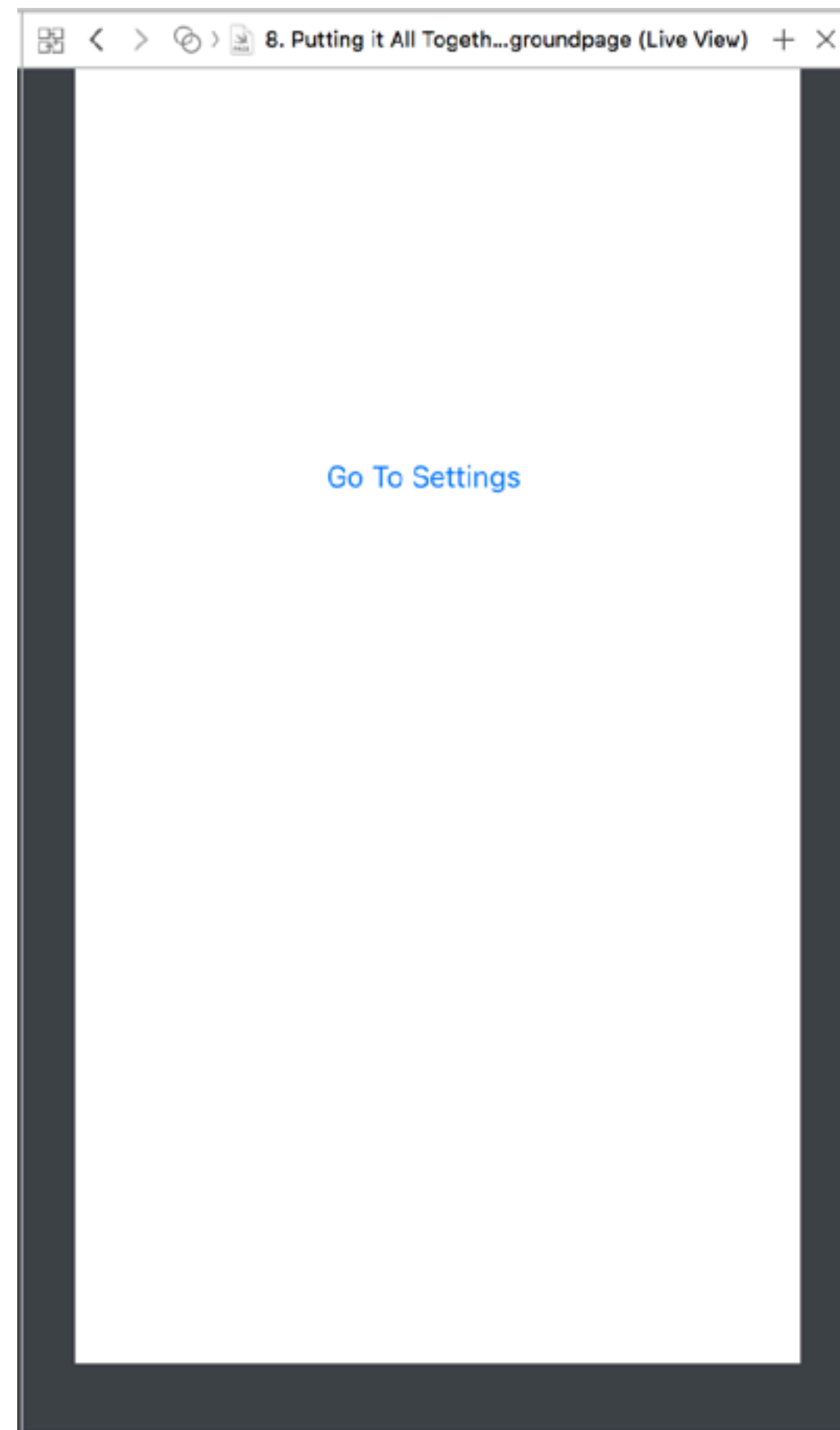


Session 18: Advanced Unidirectional Architecture

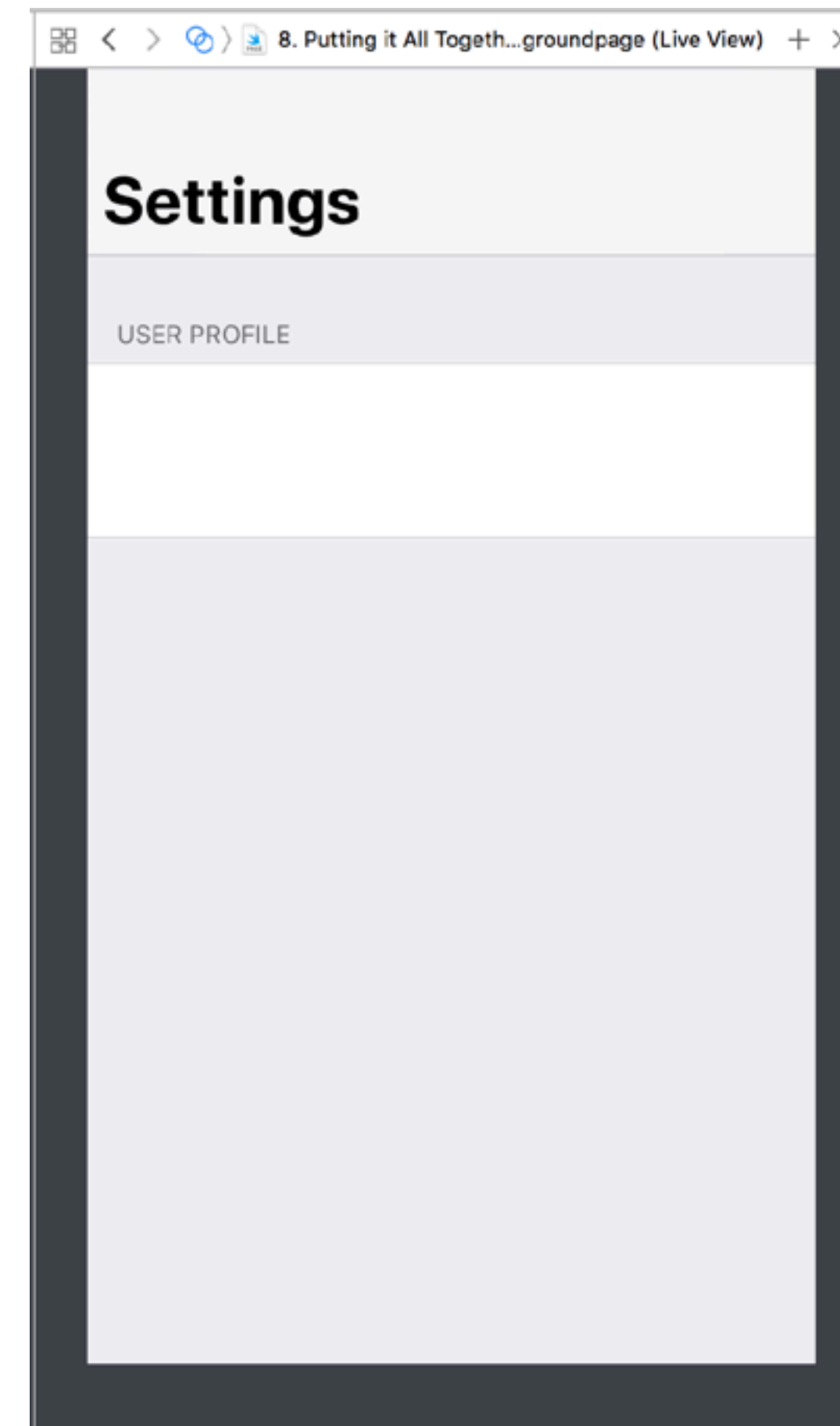


