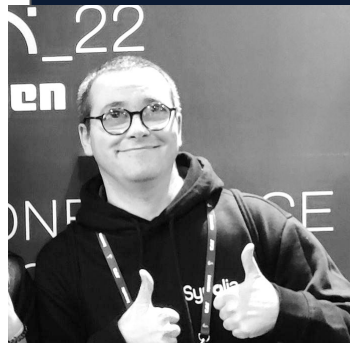# Sylius&
WEB PERFORMANCE

WHO ARE WE ?

OLIVIER ALLAIN

Solution Architect at Synolia

Sylius Early adopter

Sylius' Core Team Member

HÉLÈNA GRAVELIER

Project Directress and Sylius Solution Manager at Synolia
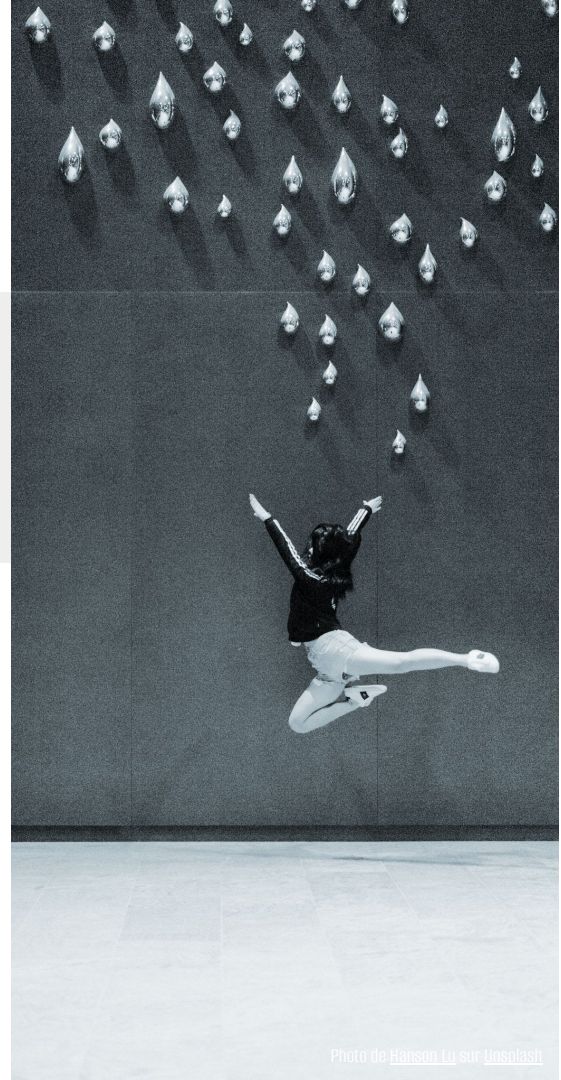
Ecommerce and UX addict

please forgive our wonderful french accent !

WHAT'S
WEB PERFORMANCE ?

The **speed** in which web pages are downloaded and **displayed** on the user's browser.

WHY SHALL WE MEASURE WEB PERFORMANCE ?

MEASURING WEB PERFORMANCE IS IMPORTANT

HEALTH CHECK

- Identify improvements
- Measure optimizations' impact
- Monitor degradation

MEASURING WEB PERFORMANCE IS IMPORTANT

HEALTH CHECK

- Detect improvements
- Measure optimizations' impact
- Monitor degradation

BETTER USER EXPERIENCE

- Increase user satisfaction
- Bring users back
- Increase sales and sales revenue

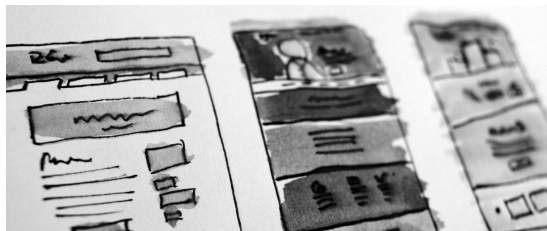# MEASURING WEB PERFORMANCE IS IMPORTANT

HEALTH CHECK

- Detect improvements
- Measure optimizations' impact
- Monitor degradation

BETTER USER EXPERIENCE

- Increase user satisfaction
- Bring users back
- Increase sales and sales revenue

POSITIVE ECOLOGICAL IMPACT

- Less data travelling
- Energy consumption

HOW DO WE MEASURE
WEB PERFORMANCE

GOOD METRICS FOR A HEALTHY SITE

CORE WEB VITALS

(Loading)

LCP
Largest Contentful Paint

GOOD	NEEDS IMPROVEMENT	POOR

2.5 sec   4.0 sec

(Visual Stability)

