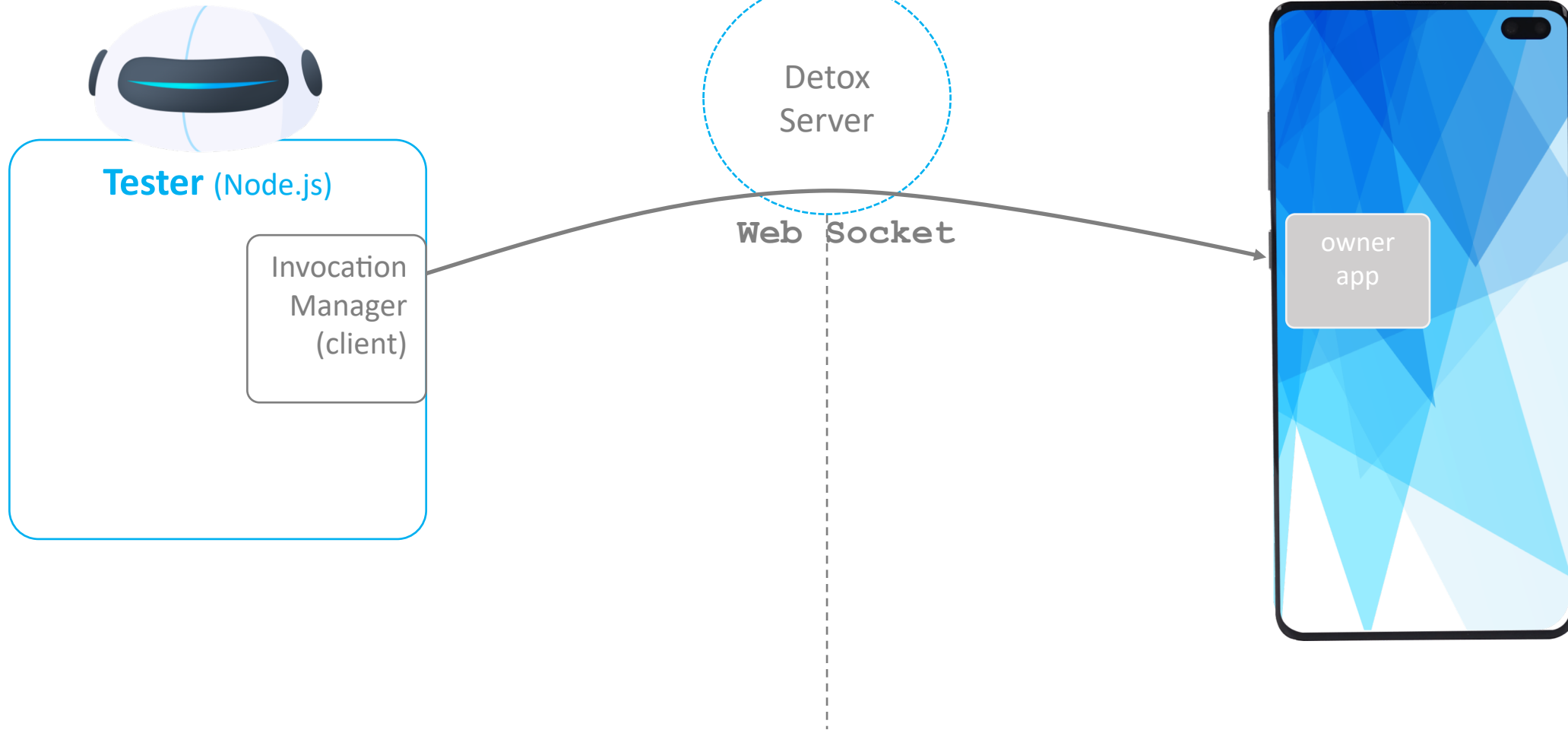


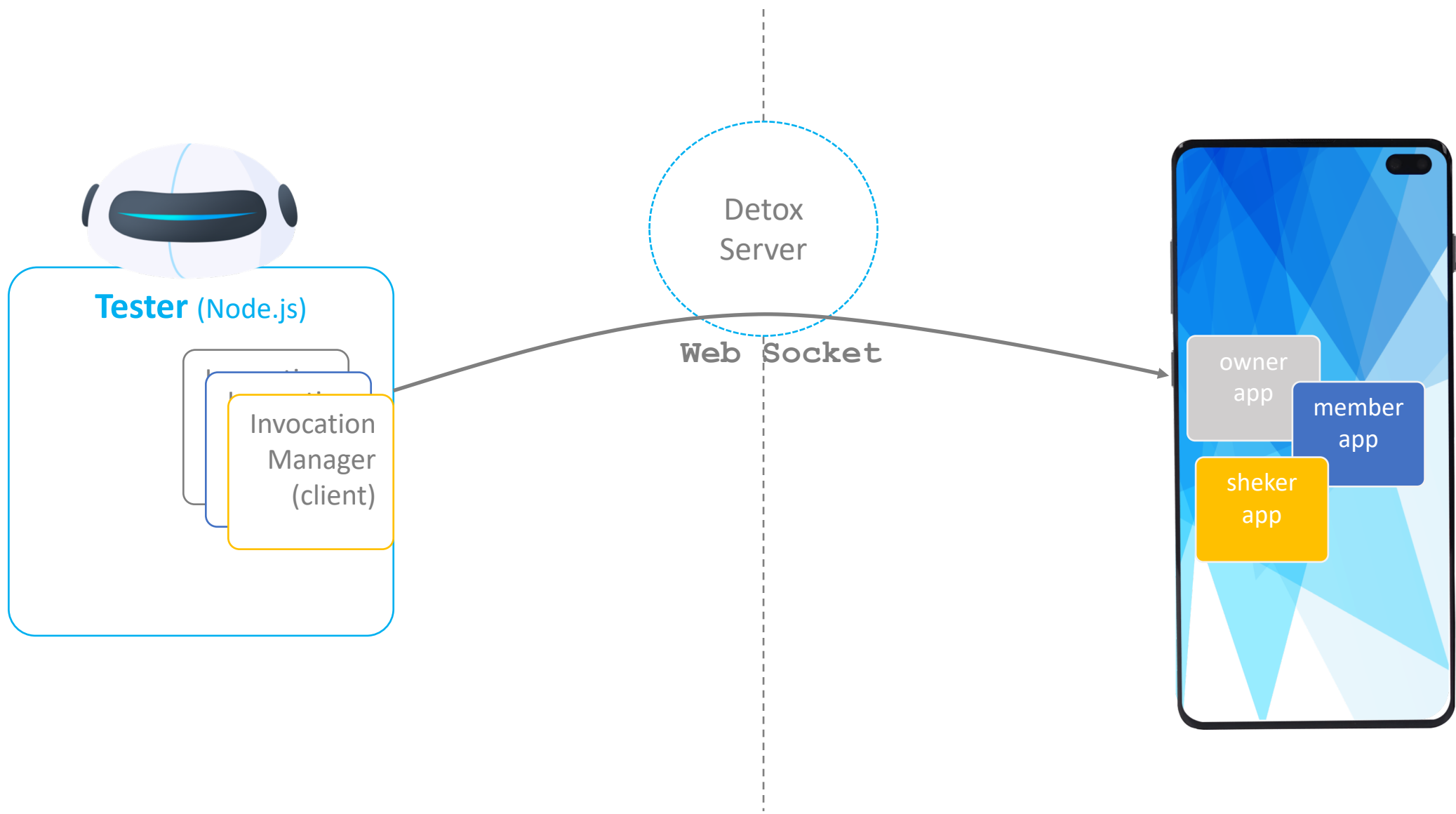
Complexity

