
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE
Further Topics of Performance Optimization
Hosted by Alex Thalhammer

Outline

1. Don't use Angular resolvers (if you ask me)
2. Smart vs Dumb Components
3. API Architecture
4. RxJS & NgRx
5. Web Workers for heavy calculations
6. Service Worker / PWA
7. Scheduling
8. Building with Nx

#1: Don't use Angular resolvers

- Better to show the title and everyting possible, even just the frame
- Instead use local spinners where data is being loaded

#2: Thought experiment

- What if <flight-card> would handle use case logic?
  - e.g. communicate with API
- Number of requests ==> Performance?
- Traceability?
- Reusability?


Smart vs. Dumb
Components

#2: Smart vs. Dumb Components

Smart	Dumb
• Use Case controller • Container	• Independent of Use Case • Reusable • Leaf

#3: API Architecture

- Try to minimize API calls
  - E.g. fetch data in list not list item
  - If possible aggregate data in backend, not frontend

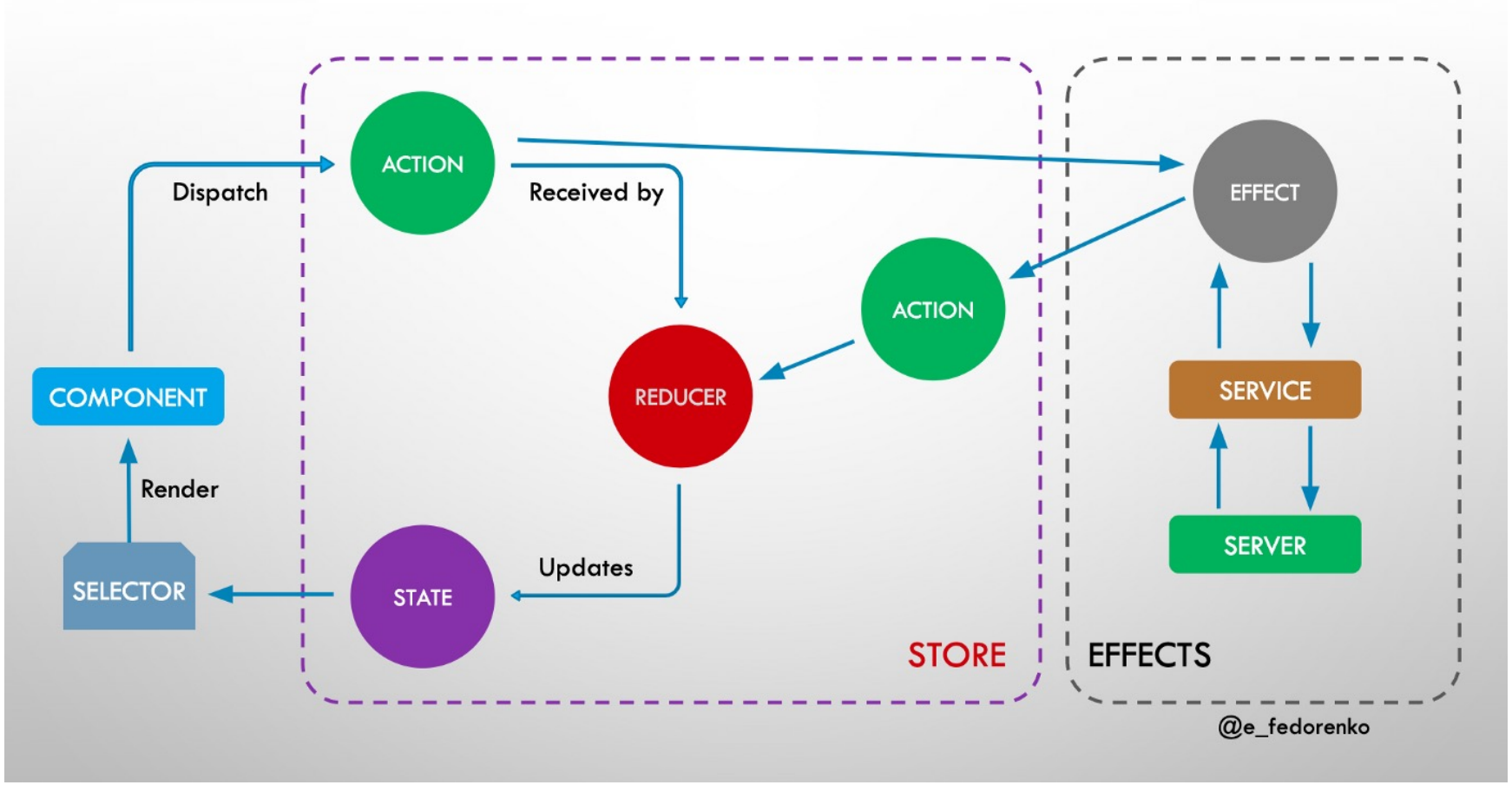
- Think about caching API calls
  - If possible, maybe valid for limited time only

#4: Use RxJS & NgRx

- Use RxJS properly
  - Share hot observables where possible
  - Pipe operaters
  - Use async pipe
  - Manage subscriptions

- Use State Management (NgRx preferred, else NGXS)
  - By using Redux libraries properly, you can improve its performance, by reducing the number of events that occur during data communication

#4: Use state management (NgRx)



https://medium.com/angular-in-depth/how-i-wrote-ngrx-store-in-63-lines-of-code-dfe925fe979b

#5: Web Workers for heavy calculations

- Problem: JS is single threaded, how to do heavy calculations?

- Solution: Delegate to web worker, it will create a new thread called the Worker Thread that will run a JS script parallel to the main thread

#5: Web Workers – Use cases

- Import external scripts
- Make XMLHttpRequest / API requests
- Use setTimeout() and setInterval()
- Spawn other workers
- Use IndexedDB, Notifications API, Web Crypto API, WebAssembly, WebSockets, WebGL, OffscreenCanvas, ImageData…
- Terminate themselves when you deem they are no longer needed
- …

#5: Web Workers – Implementations

- Worklet API
- partytown
- …

#6: Service Workers (PWA)

- Proxy or serving HTTP requests
- Background code execution
- Web push notifications
- Process payments
- …

#7: Scheduling

- Use setTimeout() to delay work
- Use setInterval() to invoke tasks continously
- Don't forget to clearTimeout() and clearInterval() on destroy

#8: Building with Nx

- For bigger / enterprise Apps use @nrwl/nx

- Nx is a 3rd party extension for Angular CLI supporting
  - Monorepo workspace
    - Split App(s) into buildable parts / libs
    - Only recompile changed parts (both during serve & build)
    - Possible to have a cloud build cache
  - Other features like
    - Schematics / generators
    - Access restrictions
    - Dependy graph
  - Out-of-the-box support for JEST & Cypress

Recap

1. Don't use Angular resolvers (if you ask me)
2. Smart vs Dumb Components
3. API Architecture
4. RxJS & NgRx
5. Web Workers for heavy calculations
6. Service Worker / PWA
7. Scheduling
8. Building with Nx