



**adaptTo()**

APACHE SLING & FRIENDS TECH MEETUP  
BERLIN, 23-25 SEPTEMBER 2013

**Become happier by using Varnish Cache**

**Stefan Maurer**

**Senior Software Engineer, Namics AG**

- **Varnish Cache at a glance**
- Use Cases and Architectures
- Limitations and Pitfalls

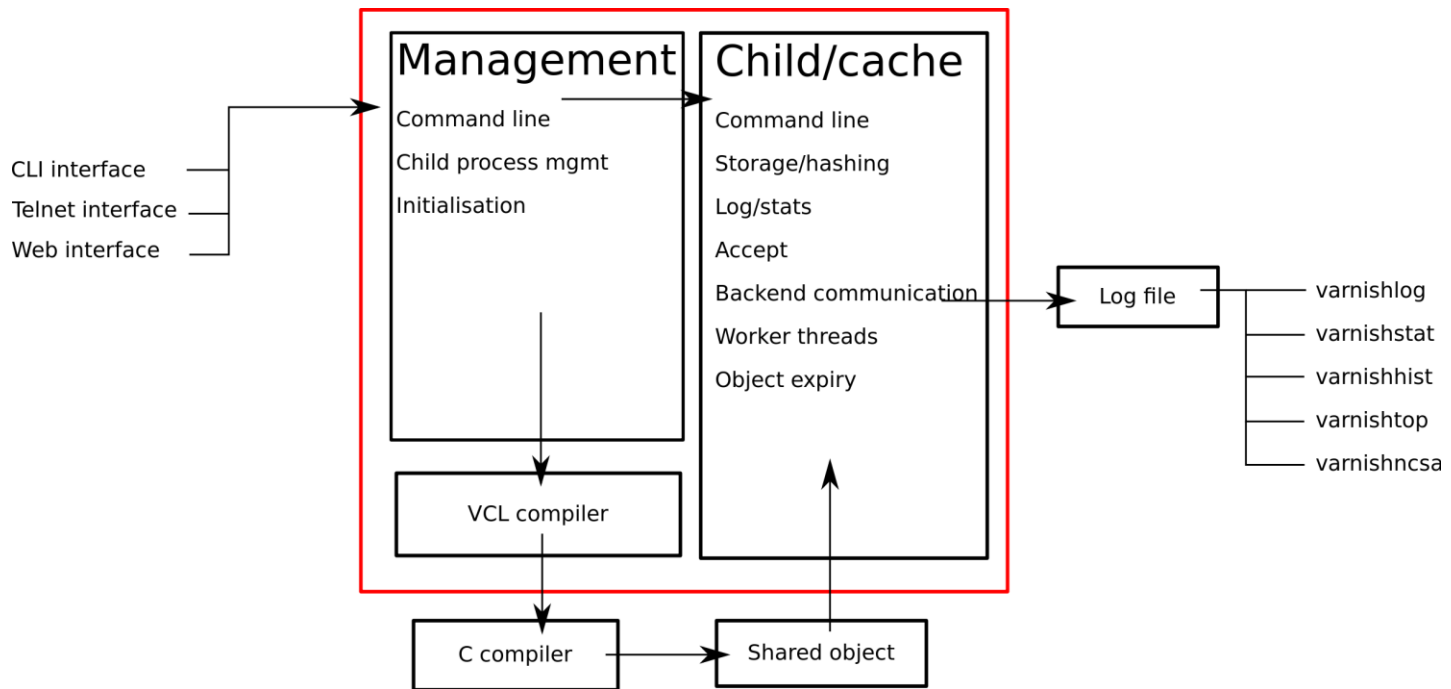
- Reverse Proxy Cache
  - 1.0.0 in 2006
  - Current version is 3.0.4
  - Free under BSD License
  - Available for various Linux Distributions, Max OS X, Open Solaris, Solaris 10, FreeBSD and Windows (with Cygwin)