Multi-apps state management

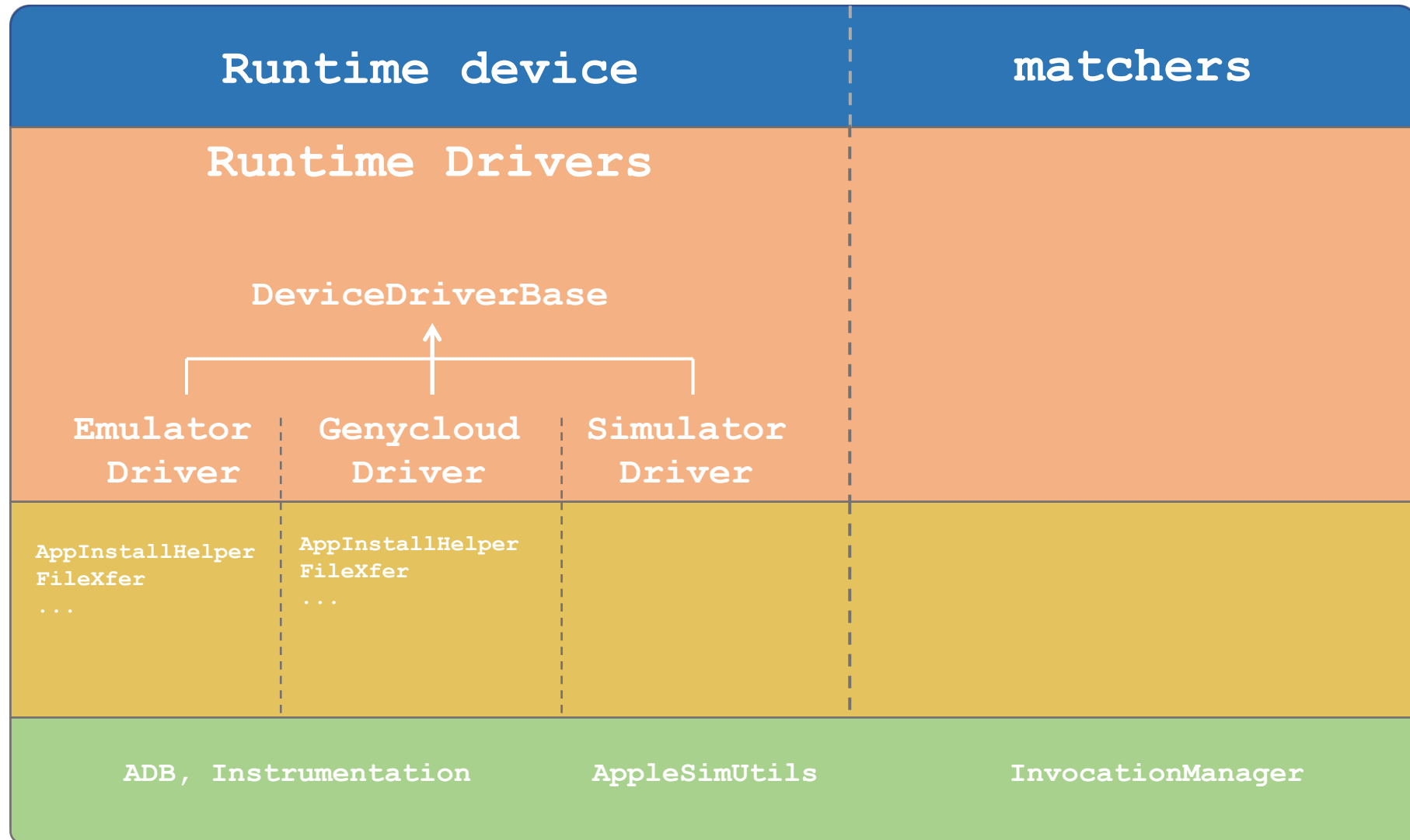
Detox High-Level



