innoQ

Transklusion

Kitt für gut geschnittene Webanwendungen

Franziska Dessart

Senior Consultant @ innoQ

franziska.dessart@innoq.com

@lapaqui

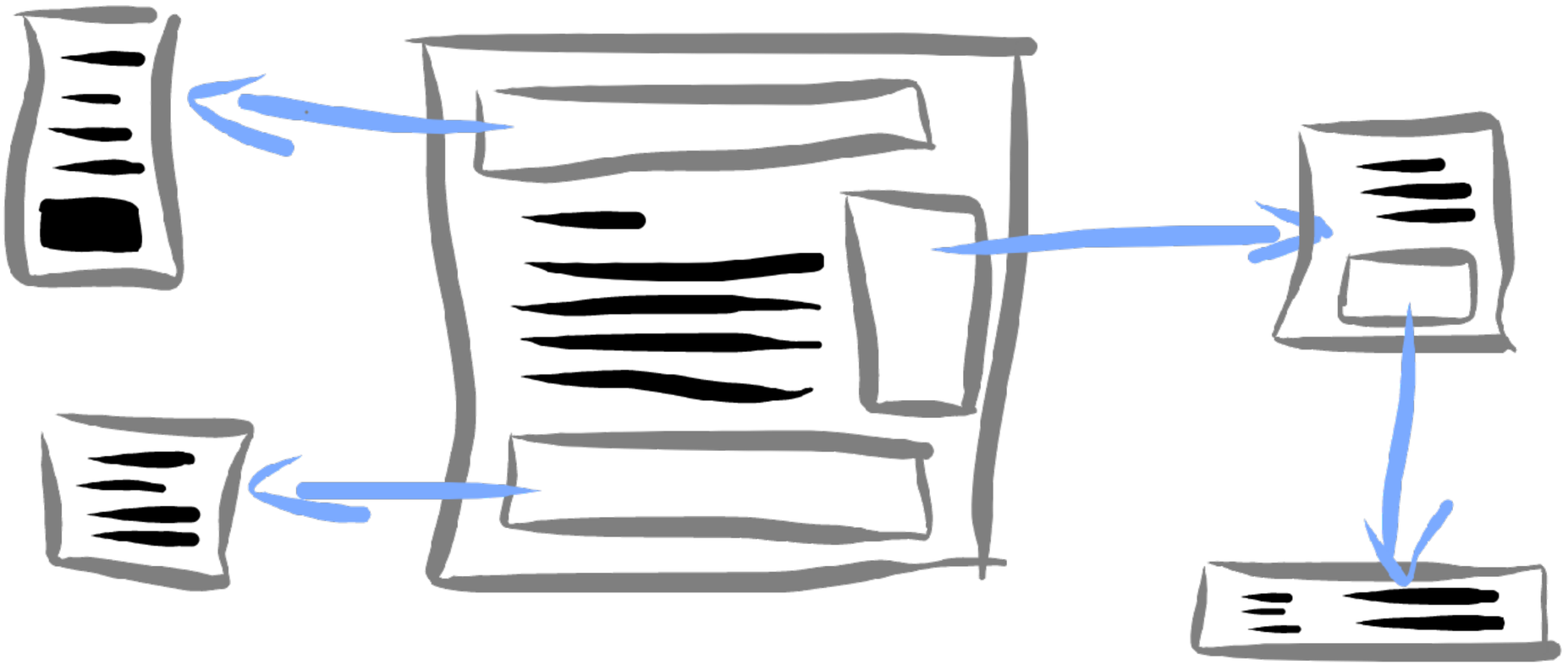
1
2
3
4
5
6
7

Was ist Transklusion?

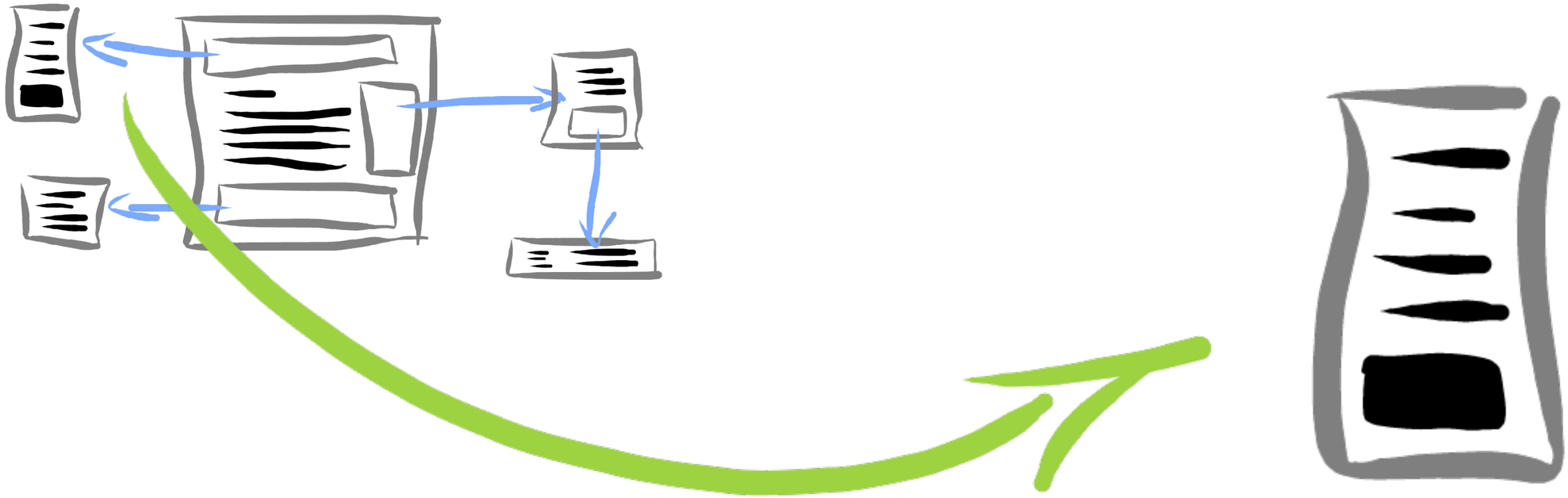
“...die Übernahme von einem elektronischen Dokument oder Teilen davon in ein oder mehrere andere Dokumente durch einen Hypertext-Verweis.”