# Varnish Cache at a glance

- Designed as HTTP accelerator – not more, not less
  - “Varnish is a modern program, designed and written for modern operating systems”  
<https://www.varnish-cache.org/trac/wiki/VarnishFeatures>
  - Storage
    - malloc or file
  - Load balancing
  - ESI (subset)

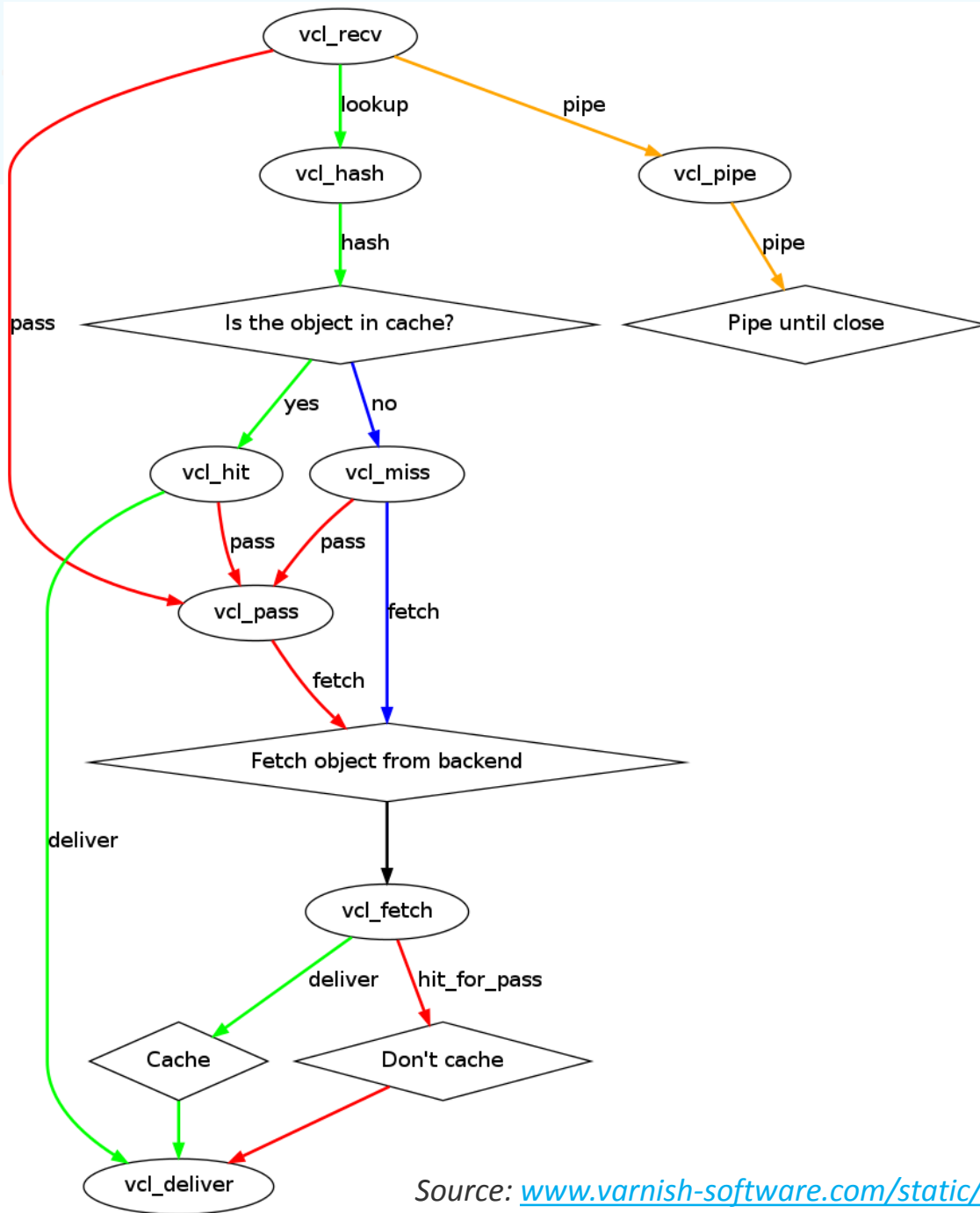
# Varnish Cache at a glance

- Process architecture
  - Management process and child process
  - Configuration change without restart



