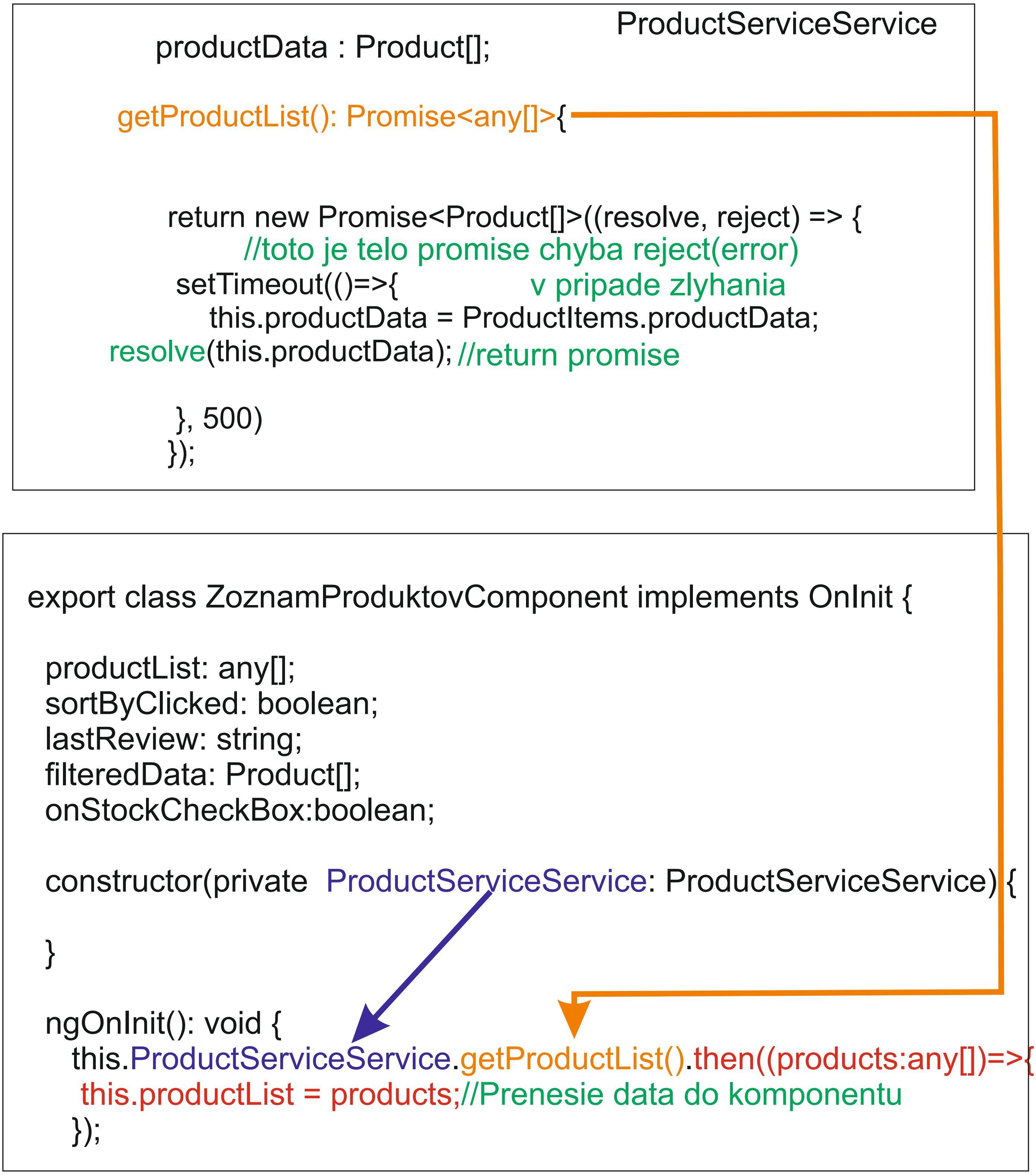
promise



```
                                          ProductServiceService
         productData : Product[];

      getProductList(): Promise<any[]>{


          return new Promise<Product[]>((resolve, reject) => {
                 //toto je telo promise chyba reject(error)
           setTimeout(()=>{              v pripade zlyhania
              this.productData = ProductItems.productData;
     resolve(this.productData); //return promise

           }, 500)
          });
```

```
export class ZoznamProduktovComponent implements OnInit {

  productList: any[];
  sortByClicked: boolean;
  lastReview: string;
  filteredData: Product[];
  onStockCheckBox:boolean;

  constructor(private  ProductServiceService: ProductServiceService) {

  }

  ngOnInit(): void {
    this.ProductServiceService.getProductList().then((products:any[])=>{
     this.productList = products;//Prenesie data do komponentu
    });
```

```
export Class Test implements OnInit, Destroy{  //moze bzt sluzba aj component

premenna: Observable<number> //premenna typu observer. ktora vracia number

constructor (private xxxxx: NazovSluzby)

ngOnInit(0: void{
this.premenna = new Observable<number>( (item)=>{

//implementacia logiky napr

let c = 0;
setInterval ( () => {
observer.next(c++);//posle premennu odoberatelom
}, 1000);

return{
unsubscribe(){//hocico potrebujeme   }  //metoda ktora zrusi odber


}

});
}
```

Trieda

```
constructor (private premenna: nazovSluzby) {     }

 premenna.subscibe ( result => {
  x = result;//result su data z observable
 });
```

unsubscibe

```
@Component({...})
export class AppComponent implements OnInit, OnDestroy {
    subscription: Subscription
    ngOnInit () {
        var observable = Rx.Observable.interval(1000);
        this.subscription = observable.subscribe(x => console.log(x));
    }
    ngOnDestroy() {
        this.subscription.unsubscribe()
    }
}
```