



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE

## Angular's Future with Signals



ManfredSteyer

# Agenda

#1  
Motivation  
& Basics

#2  
DEMO

#3  
RxJS/NGRX  
Interop

#4  
Outlook: Signal  
Components



# Motivation & Basics



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE

# Change Detection (CD) in Angular



# Drawbacks

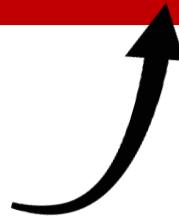
Zone.js:  
Magic

Zone.js:  
~100K

Cannot patch  
async/await

coarse  
grained CD

e.g. Components are always checked as a  
whole, even if only a tiny fraction changed



# How Do Other Frameworks Solve This?



SVELTE



SOLID



qwik

Signals!



@ManfredSteyer

# How Will Angular Solve This?

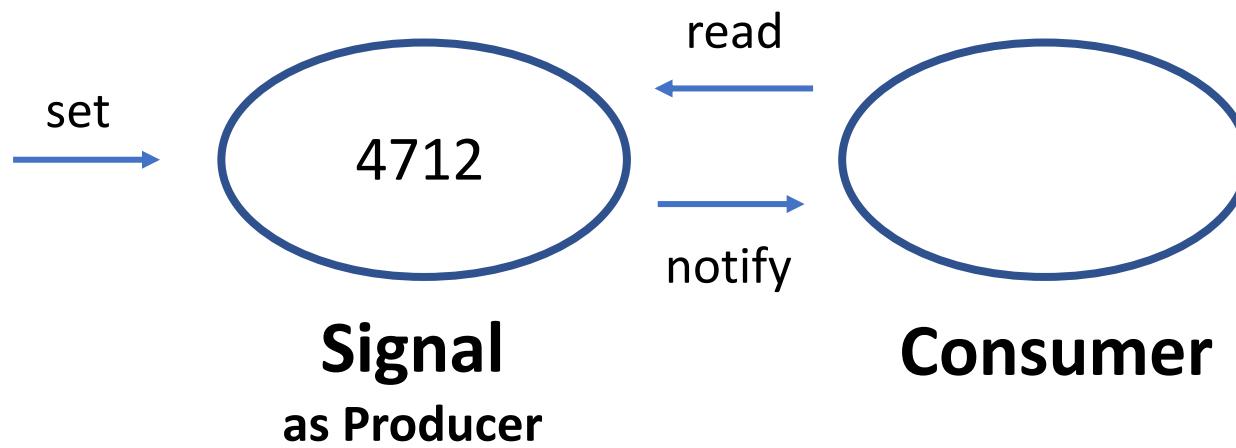


Signals!



@ManfredSteyer

# Signals: Simple Reactivity



# Component Without Signals

```
flights: Flight[] = [];

const flights = await this.flightService.findAsPromise(from, to);
this.flights = flights;

<div *ngFor="let f of flights">
    <flight-card [item]="f" />
</div>
```



# Component With Signals

```
flights = signal<Flight[]>([]);

const flights = await this.flightService.findAsPromise(from, to);
this.flights.set(flights);

<div *ngFor="let f of flights()">
    <flight-card [item]="f" />
</div>
```



# RxJS and NGRX Interop



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE

# @angular/core/rxjs-interop

toObservable(signal)

toSignal(observable\$)

takeUntilDestroyed()



# DEMO

# Outlook:

## Signal Components

(After Angular 16)



# Signal Based Components

```
@Component({
  signals: true,
  selector: 'temperature-calc',
  template: `
    <p>C: {{ celsius() }}</p>
    <p>F: {{ fahrenheit() }}</p>
  `,
})
export class SimpleCounter {
  celsius = signal(25);
  fahrenheit = computed(() => this.celsius() * 1.8 + 32);
}
```



# Signal Inputs

```
@Component({
  signals: true,
  selector: 'temperature-calc',
  template: `
    <p>C: {{ celsius() }}</p>
    <p>F: {{ fahrenheit() }}</p>
  `,
})
export class SimpleCounter {
  celsius = input(25);
  fahrenheit = computed(() => this.celsius() * 1.8 + 32);
}
```



# Signal Based Components

Work w/o  
Zone.js

Interop with  
traditional  
components

No need to  
update code!



# Conclusion

Fine-grained  
CD

Zone-less  
Future

Convertible to  
Observables  
and vice versa!

No need to  
unsubscribe!

No need to  
update code!