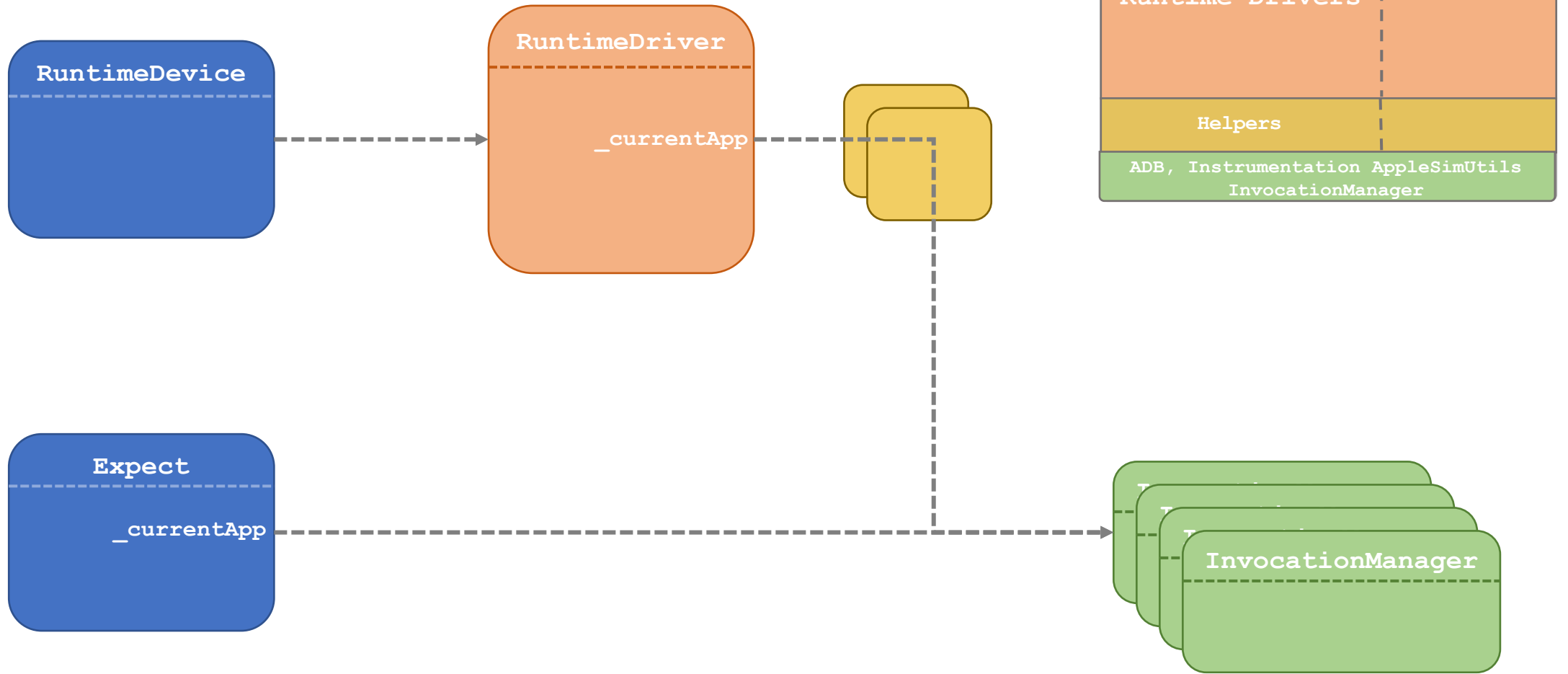
Detox High-Level



Software Architecture



Software Architecture



App Toggling

```
selectApp(alias) {  
  device.selectApp(alias);  
  matchers.selectApp(alias);  
}
```



