

SHOP.APP

ANGULAR | MATERIAL | NGRX | RXJS

01

Std

Configuration

Std

Typescript

- Option compilateur
 - `"strict": true` ([voir ici](#))
- Variable « path » pour définir des alias pour les @module

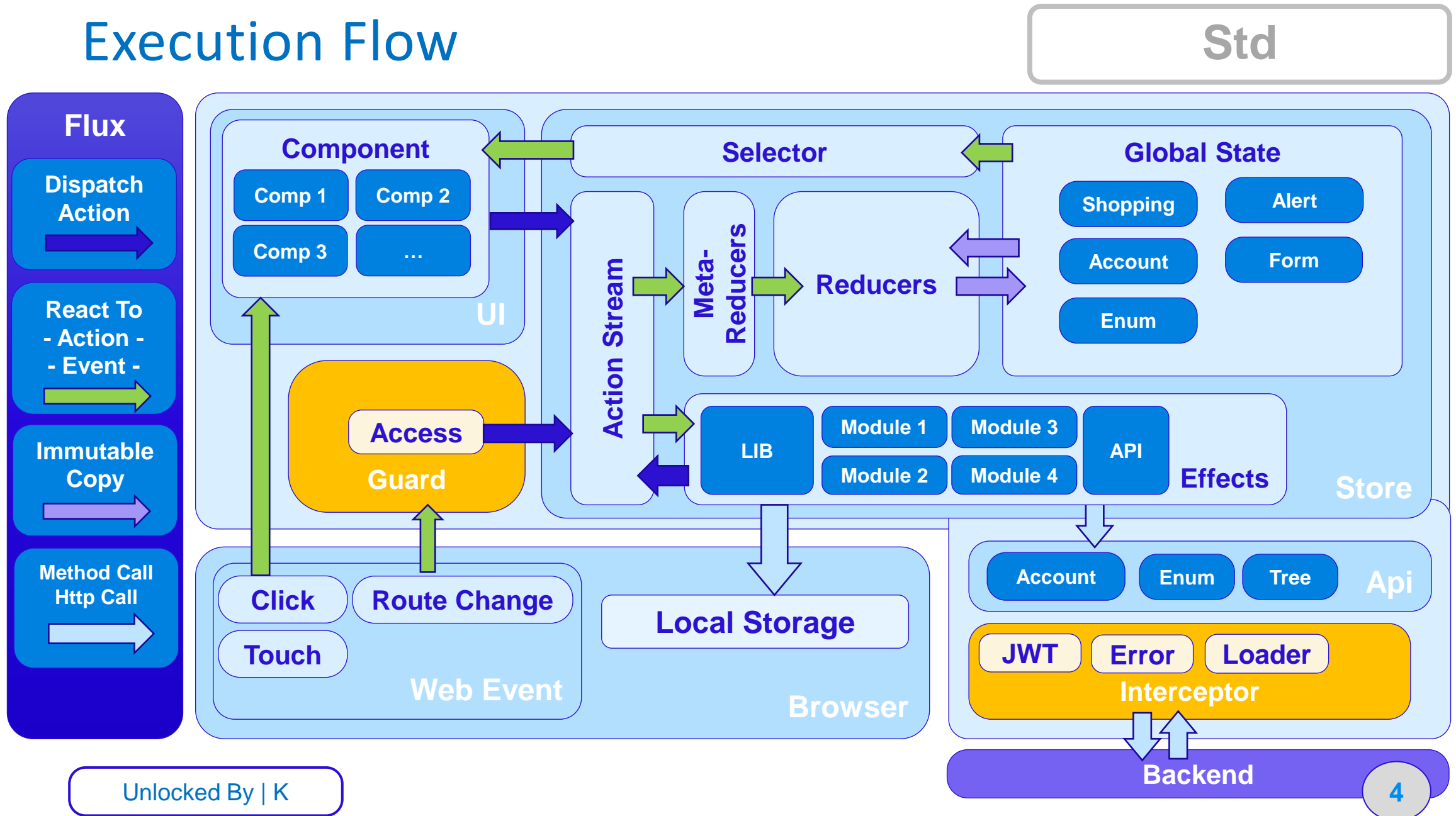
Angular

- ?

Applicative

- **Permettre une configuration selon...**
 - Une instance d'env « Dev / Int / Prd »
 - Un « Site Client »
- **Localisation de la Configuration**
 - Fichier « Environnement.ts »
 - @material ->
- **Élément Configurable**
 - Url Service Business
 - Url Service Tiers (Map, Ged, eSign...)
 - Fichiers (taille...), Timers (autologout...)