CLS
Cumulative Layout Shift

GOOD	NEEDS IMPROVEMENT	POOR

0.1   0.25

(Interactivity)

FID
First Input Delay

GOOD	NEEDS IMPROVEMENT	POOR

100 ms   300 ms

GOOD METRICS FOR A HEALTHY SITE

OTHER WEB VITALS

FCP
First Contentful Paint

GOOD	NEEDS IMPROVEMENT	POOR

1.8 sec   3.0 sec

INP
Interaction to Next Paint

GOOD	NEEDS IMPROVEMENT	POOR

200 ms   500 ms

TTFB
Time To First Byte

GOOD	NEEDS IMPROVEMENT	POOR

800 ms   1800 ms

WHAT AFFECTS
PROCESSING TIME ?

VISITOR TRAFFIC

TTFB

Time To First Byte

GOOD	NEEDS IMPROVEMENT	POOR

800 ms   1800 ms

Users	Request/s	TTFB	Server load
1	30	90 ms	20 %
10	150	110 ms	90 %
100	130	800 ms	120 %
200	80	3000 ms	200 %

VISITOR TRAFFIC

TTFB
Time To First Byte

GOOD	NEEDS IMPROVEMENT	POOR

800 ms   1800 ms

Users	Request/s	TTFB	Server load
1	30	90 ms	20 %
10	150	110 ms	90 %
100	130	800 ms	120 %
200	80	3000 ms	200 %

JUST NEED TO ADD SERVERS

LET'S MULTIPLY SERVERS

Servers	Users	Request/s	TTFB	Server load
1	10	150	110 ms	90 %
4	10	400	90 ms	40 %
1	100	130	800 ms	120 %
4	100	550	440 ms	120 %
1	300	80	3000 ms	200 %
4	300	550	910 ms	200 %

LET'S MULTIPLY SERVERS

Servers	Users	Request/s	TTFB	Server load
1	10	150	110 ms	90 %
4	10	400	90 ms	40 %
1	100	130	800 ms	120 %
4	100	550	440 ms	120 %
1	300	80	3000 ms	200 %
4	300	550	910 ms	200 %

IT'S PROBABLY NOT THE BEST IDEA!

AND IT COSTS MUCH MORE !

HOW CAN WE REDUCE
PROCESSING TIME
?
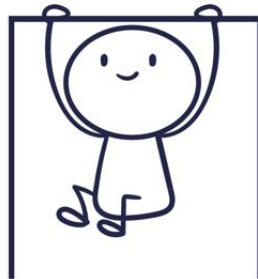
REMINDER : SIMPLE REQUEST LIFECYCLE

WEBSERVER   PHP   DATABASE
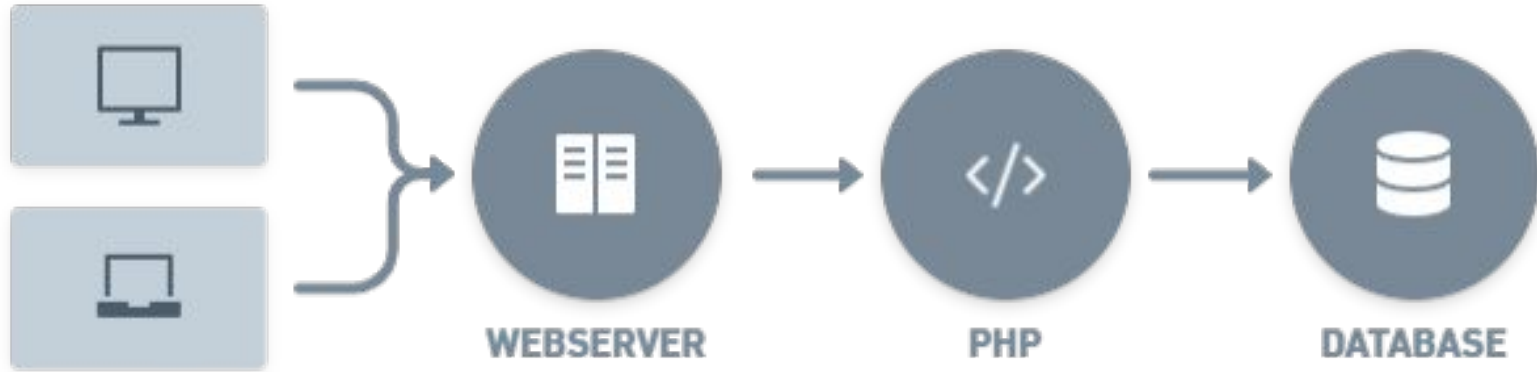
BY OPTIMIZING DATABASE REQUEST
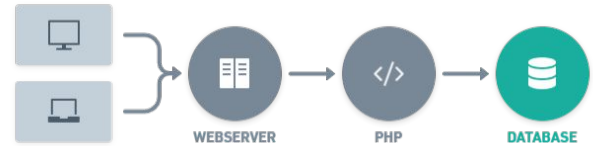
```php
class ProductRepository extends BaseProductRepository implements ProductRepositoryInterface
{
    //...
    public function findLatestByChannel(ChannelInterface $channel, string $locale, int $count): array
    {
        return $this->createQueryBuilder('o')
            ->addSelect('translation')
            ->innerJoin('o.translations', 'translation', 'WITH', 'translation.locale = :locale')
            ->andWhere(':channel MEMBER OF o.channels')
            ->andWhere('o.enabled = :enabled')
            ->addOrderBy('o.createdAt', 'DESC')
            ->setParameter('channel', $channel)
            ->setParameter('locale', $locale)
            ->setParameter('enabled', true)
            ->setMaxResults($count)
            ->getQuery()
            ->enableResultCache(300) // <- cache for 5 min
            ->getResult()
        ;
    }
}
```