— https://de.wikipedia.org/wiki/Transklusion

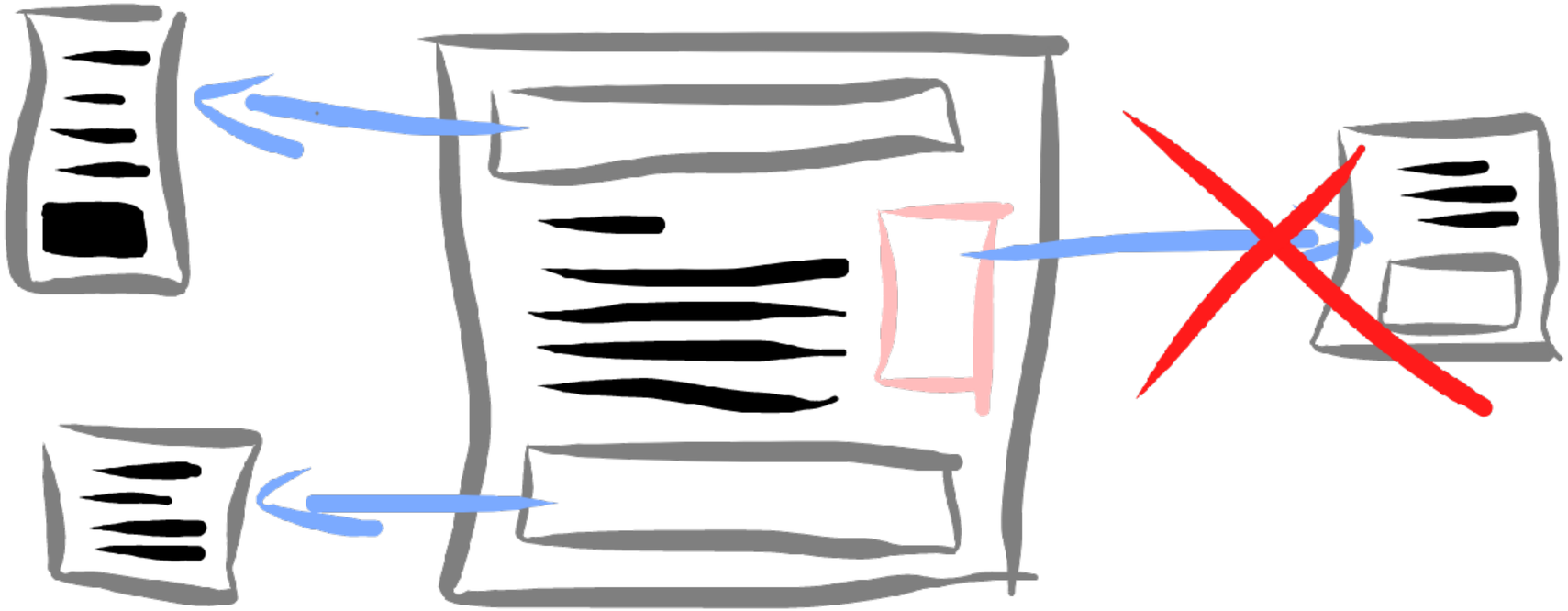
Transklusion



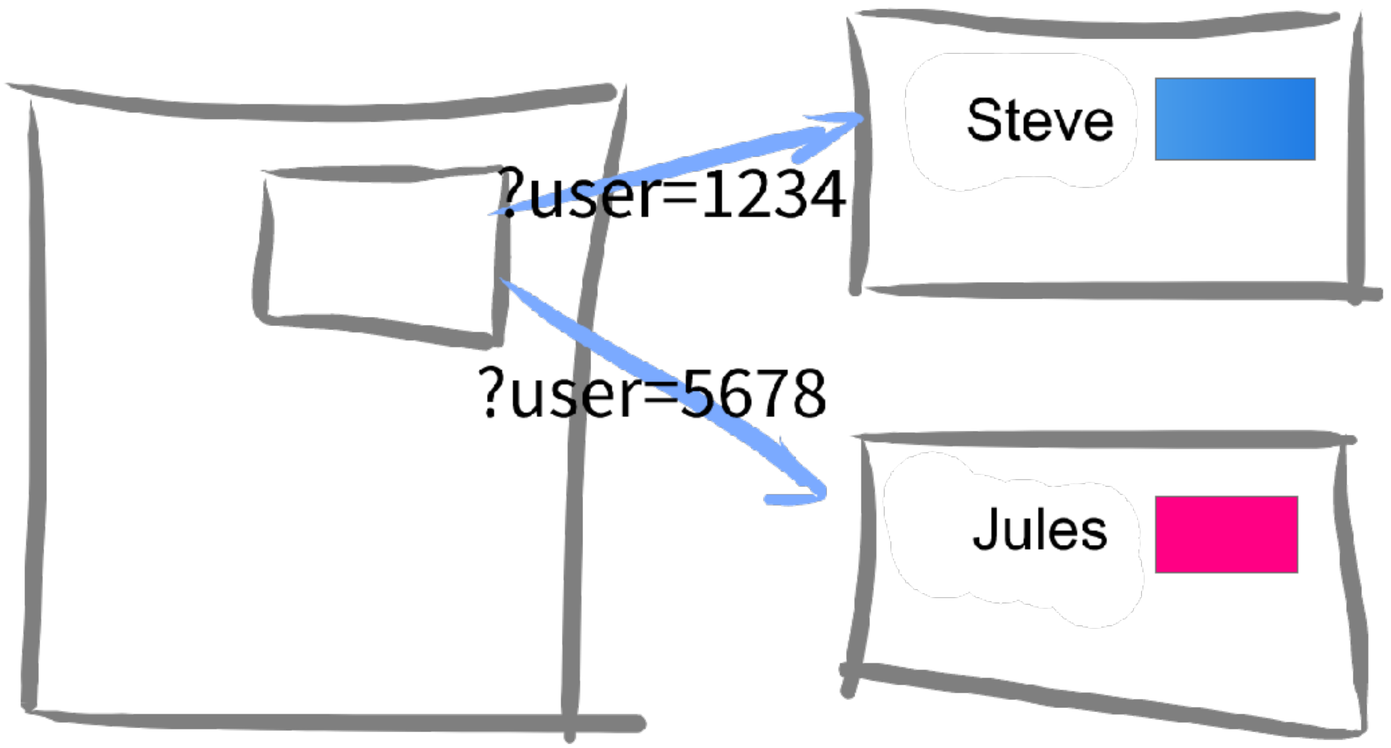
Atomisierung



Atomisierung



Parametrisierung



Warum transkludieren?

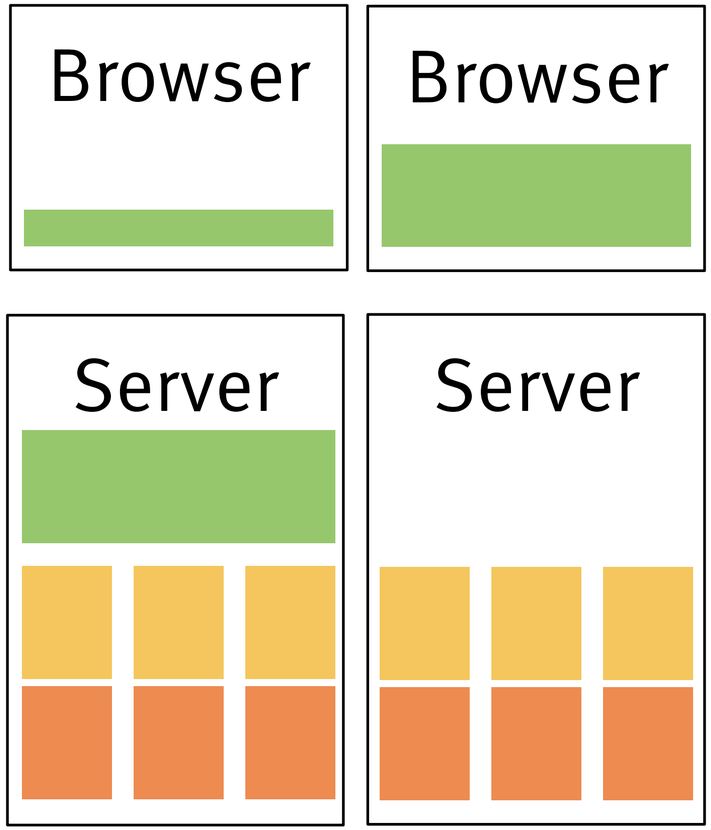
Ziele

- Architektur
  - Wartbarkeit
  - Änderbarkeit
  - kürzere Entwicklungszeiten durch Einsatz von Produkten
  - Ablösung von Altsystemen (Legacy) ohne BigBang
- Performance/Usability
  - kürzere Time to First Meaningful Paint
  - kürzere Time to First Interaction

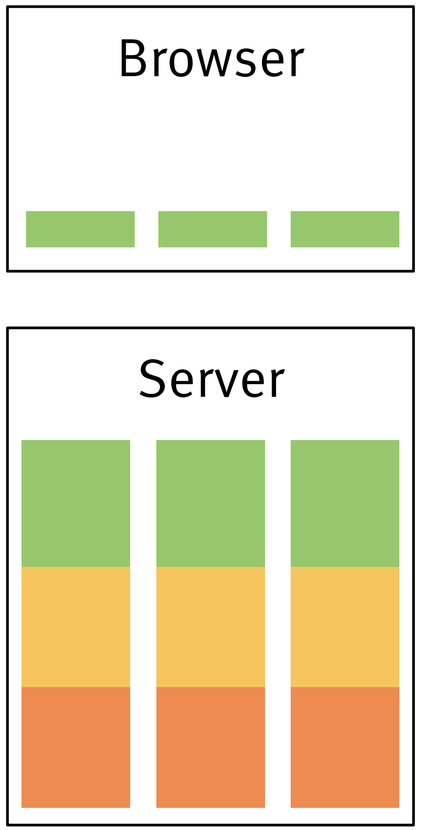
“Monolithen sind super. Deswegen sollten wir unsere großen Systeme aus ganz vielen davon bauen.”