INJECTING OBSERVABLES

FLOW

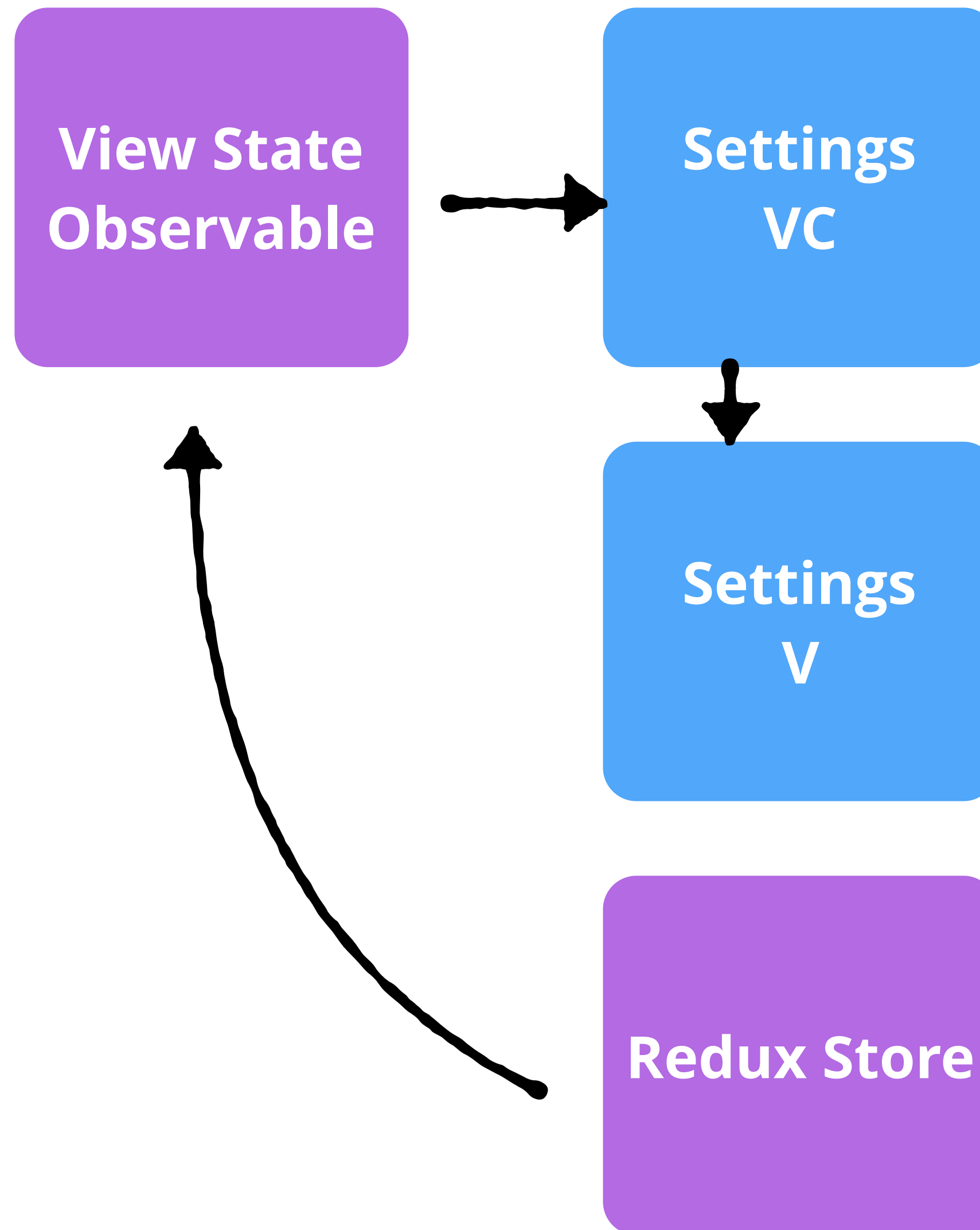


Home View Controller



Settings View Controller