BY OPTIMIZING THE SYMFONY CACHE


```php
public function latestProduct(): Response
{
    $cacheItem = $this->cache->getItem('latestProducts');

    if ($cacheItem->isHit()) {
        return $this->render('...', ['latestProducts' => $cacheItem->get()]);
    }

    $latestProducts = $this->productRepository->findLatestByChannel($channel, $locale, self::LASTEST_PRODUCTS_SHOW);
    $cacheItem->set($latestProducts);
    $cacheItem->expiresAfter(30);
    $this->cache->save($cacheItem);

    return $this->render('...', ['latestProducts' => $latestProducts]);
}
```

COULD WE PUT A CACHE
IN FRONT OF THE SERVER ?
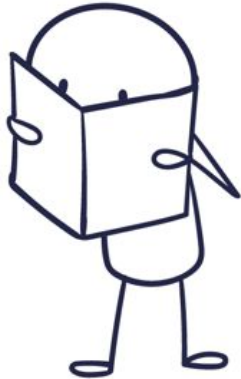
CACHING IN FRONT OF THE SERVER

WHAT ARE THE BENEFITS
OF VARNISH ?

VARNISH'S BENEFITS

Varnish respond very quickly (30ms)

Most requests are handled by Varnish

Traffic peaks are painless (commercial operations or DDOS attacks)

Supports "grace mode"

VARNISH
CACHE

WHAT ARE THE CONSTRAINTS OF VARNISH?

VARNISH'S CONSTRAINTS
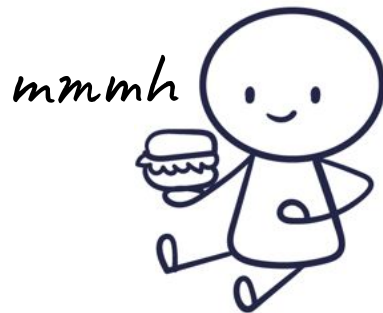
Special skills VCL language knowledges

No SSL support

Complex to measure and benchmark results

Invalidating and flushing cache management

VARNISH
CACHE

WHAT ABOUT
COOKIES
?

mmmh

MUST REMOVE THEM

```
sub vcl_recv {
    // Remove all cookies except the session ID.
    if (req.http.Cookie) {
        set req.http.Cookie = ";" + req.http.Cookie;
        set req.http.Cookie = regsuball(req.http.Cookie, "; +", ";");
        set req.http.Cookie = regsuball(req.http.Cookie, ";(PHPSESSID)=", "; \1=");
        set req.http.Cookie = regsuball(req.http.Cookie, ";[^ ][^;]*", "");
        set req.http.Cookie = regsuball(req.http.Cookie, "^[; ]+|[; ]+$", "");

        if (req.http.Cookie == "") {
            // If there are no more cookies, remove the header to get the page cached.
            unset req.http.Cookie;
        }
    }
}
```

EXCEPT PHPSESSID COOKIE

IN PRACTICE
ON SYLIUS
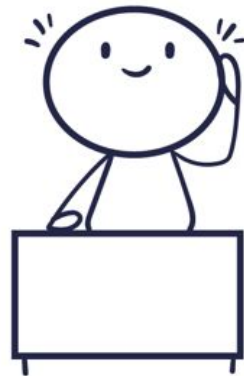
SIMPLE WAY TO DO IT

