



ManfredSteyer



I18N mit Angular

Manfred Steyer

SOFTWARE*architekt.at*

Inhalt

- Angular Compiler
- ngx-translate

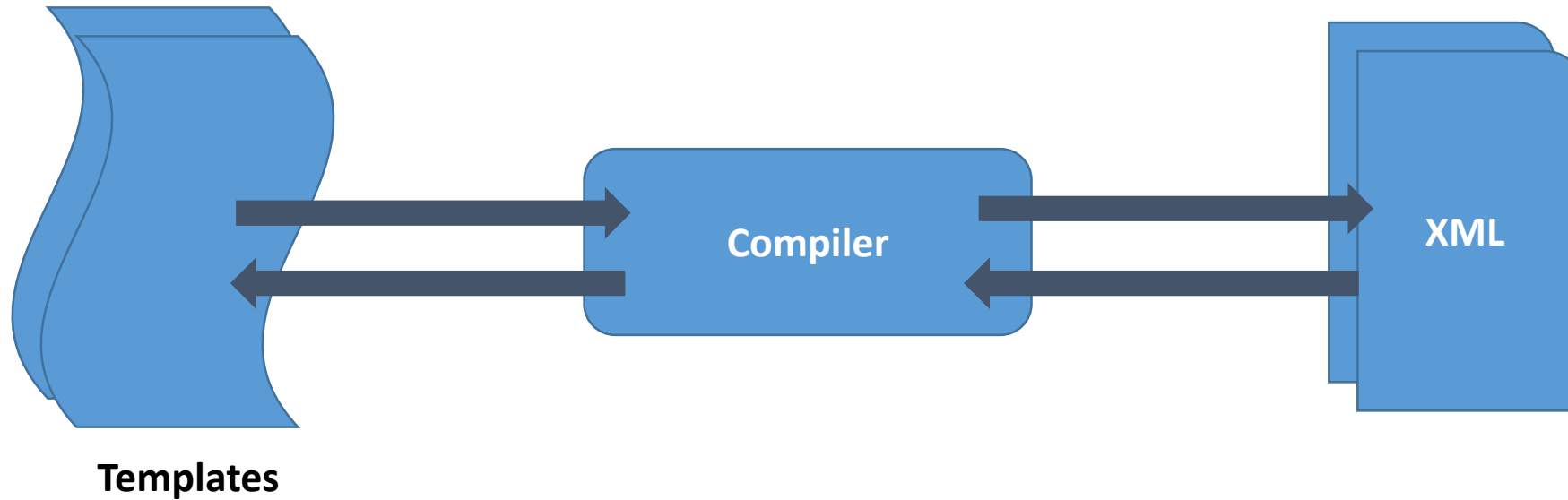
Angular Compiler



Angular Compiler

XML Localisation Interchange File Format (XLIFF, version 1.2)

XML Message Bundle (XMB)



Angular Compiler

Great
Performance

One Build per
Language

Cannot Change
Language at
Runtime!

Cannot Read
Translations in
Code

Texte markieren

```
<th i18n>From</th>  
<th i18n="Beschreibung">To</th>  
<th i18n="Bedeutung | Beschreibung">Passengers</th>
```