— Stefan Tilkov

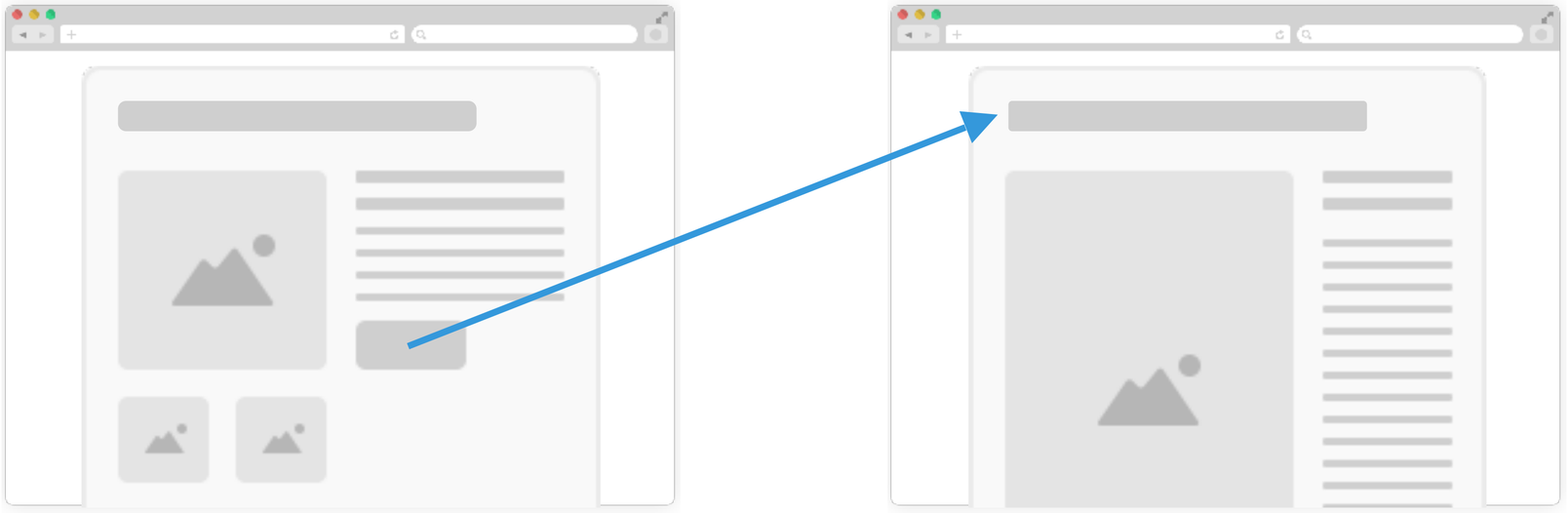
Frontend-Monolithen



SCS, µServices mit UI



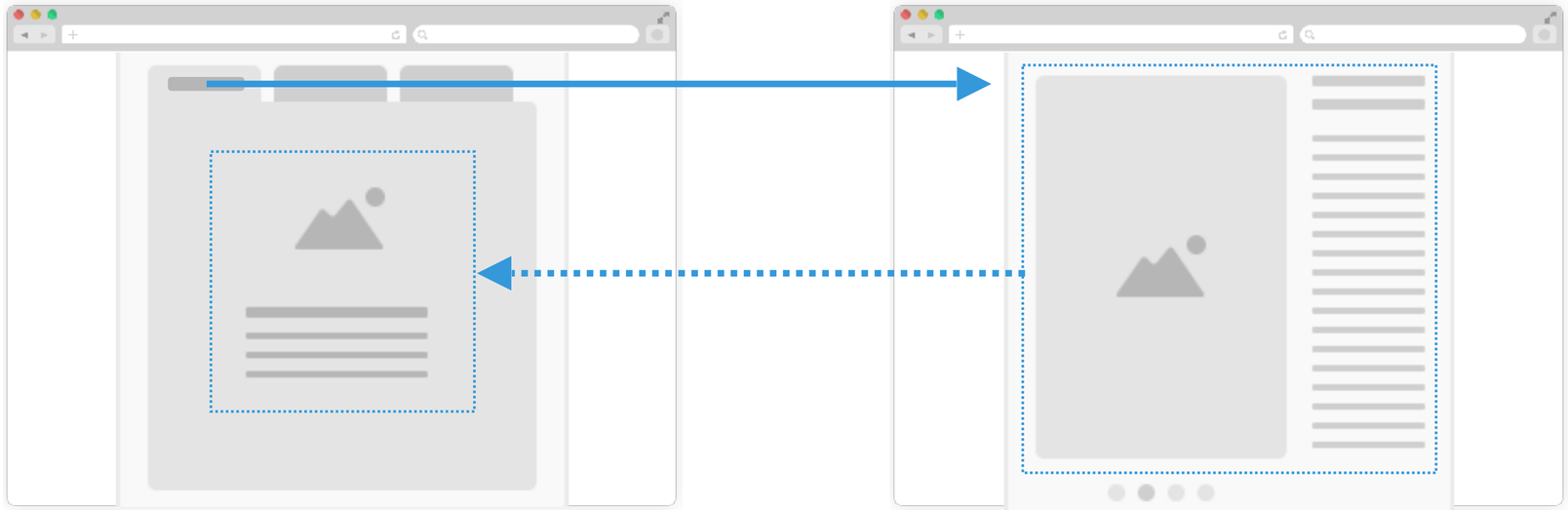
Integration über Links



System 1

System 2

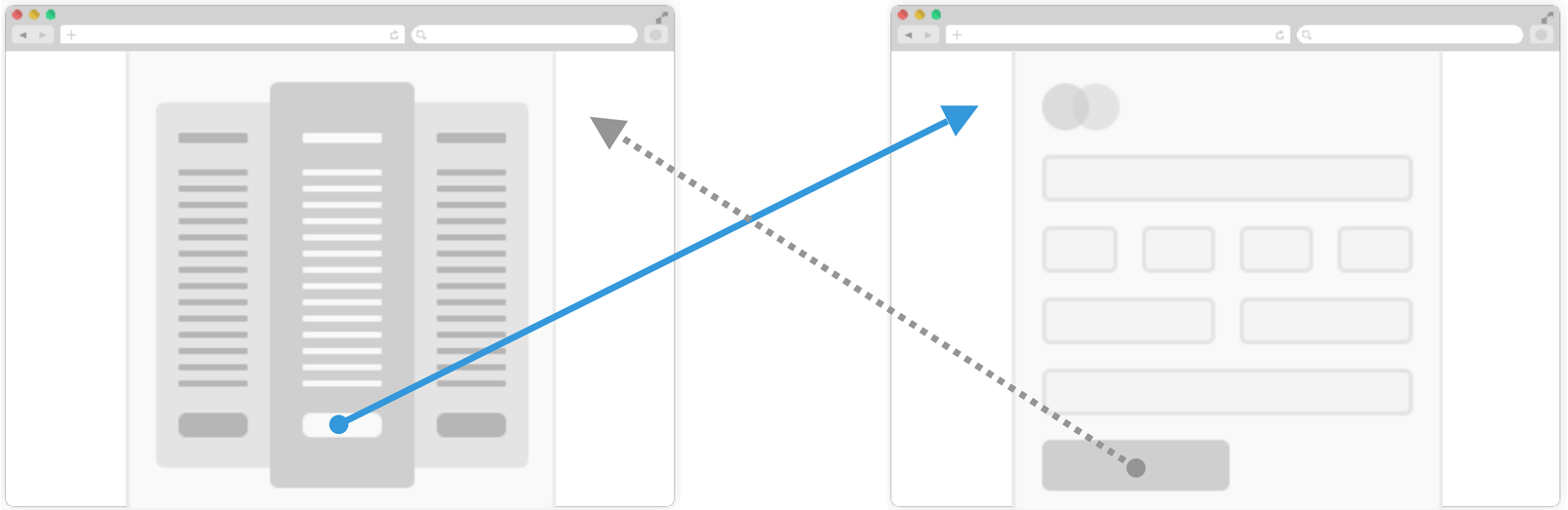
Transklusion



System 1

System 2

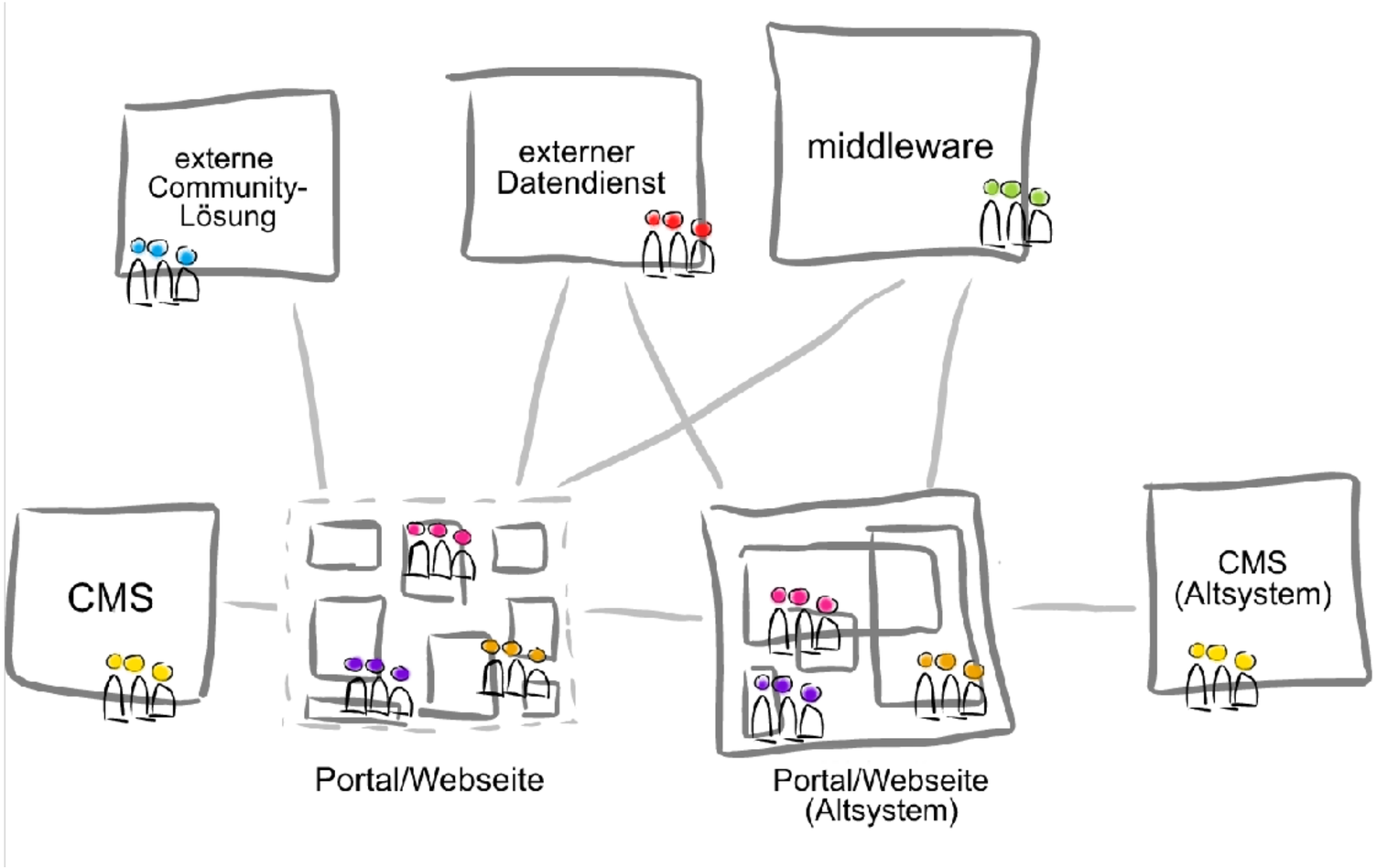
Redirection



System 1

System 2