Varnish cache key (hash) based on URL

Ready for multilingual of Sylius

Applicable to :

- Home
- Products and Products Lists
- Content
- Sitemap, robots.txt, …

Photo de Steve Lieman sur Unsplash

```php
final class VarnishSubscriber implements EventSubscriberInterface
{
    private const CACHED_ROUTES = [
        'sylius_shop_homepage',
        '...',
    ];

    public static function getSubscribedEvents(): array
    {
        return [
            'kernel.response' => [
                ['onKernelResponse', -9999],
            ],
        ];
    }

    public function onKernelResponse(ResponseEvent $event): void
    {
        // ...
    }
}
```

```php
final class VarnishSubscriber implements EventSubscriberInterface
{
    // ...

    public function onKernelResponse(ResponseEvent $event): void
    {
        if (!$event->isMainRequest()) {
            return;
        }

        $request = $event->getRequest();
        $route = $request->attributes->get('_route');
        $hasCookies = $event->getResponse()->headers->has('set-cookie');

        if ($hasCookies || !\in_array($route, self::CACHED_ROUTES, true)) {
            $event->getResponse()->setPrivate();
            $event->getResponse()->setMaxAge(0);
            $event->getResponse()->setSharedMaxAge(0);
            $event->getResponse()->headers->addCacheControlDirective('must-revalidate');
            $event->getResponse()->headers->addCacheControlDirective('no-store');

            return;
        }

        $event->getResponse()->setPublic();
        $event->getResponse()->setMaxAge(30);
        $event->getResponse()->setSharedMaxAge(30);
        $event->getResponse()->headers->removeCacheControlDirective('must-revalidate');
    }
}
```

IMPACTS FOR THE USER

EDGE SIDE INCLUDES (ESI)

Symfony configuration

```php
// config/packages/framework.php
use Symfony\Config\FrameworkConfig;

return static function (FrameworkConfig $framework): void {
    $framework->esi()
        ->enabled(true)
    ;
};
```

```vcl
sub vcl_recv {
    // Add a Surrogate-Capability header to announce ESI support.
    set req.http.Surrogate-Capability = "abc=ESI/1.0";
}

sub vcl_backend_response {
    // Check for ESI acknowledgement and remove Surrogate-Control header
    if (beresp.http.Surrogate-Control ~ "ESI/1.0") {
        unset beresp.http.Surrogate-Control;
        set beresp.do_esi = true;
    }
}
```

Varnish configuration

EDGE SIDE INCLUDES (ESI)

Twig changes

```
{# Layout/Header/_cart.html.twig  #}

<div class="right aligned column">
    {{ render(url('sylius_shop_partial_cart_summary', {'template': '@SyliusShop/Cart/_widget.html.twig'})) }}
</div>

{# Replace by #}

<div class="right aligned column">
    {{ render_esi(url('sylius_shop_partial_cart_summary', {'template': '@SyliusShop/Cart/_widget.html.twig'})) }}
</div>
```

SYLIUS' TAXONS & PRODUCTS

VARNISH USER CONTEXT HASH

User	Session ID	Role	User contexte hash
Anna		anonymous	1
Bob		authenticated	1
Peeter	**a**	vip user	2
John	**b**	vip user	2
Sofia	**c**	partner A	3
Jack	**d**	partner B	4

TIPS & TRICKS

Use AJAX instead of ESI

Beware of plugins : they can add "unwanted" cookies ...

Create a Cookie on the developer toolbar to bypass
varnish cache

Cache warm-up

VISITOR TRAFFIC

Users	Request/s	TTFB	Server load
1	30	70 ms	20 %
10	150	70 ms	30 %
100	130	70 ms	40 %
200	80	70 ms	60 %

WELL DONE !

KEY FACTS

- A high-performance website means a **successful user experience**

- A successful user experience means **satisfied, returning users and a positive impact on sales**

- There are **many tools for measuring web performance**

- Choose the **right tool for the project needs**

- Web performance can be optimised by using **caches**, but there are also other solutions, such as **CDNs**.

- **Start with improvements that have a strong impact and are quick to implement,** before tackling minor optimisations that can be more time-consuming and costly.

THANK YOU

HAVE ANY
QUESTIONS ?

LET'S STAY IN TOUCH
and have a coffee ... or a beer ?

Synolia

Olivier ALLAIN

www.linkedin.com/in/oallain/

oallain33

olivier.allain@synolia.com

Hélèna GRAVELIER

www.linkedin.com/in/helenagravelier/

LenaLscn

helena.gravelier@synolia.com