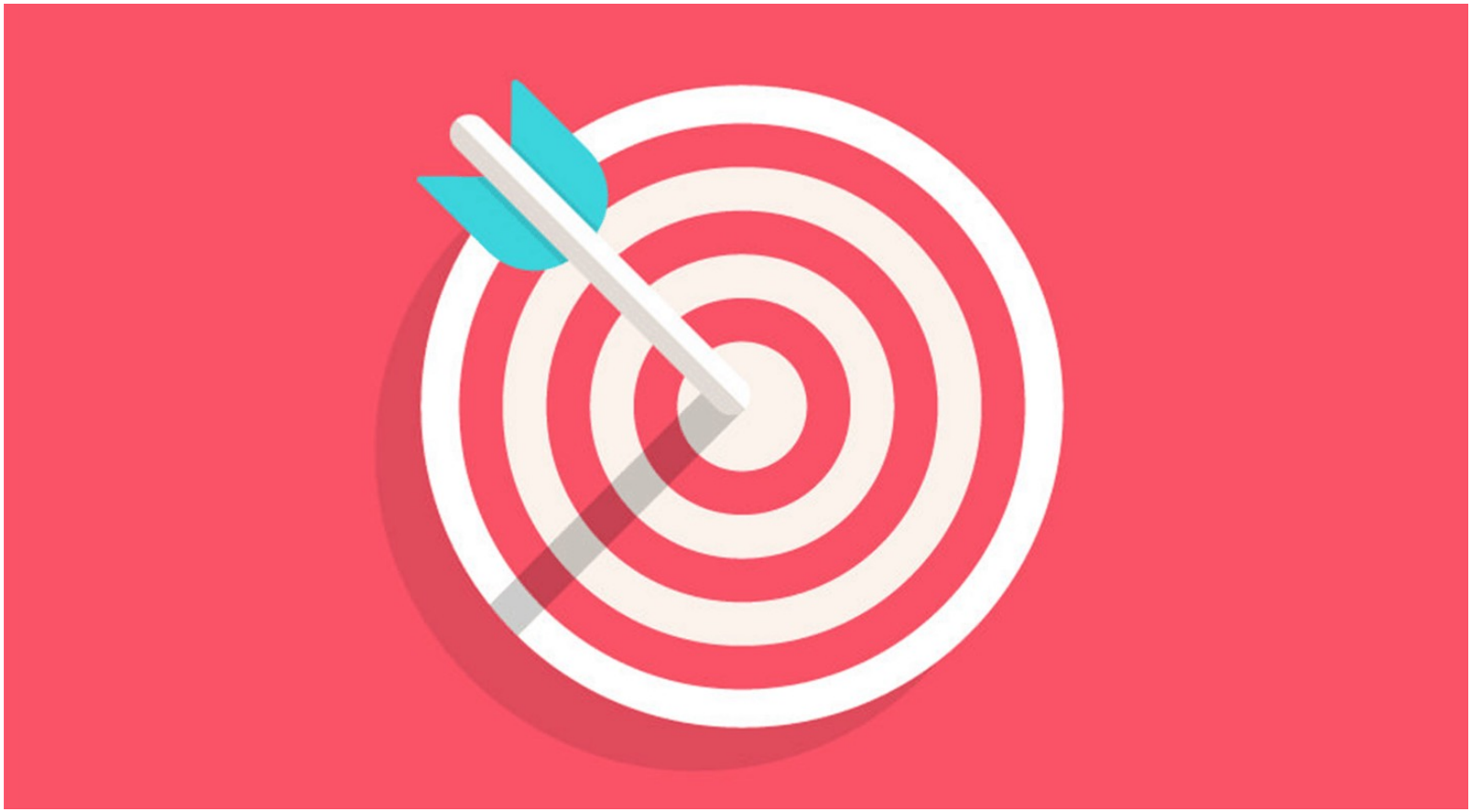
```
1
2 <h1>
3         Techno Web
4 </h1>
5
```

