

FRONTEND INTEGRATION

INHALT

- Hintergrund
- Herausforderungen
- Konzepte und Technologie
- Fazit und Diskussion

HINTERGRUND

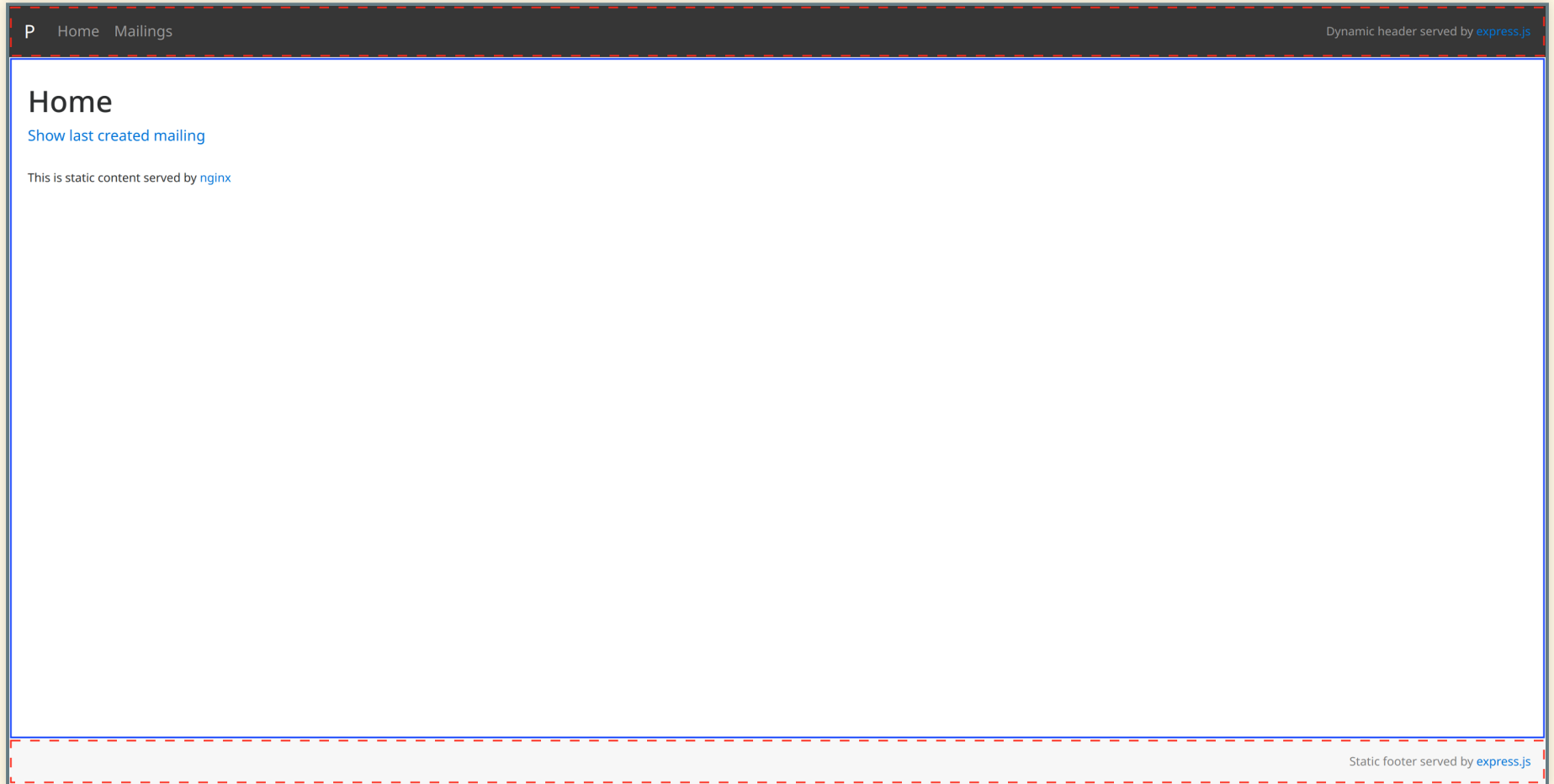
- UI ist der zentrale Kontaktpunkt mit dem Kunden
- Konzepte um verteilte (Web) UI's zu integrieren

HERAUSFORDERUNGEN

- Frankenstein UX
- Micro Service/**SCS** Praktiken auf das Frontend übertragen

TRANSCLUSION

Beispiel: Einheitliches Layout (Header, Navigation, etc.)



