



Advanced State Management with Redux and @ngrx/store

Alex Thalhammer

Outline

- Entities
- Data



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Entities



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

@ngrx/entity and @ngrx/schematics

- `ng add @ngrx/entity`
- `ng add @ngrx/schematics`
- `ng g module passengers`
- `ng g entity Passenger --module passengers.module.ts`



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Demo & Lab

NgRx Entities



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Data



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Idea

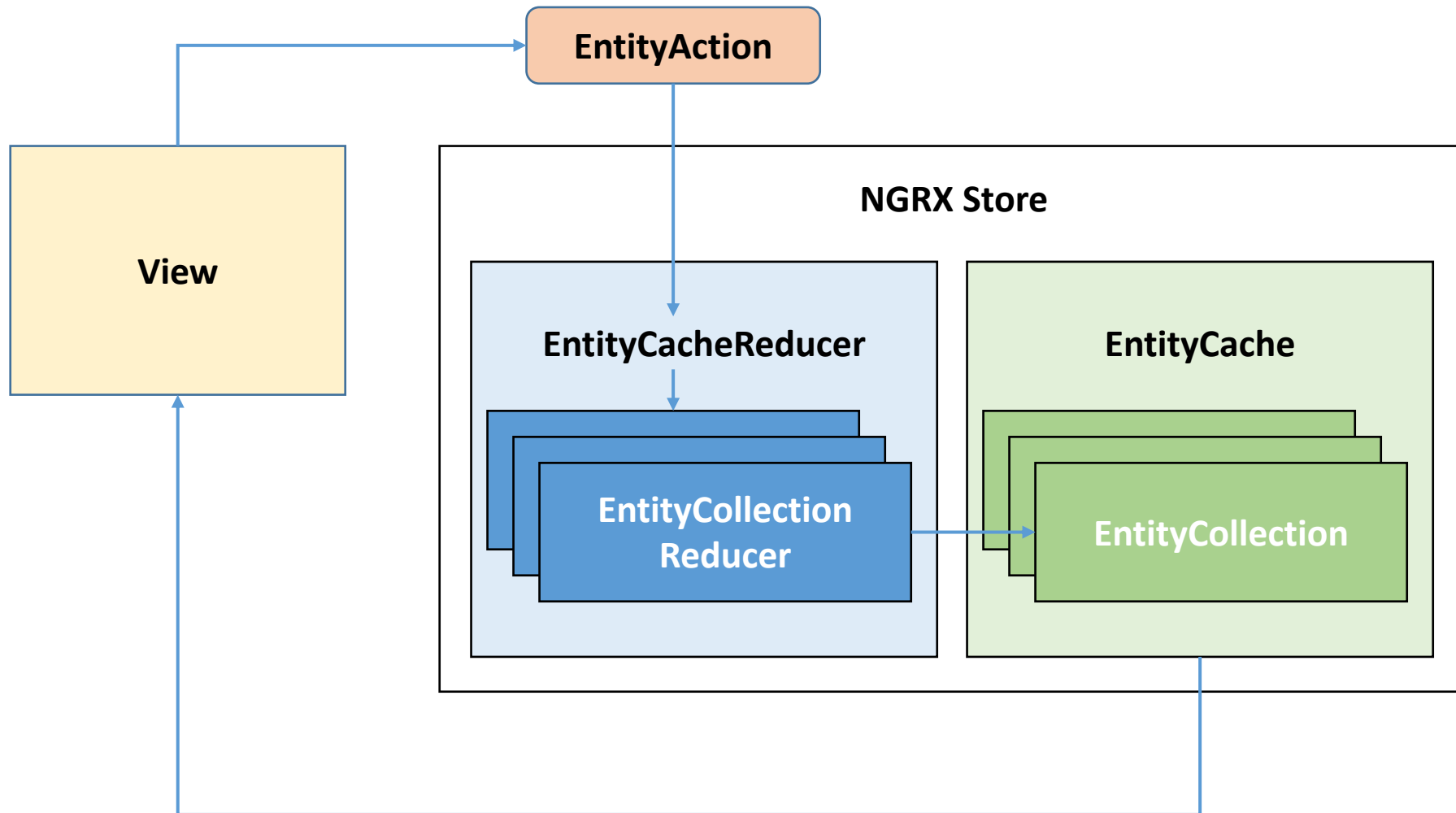
- Managing entities
- Generic solution
- Convention over configuration