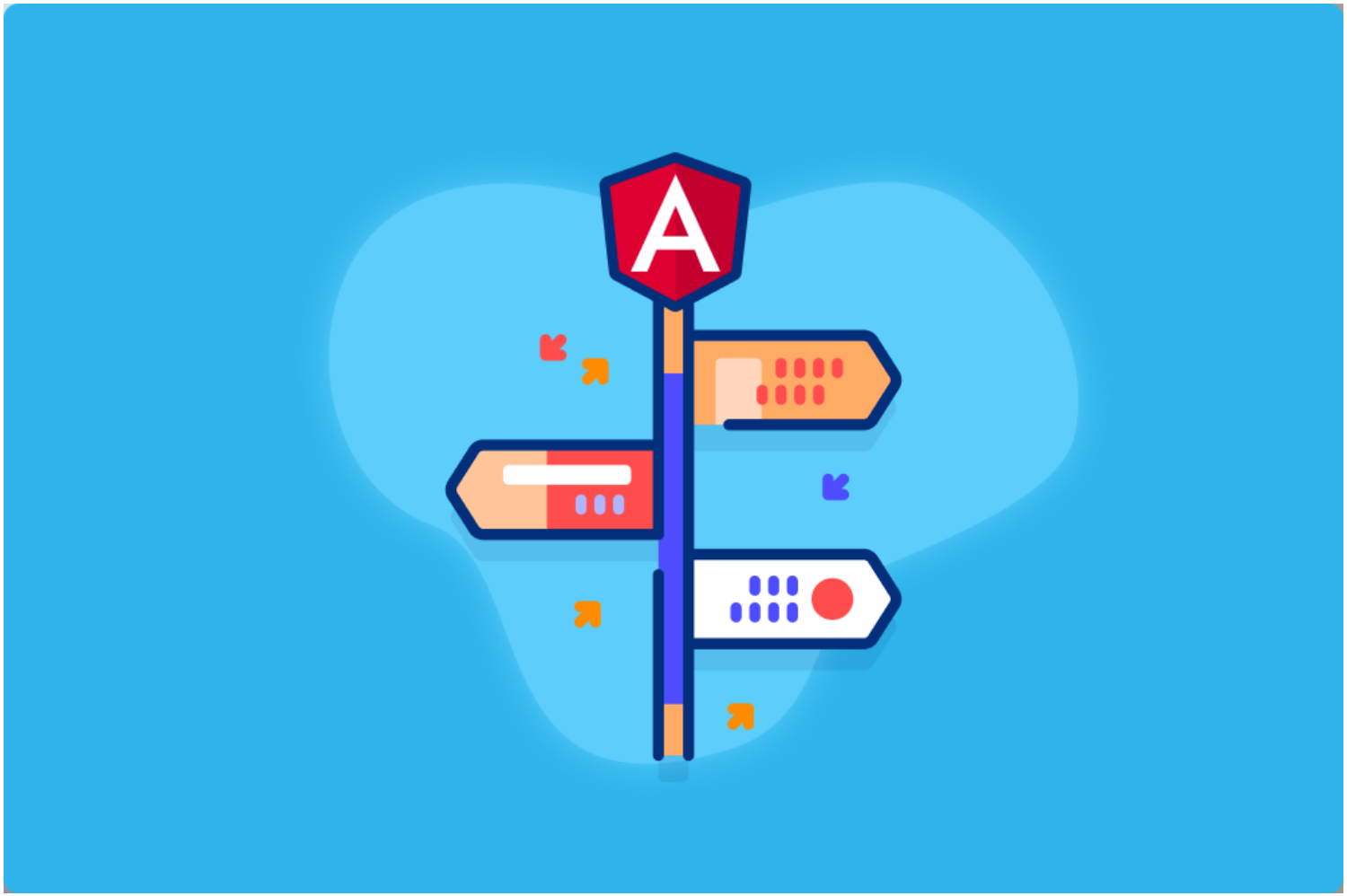
2 Cour n°4



3 Rappel

4 Go voir le code

5 Routage



6 Routage


Home
Authors
Books
Welcome to my library
You can nvaigate per author or per title
Authors
Patrick Rothfuss
Brandon Sanderson
Andrzej Sapkowski
Robert Jordan
Books
Name of the Wind - Patrick Rothfuss
The Wise Man's Fear - Patrick Rothfuss
Words of Radiance - Brandon Sanderson
Oathbringer - Brandon Sanderson
The Last Wish - Andrzej Sapkowski
The Eye of the World - Robert Jordan
Patrick Rothfuss
Nationality: American
Birth year: 1973
Books
Name of the Wind
The Wise Man's Fear
Name of the Wind
Patrick Rothfuss
DAW Books
2007

7 Routage

```
ng generate module app-routing --flat --module=app
```

*Par convention, le nom de classe du module est **AppRoutingModule** et il se situe dans le fichier **app-routing.module.ts** dans le dossier **src/app**.*