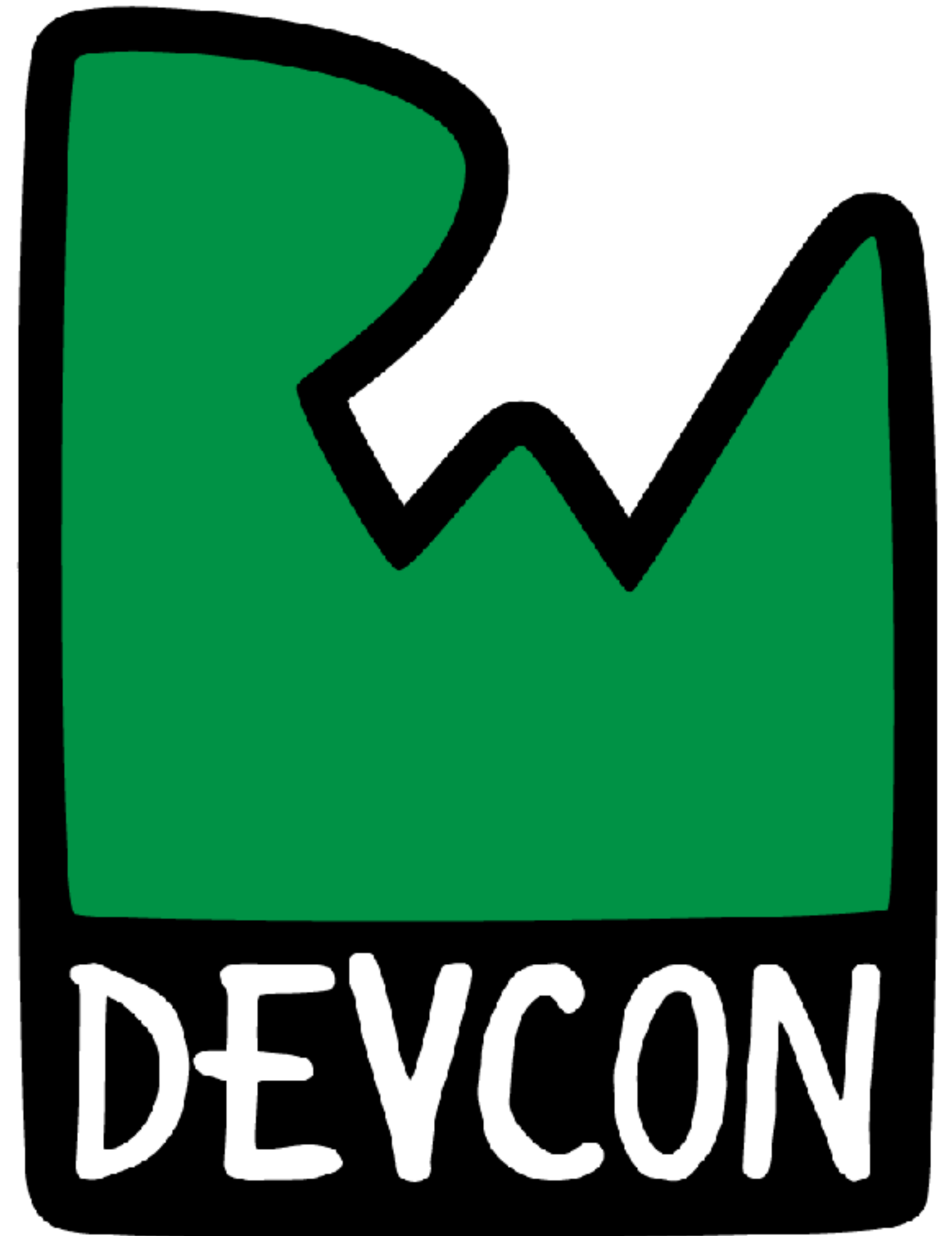
INJECTING OBSERVABLES



DEMO 1

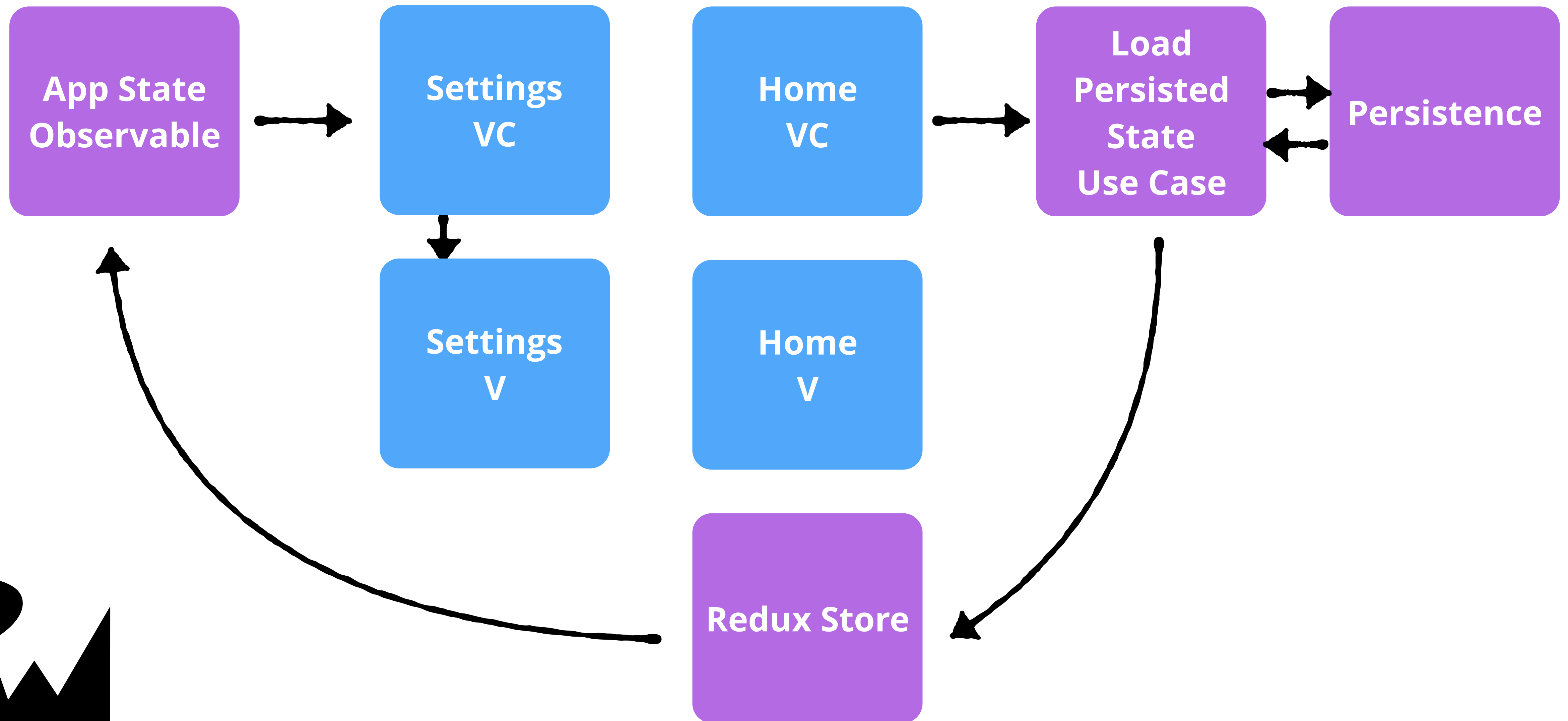