Texte extrahieren (Angular CLI)

```
ng xi18n
```

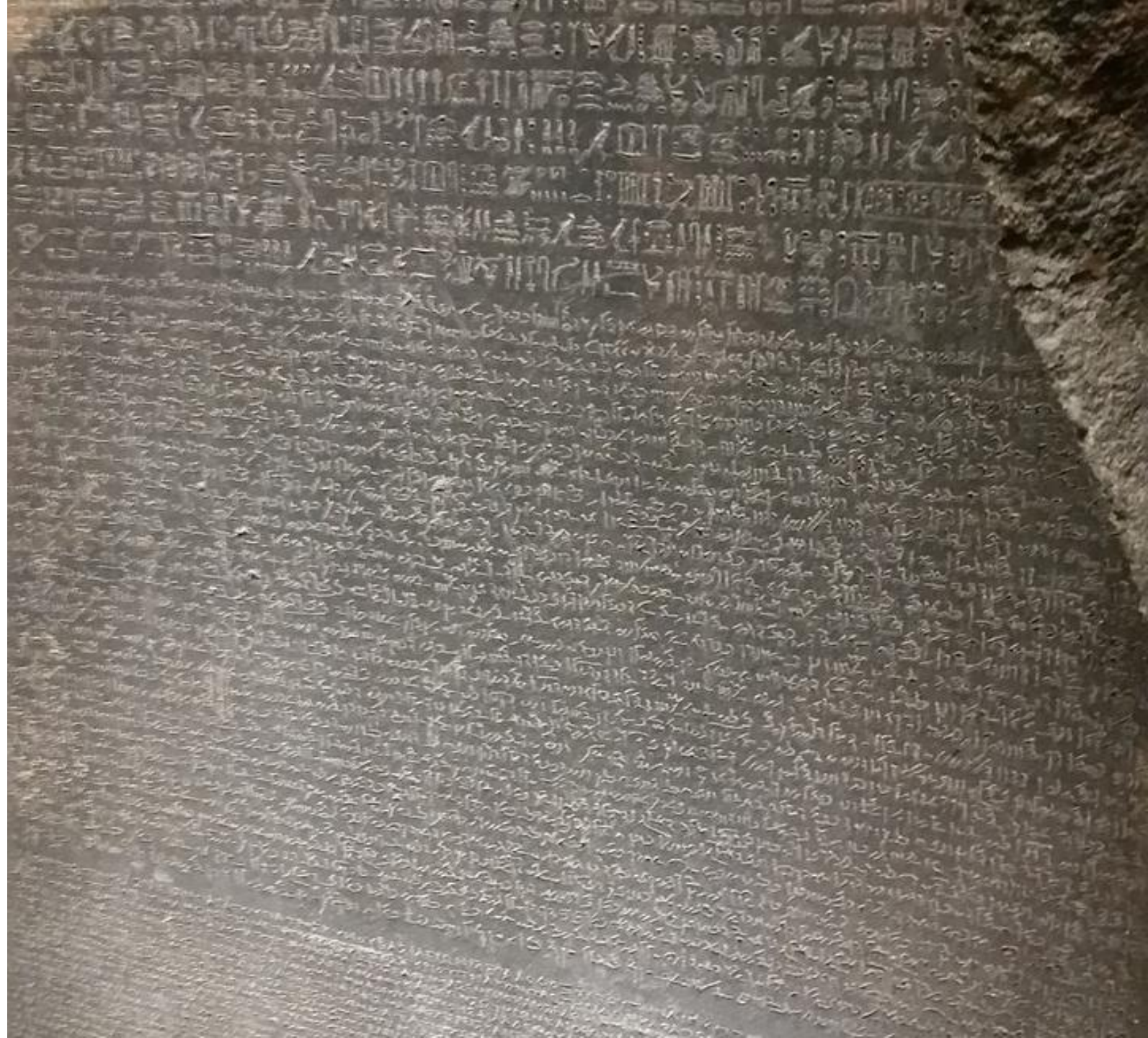
Übersetzungen und AOT (Angular CLI)

```
ng build --i18nFile=src/i18n/messages.de.xlf  
        --locale=de  
        --i18nFormat=xlf
```

