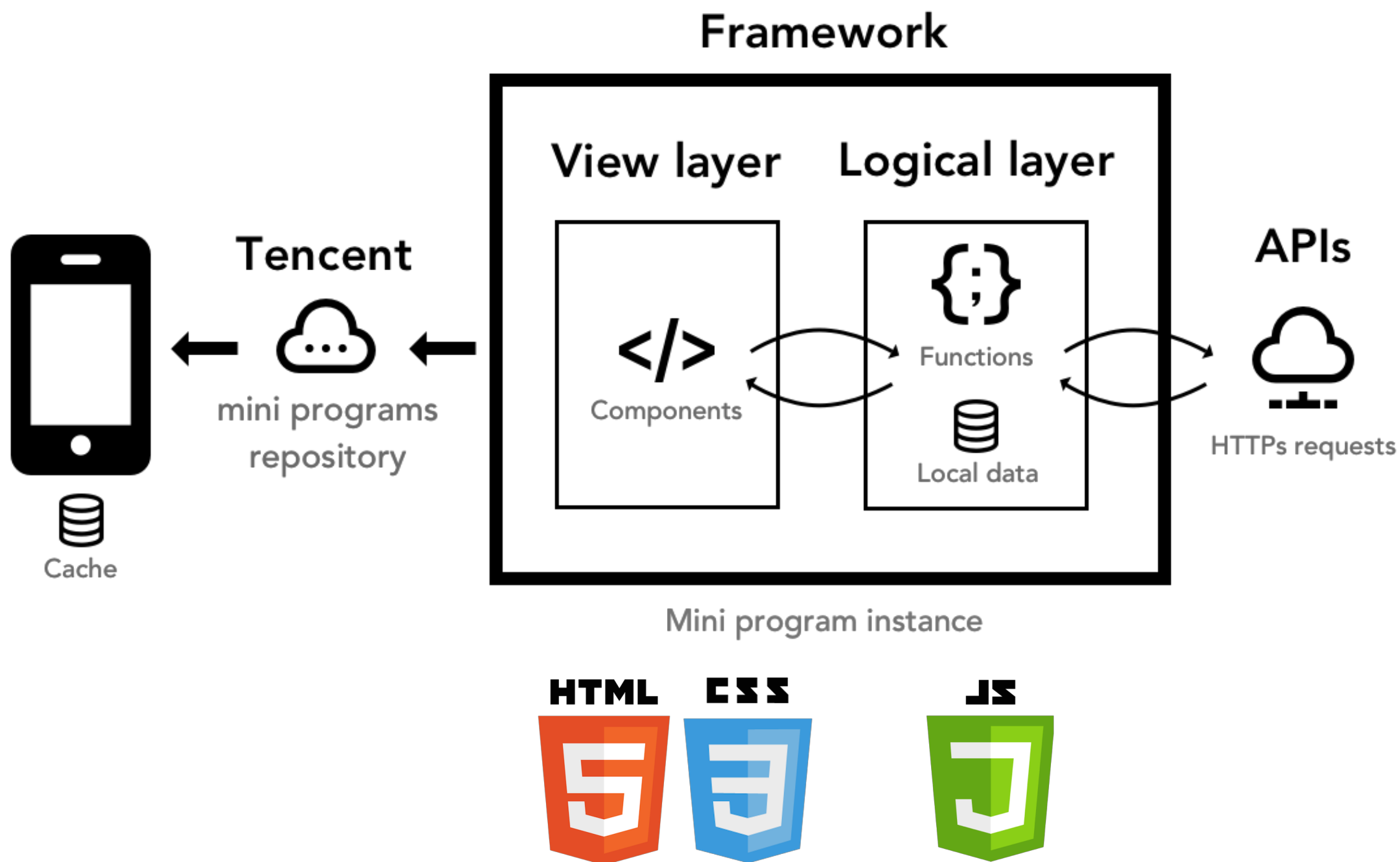




小程序

**How do you make
dynamic WeChat pages?**



Architecture

```
/
├── app.js
├── app.json
├── app.wxss
├── project.config.json
├── pages
│   ├── index
│   │   ├── index.js
│   │   ├── index.json
│   │   ├── index.wxml
│   │   └── index.wxss
│   └── logs
│       ├── logs.js
│       ├── logs.json
│       ├── logs.wxml
│       └── logs.wxss
└── utils
    └── util.js
```



Data Binding

Data binding uses mustache syntax to wrap variables.

```
<!-- .wxml -->  
<view>{{text}}</view>  
<view>{{array[0].msg}}</view>
```

```
// .js  
Page({  
  data: {  
    text: 'init data',  
    array: [{msg: 'I am message 1'}, {msg: 'I am message 2'}]  
  }  
})
```

Event Handlers

```
<!-- .wxml -->  
<button type="primary" bindtap="add">  
Incrementation: {{count}}  
</button>
```

```
// .js  
Page({  
  data: {  
    count: 1  
  },  
  add: function(e) {  
    this.setData({  
      count: this.data.count + 1  
    })  
  }  
})
```



小程序

**How do you use all
WeChat built-in features?**

APIs

👉 <https://developers.weixin.qq.com>

API - Program Document

https://developers.weixin.qq.com/miniprogram/en/dev/api/

微信公众平台 | 小程序

Mini Game Introduction ▾DesignMini Program Development ▾OperationData ▾Community中文 / EN

Easy TutorialFrameworkComponentsAPIToolsCloud Base

> Network

> Media

> File

> Data Cache

> Location

> Device

> Interface

> Third-party Platform

> Open Interface

> Data

> Update

> Multi-threading

> Monitor

API

Framework provides many WeChat native APIs that can be conveniently called to access abilities provided by WeChat, such as acquiring user information, local storage, payment functions etc.

Instruction:

- APIs that start with wx.on are API interfaces for monitoring certain events. They accept a CALLBACK function as a parameter. When the event is triggered, the CALLBACK function is called.
- If there are no special arrangements, other API interfaces all accept an OBJECT as a parameter.
- Within the OBJECT you can specify `success`, `fail`, `complete` to accept the interface call result.

Parameter Name	Type	Required	Description
success	Function	No	Callback function when interface call succeeded
fail	Function	No	Callback function when interface call failed
complete	Function	No	Callback function when interface call completed (always executed whether call succeeds or fails)

Get the current location

Use wx.getLocation

```
<!-- .wxml -->  
<button type="primary" bindtap="listenerBtnGetLocation">  
Get location</button>
```

```
// .js  
listenerBtnGetLocation: function () {  
  wx.getLocation({  
    type: 'wgs84',  
    success: function (res) {  
      console.log(res)  
    }  
  })  
}
```


Make HTTPs requests

Use wx.request

```
// .js
getCatsNow: function () {
    var that = this
    var endpoint = "https://api.giphy.com/v1/gifs/search?q=funny+cat&api_key=dc6zaTOxFJmzC"
    wx.request({
        // request sent
        url: endpoint,
        header: { 'content-type': 'application/json' },
        success: function (res) {
            console.log('success!' + res.statusCode);
            that.setData({ catData: res.data.data })
        },
        fail: function (res) {
        },
        complete: function (res) {
        }
    })
}
```

Make HTTPs requests

Now display it!

```
// .WXML
```

```
<view wx:for="{{catData}}" wx:for-index="idx"
wx:for-item="data">
  <image src="{{data.images.downsized.url}}">
  </image>
</view>
```

Live code

Let's make a map

Your turn