Solutions

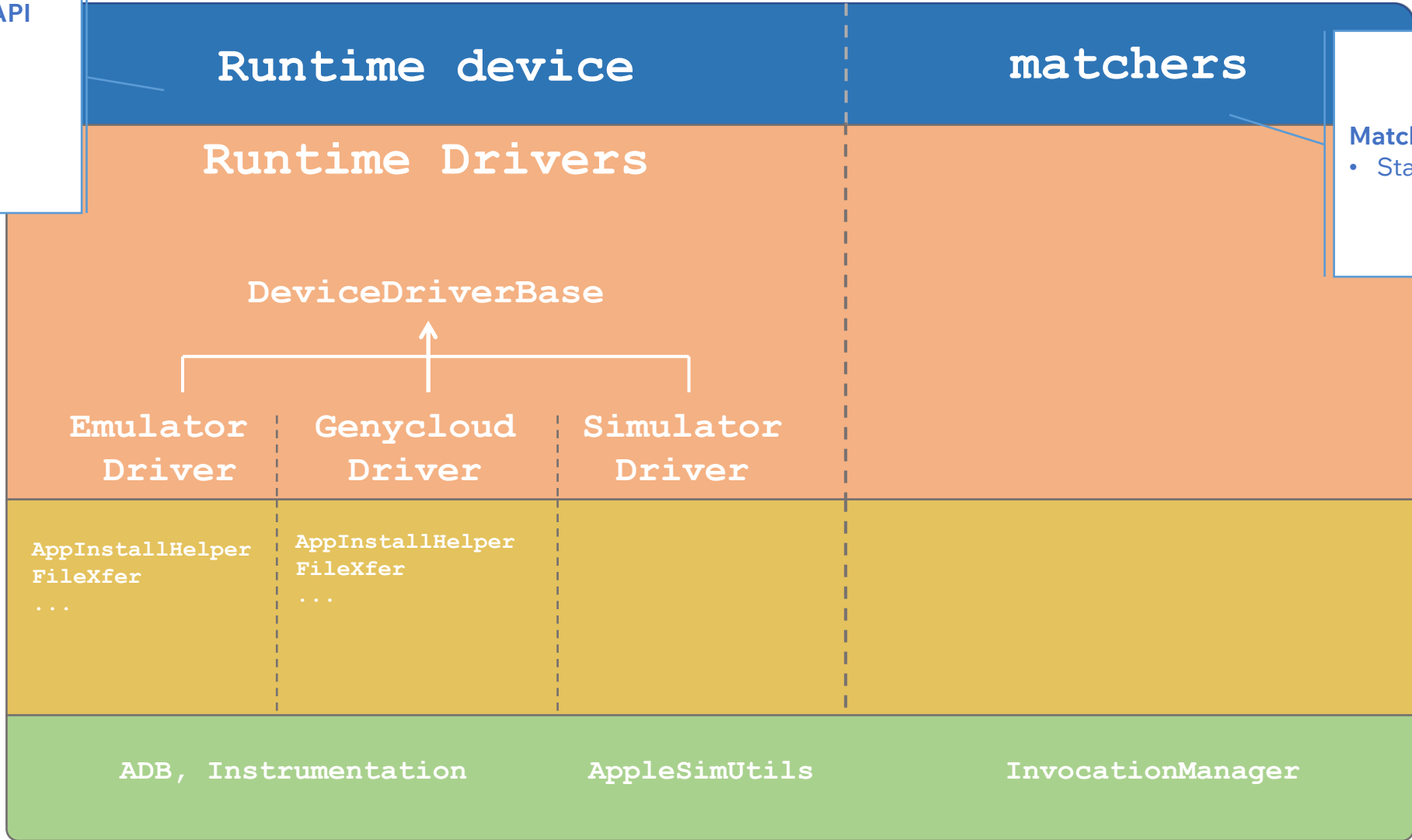
Software Architecture

Device = top-level API

- Stateless

Driver = platform-specific impl.