8

L'AppRoutingModule généré ressemble à ceci :

```
1   import { NgModule } from "@angular/core";
2   import { RouterModule, Routes } from "@angular/router";
3
4   const routes: Routes = [];
5
6   @NgModule({
7     imports: [RouterModule.forRoot(routes)],
8     exports: [RouterModule]
9   })
10  export class AppRoutingModule {}
```

9

```ts
const routes: Routes = [
  {path: 'registration', component: RegistrationComponent},
  {path: 'forgotten-password', component: ForgottenPasswordComponent},
  {path: 'login', component: LoginComponent},
  {path: 'dashboard', component: DashboardComponent, canActivate: [Authenticatio
  {path: '', pathMatch: 'full', redirectTo: '/dashboard'},
  {path: '**', redirectTo: '/dashboard'}
]
```

• **canActivate** vous permet de définir des route guards. Un route guard bloque l'activation de la route si la condition qu'il définit n'est pas vérifiée.

• **pathMatch:** 'full' force le chemin à être comparé à l'URL entière. Il est important de le faire lors de la redirection des routes à chemin vide. Sinon, parce qu'un chemin vide est un préfixe de n'importe quelle URL, le routeur appliquerait la redirection même lors de la navigation vers la destination de redirection, créant une boucle sans fin.

• **'**':** est une wildcard qui signifie que le chemin correspond à n'importe quelle URL

10 Routage

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { AccueilComponent } from './accueil/accueil.component';
import { AboutComponent } from './about/about.component';
import { ApprenantsComponent } from './apprenants/apprenants.component';
import { LoginComponent } from './login/login.component';
import { ContactComponent } from './contact/contact.component';
```

```
const routes: Routes = [
  { path: 'accueil', component: AccueilComponent },
  { path: 'about', component: AboutComponent },
  { path: 'login', component: LoginComponent },
  { path: 'apprenants', component: ApprenantsComponent },
  { path: 'contact', component: ContactComponent },
  { path: '', redirectTo: '/accueil', pathMatch: 'full'}
];
```

```
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

11 Conclusion

Angular *est un* ***framework*** *de développement* ***d'applications web*** *qui utilise le* ***routage*** *pour* ***afficher*** *différents* ***composants*** *en* ***fonction de l'URL actuelle de l'application***.

Le ***routage*** *est un* ***mécanisme*** *clé pour* ***créer*** *des applications à* ***page unique (SPA)*** *dans Angular*.

12 Conclusion

Voici les étapes de base pour mettre en place le routage dans une application Angular:

- ***Installez le module de routage*** *Angular en exécutant la commande suivante dans votre terminal:* ***ng add @angular/router***

- ***Dans votre application****, créez un* ***fichier de configuration*** *de routage nommé* ***app-routing.module.ts****. Ce fichier* ***définit les routes*** *pour l'application.*

- ***Importez les composants*** *que vous souhaitez* ***afficher*** *dans les routes. Vous pouvez utiliser* ***des composants existants*** *ou en* ***créer de nouveaux*** *pour cette fonctionnalité.*

13 Conclusion

- *Dans le fichier **de configuration de routage**, définissez les **routes** pour **chaque composant**. Par exemple, vous pouvez définir une route pour le **composant "Home"** pour l'URL **racine (/)** et une route pour le **composant "About"** pour l'URL **/about.***

- *Dans votre application, ajoutez un élément **router-outlet** à votre fichier HTML racine. Cet élément **affiche** les **composants** correspondants **en fonction de l'URL.***

- *Dans votre application, ajoutez des liens pour naviguer entre les différentes pages. Vous pouvez utiliser la **directive routerLink** pour lier les liens aux routes de votre application.*

14 Qu'est-ce qu'un observable ?

15 Qu'est-ce qu'un observable ?

Un observable *est, par définition,* ***une chose que l'on souhaite observer****.*

L'objectif *étant de* ***voir son état au fil du temps*** *et de recueillir toutes les modifications qu'elle subirait.*

À la ***différence*** *des promesses, les observables peuvent donc* ***changer un nombre infini de fois de valeur.***

Un ***observable*** *peut être* ***observé puis libéré, puis observé****, alors qu'une promesse ne peut être annulée.*

Les ***observables*** *sont* ***un flux de données*** *auquel* ***on peut souscrire****.*

L'ensemble du processus est bien divisé en deux parties distinctes : l'observable et l'observateur qui souscrit à l'observable.