- Configuration
  - Startup parameters
  - Varnish Configuration Language (VCL)

```
sub vcl_fetch {  
    if (req.url ~ "\.jpg$") {  
        set beresp.ttl = 60s;  
    }  
}
```



- Logging
  - Logs to shared memory
  - Separate application for display and write back



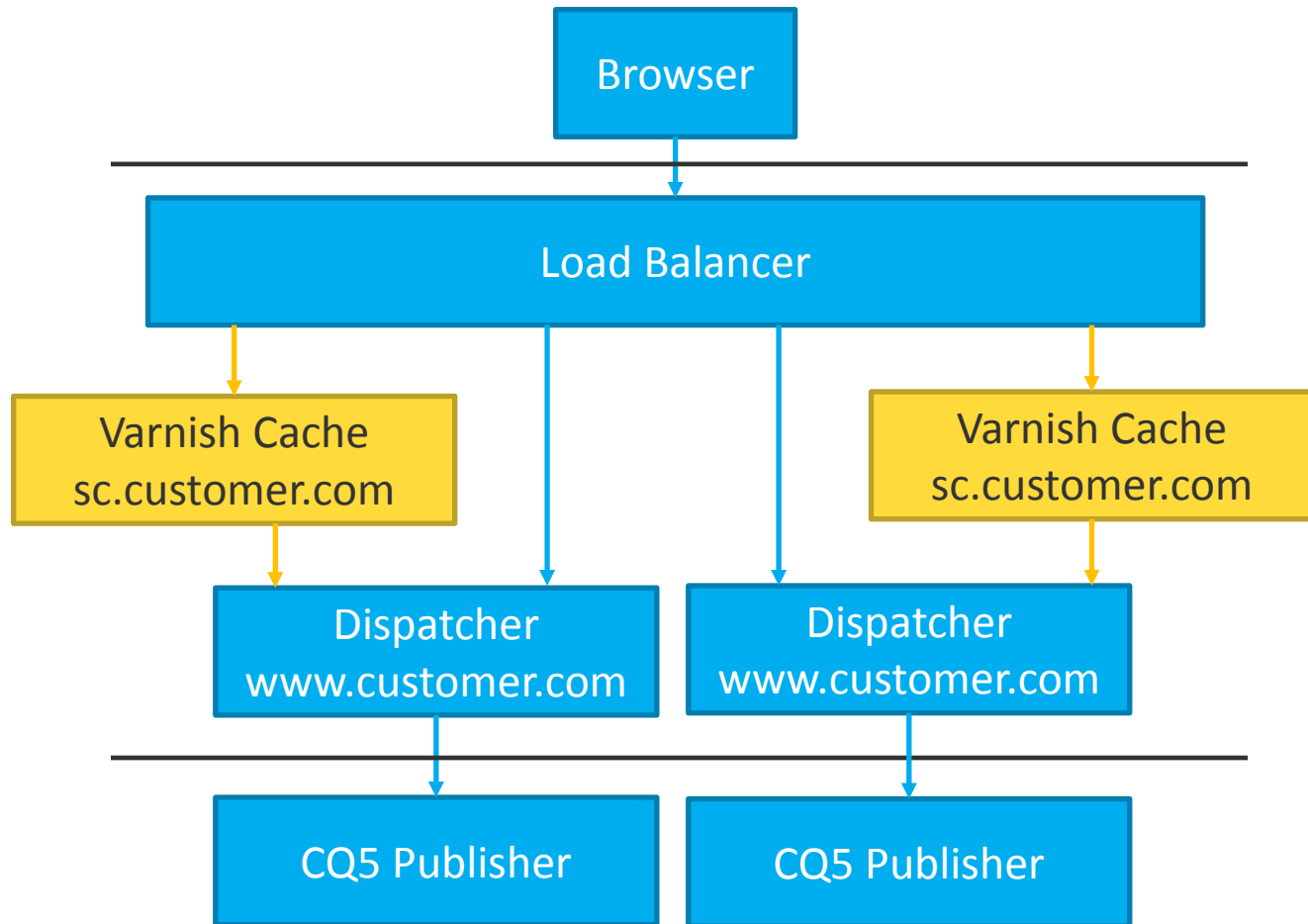
- Varnish Cache at a glance
- **Use Cases and Architectures**
- Limitations and Pitfalls