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



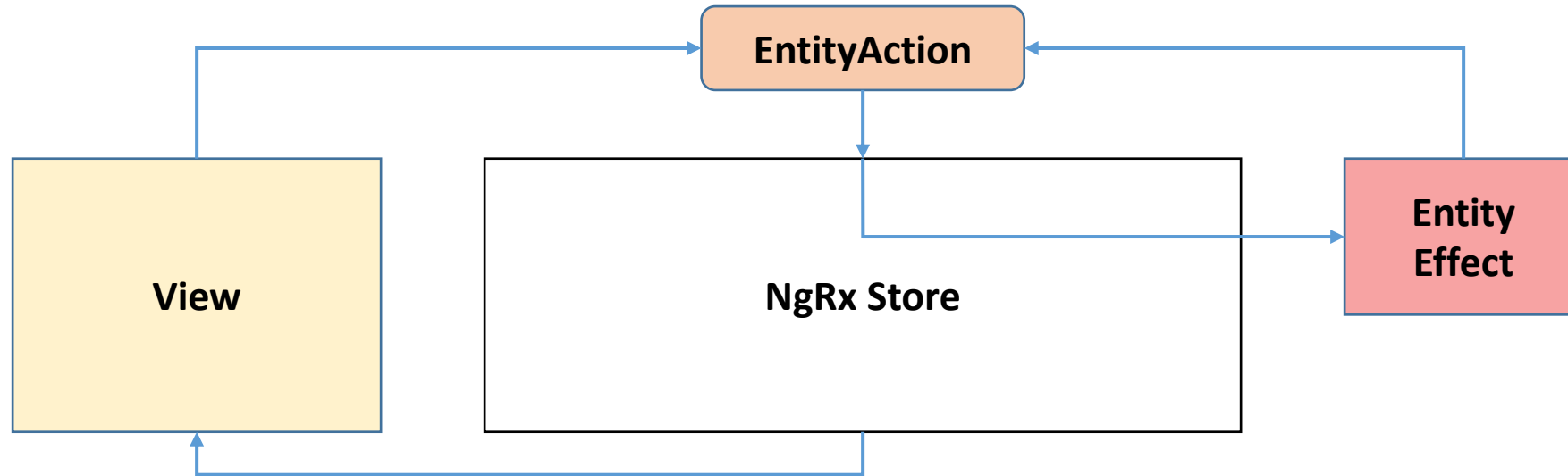
SOFTWARE
ARCHITECT



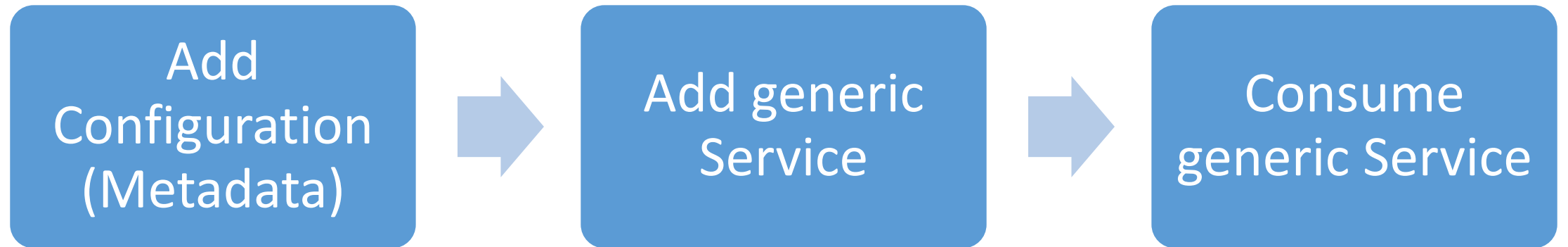
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT



Steps



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Configuration

```
const entityMetadata: EntityMetadataMap = {  
  Passenger: {  
    selectId: (p: Passenger) => p.id // == Default  
  }  
};
```



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Configuration

```
const entityMetadata: EntityMetadataMap = {  
  Passenger: {  
    selectId: (p: Passenger) => p.id // == Default  
  }  
};  
  
export const pluralNames = { Passenger: 'passenger' };
```