Session 18: Advanced Unidirectional Architecture



ADDING PERSISTENCE

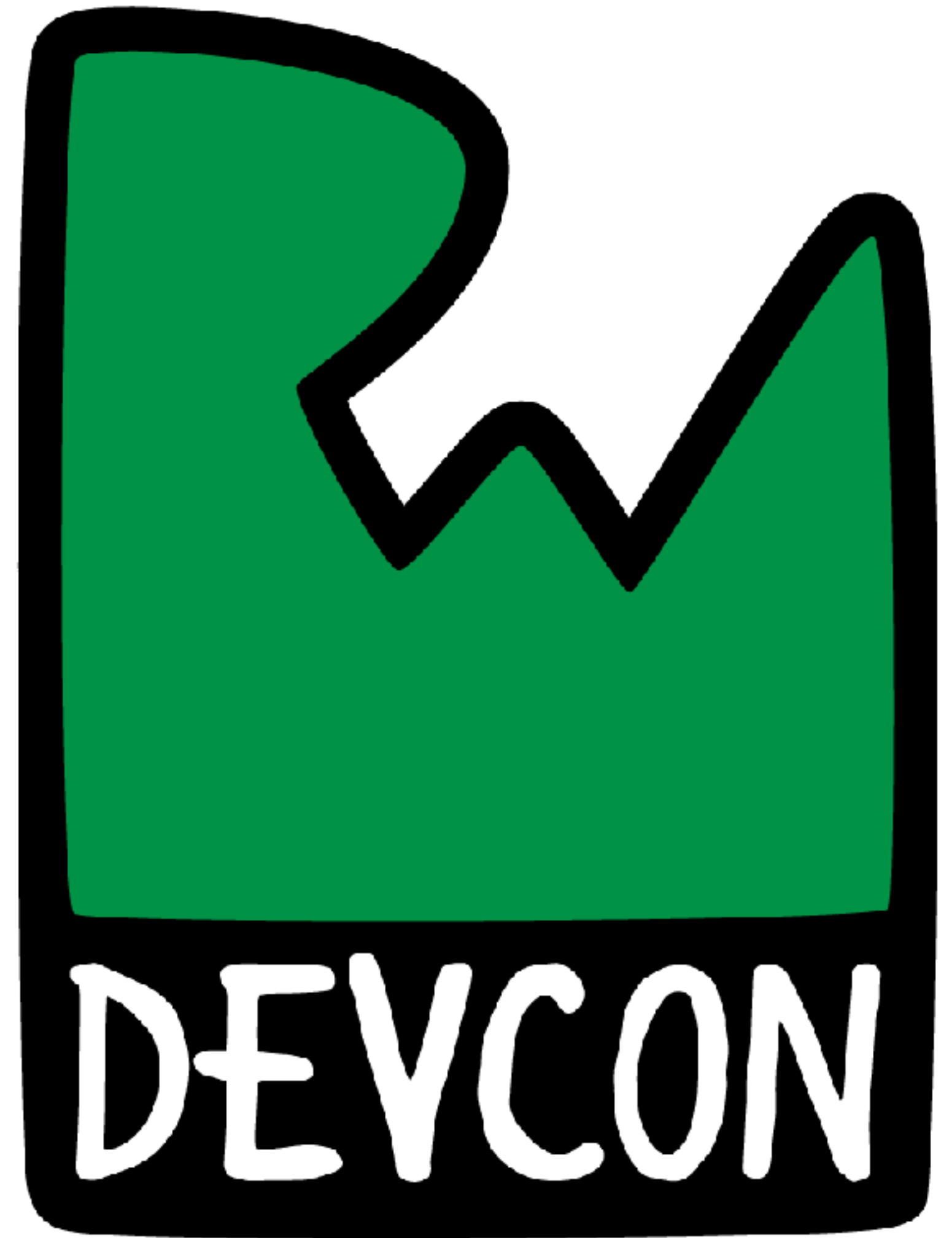
ADDING PERSISTENCE