Execution Flow



Convention de Nommage

Std

- **Dossier | Fichier**

- Minuscule + Séparateur = « - »
- Nom : <nom-fichier>.<composant>.ts (Ex:)

- **Module**

- **Alias** : Préfixé par « @ »

- **Classe | Object | Type | Variable**

- Casse : CamelCase (maVariable, monObjet)
- Observable : Suffixé par « \$ » (obs\$)
- Private : Préfixé par « _ » (_maVar)

Annotation

- **Todo.** Code à retravailler

Alias

- @alert, @env, @enum, @material, @layout, @form, @loader, @router, @timer, @token
- @core, @account, @shoppingList, @product
- @action,
- @package, @log, @deploy, @style, @lint

- **Base Technique**

- Essentiellement des modules Angular

- **Type**

- Shared. modules partagés
- Core. Cœur de l'application
- Feature. Fonctionnalités

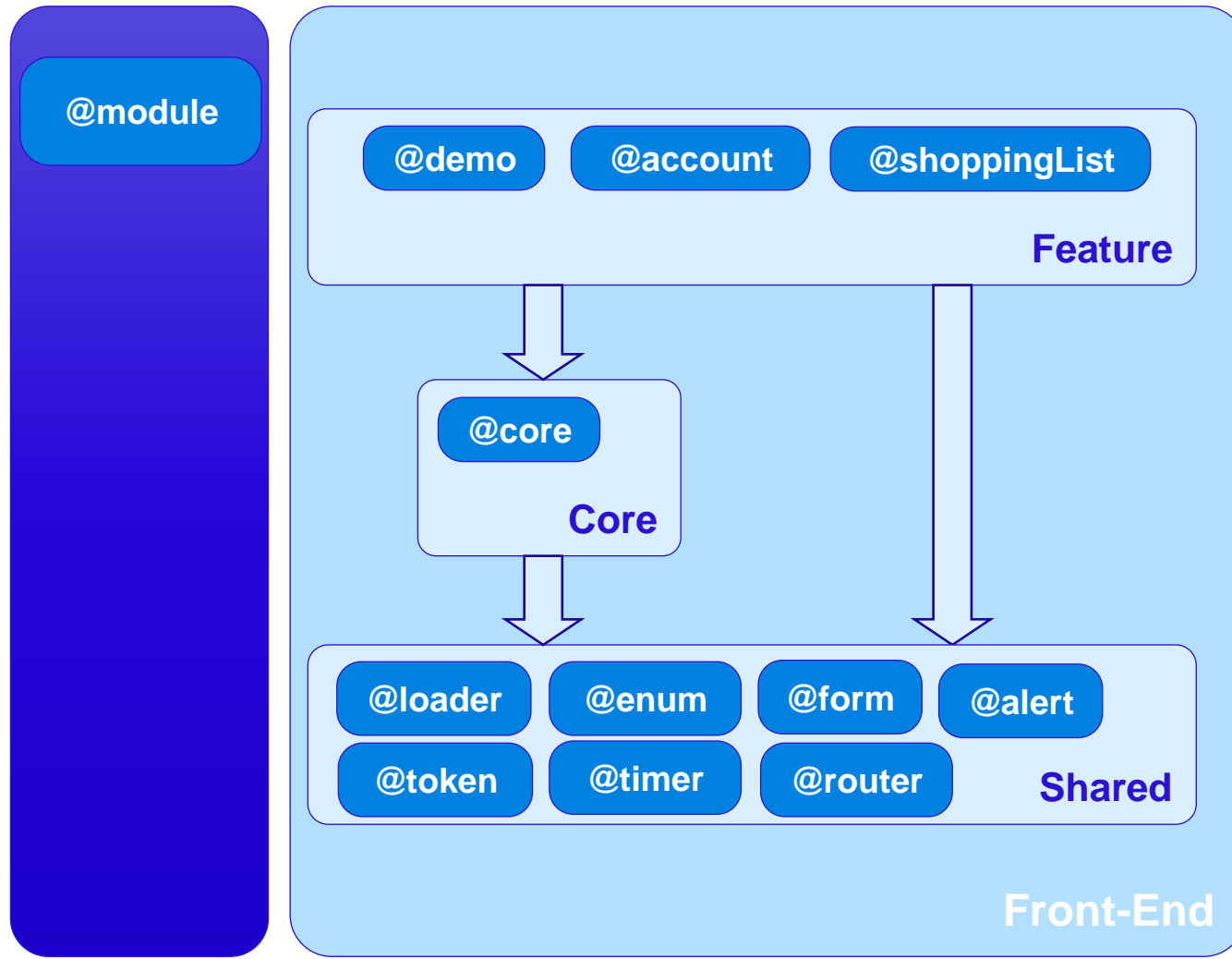
- **Import**

- 1 Alias : @NomModule
- Import * as from <Module> from '<alias>...'
- « index.ts » : Référence Effects, Composants
- Lazy-Loadés -> cf. doc angular

Code Organisation

- **Component**
- **Effect**
- **Guard**
- **Model** (Contient Model & Enum)
- **Page** (/!\ != « Page » dans SPA)
- **Service** (Contient les API)
- **Store**
 - Actions
 - Reducers
 - Selectors
 - State
 - index.ts
- <Feature>-router.module.ts
- <Feature>.module.ts

Module / Dépendance



Std

Architecture

Unlocked By | K

7

Ngrx

Std

Store

- Un fichier index pour référencer ce que le store fourni
- Ngrx rules

State

- Un **global state** (pas de local pour l'instant)
- Data **normalisés** (cf. Redux)
- Ne mettre que des plain object dans le state -> (=> pas de method, d'action...) (cf. Redux)

Reducer

- Comment update le state ? (cf. Redux)

Selector

- ?

