

Adaptive Web Engines

vicyao@tencent

Changes brought by mobile internet

Changes in the ecosystem of Internet

- Mobile phones are the main stage
 - mobile phones account for 56.8 percent of online time
 - nearly 90% of mobile usage is spent in **apps**
- Apps become the main gateway to the internet

Changes in Web on Mobile phones

- Main sources of web traffic: Not browser, but Apps
 - social media, news feeds, or e-commerce platforms
 - No complete “Website”, just fragmented pages

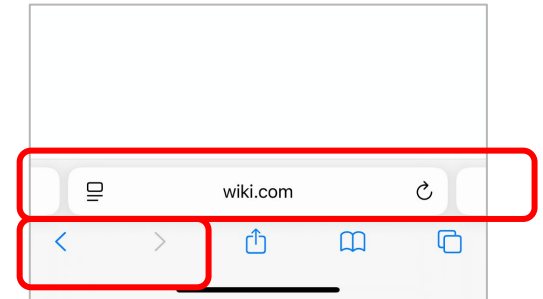
Changes of the structure of Internet

- Internet isn't a “web” anymore, it's more like a chain of isolated islands(thanks to Apps)
 - keep users inside
 - Don't jump outside

No browser, No website

But Web never change

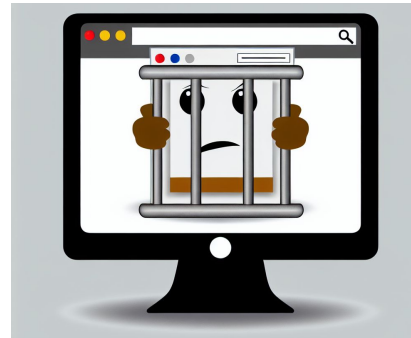
- Web was created 35 years ago
- From Desktop to mobile phone, web only make it compatible
- The interaction between laptop and mobile phone
 - mice and keyboards
 - forward/back buttons
 - URL
 - White screen waiting, sudden changes
 - touch screen
 - long list scrolling
 - gesture controls
 - smooth page transitions
- Terrible experience on mobile phone



who likes to type a url on a cell phone?

If we keep the web unchanged

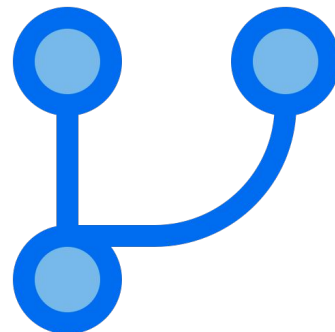
- Web, AKA the lowest performance and the most complex tech
 - A simple DIV has 246 properties and 82 methods
 - CSS display has 23 options
 - Too many features that there might not be any new browsers in the future.
- Smartphones, Car systems, watches, TVs, glasses, and more smart device will come, BUT Web could get stuck on desktop
- UI tech will become more fragmented, more RN, Flutter will come
- One web principle will be meaningless



What shall we do

Not subsetting web features

- logic and rendering threads still fighting each other
- A large number of features, low-performance, can't be resolved
- Still won't feel mobile-native
- It will grow over time eventually



A new Web engine

- Distributed as Packages, like app and mini app
- Extensibility, Mixing binary code and script code
- Mobile-Friendly by Default, for devs and users
- Reuse CSS and APIs, Redesign components

Why Packages ? It fit mobile use cases:

1. Embedded in Apps
2. Small pkgs for Real-Time Pages
3. Support Mini-App Ecosystem

Beyond Phones

- Imagine a unified, cross-device Web tech works everywhere
- Standardized, low-cost software development could spark innovation in hardware devices
- Only W3C can make this happen

MWGA

What is Mobile Friendly

WeChat drop-down refresh evens

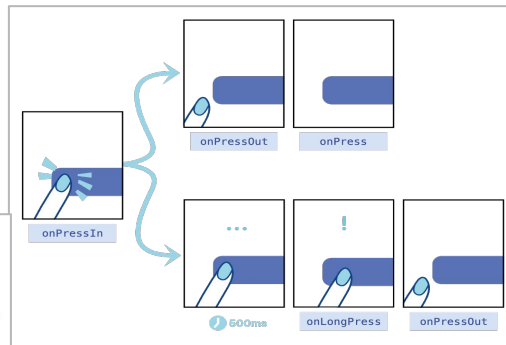
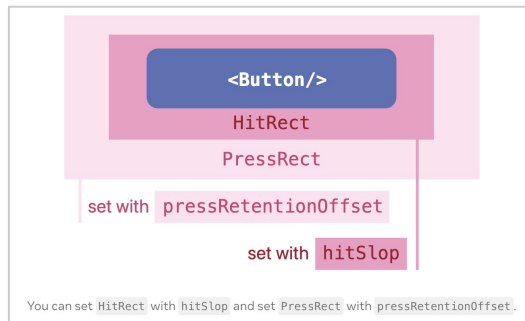
```
buildText(status: RefreshStatus) {  
  switch (status) {  
    case RefreshStatus.Idle:  
      return 'Pull to refresh'  
    case RefreshStatus.CanRefresh:  
      return 'Release to refresh, pull to enter second level'  
    case RefreshStatus.Refreshing:  
      return 'Refreshing'  
    case RefreshStatus.Completed:  
      return 'Refresh successful'  
    case RefreshStatus.Failed:  
      return 'Refresh failed'  
    case RefreshStatus.CanTwoLevel:  
      return 'Release to enter second level'  
    default:  
      return ''  
  }  
}
```

RN(so thoughtful)

KeyboardAvoidingView

This component will automatically adjust its height, position, or bottom padding based on the keyboard height to remain visible while the virtual keyboard is displayed.

RN Pressable Component



Flutter fancy scrolling

