
FRANTIC

FRANTIC

Changing the world of travel with Angular and measuring along the way

Oskari Okko Ojala
Olli Leskinen
2018-03-01

Beta.finnair.com

Built with Angular

Decoupled CMS

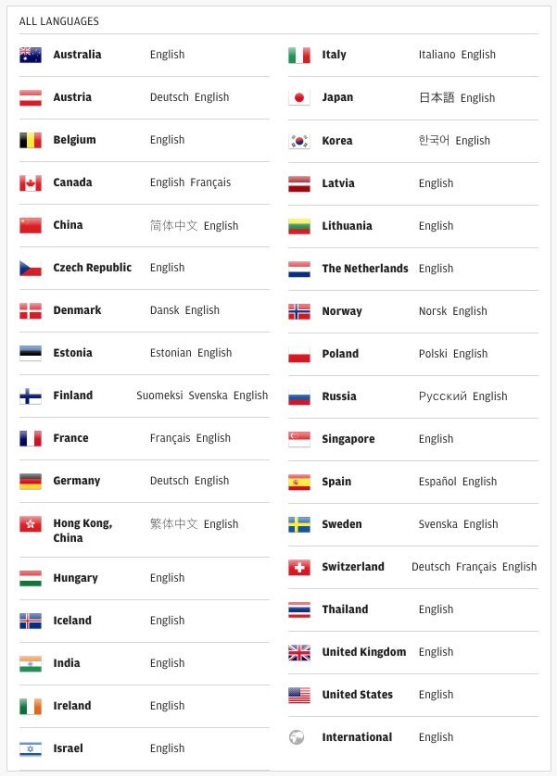
Microservice architecture

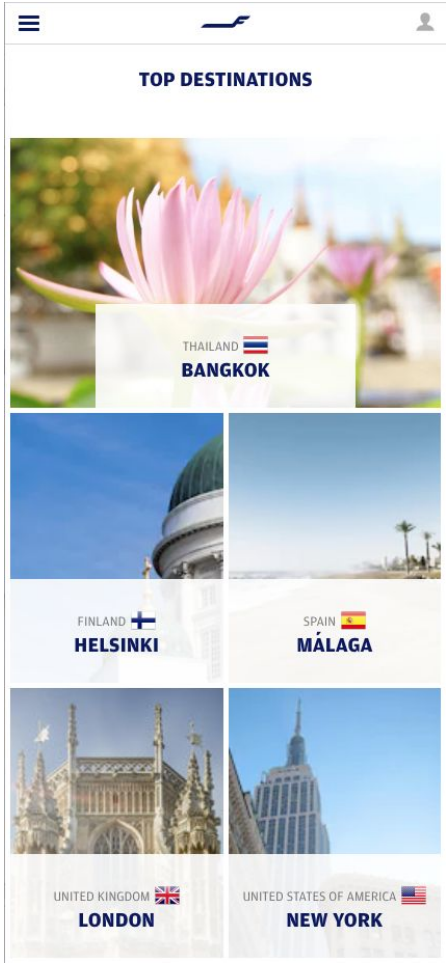
55 language variants

AWS Cloud

Amadeus

GitHub, Node, Java, Travis, Jenkins, Akamai...



