

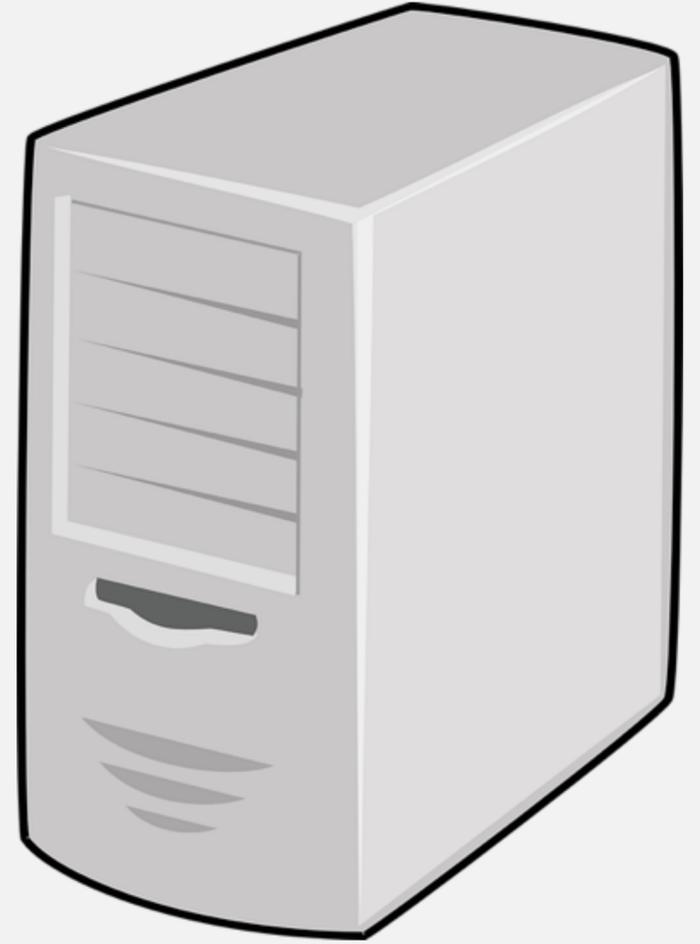
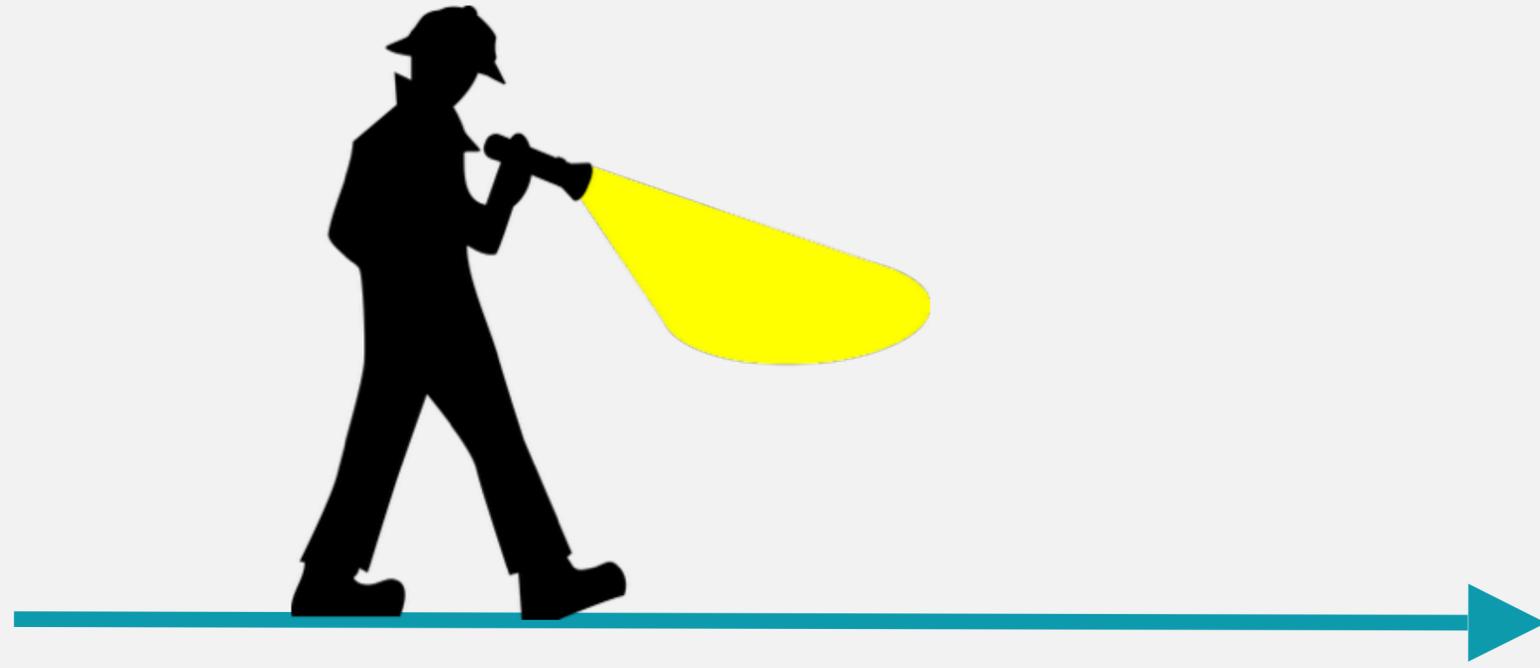
HTTPS: Securing the Foundations