„Enterprise“

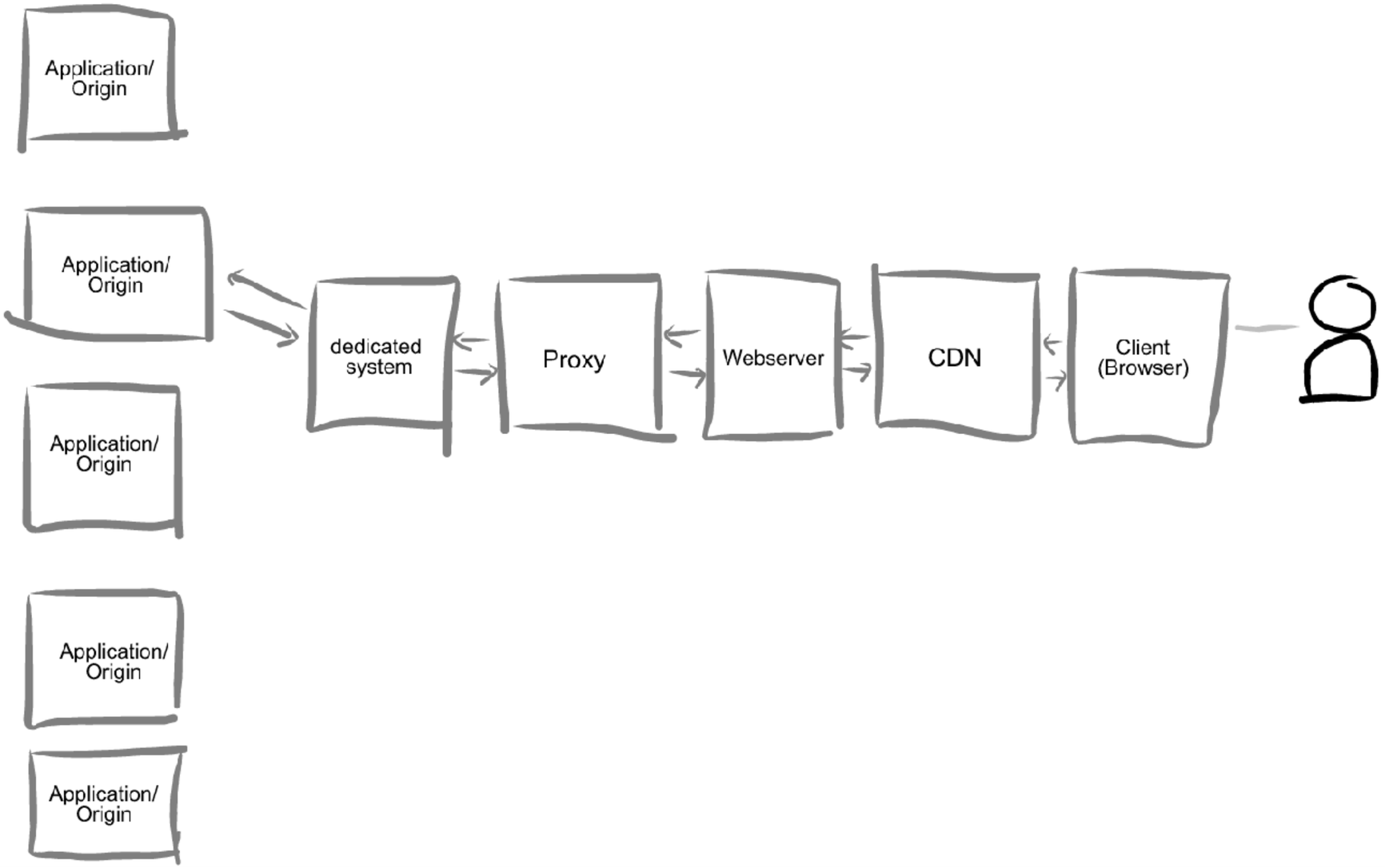

externe Community-Lösung
externer Datendienst
middleware
CMS
CMS (Altsystem)
Portal/Webseite
Portal/Webseite (Altsystem)

Performance/Usability

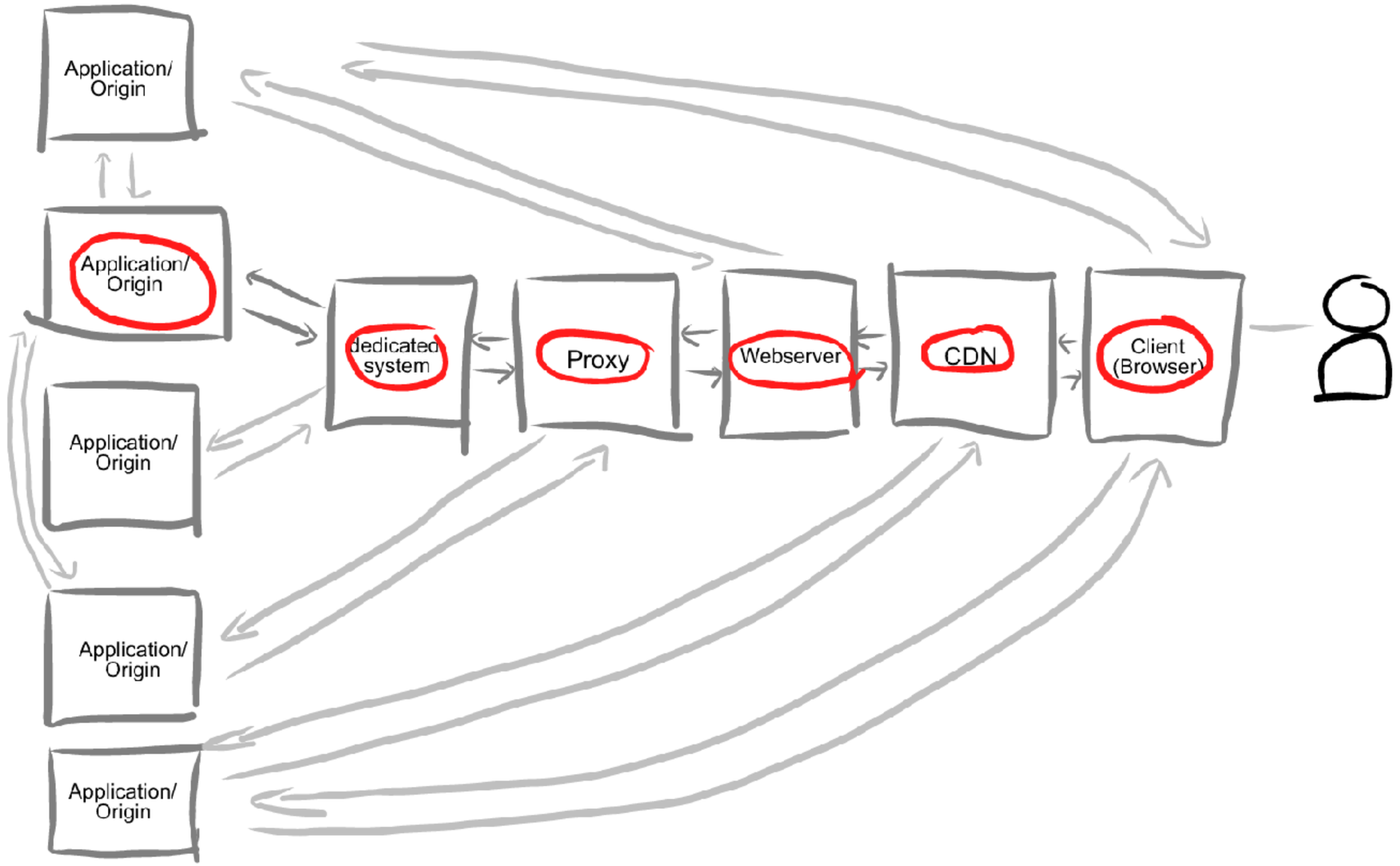
- Schnitt zwischen UIs
  - Caching von Fragmenten ermöglichen
  - Trennung von Hauptinhalten und sekundären Inhalten

Wo und was transkludieren?

Wo?

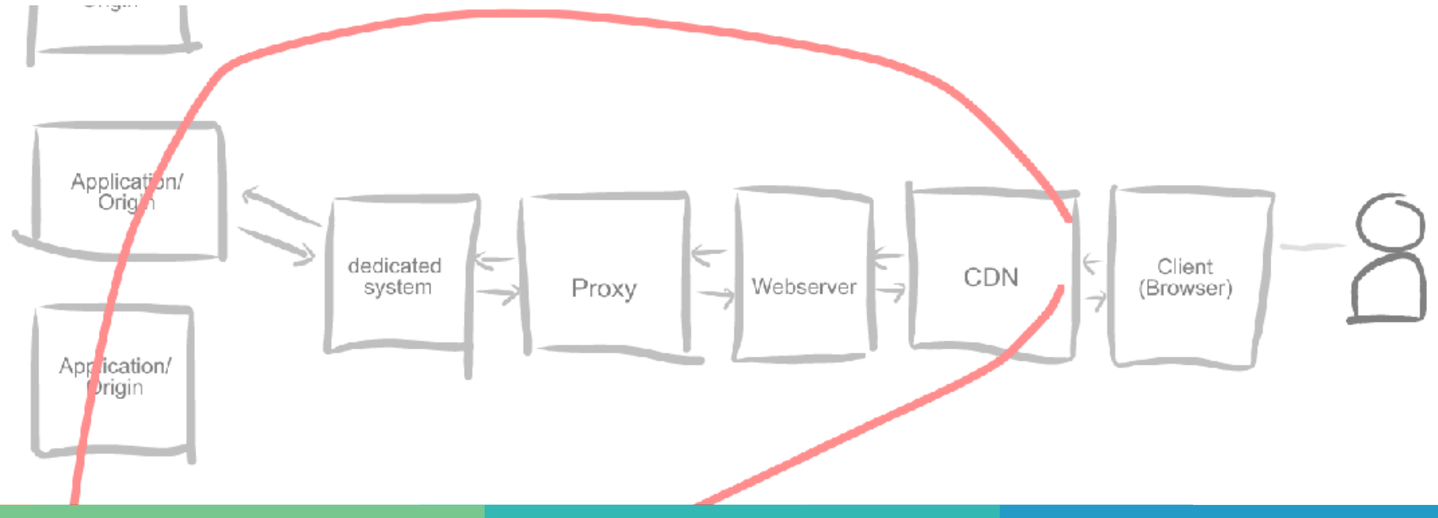

Application/
Origin
Application/
Origin
Application/
Origin
Application/
Origin
Application/
Origin
dedicated
system
Proxy
Webserver
CDN
Client
(Browser)

Wo?


Application/
Origin
Application/
Origin
Application/
Origin
Application/
Origin
Application/
Origin
dedicated
system
Proxy
Webserver
CDN
Client
(Browser)

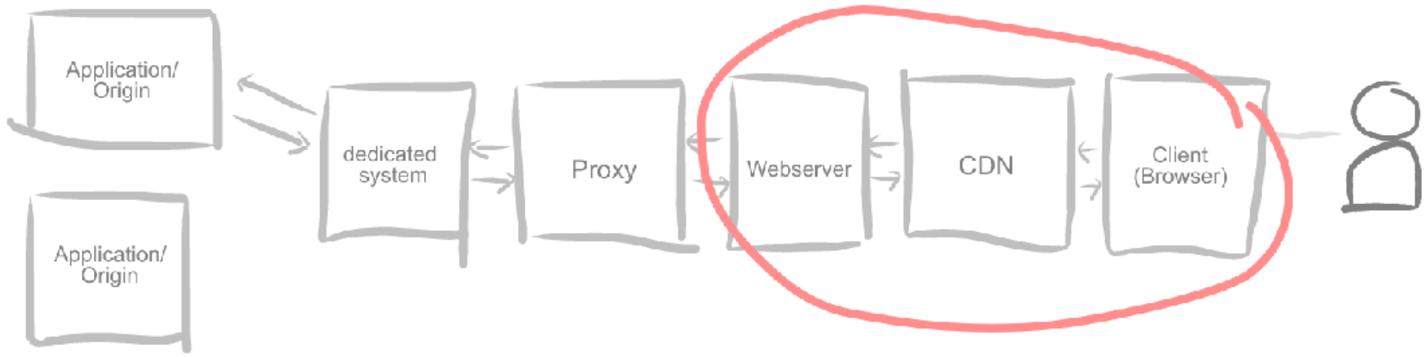
Hauptinhalte

- überall, außer clientseitig
  - Origin-Server, dedizierte Komponente
  - Webserver
  - Proxy/CDN (insbesondere, wenn cachebar)



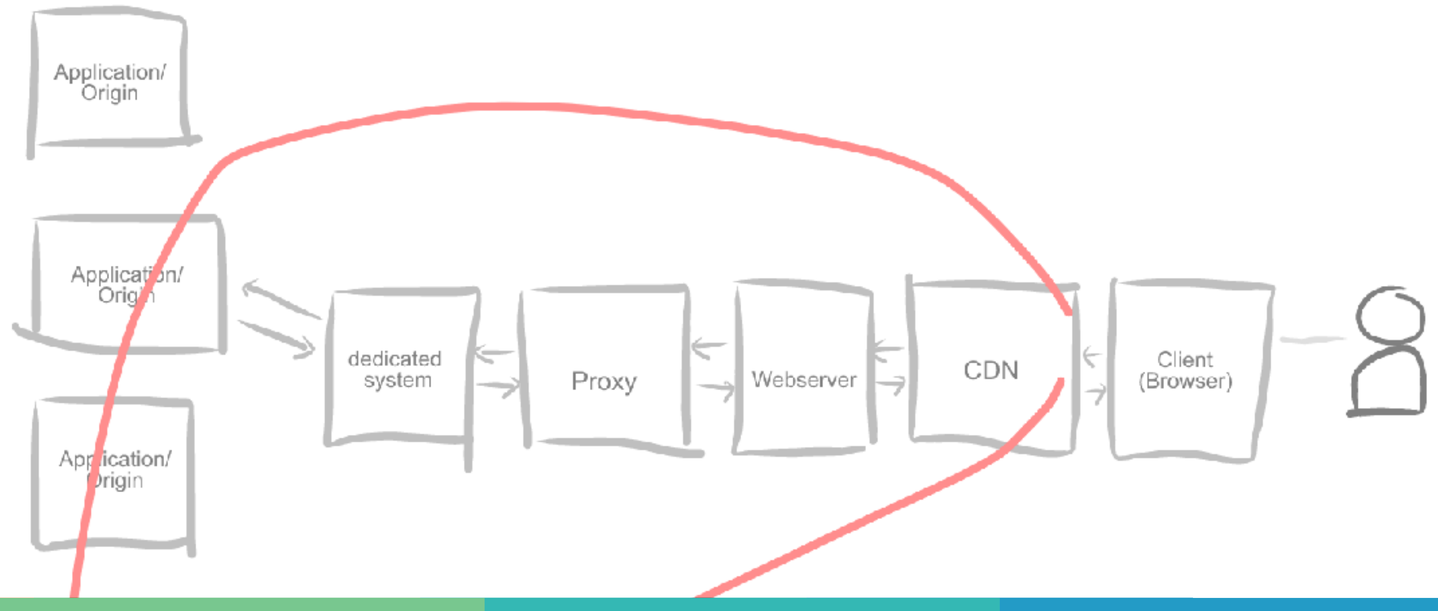
sekundäre Inhalte

- clientseitig
- Proxy/CDN/Webserver:
  wenn cachebar & nicht user-spezifisch



SEO-relevante Inhalte

- überall, außer clientseitig
  - Origin-Server, dedizierte Komponente
  - Webserver
  - Proxy/CDN