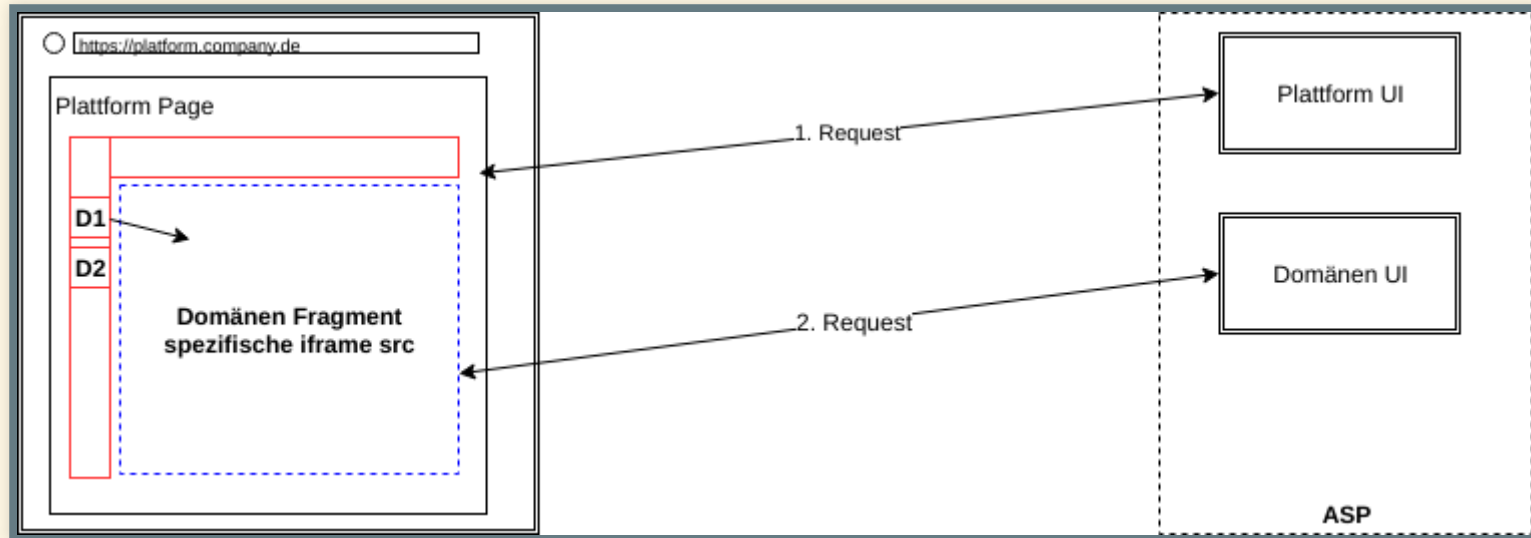
PROBLEME TRANSLUSION

- Web ist Dokumenten orientiert
- (Noch) keine Standards im Web/Browsern

ANSÄTZE

- iframe
- framework (X.js engines)
- server (SSI/ESI)
- client
(ajax/webcomponents)
- build time

IFRAME



strikte trennung

schwer nahtlos zu integrieren

kommunikation umständlich

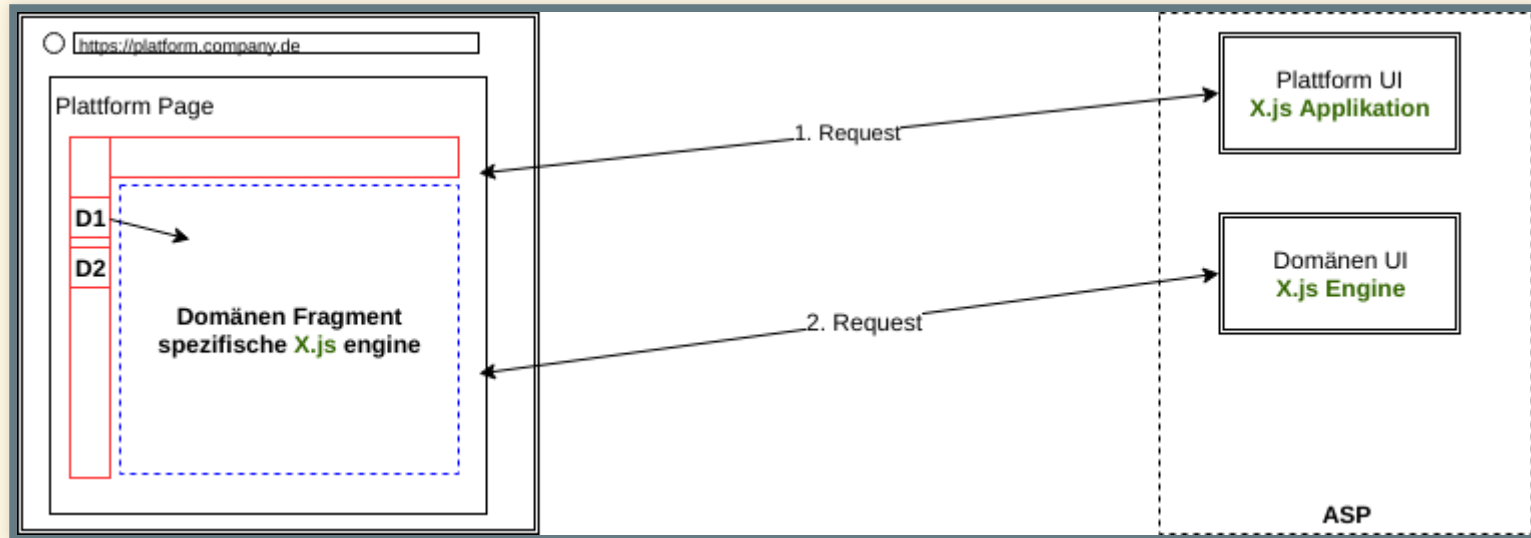
back button


```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    ... header, navigation ...

    <iframe src="https://platform.company.de/mail">

    ... footer ...
  </body>
</html>
```