- **Use case 1: Static resources**
- Use case 2: REST API calls
- Use case 3: Full website

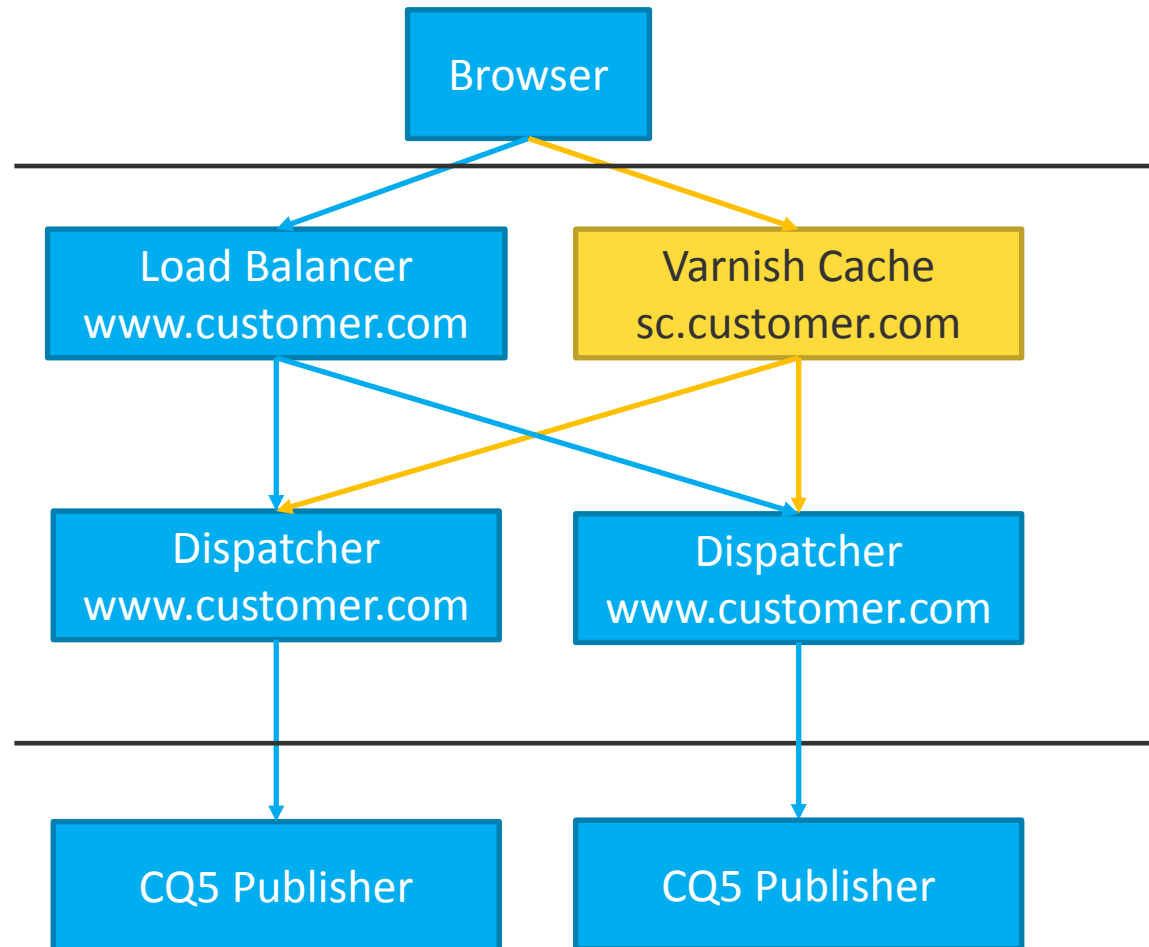
# Use Case 1: Static Resources

- Use Case
  - Cache resources like CSS / JS / static images “forever”
- Benefit with Varnish
  - Internal network traffic decreases significant
  - Do not bore backend systems ;-)
- Solution
  - Cookie-less domain [sc.customer.com](http://sc.customer.com) served by Varnish

- Architecture Sample 1



- Architecture Sample 2



- Why not directly connect Varnish with CQ5 Publisher?
  - Possible. But requires configuration for restricted paths like /system/console
- Changes in CQ
  - Load resources from static domain with a timestamp:

```
<script
type="text/javascript"
src="http://sc.customer.com/etc/designs/customer/clientlib.201309231000.js"
></script>

```

```
backend default {
    .host = "dispatcher-hostname";
    .port = "80";
}
```

```
sub vcl_recv {