PROGRESSIVE WEB APP

ROADSHOW 2016 



With great power
comes great responsibility.



Why HTTPS?



<https://www.google.com>

Why HTTPS?



<https://www.google.com>

Identity

Who are you
talking to?

Why HTTPS?



<https://www.google.com>

Identity

Who are you
talking to?

Confidentiality

Who can read your
data?

Why HTTPS?



<https://www.google.com>

Identity

Who are you
talking to?

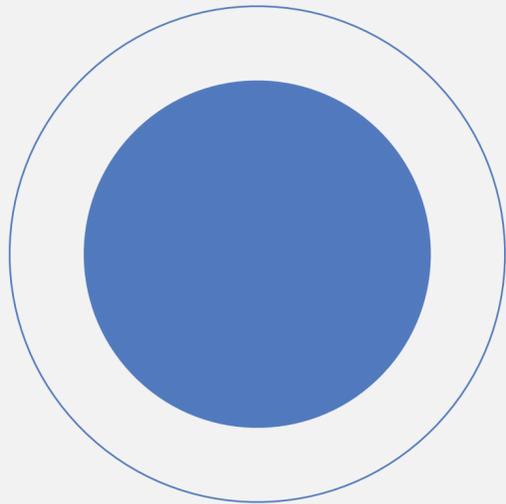
Confidentiality

Who can read your
data?

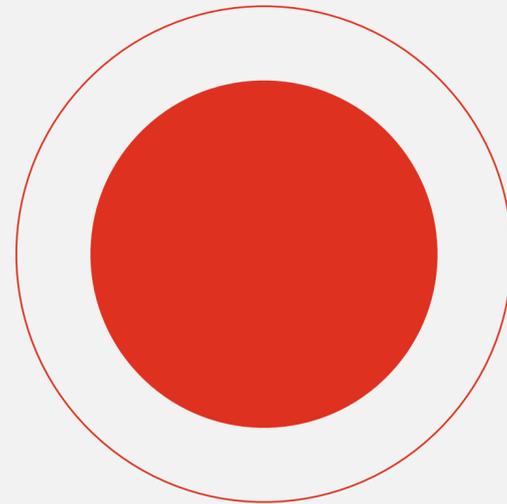
Integrity

Who can modify
your data?