Action

- **Nom.** <action><Entité>Action (ex : submitFormAction)
- **Définition.** Ngrx (Version >8.x)
- Emissible par : « **Component** » / « **Api** » / « **Store** »
- Créer au plus près de l'émetteur
- Module @action fournit des helpers
- **Action prédéfinis.** @form / @button

Effect

- **Nom.** Nommer selon ce que fait l'effect
- **Règle.** Manipulating effects
- Gérez l'activation de l'effect via
 - **ofType()** pour filtrer selon le type de l'action
 - **Filter(Fn)** pour le payload de l'action ou autre
 - withLatestFrom si besoin du state

Angular

Std

Module

- **Nom.** `<Xxxx>Module` (Ex : `FormModule`)
- **Import.** `Effect` | `Store` | `Component`

Composant

- **Selector**
 - Dans « **Shared** »
 - **Model.** `<k-<module>-<composant>>`
 - **Ex.** `<k-form-field-input>`
 - Dans « **Feature** ».
 - **Model.** `<<feature>-<composant>>`
 - **Ex.** `<demo-alert>`
- **Héritage**

- **Module.** Load method -> lazyloaded
- **Router.** Sync avec `ngrx-router-store`
- **Form.** `Reactive Form` | `Forms` (NON)
- **I18n.** `Tag...` (à creuser)
- **Accessibilité.** `aria-label...` (à creuser)
- **Resolvers.** (à creuser)

Autre

- **Service (DI)**
- **Router**
- **Guard**
- **Directive**
- **HTTP Client**
- **HTTP Interceptor**
- **Pipe**
- **Resolver**

A Creuser

- **Web Worker**
- **Service Worker**
 - Pour PWA !
- **Angular Universal**

Autres Fonctionnalités

Std

Layout

- **Library.** Angular-flex-layout
 - UI Example | otherDoc
 - Declarative API | Responsive API
- Chaque feature HomePage à la responsabilité de son layout (à noter que le root Component d'un module se trouve de facto dans un Flex/Column)
- Distinction de 2 usage (Choix Conception) :
 - Mobile : -> **.xs**
 - Le reste -> **.gt-xs**
- Implémentation
 - Soit dans un composant commun ac flex-layout

Code Quality

- **Analyse Statique** (Use Sonar ?)
 - Formatage du code
 - Utilisation Prettier & ESLint

Log

- **Library.** Ngrx-store-logger
- **TODO**

Environnement Supporté

- **Browser.** Chrome, Edge, Firefox, Opera ?
- **Device.** PC, Mobile, Tablette ?
- **Ecran.** Longueur, largeur, définition, résolution ?

Package (See « *package.json/dependencies* »)

Std

Angular

- @angular/common => http
- @angular/core
- @angular/forms
- @angular/material
- @angular/material-luxon-adapter
- @angular/router
- @angular/animation

Ngrx

- @ngrx/effects
- @ngrx/entity
- @ngrx/router-store
- @ngrx/store
- @ngrx/store-devtools (DEV Only)

Autre

- DateTime | TimeZone. luxon / intl (JS)
- Font | Icon. fonts.google
- Layout. angular-flex-layout
- Object Normalizer. normalizr
- Utils (Array, Math...). lodash

Ngrx Tiers Library

- Debug. ngrx-store-freeze (DEV Only)
- Form. ngrx-forms
- Logger. ngrx-store-logger
- ORM. ngrx-entity-relationship
- State Sync. ngrx-store-localstorage

Unlocked By | K

Library Status

- intégré | majeur | à creuser

Merci !

Angular

- **Module**
- **Component** (.html, .css, .ts)
- **Service** : Injectable
- **Model** | Enum
- **Guard** | **Interceptor**

NgRx

- **Store** | **State**
- **Action** | **Reducer** | **Selector** | **Effect**,
- **Entity** | **Adapter**
- **Router-store** | **Store-devtools**

Material

- **mat-form-field**
- **mat-icon**
- **mat-button**
- ...

Rxjs

- **Observable** / **Subject**
- **tap**, **map**, **filter**, **of**, **first**
- **switchMap**, **exhaustMap**...

Html

- **Balise, Attribut** `<balise></balise>`
- `<head><body>`
- `<a>`, `<p>`, `<button>`, `<video>...`

Css

- **Selector, attribute, class**
- **Pseudo-elt** (`:`), **Pseudo-class** (`::`)
- **Display.** inline, block, ***flex*** / ***grid***...
- **Unité.** px, em, rem, %, fr
- **Flex.** container / direction / wrap / grow...
- Position / margin / padding...

Typescript

- **Variable, type, enum**
- **Function**
- **Class, Object, Property**
 - Cloner un objet
- **Interface & Héritage**
- **Public | Privé**
- **Modifiers** (readonly, optionnal...)
- **Module**
- **Decorator** (~Annotation de classe)
- ...