DEMO 2

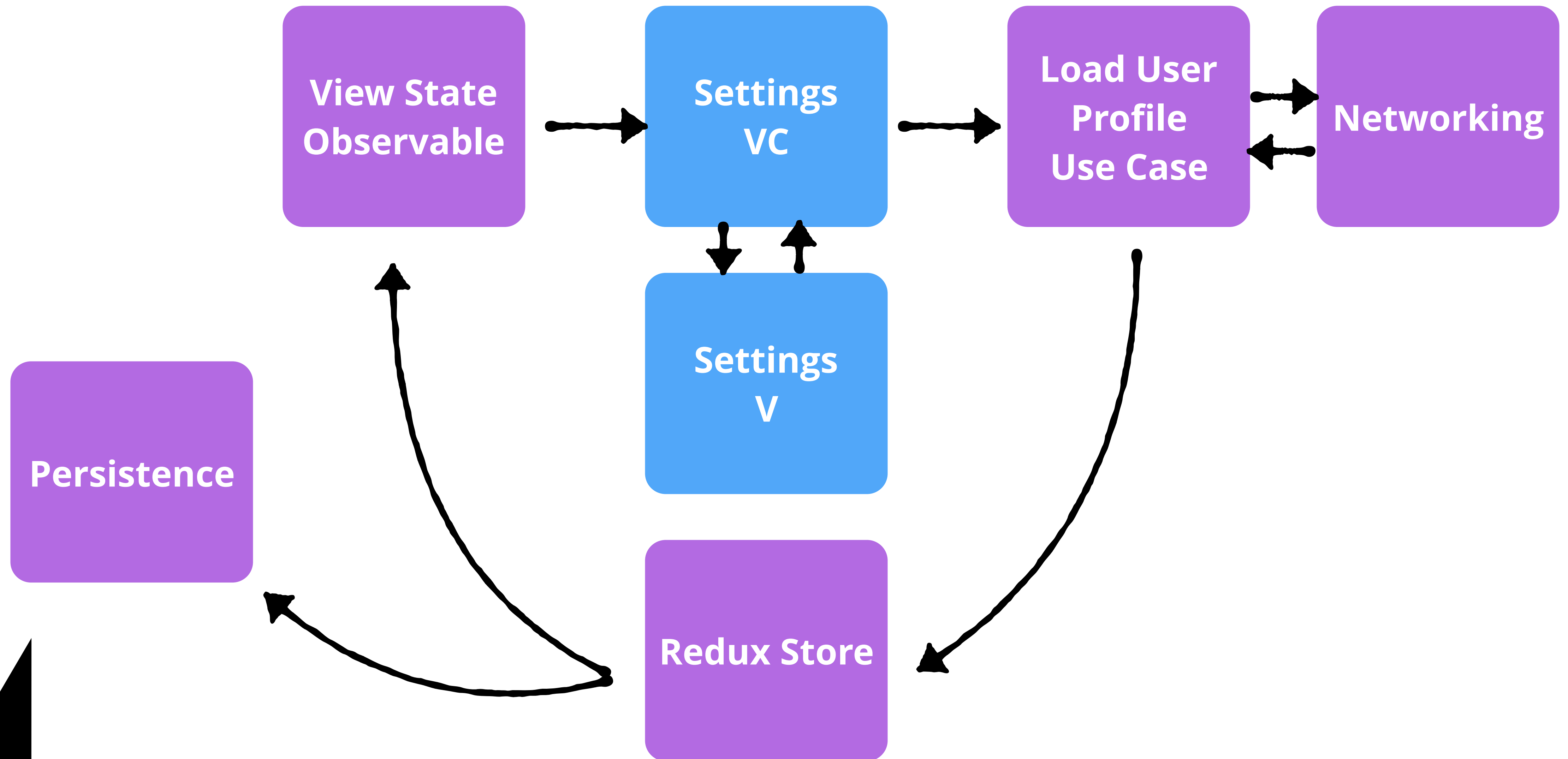


Session 18: Advanced Unidirectional Architecture



NETWORKING & USER IX