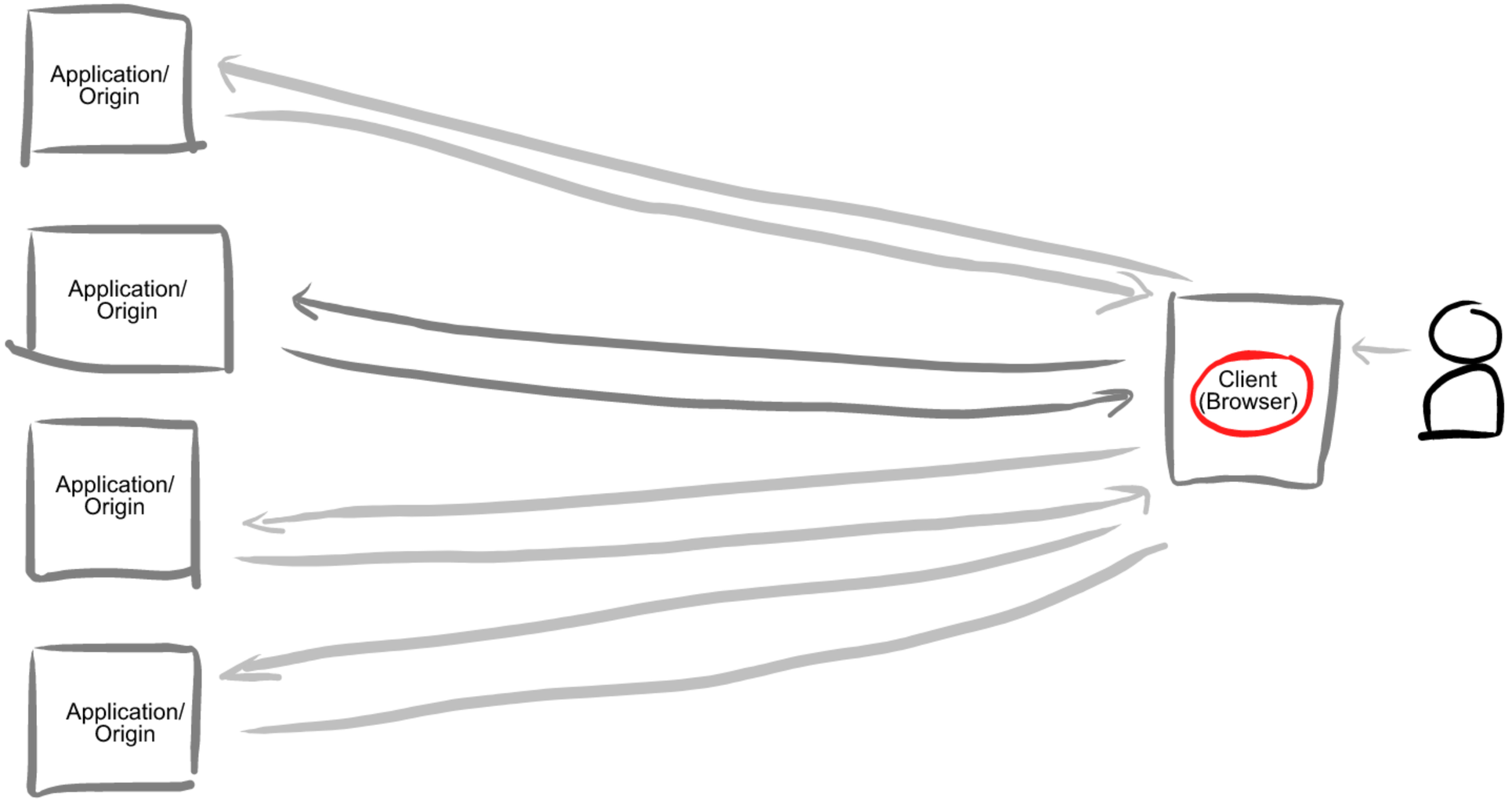


cachebare Inhalte

- nicht user-spezifisch
  - Proxy
  - CDN
  - Webserver
- user-spezifisch
  - clientseitig

Wie transkludieren?

clientseitig



clientseitig

- iframe

```
<iframe src="https://www.google.com/maps/embed?pb=…" width="100%" height="200"
        frameborder="0" style="border:0"></iframe>
```

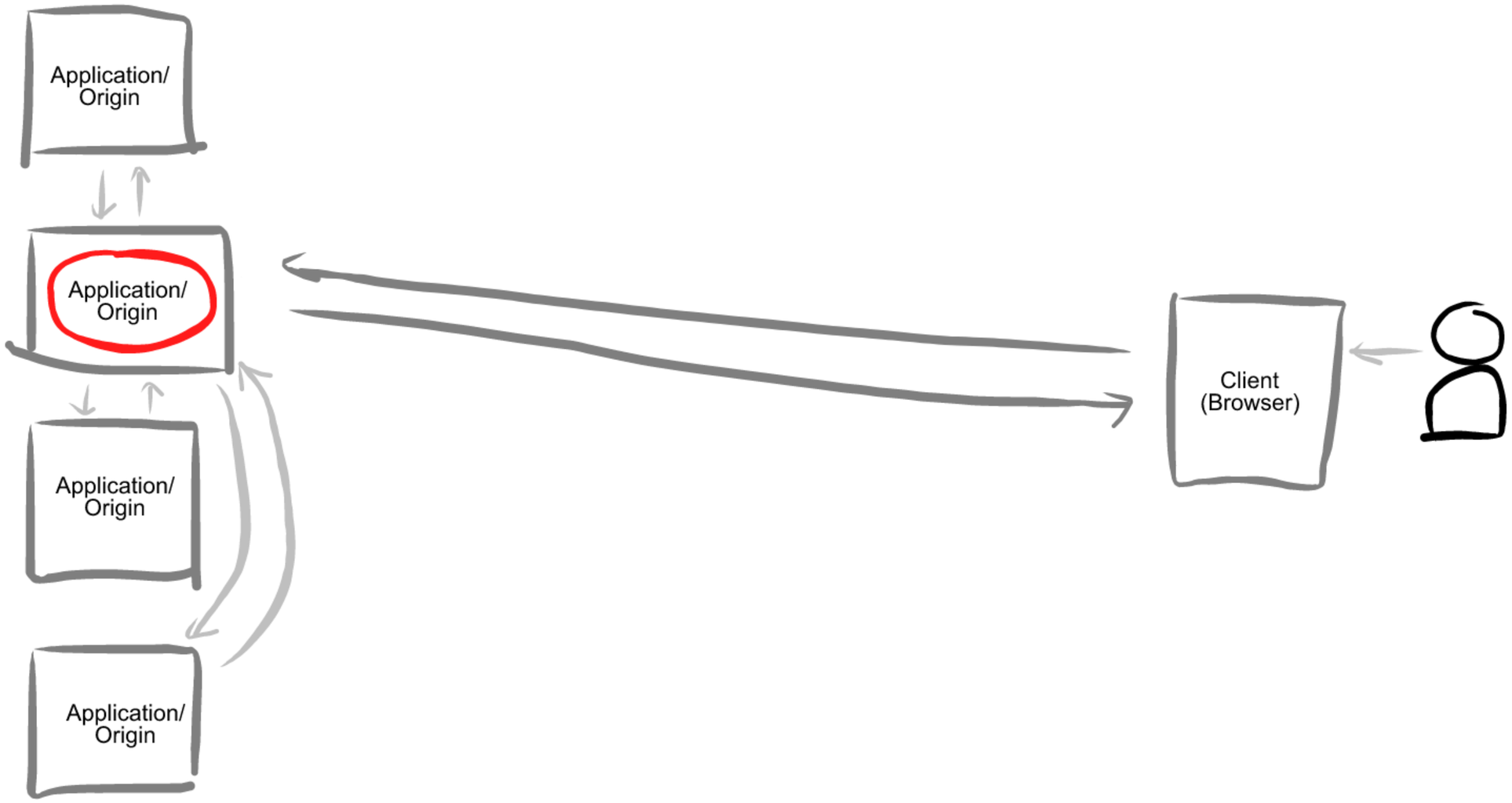
- h-include - https://github.com/gustafnk/h-include

```
<h-include src="/other/document/here.html"></h-include>
```