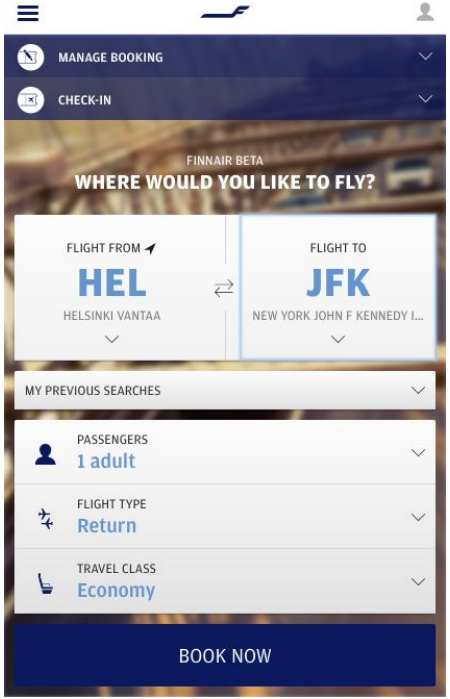


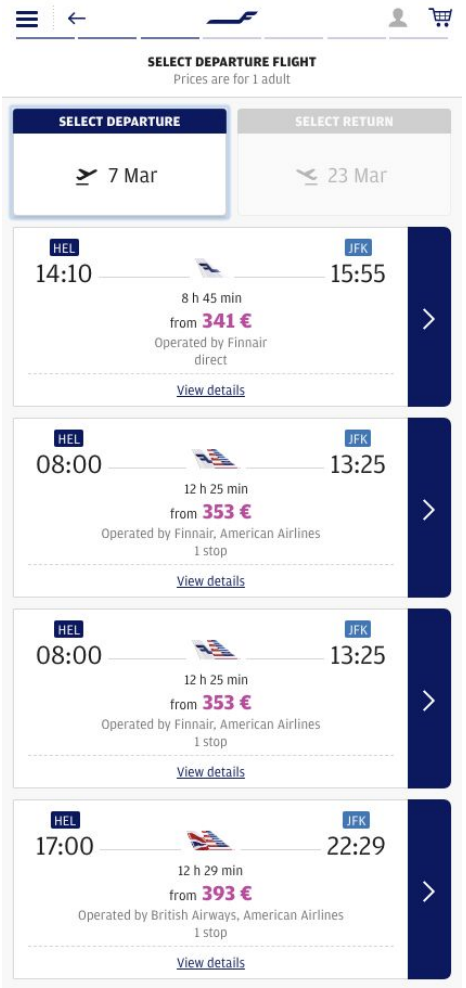
Beta.finnair.com

Hypothesis driven design

Multi-vendor team

- Finnair
- Frantic
- Nitor
- Houston Inc
- CGI
- Nextage Design





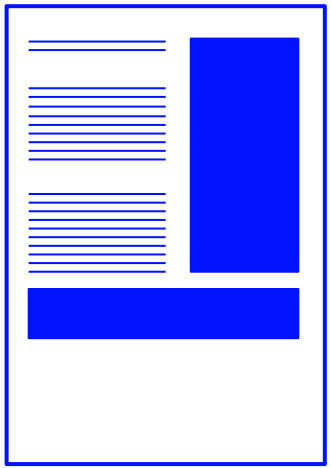
FRANTIC

Approaches

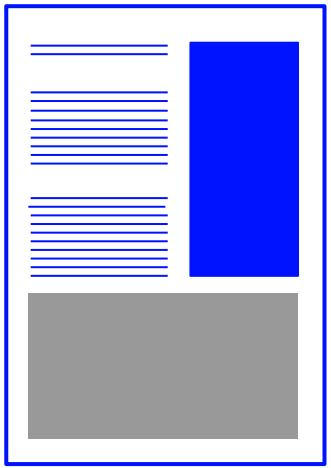
CMS

CLIENT-SIDE APP

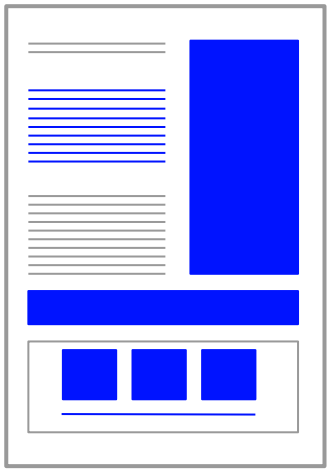
CMS-generated page



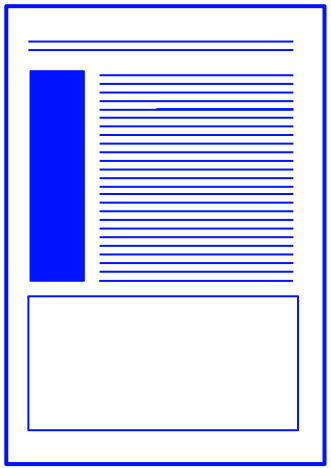
CMS-generated page with an in-page app



Client-side app with included CMS content



App only



FRANTIC

Seamless UX flow

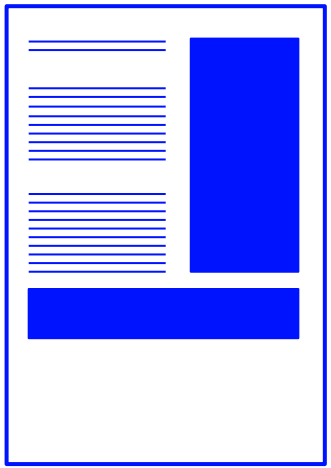
FRANTIC

Approaches

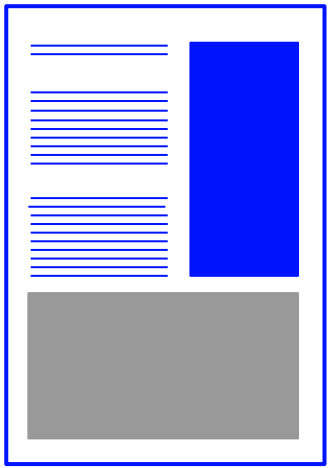
CMS

CLIENT-SIDE APP

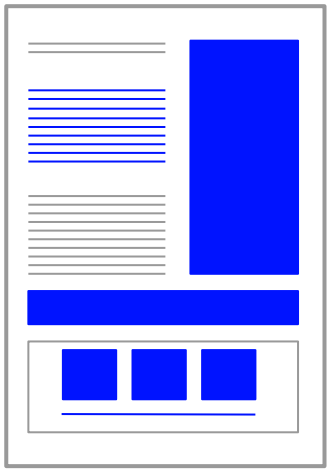
CMS-generated page



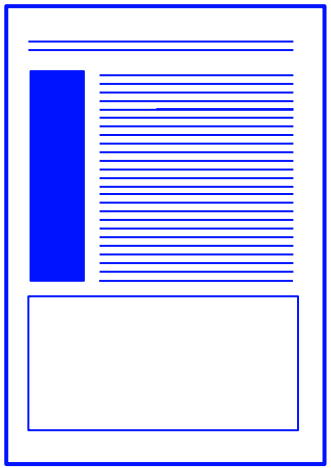
CMS-generated page with an in-page app



Client-side app with included CMS content



App only



“Beta means we’re now confident that most developers can be successful building large applications using Angular 2.”

-Angular 2 Beta announcement, December 2015