    if (req.url !~ "^/etc/designs/" && req.url !~ "^/content/dam/") {
        error 405 "Not allowed.";
    }
    unset req.http.cookie;
}
```

```
sub vcl_fetch {
    unset beresp.http.set-cookie;
    unset beresp.http.expires;
    unset beresp.http.Etag;
    set beresp.http.Cache-Control = "max-age=86400";
    set beresp.ttl = 4w;
    set beresp.http.magicmarker = "1";
}
```

```
sub vcl_deliver {
    if (resp.http.magicmarker) {
        unset resp.http.magicmarker;
        set resp.http.age = "0";
    }
}
```

# Use Cases and Architectures

- Use case 1: Static resources
- **Use case 2: REST API calls**
- Use case 3: Full website



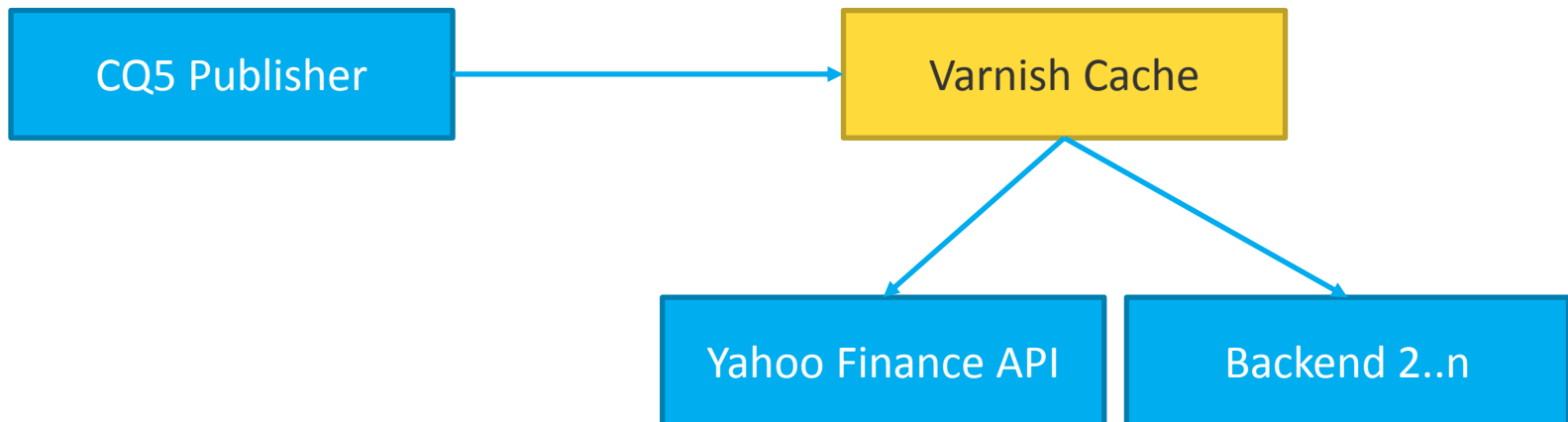
# Use Case 2: REST API Calls

- Use Case
  - Cache requests to backend APIs
- Benefit with Varnish
  - Reduce load on backend systems
  - Reduce response time
  - Handle backend downtime
  - Custom time to life for each backend
- Solution
  - Call backend through Varnish Cache