- „selbst implementiert“ -
  z.B. wie in https://github.com/FND/universal-web-transclusion (PoC)

```
<a is="embeddable-link" href="http://localhost:8000/user">current user</a>
```

Origin Server

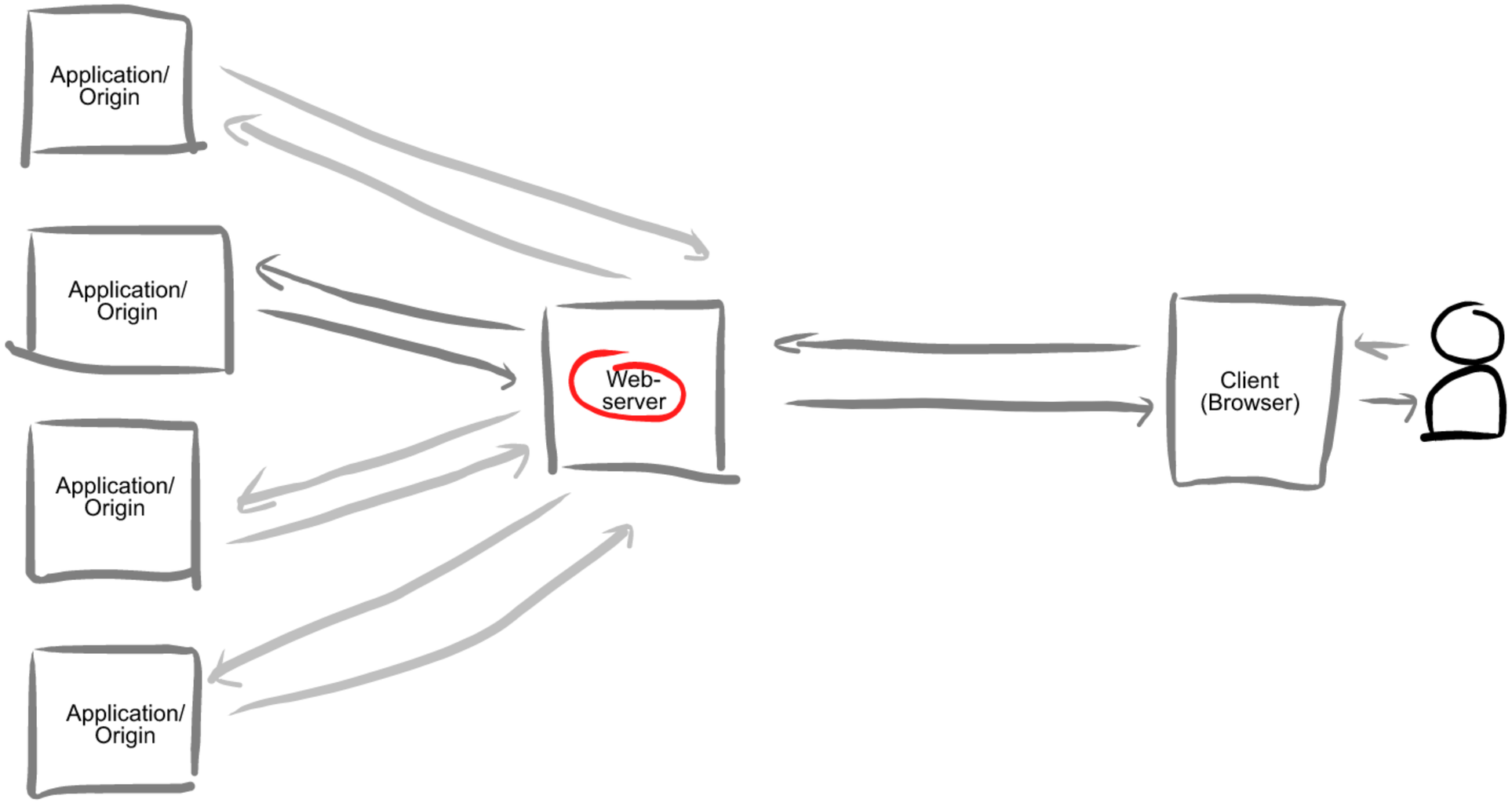


Origin Server

- „selbst implementiert“ -
  z.B. wie in https://github.com/FND/universal-web-transclusion (PoC)

```
<a is="embeddable-link" href="http://localhost:8000/user">current user</a>
```

Webserver

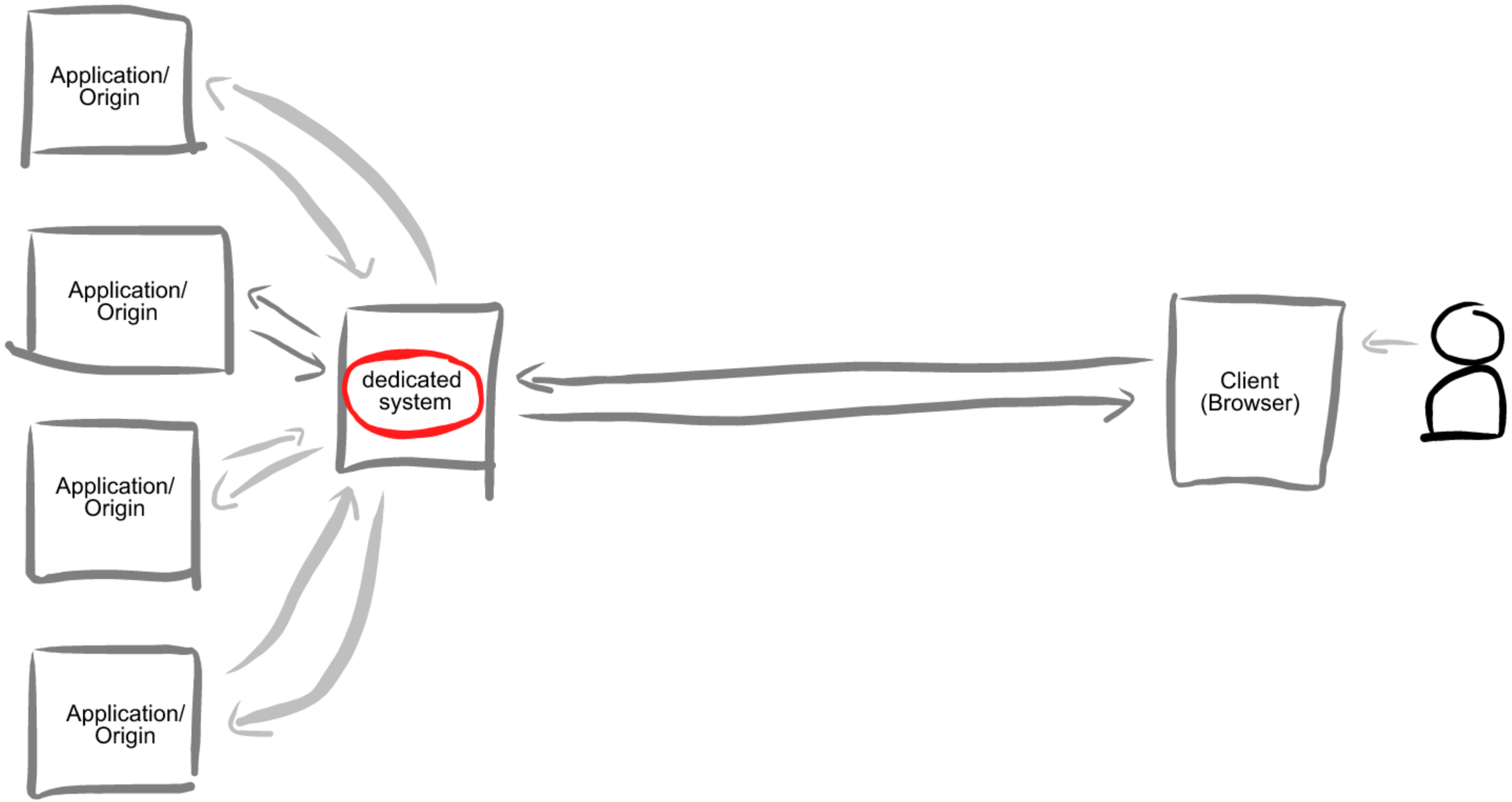

Application/
Origin
Application/
Origin
Application/
Origin
Application/
Origin
Web-
server
Client
(Browser)

