

CDNs implement ESI

- CDNs implement ESI

CDNs run servers at several Internet nodes so that every user can load web pages and images from a nearby server, thus, **reducing loading times**.

CDNs and caches implement ESI to be able to assemble web pages from different fragments. Static parts can be cached, even if other parts must be dynamically generated.

ESI not only offers features for frontend integration but also **features especially useful for caching**.

Varnish is a web cache and is used as an ESI implementation in the example.

Varnish configuration

```
vcl 4.0;

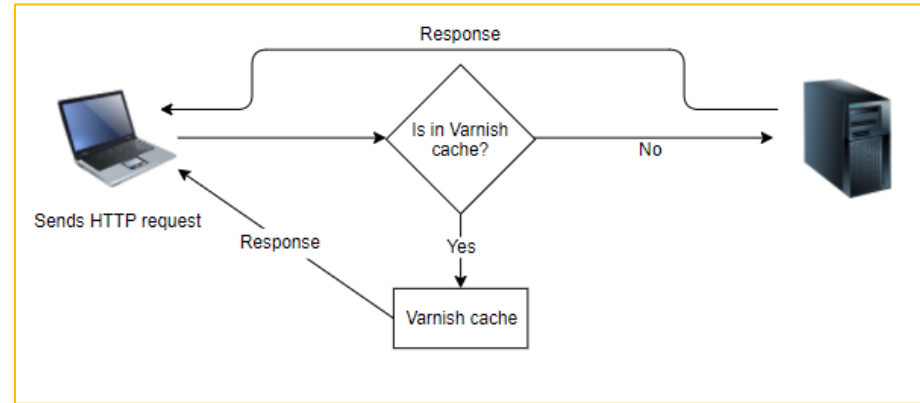
backend default {
    .host = "order";
    .port = "8080";
}

backend common {
    .host = "common";
    .port = "8180";
}

sub vcl_recv vcl_recv {
    if (req.url ~ "^/common") {
        set req.backend_hint = common;
    }
}

sub vcl_backend_response {
    set beresp.do_esi = true;
    set beresp.ttl = 30s;
    set beresp.grace = 15m;
}
```

```
<html>
<head>
    ...
    <esi:include src="/common/header"></esi:include>
</head>
<body>
    <div class="container">
        <esi:include src="/common/navbar"></esi:include>
        ...
    </div>
    <esi:include src="/common/footer"></esi:include>
</body>
</html>
```



```
<html>
<head>
    ...
    <link rel="stylesheet"
    href="/common/css/bootstrap-3.3.7-dist/css/bootstrap.min.css" />
    <link rel="stylesheet"
    href="/common/css/bootstrap-3.3.7-dist/css/bootstrap-theme.min.css" />
</head>
<body>
    <div class="container">
        <a class="brand"
        href="https://github.com/ultraq/thymeleaf-layout-dialect">
            Thymeleaf - Layout </a>
        Mon Sep 18 2017 17:52:01 </div></div>
    ...
</div>
<script src="/common/css/bootstrap-3.3.7-dist/js/bootstrap.min.js" />
</body>
</html>
```

CDNs implement ESI with Variations

Different ESI implementations

instead of Varnish, a different ESI implementation could be used by [Squid](#) or by a CDN like [Akamai](#).

SSI(Server Side Includes) : CDN like Akamai and Squid(<http://www.squid-cache.org/>) using this for ESI like output.



Tailor

[Tailor](#) is a system for server-side frontend integration that Zalando implemented as part of [Mosaic](#). It is optimized for showing the user the first parts of the HTML page as quickly as possible. For e-commerce, the rapid display of a web page is very important to keep users and can increase sales.

To achieve this, Tailor implements a [BigPipe](#). First, very simple HTML code is transferred to the user in order to be able to display a simple page very quickly. JavaScript is used to load more details step by step. Tailor implements this with asynchronous I/O using Node.js streams.

Client-side integration - for **optional** elements makes sense. Dealing with failed services is then a task for the client code.

Additional integration - Pure frontend integration is rarely enough. Therefore, a system will combine backend integration with synchronous or asynchronous communication mechanisms with frontend integration.