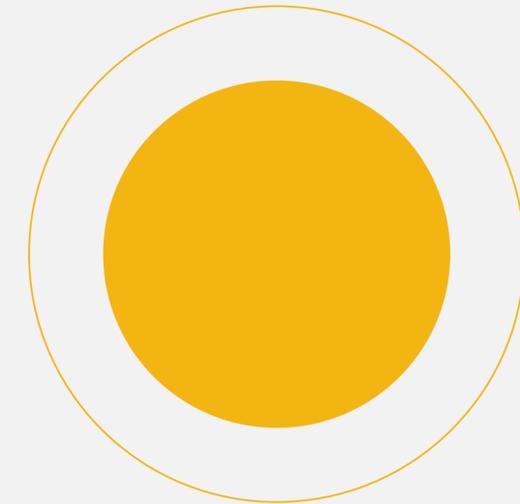
But what about...



Performance



Cost



Maintenance

But HTTPS is too hard...

1. My site doesn't need it!
2. It'll cause performance problems and I'll lose money
3. It costs too much to implement
4. Critical 3rd party dependancies may not support HTTPS yet

But HTTPS is too hard...

- 1. My site doesn't need it!**
2. It'll cause performance problems and I'll lose money
3. It costs too much to implement
4. Critical 3rd party dependancies may not support HTTPS yet

APIs that require HTTPS

- Service Workers
- getUserMedia
- Push Notifications
- App Cache
- Encrypted Media Extensions
- Geo Location
- HTTPS/2

For more information, see:

<https://www.chromium.org/Home/chromium-security/prefer-secure-origins-for-powerful-new-features>

APIs that require HTTPS

- **Service Workers**
- getUserMedia
- Push Notifications
- App Cache
- Encrypted Media Extensions
- Geo Location
- HTTPS/2

For more information, see:

<https://www.chromium.org/Home/chromium-security/prefer-secure-origins-for-powerful-new-features>

HTTPS is a crucial part
of the user experience

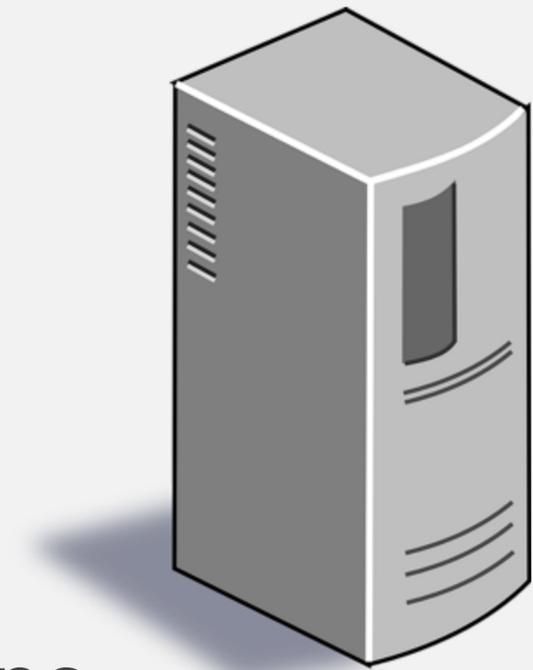
It's no longer for especially
important or security-sensitive
sites only.

But HTTPS is too hard...

1. My site doesn't need it!
- 2. It'll cause performance problems and I'll lose money**
3. It costs too much to implement
4. Critical 3rd party dependancies may not support HTTPS yet

HTTP Strict Transport Security

```
Strict-Transport-Security: max-age=2592000;  
includeSubDomains
```



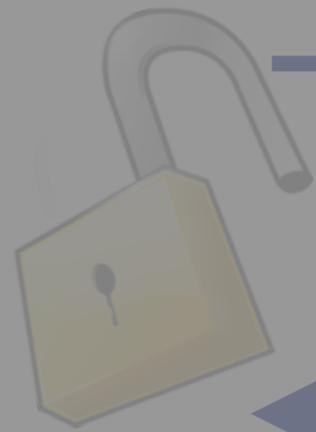
“Only access my site and all its subdomains
over HTTPS for the next month.”