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Configuration

```
const entityMetadata: EntityMetadataMap = {  
  Passenger: {  
    selectId: (p: Passenger) => p.id // == Default  
  }  
};  
  
export const pluralNames = { Passenger: 'passenger' };  
  
export const entityConfig: EntityDataModuleConfig = {  
  entityMetadata,  
  pluralNames  
};
```



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Configuration

```
export const defaultDataServiceConfig: DefaultDataServiceConfig = {  
  root: 'http://www.angular.at/api'  
}
```



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Configuration

```
@NgModule({  
  imports: [  
    [...]  
    EntityDataModule.forRoot(entityConfig),  
  ],  
  providers: [  
    {  
      provide: DefaultDataServiceConfig,  
      useValue: defaultDataServiceConfig  
    },  
  ],  
})  
export class AppModule {}
```



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Generic Data Service

```
@Injectable({ providedIn: 'root' })  
export class PassengerService  
    extends EntityCollectionServiceBase<Passenger> {  
  
    [...]  
}
```



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Generic Service

```
@Injectable({ providedIn: 'root' })
export class PassengerService
    extends EntityCollectionServiceBase<Passenger> {

    constructor(serviceElementsFactory:
        EntityCollectionServiceElementsFactory)
    {
        super('Passenger', serviceElementsFactory);
    }

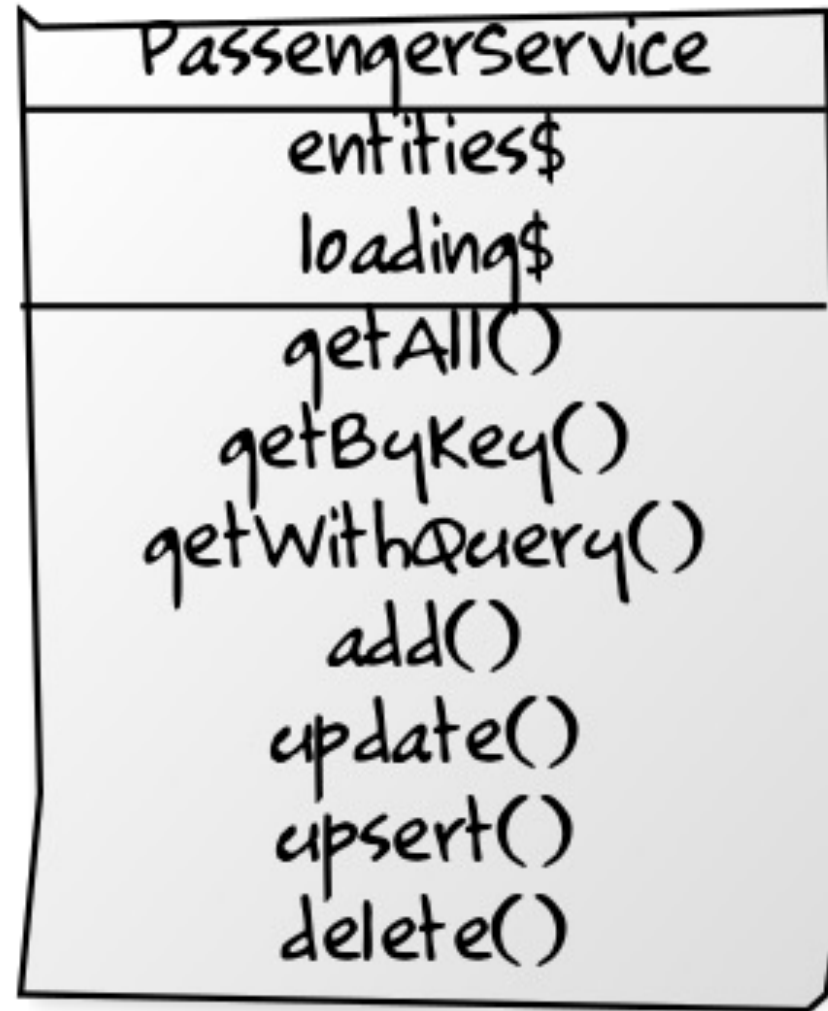
}
```



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT



Demo & Lab

NgRx Data



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT