FRONT-END SERVERS

simply more front-end

MORE FRONT-END

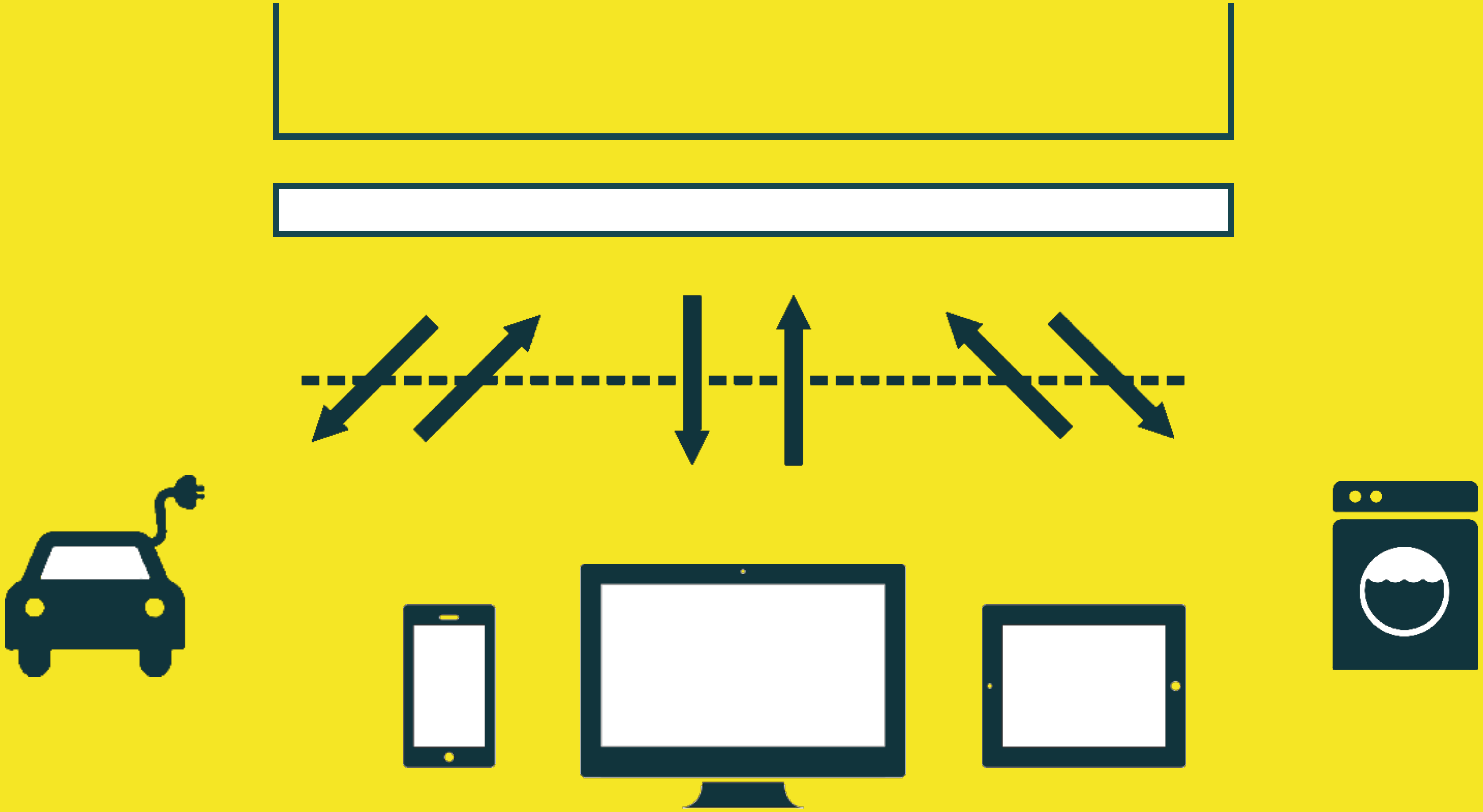
MORE FRONT-END

IN THE BROWSER

ON THE SERVER

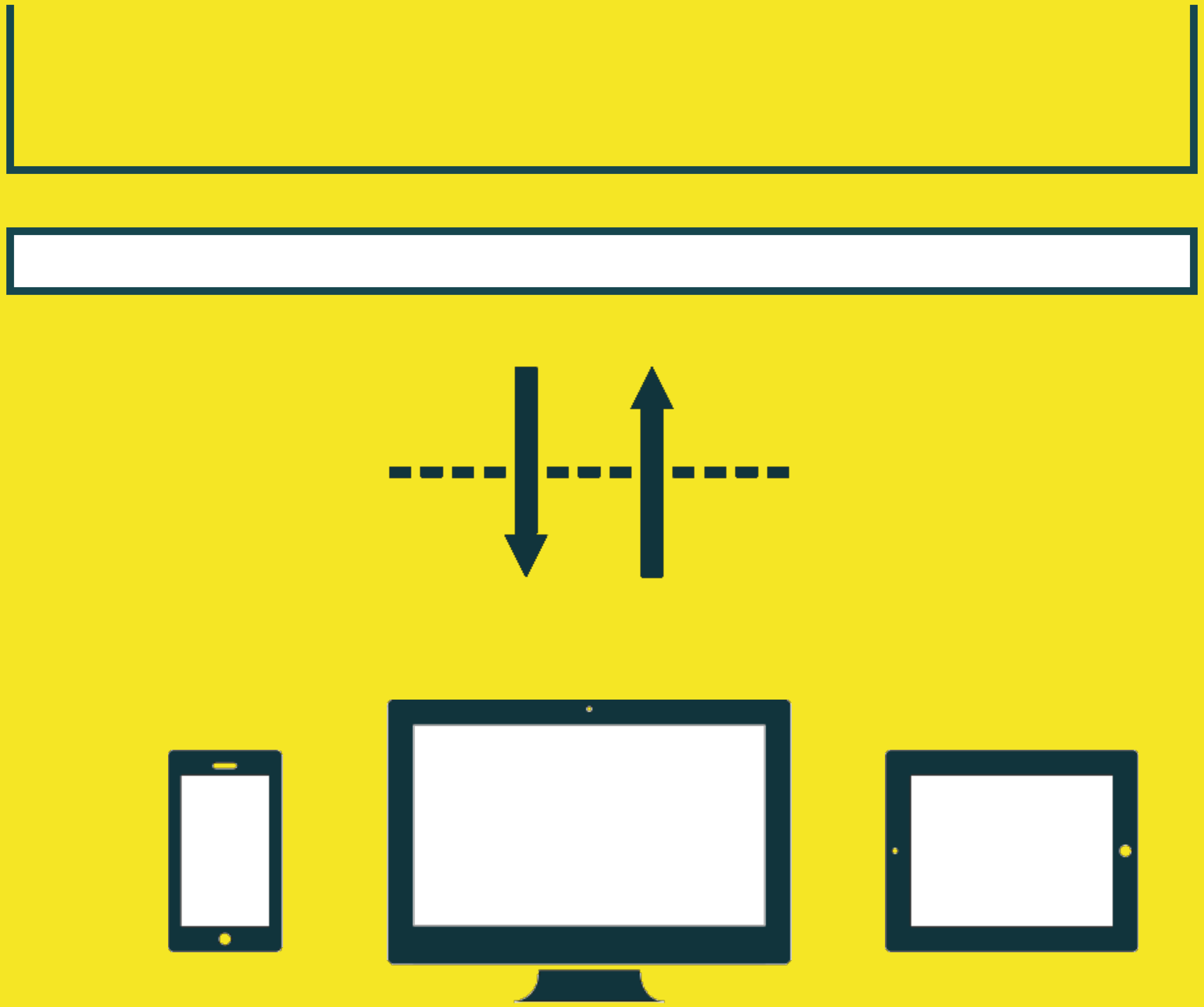
IN BETWEEN

AND OTHER DEVICES?

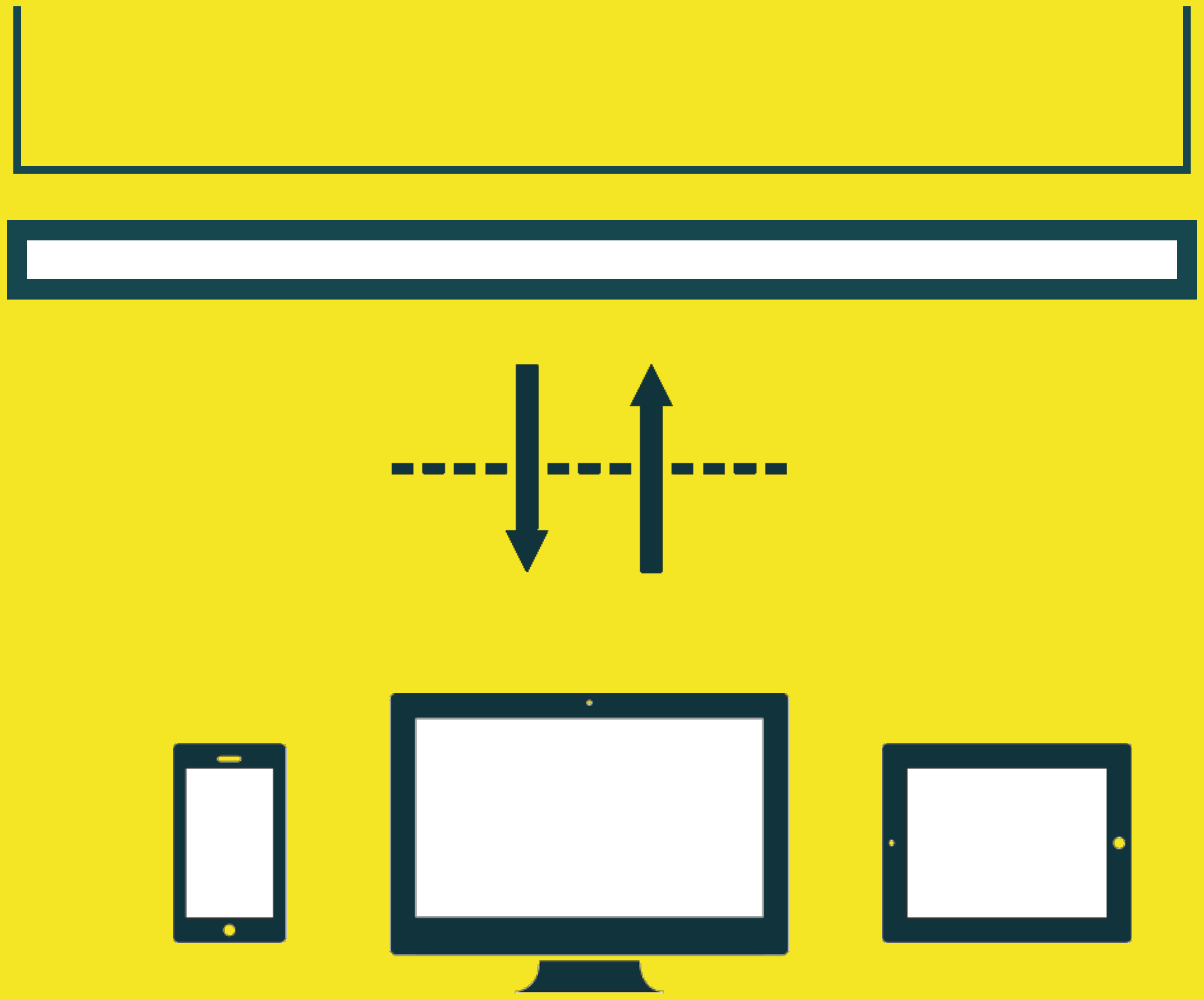


FRONT-END SERVERS

ON THE SERVER



ON THE **FRONT-END** SERVER



FRONT-END SERVER

is dedicated to the
best experience in the browser
handles **routing**, data fetching,
template **rendering**, static assets,
optimises responses and caches

FRONT-END SERVER

is **NOT** aiming to

manage **business logic**

handle **databases**, search indexing

user **auth**entication & authorisation

check-out & order processing

etcetera

HOW?

MICRO FRAMEWORKS

NodeJS Express + Nunjucks

Python Flask + Jinja2

PHP Silex/Symfony + Twig

Golang Revel + Go HTML/Template ?

Java Spark + Jinjava ?

Ruby Sinatra + Liquid ?

C# .NET Nancy + Razor ?

HOW TO CHOOSE?

technical requirements

experience of development team

available resources

who will maintain the app?

who will manage server operations?

whatever you feel is relevant

ROUTE - FETCH - RENDER - RESPOND

```
router.get('/blog/', function (request, response) {
    fetchPosts(request.query)
        .then(function (posts) {
            var html = nunjucks.render('blog.html', posts);
            response.end(html));
        })
        .catch(/* handle errors */);
});
```

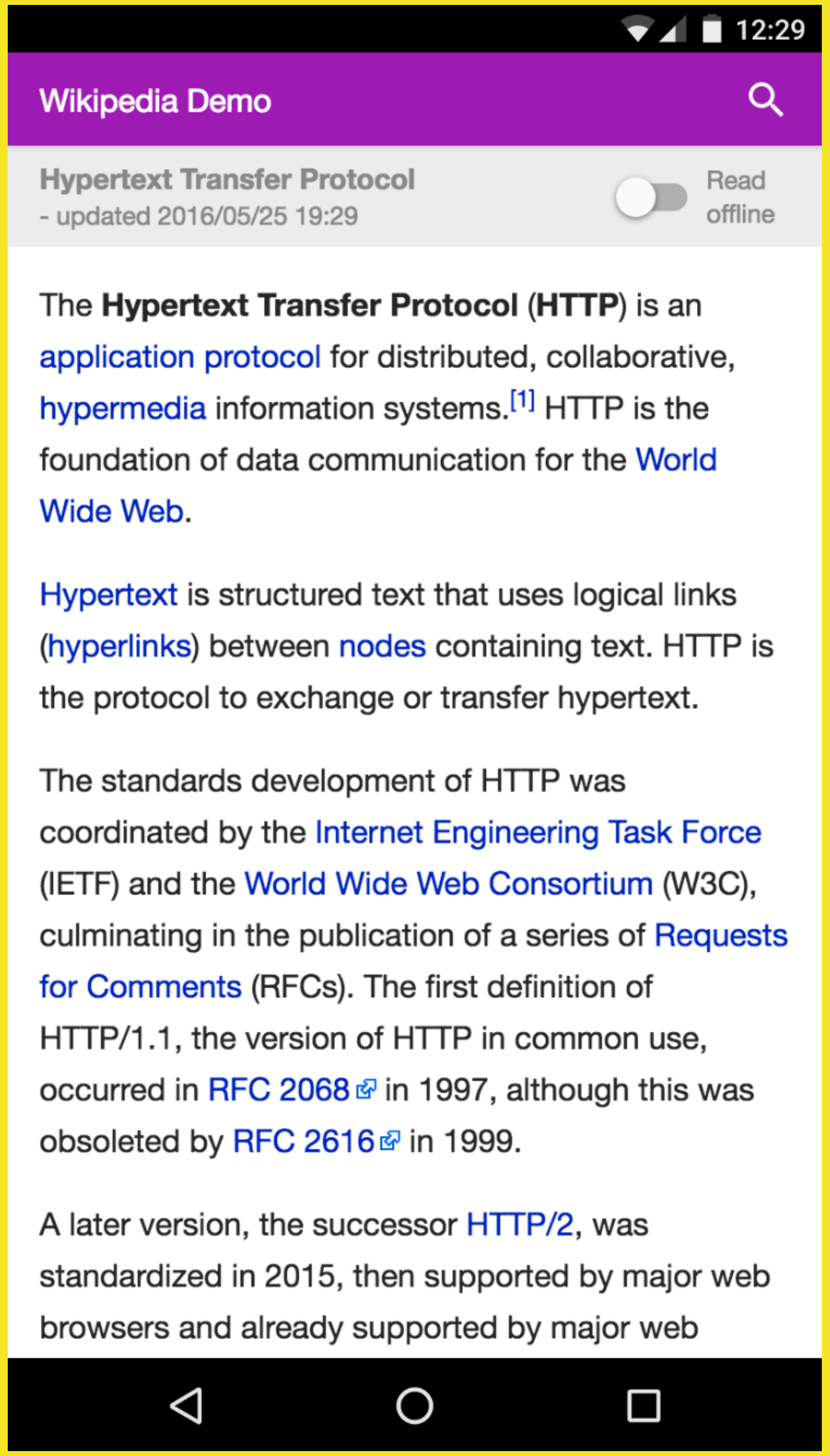
ADVANCE & ENHANCE

FRONT-END ENHANCEMENTS

pre-rendering
optimising JSON responses
aggressive caching
file-level cache revision
lazy loading assets
cookies & critical CSS / JS
speedy serving with HTTP/2
combine with Service Worker
and more

EXAMPLE

EXAMPLE - **OFFLINE WIKI**



Hypertext Transfer Protocol
- updated 2016/05/25 19:29

Read offline

The **Hypertext Transfer Protocol** (**HTTP**) is an application protocol for distributed, collaborative, hypermedia information systems.[1] HTTP is the foundation of data communication for the World Wide Web.

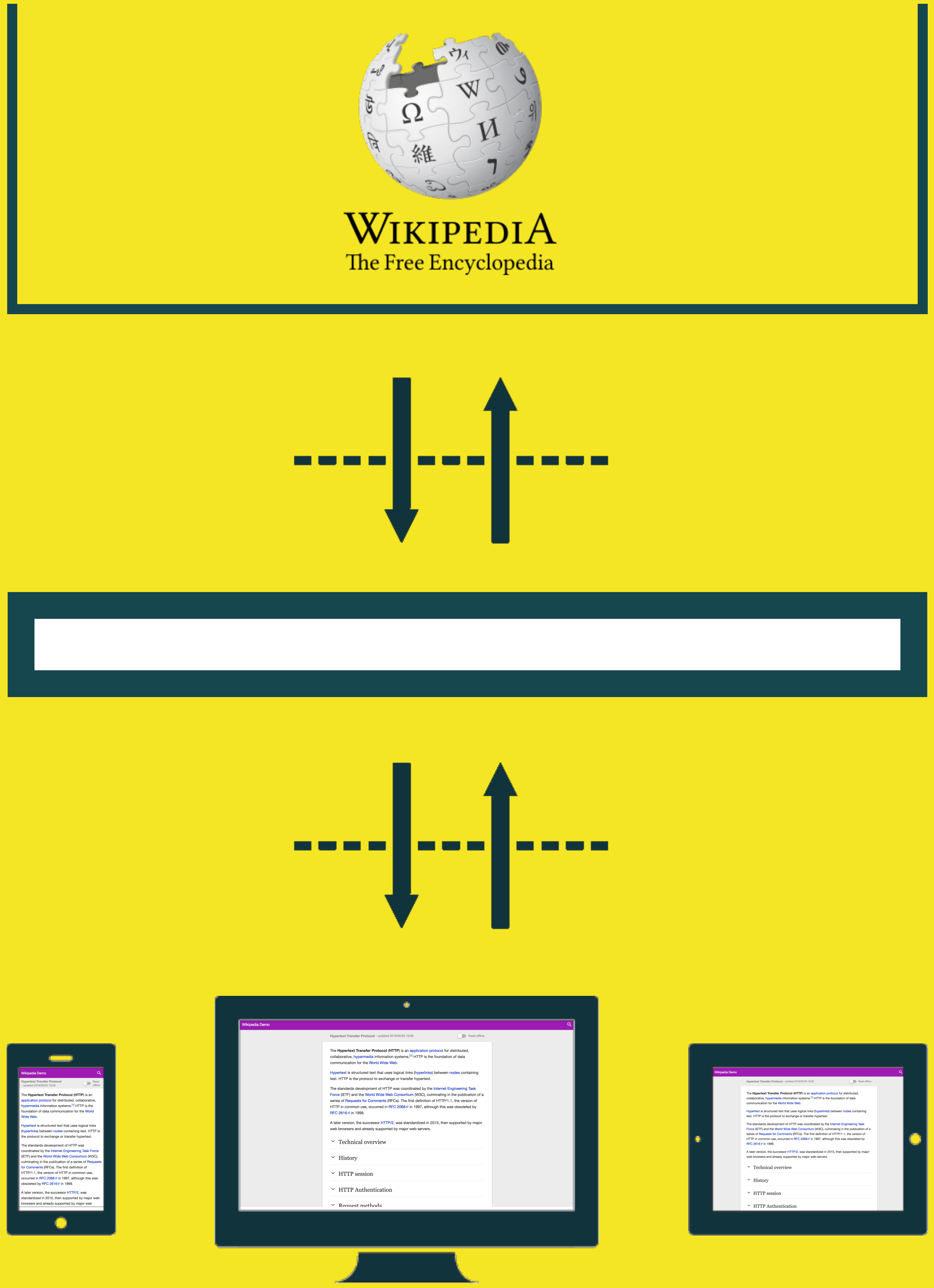
Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text. HTTP is the protocol to exchange or transfer hypertext.

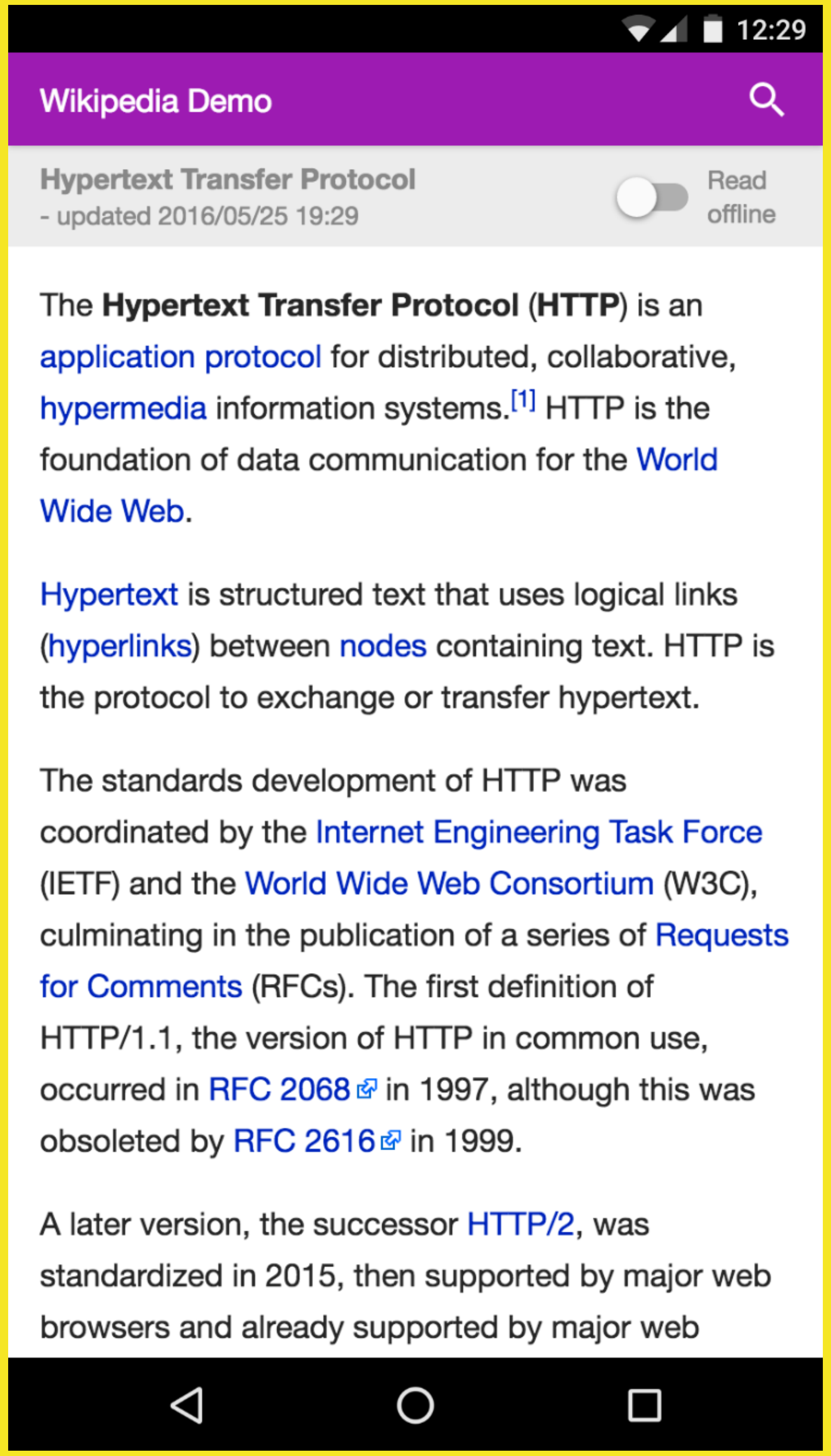
The standards development of HTTP was coordinated by the Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C), culminating in the publication of a series of Requests for Comments (RFCs). The first definition of HTTP/1.1, the version of HTTP in common use, occurred in RFC 2068 in 1997, although this was obsoleted by RFC 2616 in 1999.

A later version, the successor HTTP/2, was standardized in 2015, then supported by major web browsers and already supported by major web

jakearchibald/offline-wikipedia

EXAMPLE - **OFFLINE WIKI**





Hypertext Transfer Protocol
- updated 2016/05/25 19:29

Read offline

The **Hypertext Transfer Protocol** (**HTTP**) is an application protocol for distributed, collaborative, hypermedia information systems.[1] HTTP is the foundation of data communication for the World Wide Web.

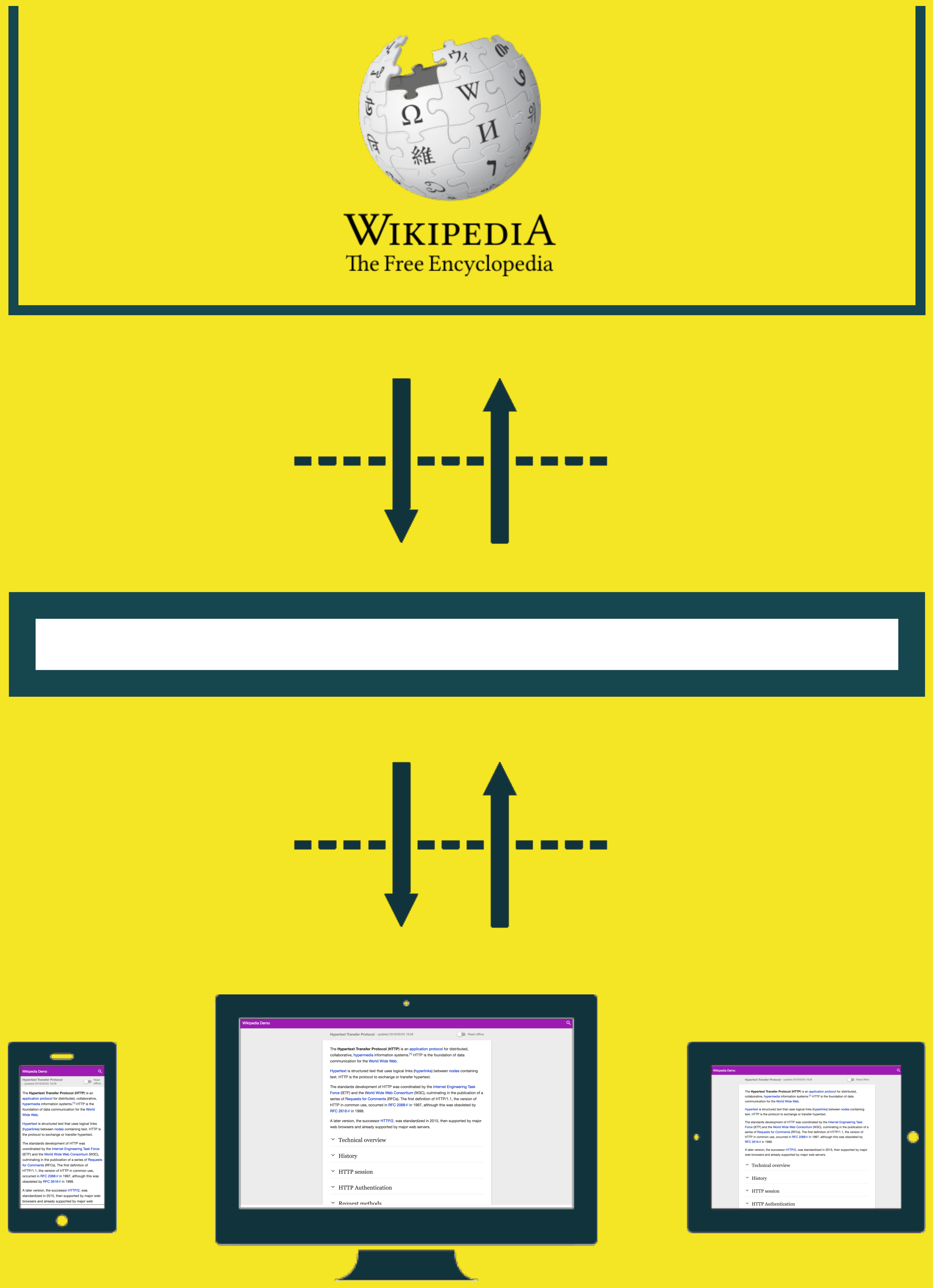
Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text. HTTP is the protocol to exchange or transfer hypertext.

The standards development of HTTP was coordinated by the Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C), culminating in the publication of a series of Requests for Comments (RFCs). The first definition of HTTP/1.1, the version of HTTP in common use, occurred in RFC 2068 in 1997, although this was obsoleted by RFC 2616 in 1999.

A later version, the successor HTTP/2, was standardized in 2015, then supported by major web browsers and already supported by major web

jakearchibald/offline-wikipedia

EXAMPLE - **OFFLINE WIKI**



Wikipedia API:

- *(open) search as JSON*
- *article meta data as JSON*
- *article content as HTML*

Front-end Server:

- *request data from API*
- *transform data*
- *use cookie & header data*
- *render templates*
- *optimise browser responses*

Front-end in browser:

- *optimised user experience*

R.I.P. FRONT-END GUIDE ?

FRONT-END GUIDE

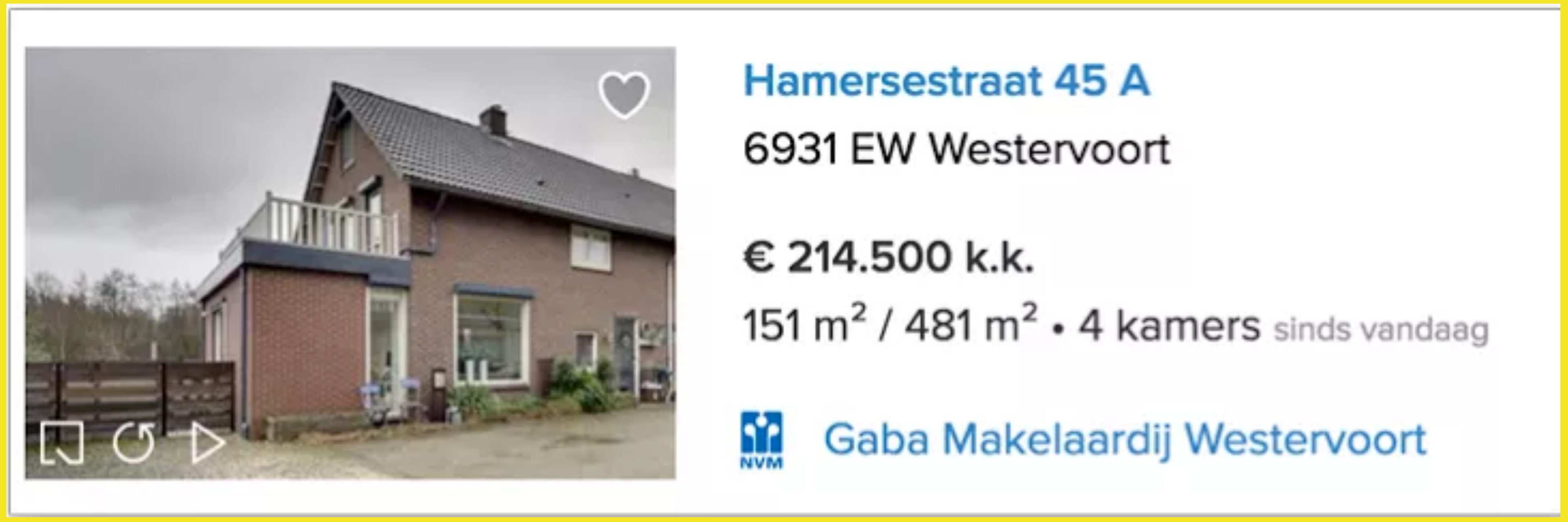
Structure (front-end) code
as small and **reusable modules**
with **demos** & documentation
bundled with **testing** suite
and code **quality** tools

REINCARNATION

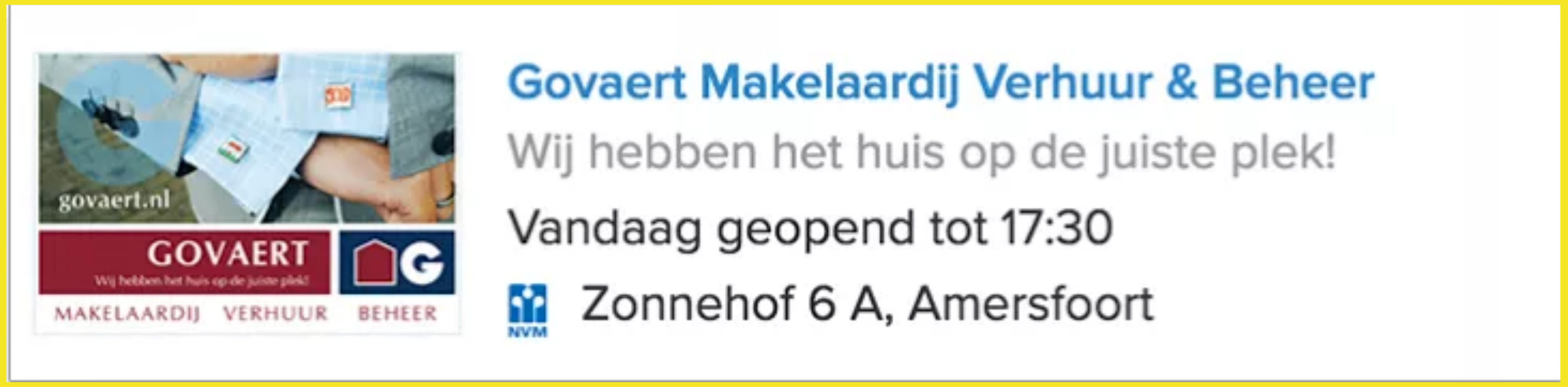
keep front-end guide **concepts**
split it into smaller **parts**
apply to front-end servers

MODULE STRUCTURE

```
modules/
    search-result/
        search-result.html
        search-result.demo.html
        search-result.css
        search-result.js
        search-result.test.js
        README.md
```



Hamersestraat 45 A
6931 EW Westervoort
€ 214.500 k.k.
151 m² / 481 m² • 4 kamers sinds vandaag
Gaba Makelaardij Westervoort



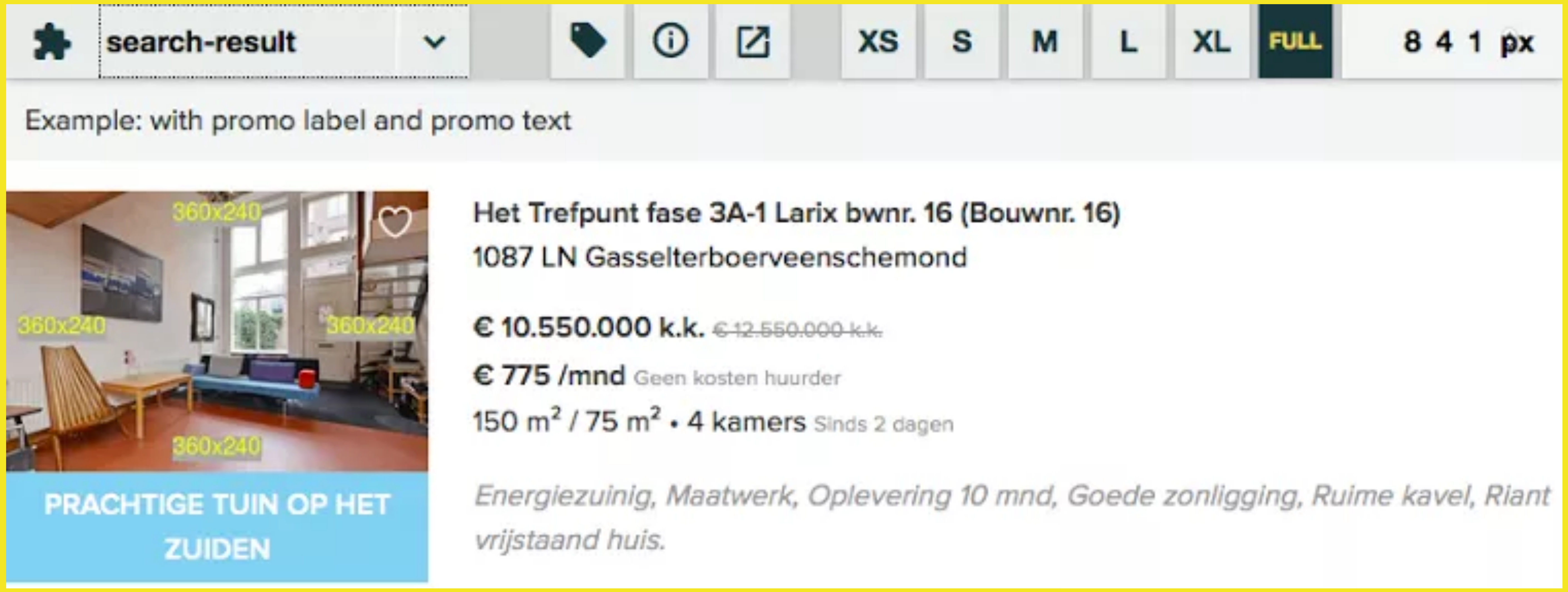
Govaert Makelaardij Verhuur & Beheer
Wij hebben het huis op de juiste plek!
Vandaag geopend tot 17:30
Zonnehof 6 A, Amersfoort

TEMPLATING LOGIC

```
{% block content %}
    <article><h1>{{ post.title }}</h1>
        {{ post.body | safe }}
    </article>
    <section><h2>Comments</h2>
        {% for comment in comments %}
             {{ cardMacro(title = comment.name, body= comment.text) }}
        {% endfor %}
    </section>
{% endblock %}
```

DEMO VIEWER

select a module *info & more* *test responsive behaviour*



Example: with promo label and promo text

Het Trefpunt fase 3A-1 Larix bwnr. 16 (Bouwnr. 16)
1087 LN Gasselterboerveenschemond

€ 10.550.000 k.k. ~~€ 12.550.000 k.k.~~
€ 775 /mnd Geen kosten huurder
150 m² / 75 m² • 4 kamers Sinds 2 dagen

Energiezuinig, Maatwerk, Oplevering 10 mnd, Goede zonligging, Ruime kavel, Riant vrijstaand huis.

preview in variations (using dummy data)

TASK RECIPES

CSS & JS pipelines
sourcemaps
image optimisers
icon generators
file watchers
critical css generators
test runners
etcetera

PROOF OF CONCEPT

DEMO-SERVER-EXPRESS

DE VOORHOEDE

front-end developers