“Beta means *the best* developers can *start* building large applications using Angular 2.”

-There, fixed it for ya

Angular 2 (early 2016)

- SPEED & PERFORMANCE
- SIMPLE & EXPRESSIVE
- CROSS-PLATFORM
- FLEXIBLE DEVELOPMENT: THE CHOICE OF LANGUAGE IS UP TO YOU
- COMPREHENSIVE ROUTING
- DEPENDENCY INJECTION
- LEGACY BROWSER SUPPORT
- ANIMATIONS (LATER)
- INTERNATIONALIZATION (I18N) & ACCESSIBILITY (LATER)

FRANTIC

SPEED & PERFORMANCE

FRANTIC

...once you've gotten it loaded

FRANTIC

NO

Angular Universal

FRANTIC

SIMPLE & EXPRESSIVE

FRANTIC

NO
documentation

FRANTIC

FALSE

documentation

FRANTIC

STEEP

learning curve

FRANTIC

EFFORT

to create
new components

FRANTIC

COMPREHEN-SIVE ROUTING

FRANTIC

AOT &
huuuuge site

FRANTIC

LEGACY BROWSER SUPPORT

FRANTIC

BYGONES
‘cause
PCI & IE EOL

FRANTIC

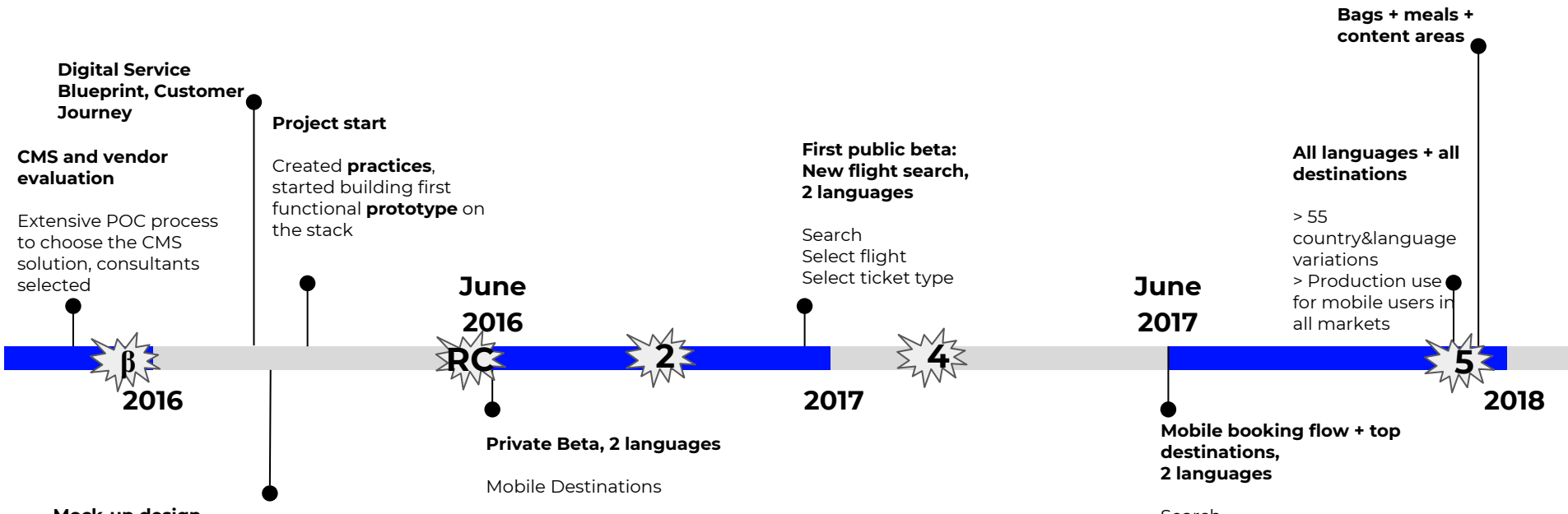
Our current challenges

- Async dynamic routing because of CMS content
  <-> Angular Router-based routing <-> AOT <-> lazy loading
- Unzipped 2MB of JS is not small
- Learning curve is still steep
- Zone.js issues are painful to debug