GET / HTTP/1.1



HTTP/1.1 301 Moved Permanently

Location: ~~https://bob site.com~~



ClientHello



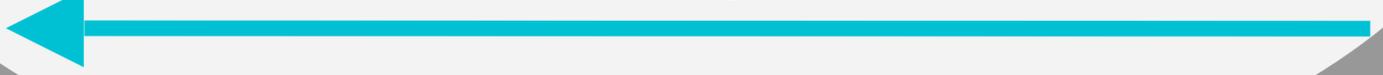
ServerHello
Certificate



Finished



Finished



GET / HTTP/1.1



...

GET / HTTP/1.1



HTTP/1.1 301 Moved Permanently

Location: https://bob-site.com



ClientHello



ServerHello

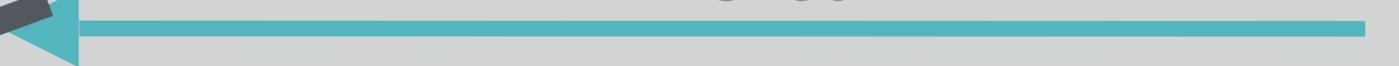
Certificate



Finished



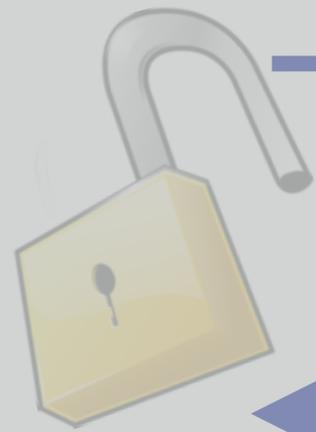
Finished



GET / HTTP/1.1



TLS False Start



GET / HTTP/1.1



HTTP/1.1 301 Moved Permanently

Location: https://bob.site.com



ClientHello (with session id)



ServerHello
Finished



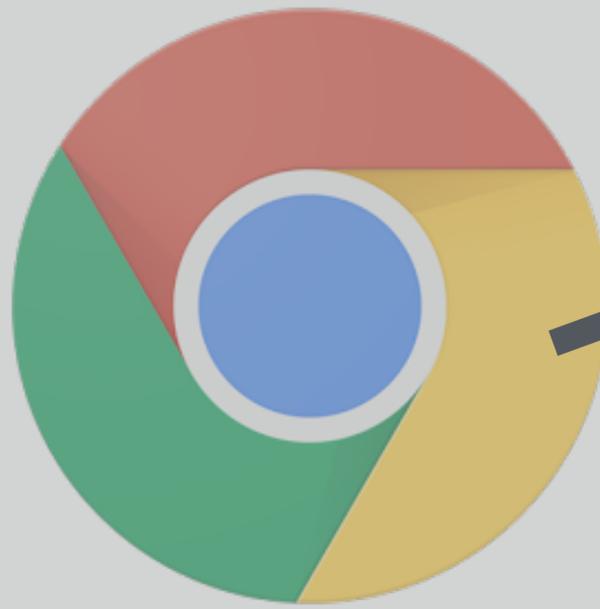
Finished



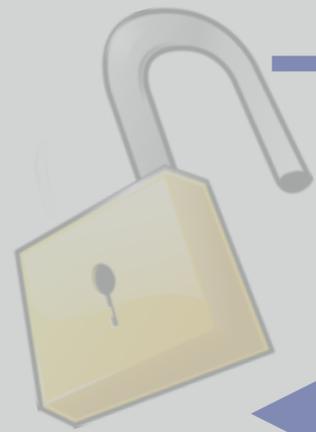
GET / HTTP/1.1



...