FRAMEWORK (X.JS ENGINES)



footprint (assets, framework code, etc.)

Eine Technologie

Build Zeit Abhängigkeit

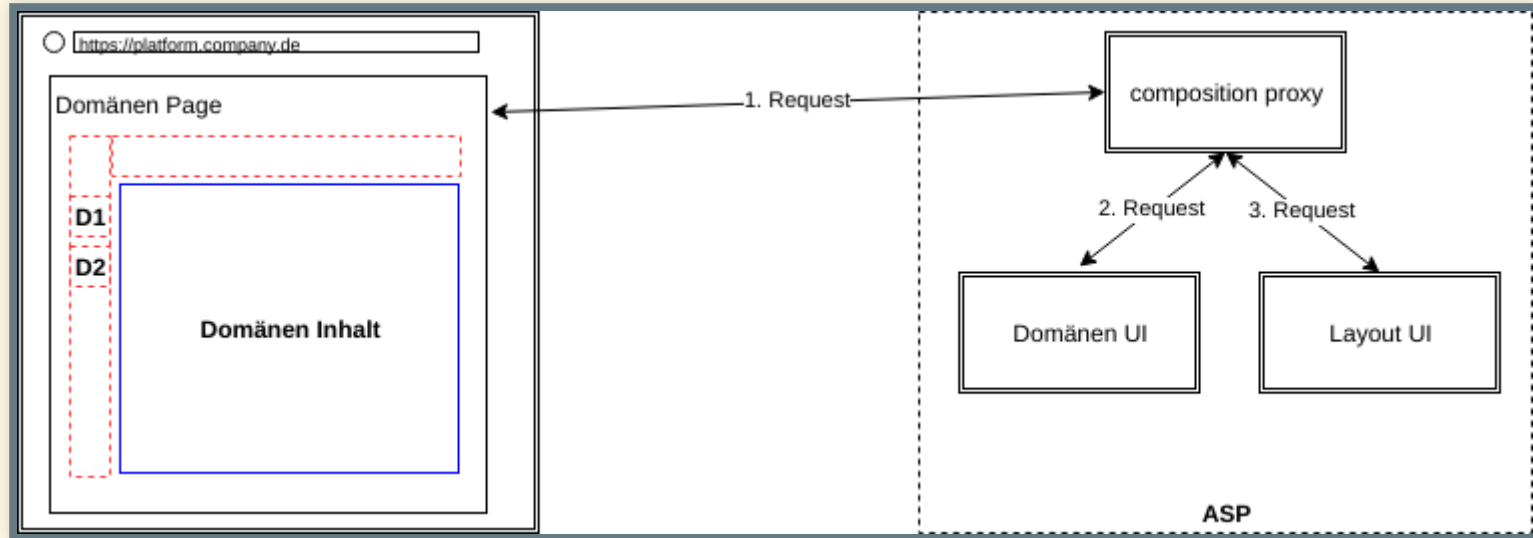
Keine Technologie Autonomie

```
"dependencies": {  
  "maileditor-engine": "1.2.3"  
}
```

```
this.mount( 'maileditor-engine', { path: '/mail' } );
```

<https://github.com/ember-engines/ember-engines>

SERVER SIDE (ESI/SSI)



SEO friendly

DRY (Deklarativ, Resilienz, Caching)