## Use Case 2: REST API Calls

- **Sample: Yahoo Finance API**
  - <http://finance.yahoo.com/d/quotes.csv?s=ADBE&f=snd1l1c1p2>
  - Last trade date, price, change and percents
  - But: has limit of requests/day
- **CQ Publisher calls Varnish with prefix “finance”**
  - <http://varnish/finance/d/quotes.csv?s=ADBE&f=snd1l1c1p2>
- **Varnish executes API Call and caches the response for one hour**

## ■ Architecture



```
backend yahoofinance {
    .host = "finance.yahoo.com";
    .port = "80";
}

sub vcl_recv {
    if (req.url ~ "^/finance") {
        set req.backend = yahoofinance;
        set req.url = regsub(req.url, "^/finance", "");
        unset req.http.cookie;
    }
}

sub vcl_fetch {
    if(req.backend == yahoofinance) {
        set beresp.ttl = 1h;
        unset beresp.http.set-cookie;
        return(deliver);
    }
}
```

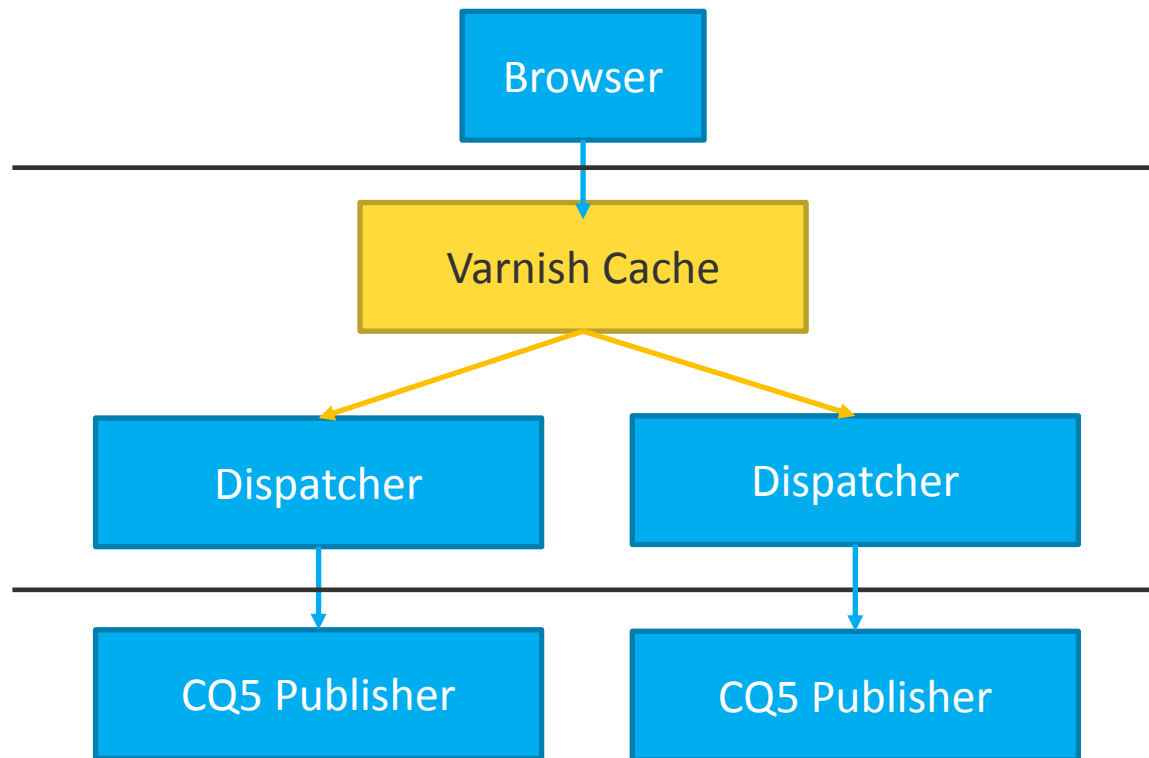
# Use Cases and Architectures

- Use case 1: Static resources
- Use case 2: REST API calls
- **Use case 3: Full website**