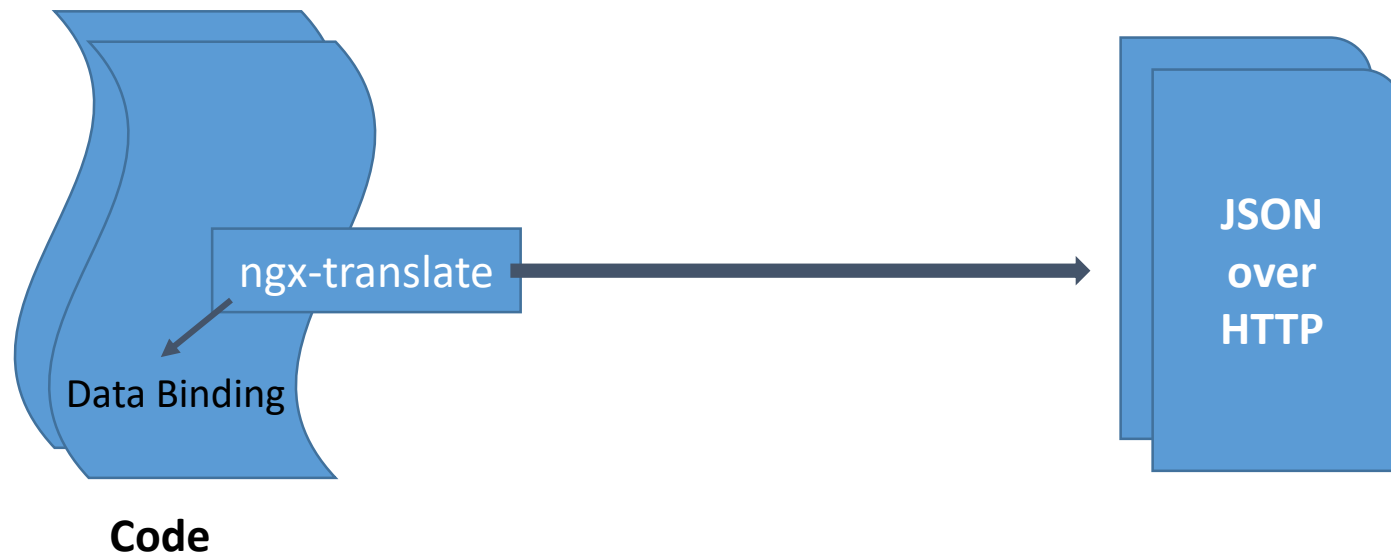

Übersetzungen und JIT

```
providers: [  
  {provide: TRANSLATIONS, useValue: require('./i18n/messages.de.xlf')},  
  {provide: TRANSLATIONS_FORMAT, useValue: 'xlf'},  
  {provide: LOCALE_ID, useValue: 'de'}  
]
```

ngx-translate



ngx-translate



Data format and way of loading are adaptable

ngx-translate

One Build

Change
Language at
Runtime!

Read
Translations in
Code

Data Binding
Influences
Performance

Import

```
export function createLoader(http: HttpClient) {  
  return new TranslateStaticLoader(http, './i18n', '.json');  
}
```

Import

```
export function createLoader(http: HttpClient) {  
  return new TranslateStaticLoader(http, './i18n', '.json');  
}
```

```
@NgModule({  
  imports: [  
    TranslateModule.forRoot({  
      provide: TranslateLoader,  
      useFactory: createLoader,  
      deps: [HttpClient]  
    })  
  ],  
  [...]  
})  
export class AppModule {  
}
```

Initialize

```
this.translate.addLangs(['en', 'de']);  
this.translate.setDefaultLang('de');  
this.translate.use('de');
```

Resource File

```
{  
  "FLIGHTS": {  
    "from": "From",  
    "to": "to",  
    "search": "Search",  
    "found": "{{count}} flights found."  
  }  
}
```


Translation

```
<th>{{'FLIGHTS.from' | translate}}</th>
```

```
<th>{{'FLIGHTS.found' | translate:{count: 7} }}</th>
```

DEMO



Local Formats

Pipes and local Formats

```
{{ flight.date | date:'long' }}
```

Import format information

```
import { registerLocaleData } from '@angular/common';  
import localeDe from '@angular/common/locales/de';  
import localeDeAt from '@angular/common/locales/de-AT';  
import localeEs from '@angular/common/locales/es';
```


```
registerLocaleData(localeDe);    // de-DE  
registerLocaleData(localeDeAt);  // de-AT  
registerLocaleData(localeEs);    // es-ES
```

Define default locale

```
providers: [  
  [...]  
  { provide: LOCALE_ID, useValue: 'de' },  
]
```


Use Formats

```
<p>Datum (Default=de): {{ item.date | date:'long' }}</p>
```

```
<p>Datum (de-AT): {{ item.date | date:'long': 'de-AT' }}</p>
```

```
<p>Datum (es): {{ item.date | date:'long':':': 'es' }}</p>
```


I18N and Inputs?

- Pipes are just for outputs
- Use component lib (like Angular Material) 
- General concept: ControlValueAccessor 