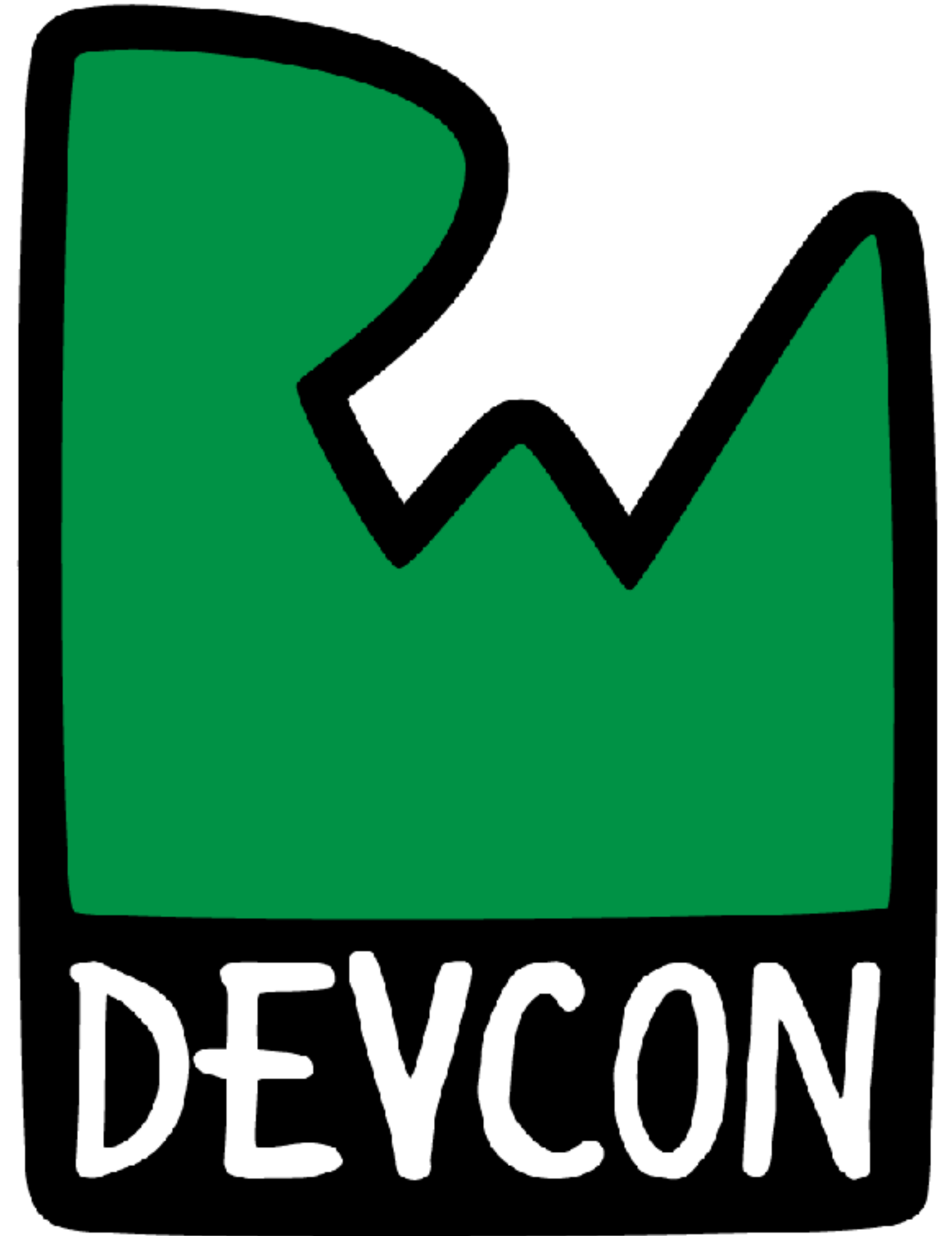
NETWORKING



DEMO 3



Session 18: Advanced Unidirectional Architecture



CONCLUSION

WHAT YOU LEARNED

- ⚙ **Demo 1:** Injecting Observables
- ⚙ **Demo 2:** Persistence
- ⚙ **Demo 3:** Networking & User IX