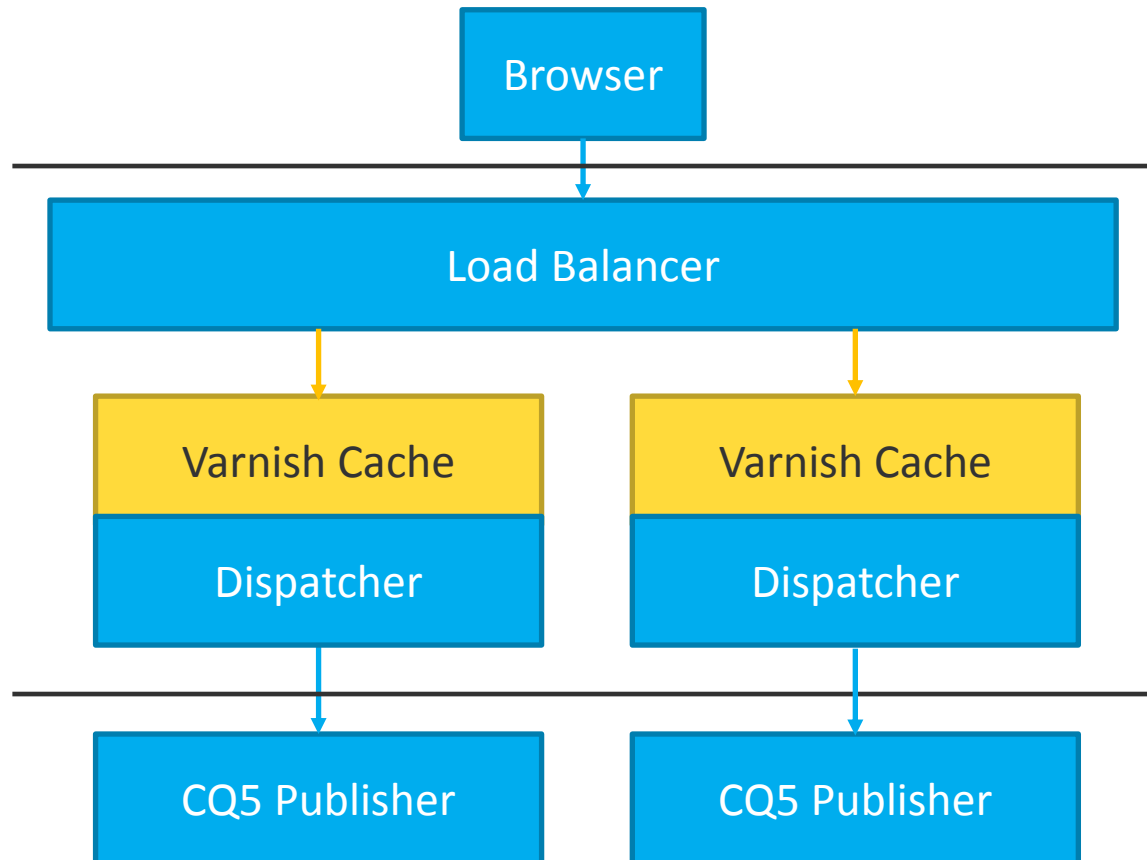
# Use Case 3: Full Website

- Use Case
  - Cache generated HTML and related resources
- Benefit with Varnish
  - Reduce load on backend systems
  - Reduce response time
  - Granular cache invalidation

