



JONAS SCHMEDTMANN

# THE ULTIMATE REACT COURSE

 @JONASSCHMEDTMAN

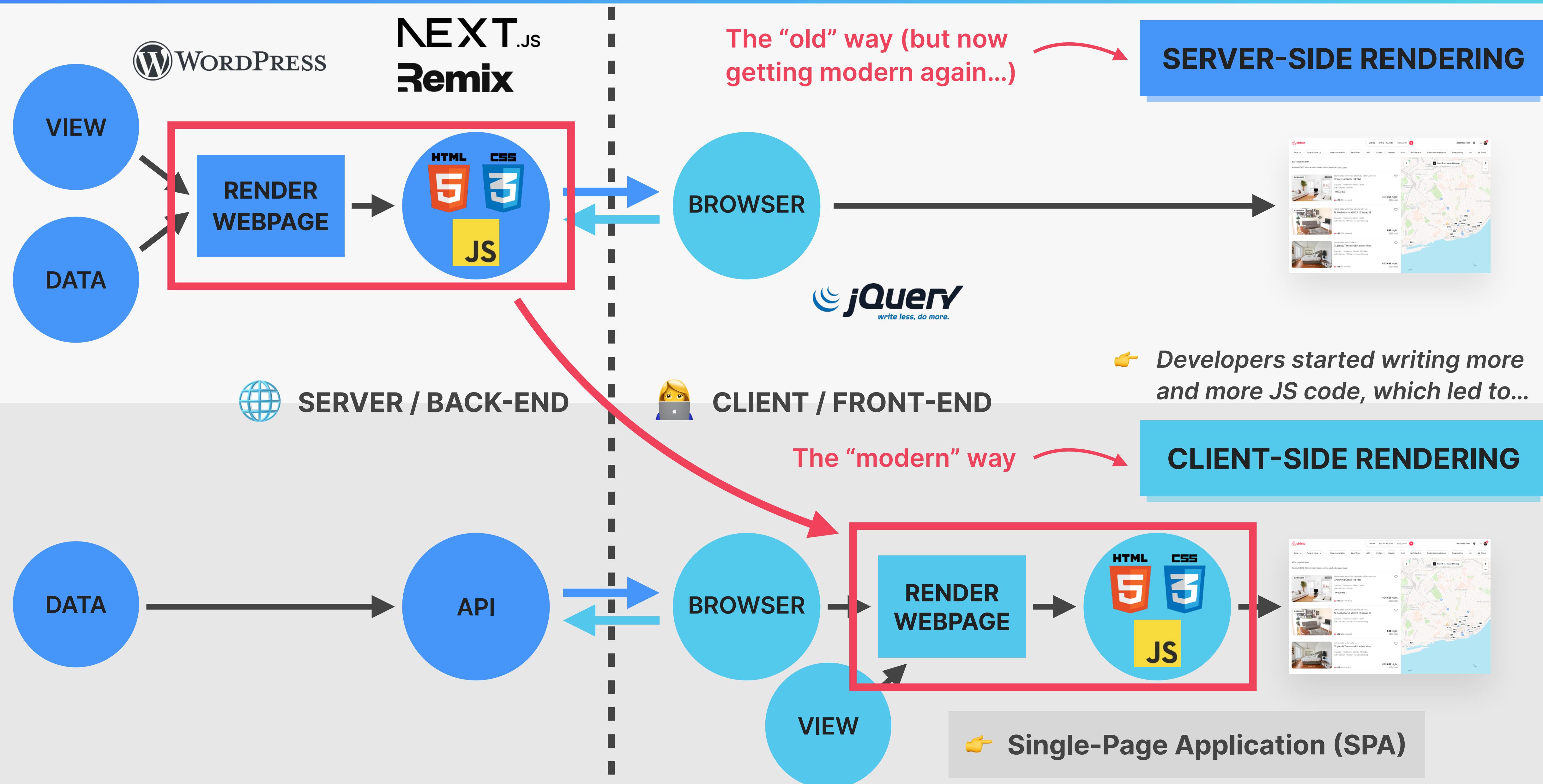
**SECTION**

A FIRST LOOK AT REACT

**LECTURE**

WHY DO FRONT-END  
FRAMEWORKS EXIST?

# THE RISE OF SINGLE-PAGE APPLICATIONS



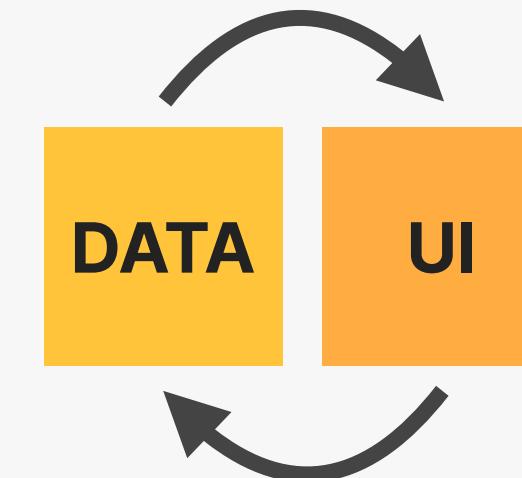
# SINGLE-PAGE APPLICATIONS WITH VANILLA JAVASCRIPT?

👉 *Front-end web applications are all about...*

**Handling data + displaying  
data in a user interface**

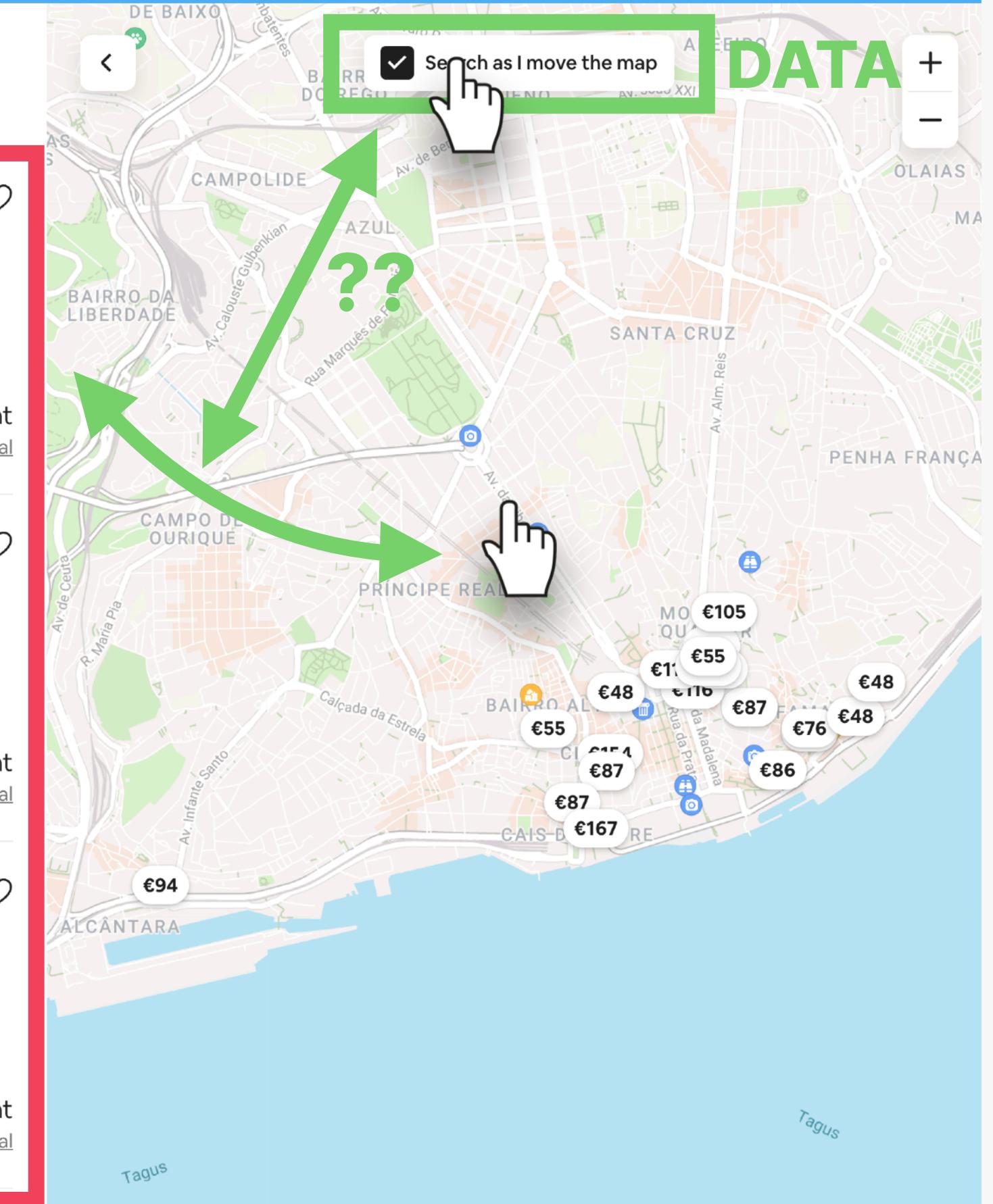
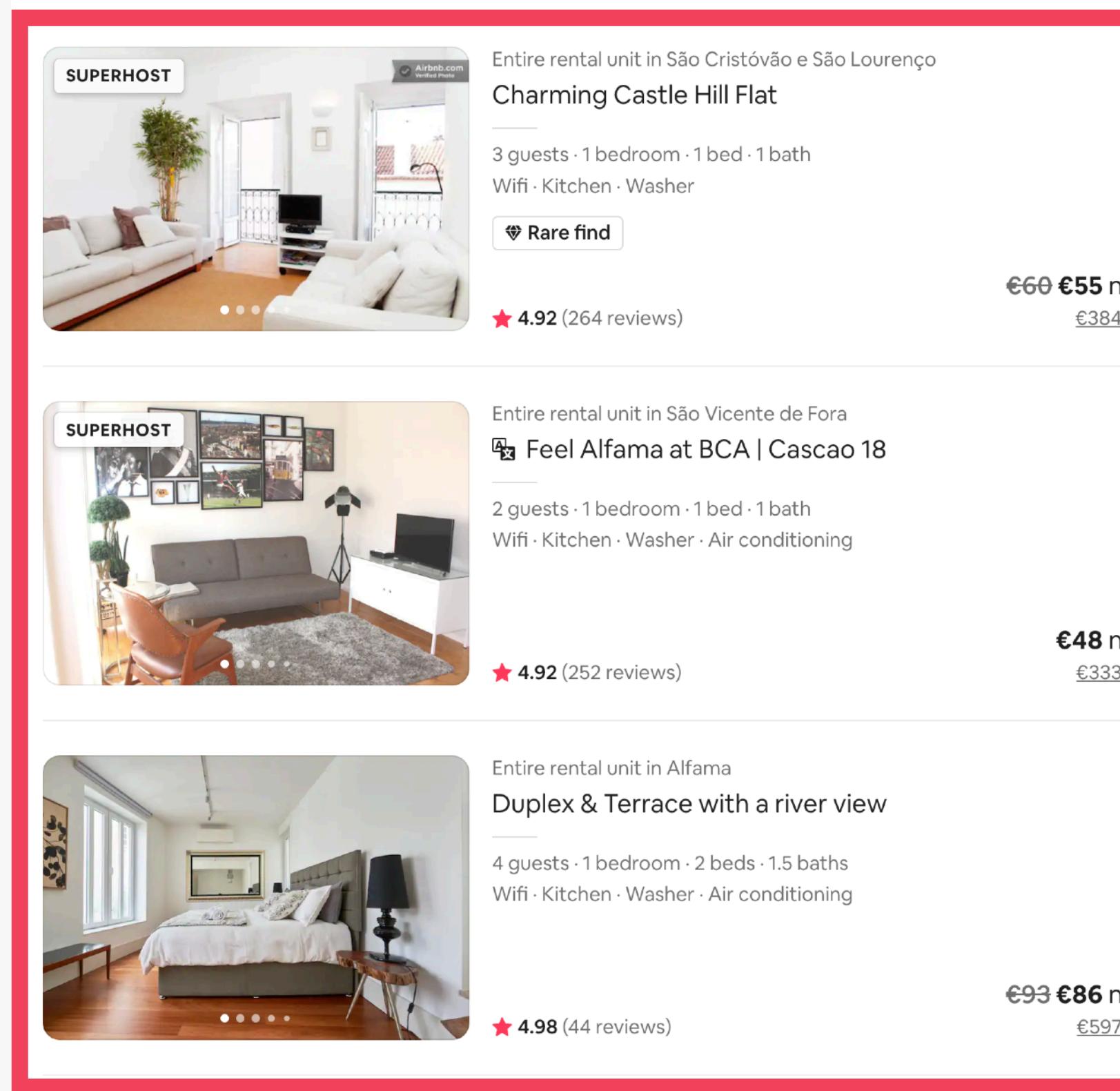
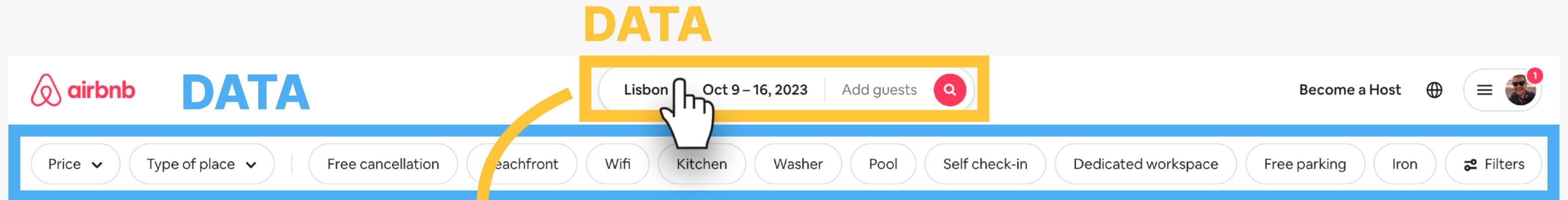


User interface needs to  
**stay in sync** with data



**Very hard problem** to solve!

# KEEPING UI IN SYNC WITH DATA



The image shows the Airbnb search interface and listing details page. A yellow arrow points from the search bar to the word 'DATA'. The listing details page has three listings highlighted with red boxes and labeled 'DATA'. A green arrow points from the search bar to the word 'DATA' on the map.

👉 Keeping UI  
and data in sync  
would be virtually  
impossible with  
just vanilla  
JavaScript

Piece of data  
=  
Piece of state

```
const data = {  
  currency: {  
    currencyCountries: [],  
    currencySelectorExpanded: false,  
    selectedCurrencyCode: 'EUR',  
    loadingCurrencies: false,  
  },  
  footer: {  
    footerExpanded: false,  
  },  
  map: {  
    hoveredListingId: null,  
    hoveredDestinationPlaceId: null,  
    clickedListingId: null,  
    expandMapToItLocations: null,  
  },  
  saveToListModal: {  
    entity: null,  
    entityId: null,  
    entityType: null,  
    fetchListsError: false,  
    isCreatingList: false,  
    isFetchingLists: false,  
    isListsCacheValid: true,  
    entityMap: {  
      'listing-36109352': {},  
      'listing-34888453': {},  
      'listing-13357289': {},  
      'listing-29842619': {},  
      'listing-21693919': {},  
      'listing-44308963': {},  
      'listing-45085968': {},  
    },  
    lastError: null,  
    lastListSavedTo: null,  
    lastListSavedToOperation: null,  
    lists: {  
      id: 870054853,  
      name: 'Bons',  
      listing_ids: [  
        36109352, 34888453, 13357289, 29842619, 21693919, 44308963, 45085968,  
      ],  
      mt_template_ids: [],  
      place_activity_ids: [],  
      place_ids: [],  
      article_ids: [],  
      mt_scheduled_template_ids: [],  
      is_china_wishlist_home_collection: false,  
      settings_disabled: false,  
      airbnb_canonical_place_ids: [],  
      listing_id_str: [  
        '36109352',  
        '34888453',  
        '13357289',  
        '29842619',  
        '21693919',  
        '44308963',  
        '45085968',  
      ],  
      newListName: null,  
      newTask: {  
        actionText: '',  
        listRef: null,  
        message: null,  
      },  
      requiresSignup: false,  
      savingFrom: null,  
      visible: false,  
    },  
    ui: {  
      hideMap: true,  
      openedFilterId: null,  
      openedSearchInputField: null,  
      shouldLoadInterceptSurvey: false,  
      visiblePromos: {},  
    },  
    header: {  
      user: {  
        isLoggedIn: true,  
        profilePicUrl:  
          'https://a0.musache.com/im/pictures/user/9ed27941-a4f0-4c5e-b347-959a7f  
        name: 'Jonas',  
        currency: 'EUR',  
        isHost: false,  
        guidebooksCount: 0,  
      },  
      dynamicColorTheme: null,  
      activeNavigationItem: null,  
      navigationItemsWithNotifications: 0,  
      flyoutMenuIsOpen: false,  
    };  
};
```

# SINGLE-PAGE APPLICATIONS WITH VANILLA JAVASCRIPT?

👉 *Front-end web applications are all about...*

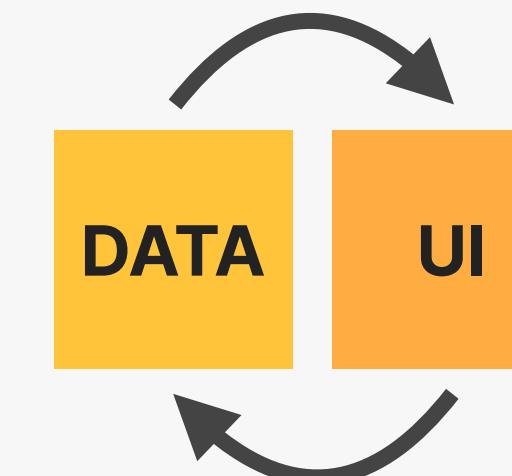
**Handling data + displaying  
data in a user interface**



User interface needs to  
**stay in sync** with data



**Very hard problem** to solve!



## PROBLEMS WITH **jQuery**

1

Requires lots of direct DOM manipulation and  
traversing (*imperative*) ➡ “Spaghetti code” 🍝

```
const guestsEl = document.querySelector('.guests');
const guestsPickerEl = document.querySelector('.picker');

guestsEl.addEventListener('click', function () {
    guestsEl.classList.toggle('inactive');
    guestsEl.classList.toggle('active');

    if (guestsPickerEl.style.display === 'block') {
        guestsPickerEl.style.display = 'none';
        guestsEl.firstElementChild.textContent = 'Add guests';
    } else {
        guestsPickerEl.style.display = 'block';
        guestsEl.firstElementChild.textContent = '';
    }
})
```

2

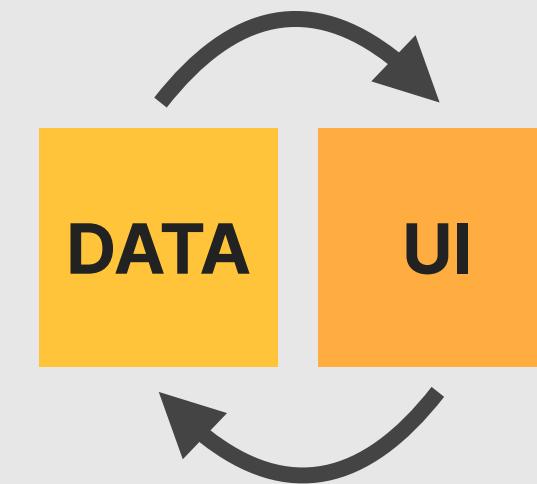
Data (state) is usually **stored in the DOM**, shared  
across entire app ➡ Hard to reason + bugs 🐛

# WHY DO FRONT-END FRAMEWORKS EXIST?

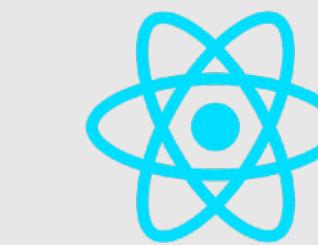
1

JavaScript front-end frameworks exist because...

KEEPING A USER INTERFACE IN SYNC WITH DATA  
IS REALLY HARD AND A LOT OF WORK



Front-end frameworks **solve this problem** and take hard work away from developers 🎉



←

Different approaches, same goal

2

They enforce a “**correct**” way of structuring and writing code (therefore contributing to solving the problem of “spaghetti code” 🍝)

3

They give developers and teams a **consistent** way of building front-end applications