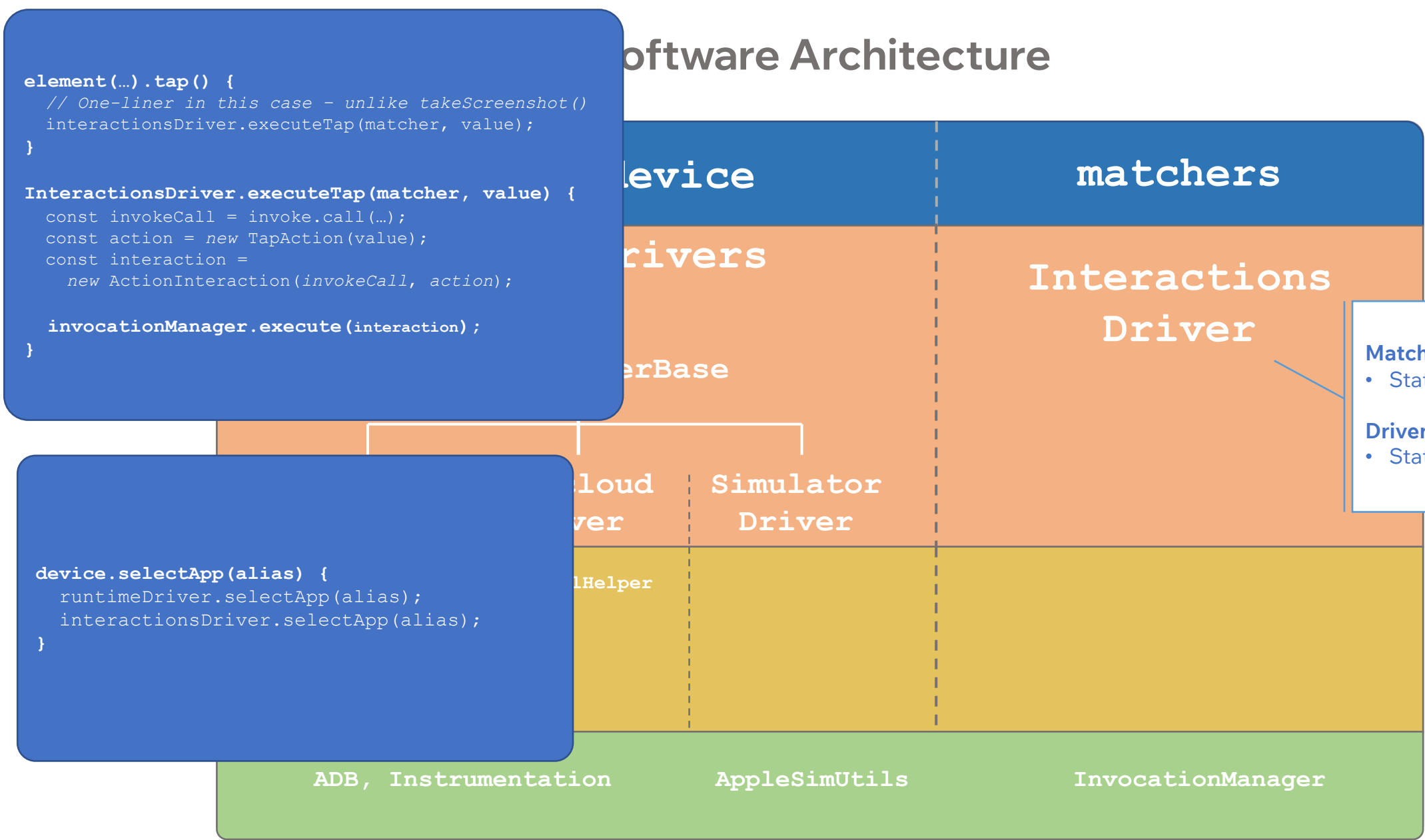
- Stateful



Matchers = top-level API

- Stateful ⚠

Software Architecture



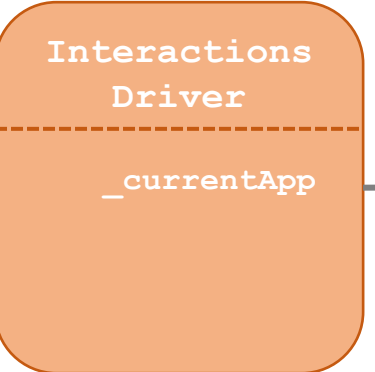
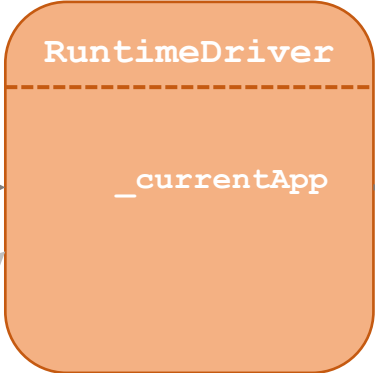
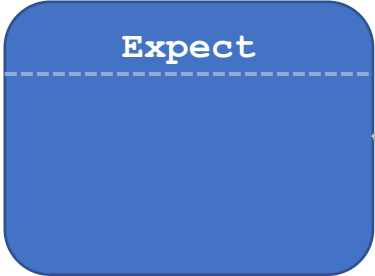
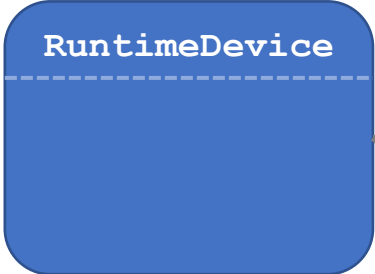
Matchers = top-level API

- Stateless ✓

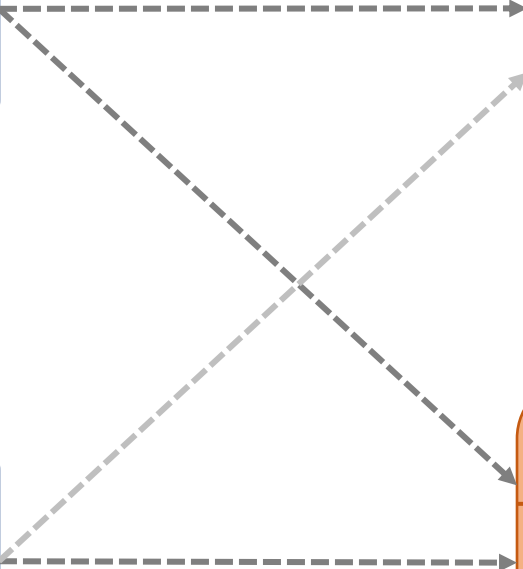
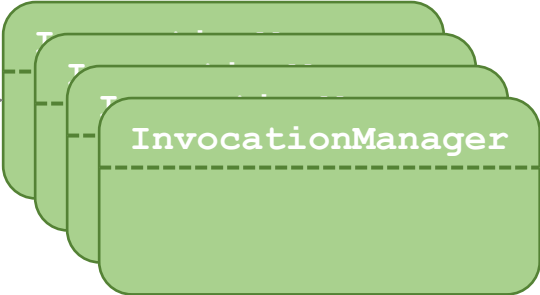
Driver = platform-specific

- Stateful ✓

Software Architecture



Runtime device	matchers
Runtime Drivers	
Helpers	
ADB, Instrumentation	AppleSimUtils InvocationManager



Profit

- Pushes state down
 - Decouples *external* API from *internal* architecture
 - Solves multi-app case
- Matchers: Inter-platform code reuse via inheritance
- Eliminates existing matchers → device relation

Other Solutions

- Connections Abstraction
 - `selectApp()` → `connections.selectActive()`
- Add `App` to main objects model
 - Semantic
- Flux-like
 - Centrally manage apps as a state
 - Entities to observe mutations