16 Qu'est-ce qu'un observable ?

Observable *est une classe TypeScript située dans un paquet de la librairie rxjs.*

Celle-ci permet d'utiliser l'asynchrone et les observables plus facilement.

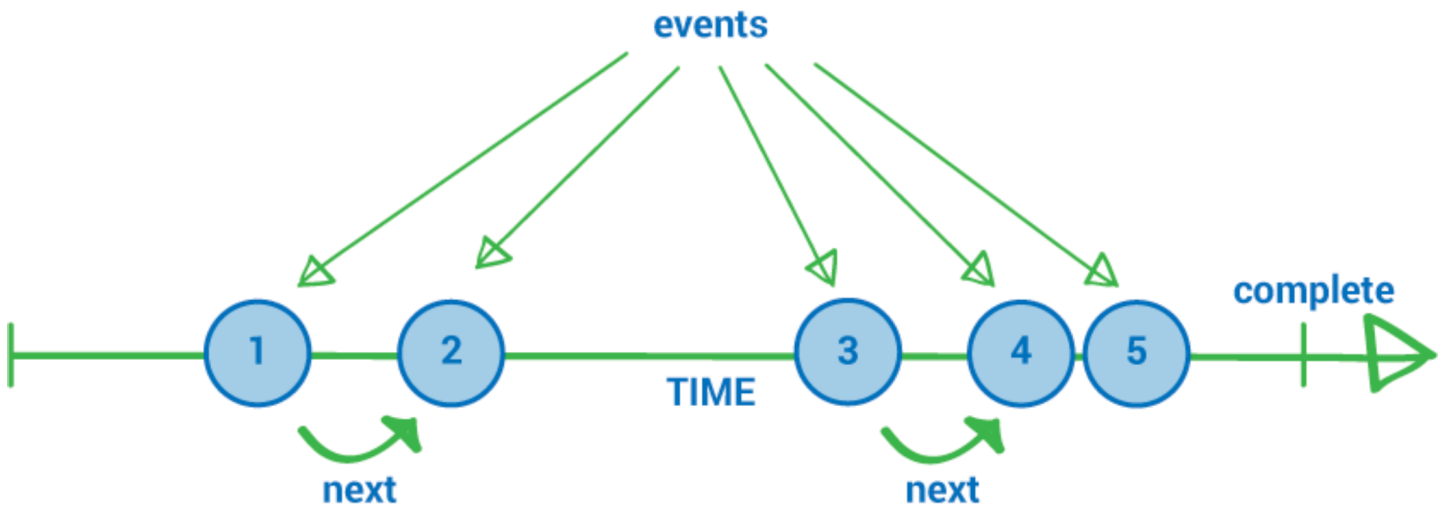
maVariable = new Observable(fonction () { });

La fonction est une série d'évènements que l'observable émettra.
Chaque émission s'effectue grâce à la méthode next().

17

```
monObservable = new Observable(obs => {
   obs.next(1);
   obs.next(2) ;
   obs.next(3) ;
 });
```

Ici, l'observable émet la valeur 1, puis 2, puis 3.