Webserver

- Server Side Includes (SSI) - z.B. auf Apache HTTP Server, nginx

```
<!--#include virtual="path/to/a/resource" -->
```

dediziertes System



dediziertes System

z.B. Zalando Tailor

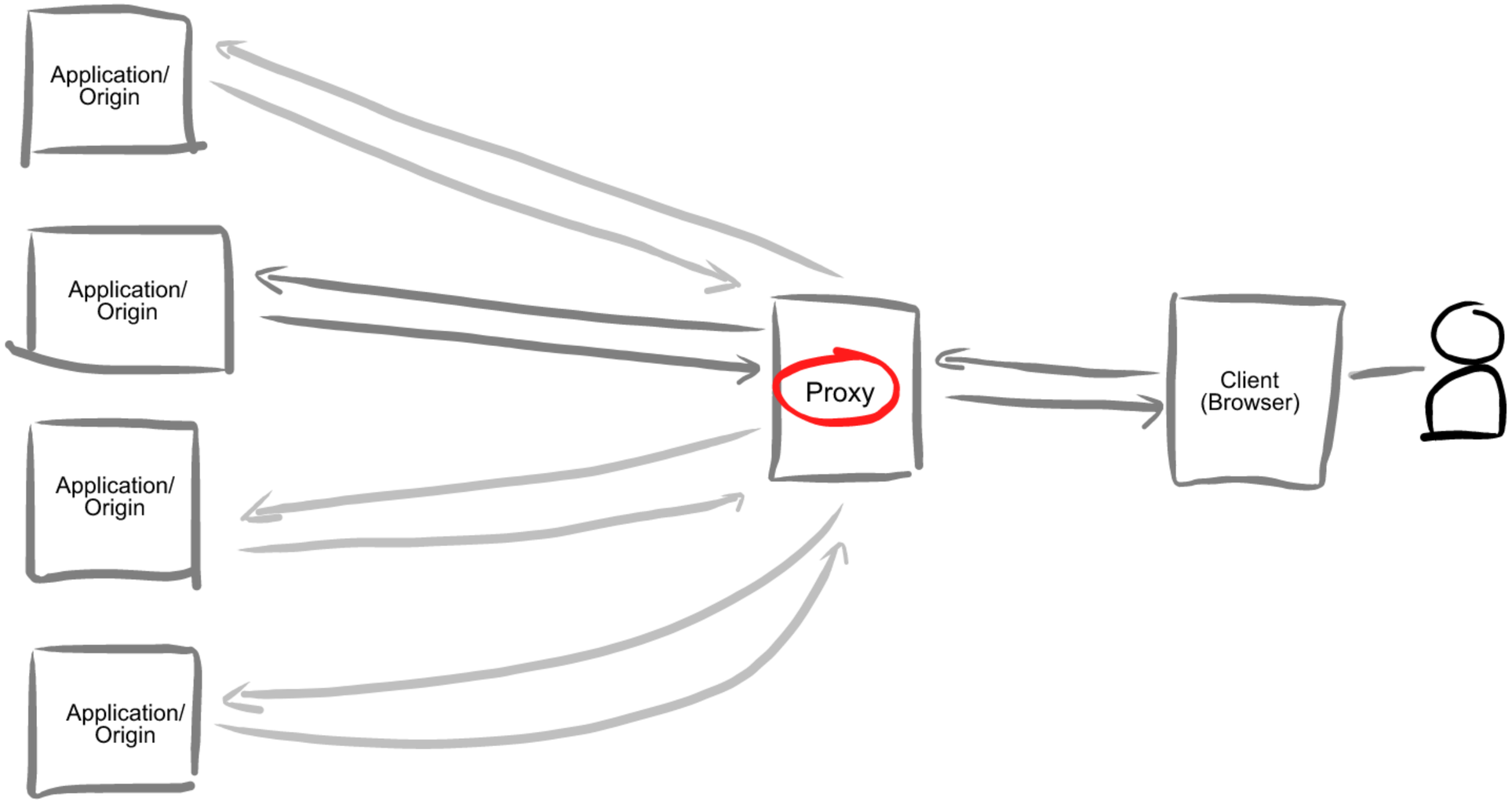
```
<fragment src="http://localhost:8088"
          fallback-src="http://localhost:8081" primary></fragment>

<fragment src="http://localhost:8082" async></fragment>

<fragment slot="body-start" src="http://localhost:8083"
          timeout="1000"></fragment>
```

https://github.com/zalando/tailor

Caching Proxy/CDN



Caching Proxy/CDN

- Edge Side Includes (ESI)

  z.B. auf Varnish, squid, Akamai, Oracle Web Cache

```
<esi:include src="http://example.com/1.html"
             alt="http://bak.example.com/2.html" onerror="continue"/>
```

https://www.w3.org/TR/esi-lang

FAQ

Assets

- dürfen Fragmente Styles und Skripte mitliefern?
  - ja
    - Syntaxfehler crashed ganze Seite
    - ggf. Versionskonflikte von genutzten Bibliotheken
  - nein
    - nur Markup transkludieren

Assets

- zentrale Assets-Bibliothek (Pattern-Library/Living-Style-Guide)
  - push vs. **pull**
    - Versionierung
    - Abwärtskompatibilität
  - inkludiert zur
    - Laufzeit
    - **Buildzeit**
      - Feature Toggles

Caching

- Cache-Control Header setzen
- Proxy/CDN: Surrogate Keys nutzen und bei Änderungen von Ressourcen aktiv invalidieren (zuvor endlos cachen)

```
HTTP/1.1 200 OK
Surrogate-Key: 1234566789 192837465 987654321
Content-Type: text/html
…
```

```
PURGE 192837465
…
```

Fallbacks

- letzter Fallback geht immer:
  graceful degradation zu

```
<a href="http://example.org/userinfo">Hier geht’s zur Userinfo</a>
```

rekursives Laden

- geladenes Dokument enthält wiederum Referenzen
  - grundsätzlich überall möglich:
    ESI, SSI, h-include, Eigenlösung
  - Ausnahme: Tailor
    nur Verschachteln von Base- und Page-Templates
  - Achtung: Performance!

Cookies, Header

- in der Regel nur vom einbettenden System zu setzen
- Setzen von Cookies ggf. über Redirects lösbar

Tests

- Integrationstest
  - Gutfall (alle Systeme da und fehlerfrei)
  - Schlechtfall
    - System ausgefallen/Netzwerkprobleme
    - System antwortet mit ungültiger/fehlerhafter Antwort
    - funktioniert der Fallback?

Beispiel

Beispiel

- Crimson Insurance

https://crimson-portal.herokuapp.com/ (deployt)

https://github.com/innoq/lvm-roca-prototype (Einzelprojekte)

https://github.com/ewolff/crimson-assurance-demo (mit Docker)

http://roca-style.org/

Verwandtes

chunked HTML

- Header `Transfer-Encoding: chunked`
- Konzept: sekundäre, aufwendige Inhalte nachgelagert ausliefern
  - Hauptinhalte werden sofort geflusht
  - am Ende des HTML bodys sekundäre Inhalte anhängen und im Client an die eigentliche Stelle kopieren

- https://en.wikipedia.org/wiki/Chunked_transfer_encoding
- https://de-de.facebook.com/notes/facebook-engineering/bigpipe-pipelining-web-pages-for-high-performance/389414033919/
- https://github.com/zalando/tailor
- http://www.bigpipe.io

chunked HTML - Tailor

```
<fragment src=„http://localhost:8088" async></fragment>
```

chunked HTML - Tailor

```
<html>
  …
<body>
  …
 <script data-pipe>_p941.placeholder(2)</script>
  …

 <script>_p941.loadCSS("http://localhost:8082/fragment.css")</script>
 <script data-pipe>_p941.start(2, "http://localhost:8082/fragment.js")</script>
  <div class="fragment-world">
   Fragment world
  </div>
 <script data-pipe>_p941.end(2, "http://localhost:8082/fragment.js", "2")</script>
</body>
</html>
```

HTTP/2 Server Push

- Server schickt Push-Ressourcen, bevor diese vom Client angefordert werden
- gesteuert über Link Header

```
Link: </css/styles.css>; rel=preload; as=style,
      </js/scripts.js>; rel=preload; as=script,
      </img/logo.png>; rel=preload; as=image,
      </app/transclude/resource>; rel=preload; as=fetch
```

https://w3c.github.io/preload/

https://www.innoq.com/de/talks/

Franziska Dessart | @lapaqui
franziska.dessart@innoq.com

Dankeschön!

Fragen?

Anmerkungen?

innoQ
www.innoq.com

innoQ Deutschland GmbH

Krischerstr. 100
D-40789 Monheim am Rhein
Germany
Phone: +49 2173 3366-0

Ohlauer Straße 43
D-10999 Berlin
Germany
Phone: +49 2173 3366-0

Ludwigstr. 180 E
D-63067 Offenbach
Germany
Phone: +49 2173 3366-0

Kreuzstr. 16
D-80331 München
Germany
Telefon +49 2173 3366-0

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland
Phone: +41 41 743 0116