- Architecture Sample 1



- Architecture Sample 2





## ■ Cache Invalidation

	Short TTL (1 minute)	Long TTL (1 month)
Frequently called pages	+	+
Seldom called pages	-	+
Instant invalidation	-	+
Low complexity	+	-

## ■ Excludes from cache

```
sub vcl_recv {
    if (req.url ~ "\?"
        || req.url ~ "\/cug_") {
        return(pass);
    }
}

sub vcl_fetch {
    if (beresp.http.Dispatcher ~ "no-cache"
        || beresp.http.cache-control ~ "(no-cache|private)"
        || beresp.http.pragma ~ "no-cache") {
        set beresp.ttl = 0s;
    }
    else {
        set beresp.ttl = 4w;
    }
}
```

- Invalidate a page with “smart bans”

```
acl purgers {
    "127.0.0.1";
    "192.168.0.0"/24;
}

sub vcl_recv {
    if (req.http.X-Purge-URL) {
        if (!client.ip ~ purgers) {
            error 405 "Method not allowed";
        }
        ban("obj.http.x-url == " + req.http.X-Purge-URL);
        error 200 "Banned URL";
    }
}

sub vcl_fetch {
    set beresp.http.x-url = req.url;
}
```

## Use Case 3: Full Website

- Invalidate from CQ Publisher with Modification Listener
  - **Modify:** `/content/customer/en/news/adaptto`
  - **Risky: Purge only page and sub pages**  
`/content/customer/en/news/adaptto.*`
  - **Safe: Purge whole language tree**  
`/content/customer/en.*`

# Agenda

- Varnish Cache at a glance
- Use Cases and Architectures
- **Limitations and Pitfalls**

- **SSL termination**
  - **Problem**

Varnish has no SSL termination
  - **Justification**

Varnish source code: 80'000 lines of code  
OpenSSL: 340'000 lines of code
  - **Solution**

Use another tier in front of Varnish (nginx, pond, ...)

- Avoid “double purge”
  - **Assumed situation**  
2 Publisher and 1 Varnish
  - **Problem**  
Modification listener is executed on both Publishers → 2 purge requests
  - **Solution**  
Define “master” publisher and handle failover

- **Proxy**
  - **Assumed situation**
    - Varnish is used to cache REST API Calls
    - Your customer force to use proxy for all external requests
  - **Problem**

Varnish has no configuration for backend through proxy
  - **Solution**

Set proxy as backend and tweak URL



## ■ Proxy solution

```
backend proxy {
    .host = "corporate-proxy-hostname";
    .port = "8080";
}

sub vcl_recv {
    set req.backend = proxy;
    set req.http.X-Forwarded-For = client.ip;
    set req.http.host = "dispatcher-hostname";
    set req.http.port = 80;

    set req.url = "http://" + req.http.host + ":" +
                  req.http.port + req.url;
}
```

Thank you

- **5 Locations in Germany and Switzerland**  
Frankfurt, Hamburg, München, Zürich, St. Gallen
- **Many interesting projects with CQ5 and other technologies**
- **400 + 1 (?) Employees**  
[www.namics.com/jobs](http://www.namics.com/jobs)

# Resources and Further Links

- <https://www.varnish-cache.org/trac/wiki/VCLExampleLongerCaching>
- <https://www.varnish-software.com/static/book/Tuning.html>
- [https://www.varnish-software.com/static/book/\\_images/vcl.png](https://www.varnish-software.com/static/book/_images/vcl.png)
- <http://blog.namics.com/2013/01/www-zh-ch-ist-50x-schneller-und-stabiler.html>
- <http://blog.namics.com/2010/12/varnish-cache.html>

# Variable availability in VCL

Variable	recv	fetch	pass	miss	hit	error	deliver	pipe	hash
req.*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
bereq.*		R/W	R/W	R/W				R/W	
obj.hits					R		R		
obj.ttl					R/W	R/W			
obj.grace					R/W				
obj.*					R	R/W			
beresp.*		R/W							
resp.*						R/W	R/W		