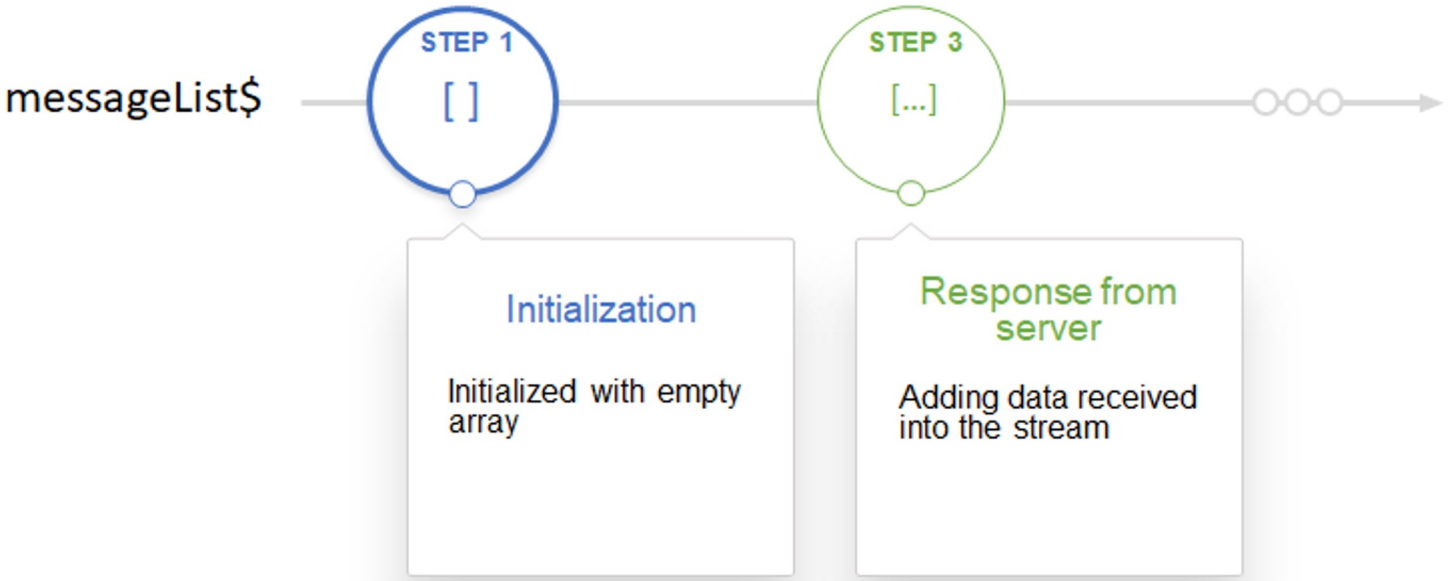
Observable Sequence

	SINGLE	MULTIPLE
SYNCHRONOUS	Function	Enumerable
ASYNCHRONOUS	Promise	Observable

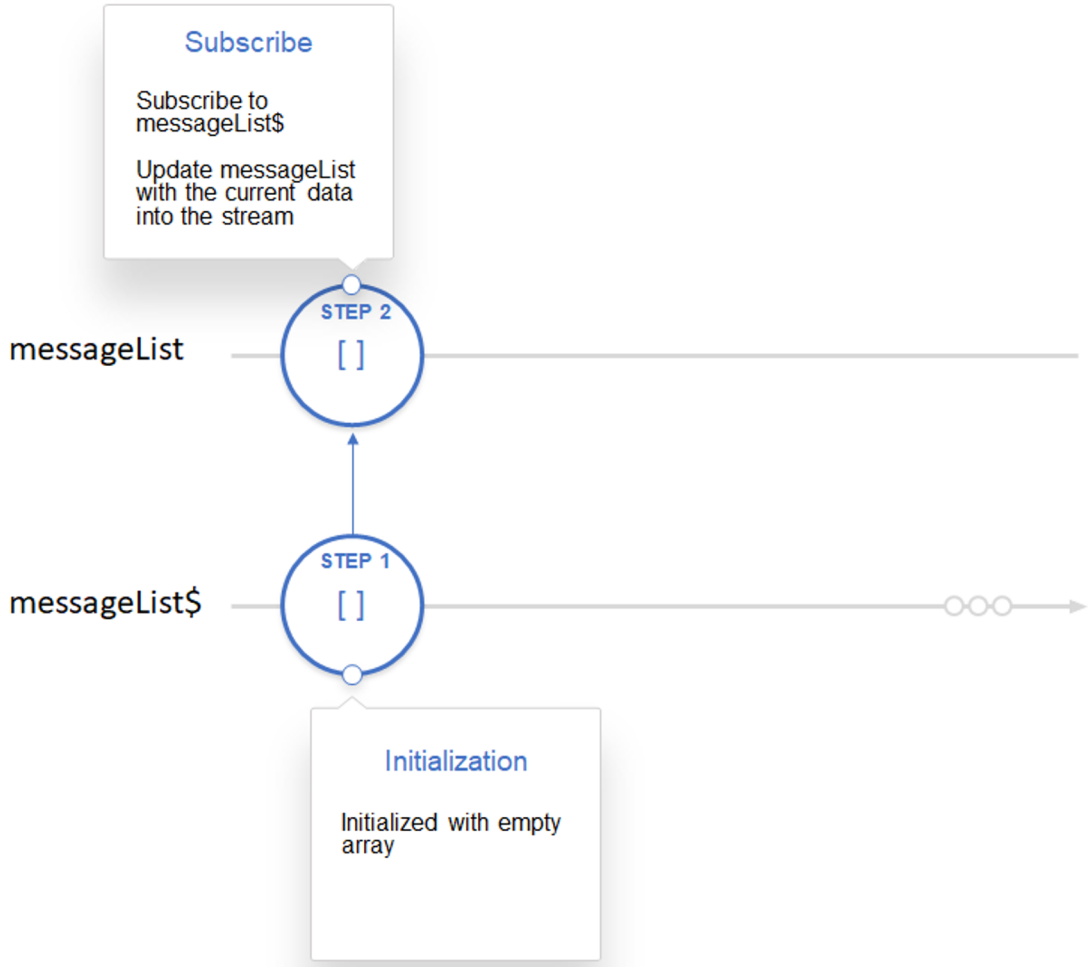
Time + Value

20 Observables

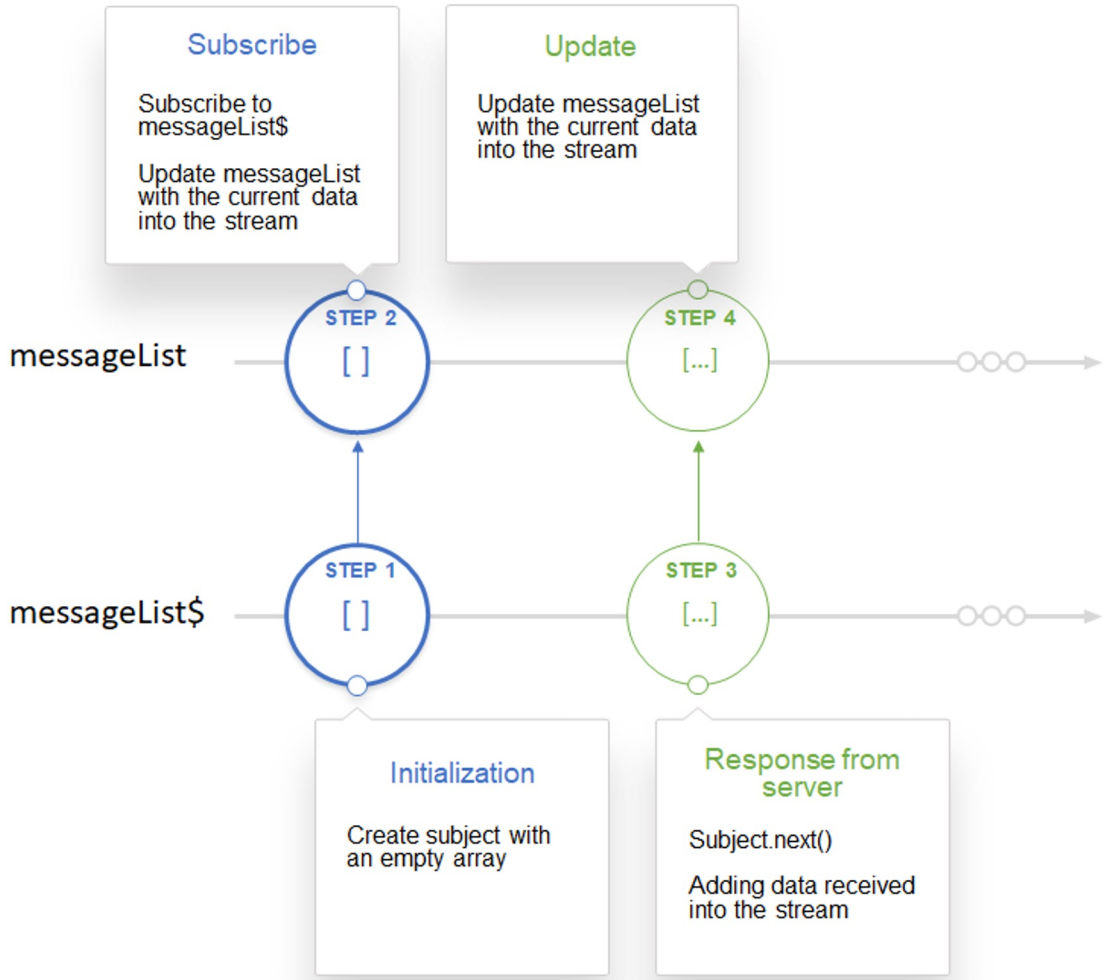
21 Observables



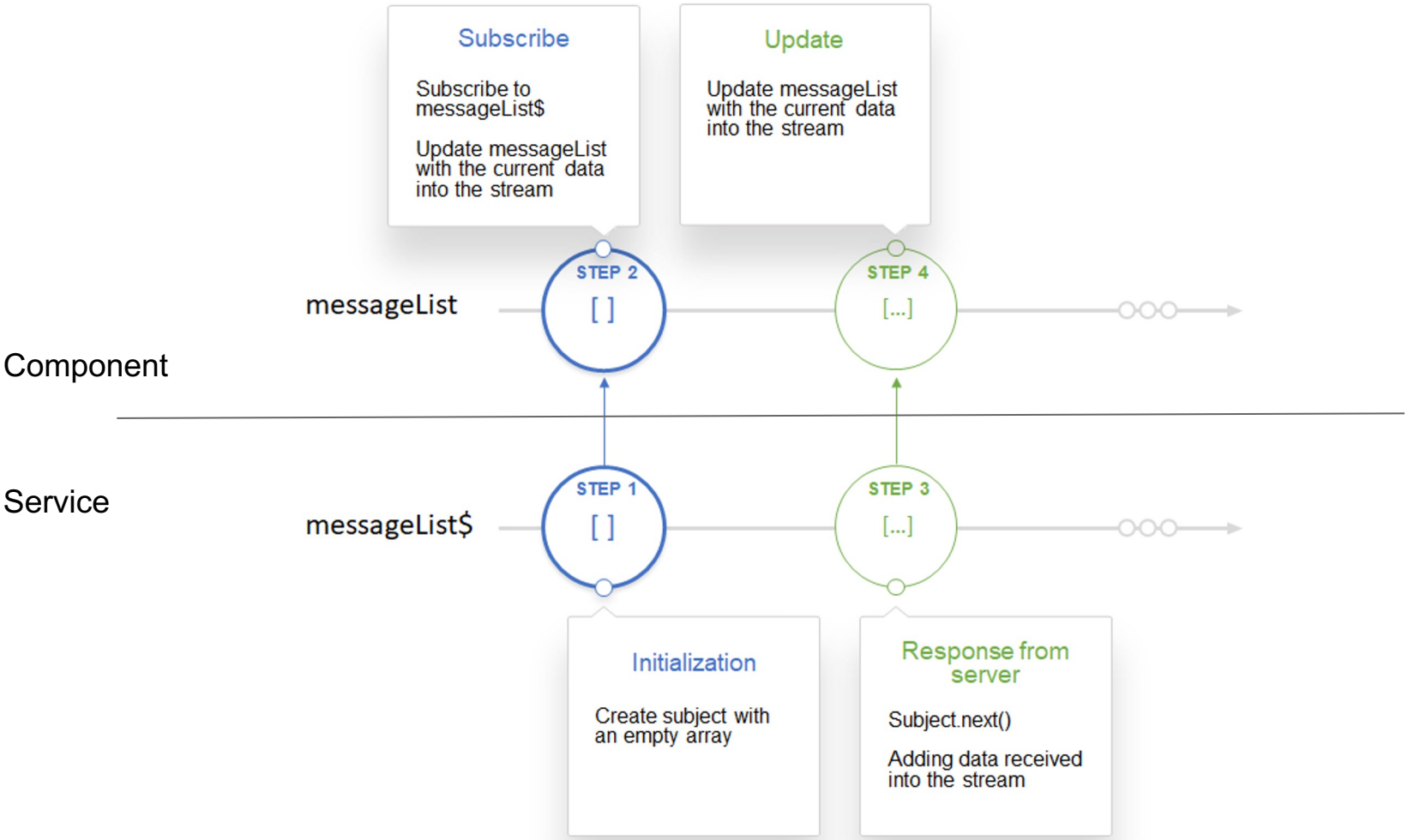
22 Observables



23 Observables



24 Observables



25 Observables

Service

```
@Injectable()
export class MessageService {

  public messageList$: BehaviorSubject<MessageModel[]>;

  constructor() {
    this.messageList$.next( value: []);
  }

  updateMessageList(messageList: MessageModel[]) {
    this.messageList$.next(messageList);
  }

}
```

1

3

Component

```
export class MessageListComponent implements OnInit {

  public messageList: MessageModel[];

  constructor(private messageService: MessageService) {
      this.messageService.messageList$.subscribe(
        next: (messageList) => this.messageList = messageList
      );
  }

  ngOnInit() { }
}
```

2


QUESTIONS?