Orchestrierung von internet requests

Verwaltung/Konfiguration des Services (SPOF)

Response Zeit abhängig von einzelnen backends

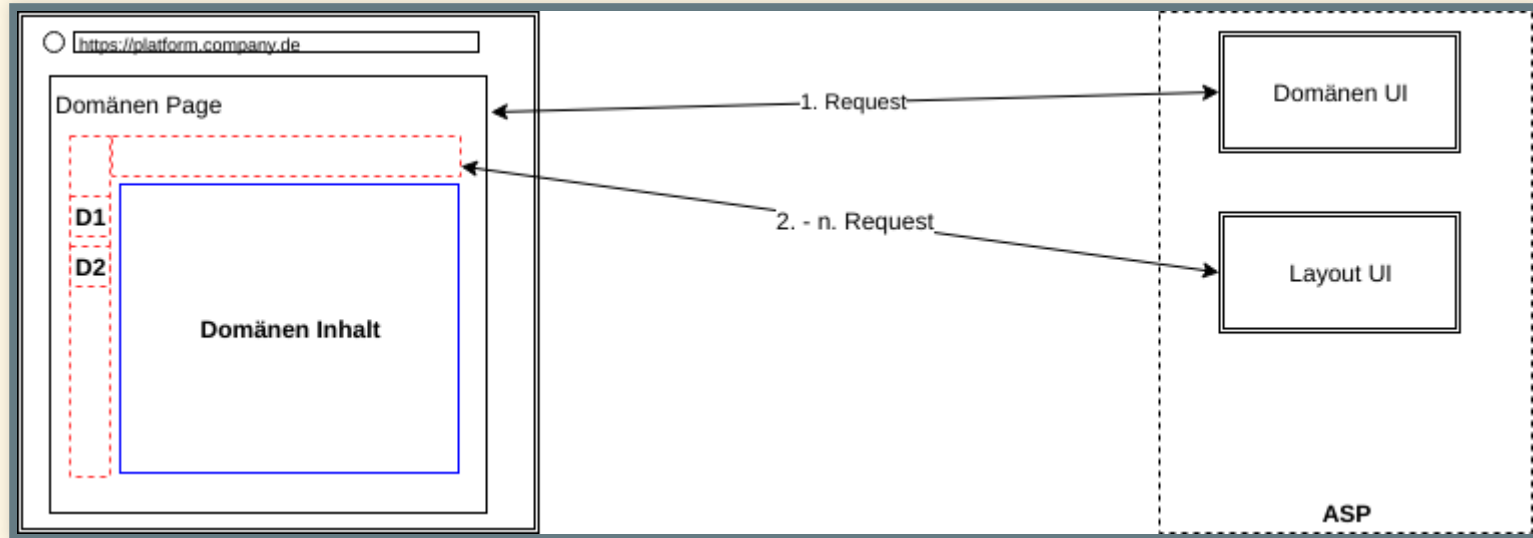
```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <div cx-url="https://platform.company.de/header" cx-timeout="1s" cx-cache-ttl="10s">
      This content will be replaced on the way through ...
    </div>

    Domänen Inhalt (Server Side, SPA, etc.)

    <div cx-url="https://platform.company.de/footer" cx-timeout="1s" cx-cache-ttl="1h">
      Default content can be placed here, in case of errors ...
    </div>
  </body>
</html>
```

<https://github.com/tes/compoxure>

CLIENT SIDE (AJAX/WEBCOMPONENTS)



Nutzung der Browserplattform (polyfills)

DRY (Deklarativ, Resilienz, Caching)

(Keine Abhängigkeiten)

Mehrere internet requests

footprint (framework code)

```

<script>
(function() {
    var FragmentPrototype = Object.create( HTMLElement.prototype );

    FragmentPrototype.attachedCallback = function() {
        this.appendChild( this.getAttribute( 'ref' ) );
    }

    FragmentPrototype.appendChild = function( ref ) {
        $(this).load( ref );
    }

    document.registerElement( 'x-fragment', {
        prototype: FragmentPrototype
    } );
})();
</script>

```

```

<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <x-fragment ref="https://platform.company.de/header"></x-fragment>

    Domänen Inhalt (Server Side, SPA, etc.)

    <x-fragment ref="https://platform.company.de/footer"></x-fragment>
  </body>
</html>

```

<https://www.innoq.com/de/blog/transclusion/>

FAZIT

- Fragmente/Glue technologie neutral halten
- Resoure orientiert UI's
- Nicht dogmatisch sein
- No free lunch

DEMO

Demo

DEMO ARCHITEKTUR