(Source: Imgur)

FRANTIC

Timeline



FRANTIC

Why we <3 Angular

Why we <3 Angular in a large team

- Expressive - less loc = less bugs
- TypeScript's typing prevents errors
- Encourages modular components
- Automated testing of components is built-in
- Angular 2 templates
- SCSS styles isolation per-component
- Dependency injections
- Rxjs

FRANTIC

TAKEAWAYS

FRANTIC

Validate designs with real users

Framework saves time when creating the conventions

Do fat microservices

FRANTIC

Give thin responses

Minimize

the client-side business logic

Write

tests

FRANTIC

PR PRs

FRANTIC

TypeScript

is awesome

FRANTIC

Abandon

vi and emacs

- you need an IDE

FRANTIC

Tracking & Analytics

Tracking

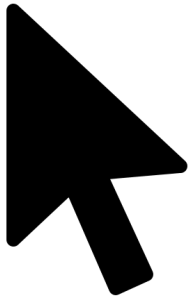
Page views

Custom events

Business stuff

FRANTIC

Page views
Router



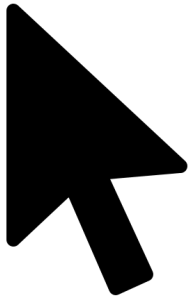
Custom events
Directives

Business stuff
Store

FRANTIC

Page views

Router



Custom events

Directives

Business stuff

Store

Tracking
Service

FRANTIC

Page views
Router

Custom events
Directives

Business stuff
Store

Tracking
Service

Google Tag Manager

Google Analytics

Tracking service

- Does nothing

FRANTIC

Tracking service

Does:

- Provide event specific functions
- Formatting and data minimisation
- Push to a global array created by GTM

```
this.dataLayer = this.windowRef.nativeWindow['dataLayer'] = this.windowRef.nativeWindow['dataLayer'] || [];
checkoutFlights(cart: CartData): void {
   // Privacy filtering, manipulate & format
   this.pushEventToDataLayer(GtmEvent.CART_CHECKOUT, formattedCart);
}
private pushEventToDataLayer(event: GtmEvent, eventData: GtmEventData): void {
  this.dataLayer.push({ event, [event]: eventData });
}
```

FRANTIC

The router

- No page refreshes in a SPA ➜ Use router!
- You can subscribe to router's events

```
this.router.events
  .filter((event: Event) => event instanceof NavigationEnd)
  .throttleTime(200)
  .subscribe((event: NavigationEnd) => {
    this.trackingService.pageView(event.urlAfterRedirects);
  });
```

The store - we use ngrx/store

- Good place for following business relevant stuff, **like the payment process**
- Access store as observables using **.let operator** (and now also: **.pipe**)

  or

- Know when to call tracking service functions using **ngrx @effect -side effects**

FRANTIC

The store

```
this.store
  .let(cartData())
  .filter(Boolean)
  .subscribe(cart => {
    this.checkoutFlights(cart);
  });

--- OR ---

@Effect cartCheckout$: Observable<Action> = this.actions$
  .ofType(CartAtions.SET_CART_DATA)
```

FRANTIC

Attribute directives

- Not everything happening in the UI is changes the app state to store.
- Attribute directive can easily:
  - Utilise component's existing event bindings
  - Utilise stuff existing in the template

FRANTIC

Attribute directives

Template

```
<li *ngFor="let flights of flightList; let i = index">
   <button (click)="clickFlight(flight)"
       [elemTracker]="'flight-selection-item'"
       [elemIndex]="(i+1)"
       [elemModifier]="flight.stops + ' stops'"
       [elemType]="ElementTypes.BUTTON">
</li>
```

Directive

```
@Input('elemTracker') trackingId: string;
@Input('elemIndex') trackingIndex: number;
@Input('elemModifier') trackingModifier: string;
@Input('elemType') trackingElemType: ElementTypes;

@HostListener('click', ['$event'])
onClick(event) {
  this.trackingService.trackElement(
    this.trackingId,
    this.trackingIndex,
    this.trackingModifier,
    this.trackingElemType.
  );
}
```

Tracking conclusions

- Router
- @ngrx/store if you are using it ❤️
- Directives can be used for UI tracking
- Service to format and push data forwards
- Easy data layer specs based on your event interfaces

FRANTIC

beta.finnair.com

frantic.com/jobs
finnair.com/jobs