TLS Session Resumption



HTTPS unlocks HTTP/2

HTTP/2 unlocks massive
performance wins.



When we launched [HTTPS], we saw an average of a **50ms hit for negotiation**... it was more than offset when we activated HTTP/2 a month later and saw an overall drop of ~250ms per pageview on supported devices.

- [weather.com](https://www.weather.com)

But HTTPS is too hard...

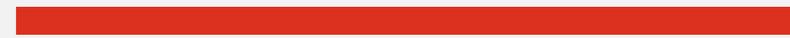
1. My site doesn't need it!
2. It'll cause performance problems and I'll lose money
- 3. It costs too much to implement**
4. Critical 3rd party dependancies may not support HTTPS yet

The Cost of HTTPS

Certificate



Search Ranking



SSLMate Pricing

https://sslmate.com/pricing

sslmate Features Pricing Blog Docs

Pricing

<h3>Standard SSL</h3> <p>\$15⁹⁵ / year</p> <p>Secures one hostname [?]</p> <p>Domain Validation</p> <p>One Minute Issuance</p> <p>Automated Purchase and Renewal</p>	<h3>Multi-Host SSL</h3> <p>\$24⁹⁵ / hostname / year</p> <p>Multiple hostnames [?] (For \$24.95 / hostname)</p> <p>Domain Validation</p> <p>One Minute Issuance</p> <p>Automated Purchase and Renewal</p>
--	--



Let's Encrypt

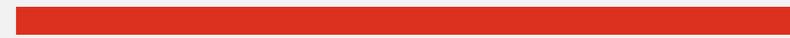
Let's Encrypt is a trademark of the Internet Security Research Group.