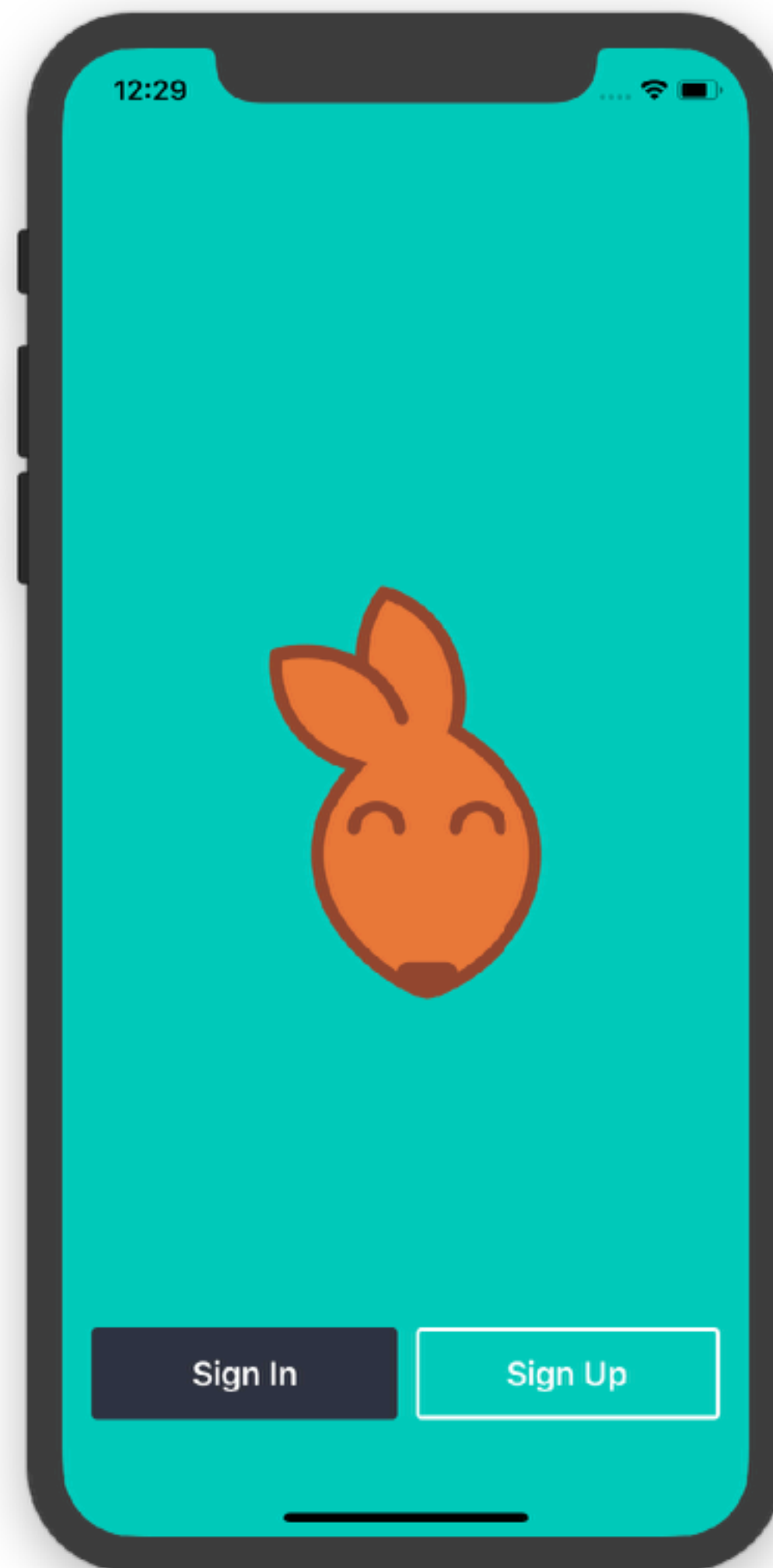


WHY?

- ⚙ Parallel Workstream
- ⚙ Easily change with requirements
- ⚙ Testability
- ⚙ GraphQL



KOOBER



WHERE TO GO FROM HERE?

- ⚙️ [ReSwift at GitHub](#)
- ⚙️ [ReSwift RW Tutorial](#)
- ⚙️ [RxSwift at GitHub](#)
- ⚙️ [RxSwift book](#)
- ⚙️ [Dependency Injection on YouTube](#)
- ⚙️ [Twitter: @rcachATX](#)



THAT'S IT!