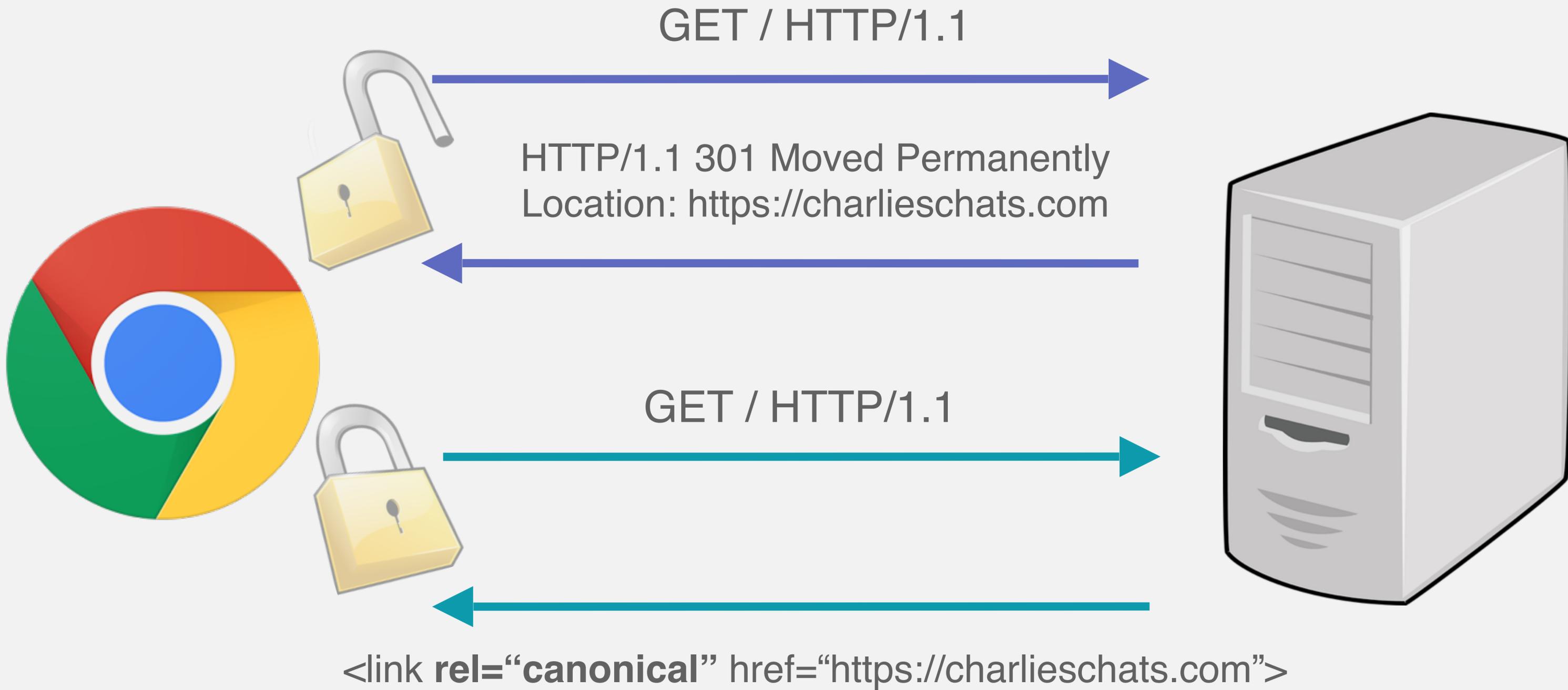
The Cost of HTTPS

Certificate



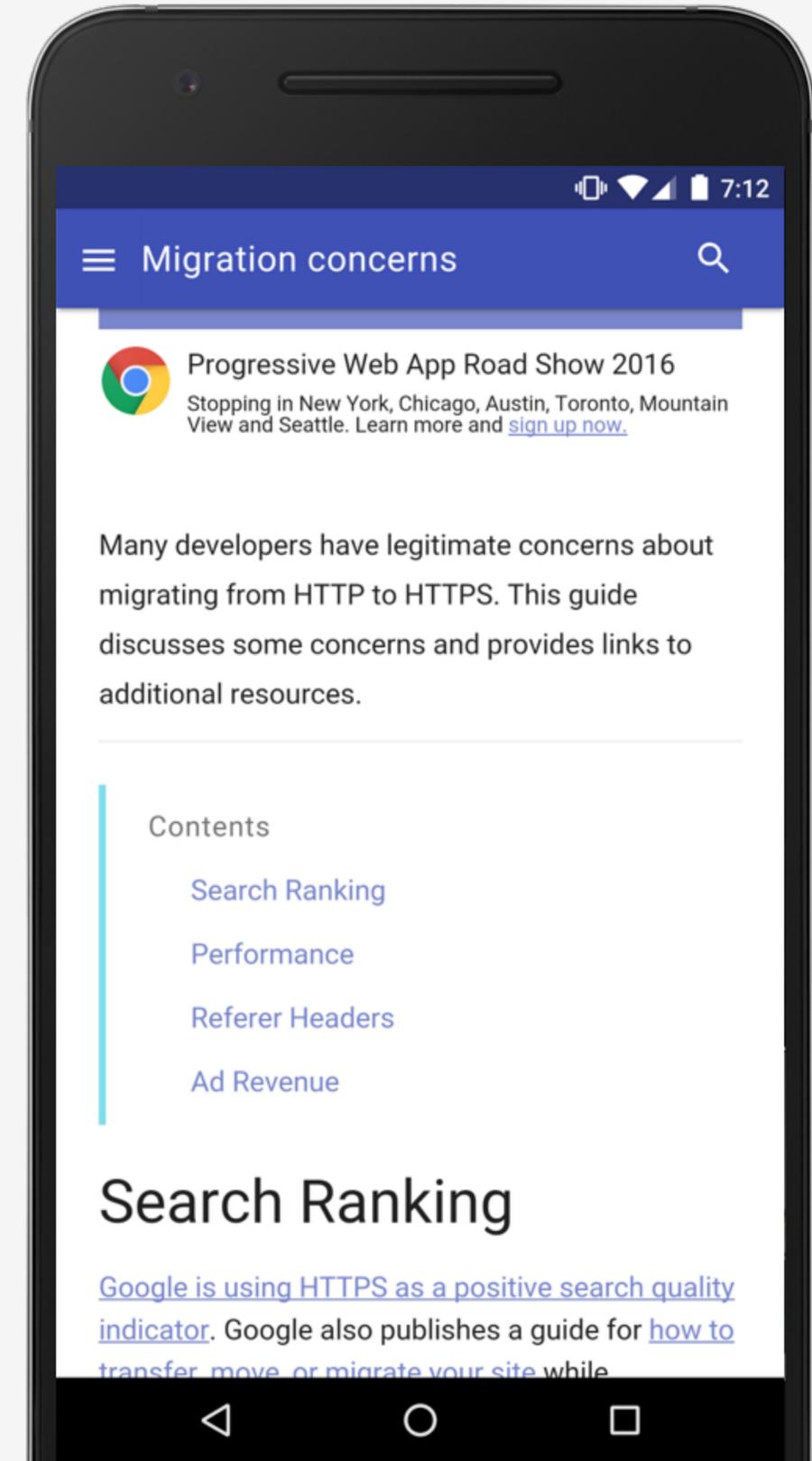
Search Ranking





Search Ranking Guidance

[developers.google.com/web/fundamentals/
security/encrypt-in-transit/migration-concerns](https://developers.google.com/web/fundamentals/security/encrypt-in-transit/migration-concerns)



But HTTPS is too hard...

1. My site doesn't need it!
2. It'll cause performance problems and I'll lose money
3. It costs too much to implement
- 4. Critical 3rd party dependancies may not support HTTPS yet**



[A] survey of our membership late last year showed **nearly 80% of member ad delivery systems supported HTTPS.**

- IAB (Interactive Advertising Bureau),
- “Adopting Encryption: The Need for HTTPS”

Chrome DevTools Security Panel

The screenshot displays the Chrome DevTools Security Panel for the URL `https://very.badssl.com/`. The panel is divided into a left sidebar and a main content area. The sidebar shows the 'Overview' tab with a list of origins: 'Main Origin' `https://very.badssl.com` (secure) and 'Non-Secure Origins' `http://http.badssl.com` (insecure) and `http://very.badssl.com` (insecure). The main content area, titled 'Security Overview', shows a red padlock icon with an 'x' indicating an insecure page. Below this, three security issues are listed:

- SHA-1 Certificate**: The certificate for this site expires in 2017 or later, and the certificate chain contains a certificate signed using SHA-1. A 'View certificate' button is provided.
- Mixed Content**: The site includes HTTP resources. A link to 'View 1 request in Network Panel' is provided.
- Blocked mixed content**: Your page requested insecure resources that were blocked. A link to 'View 1 request in Network Panel' is provided.

The bottom of the panel shows the Console with two error messages:

- Mixed Content**: The page at `'https://very.badssl.com/'` was loaded over HTTPS, but requested an insecure image `'http://very.badssl.com/image.jpg'`. This content should also be served over HTTPS. Location: `very.badssl.com/:28`
- Mixed Content**: The page at `'https://very.badssl.com/'` was loaded over HTTPS, but requested an insecure script `'http://http.badssl.com/test/imported.js'`. This request has been blocked; the content must be served over HTTPS. Location: `very.badssl.com/:1`

What's Next?

More resources to help

Encrypting Data in Transit

developers.google.com/web/fundamentals/security/encrypt-in-transit/

Using Content Security Policy

developers.google.com/web/fundamentals/security/csp/

Preventing Mixed Content

developers.google.com/web/fundamentals/security/prevent-mixed-content/

Thank You!

PROGRESSIVE WEB APP

ROADSHOW 2016 