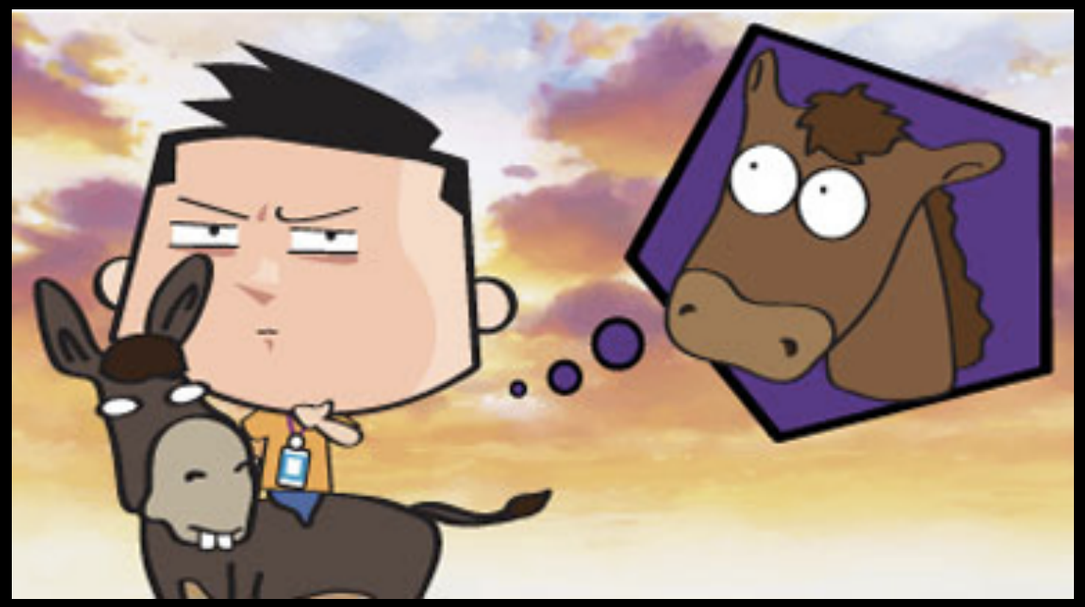
Hot Swapping Architecture

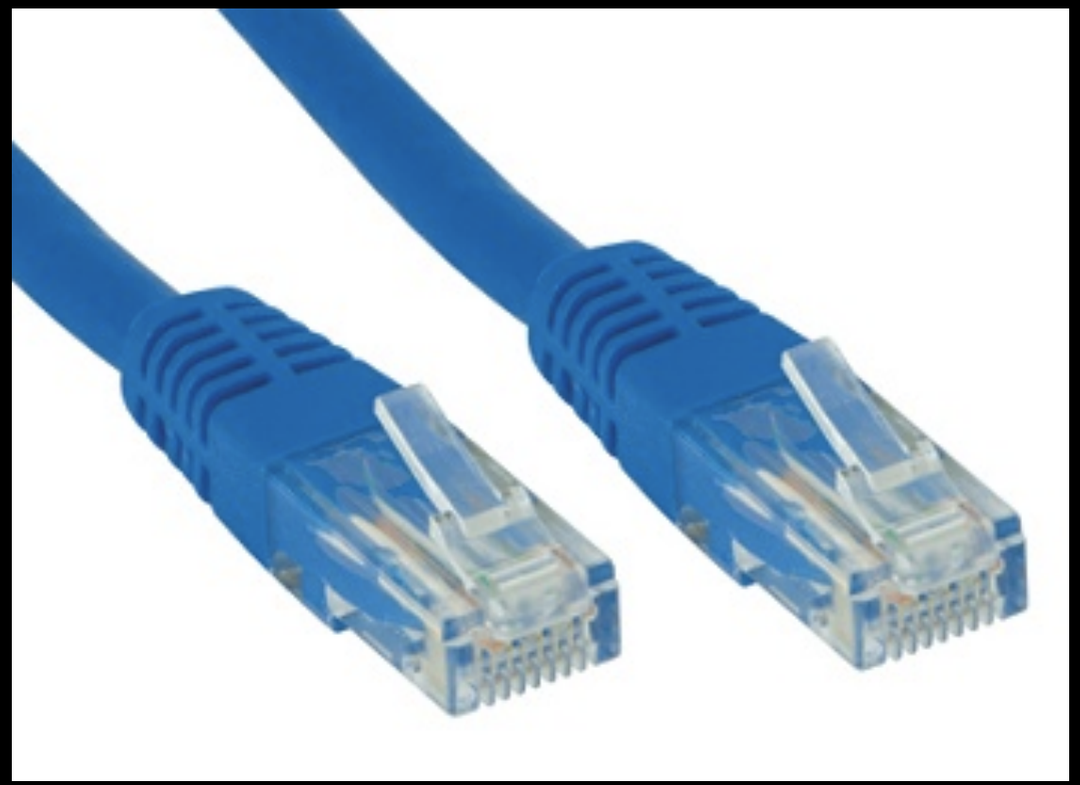
Part 1

Hot swapping (frequently called hot plugging) is replacing or adding components without stopping or shutting down the system.

–Wikipedia


PAULTAN.ORG

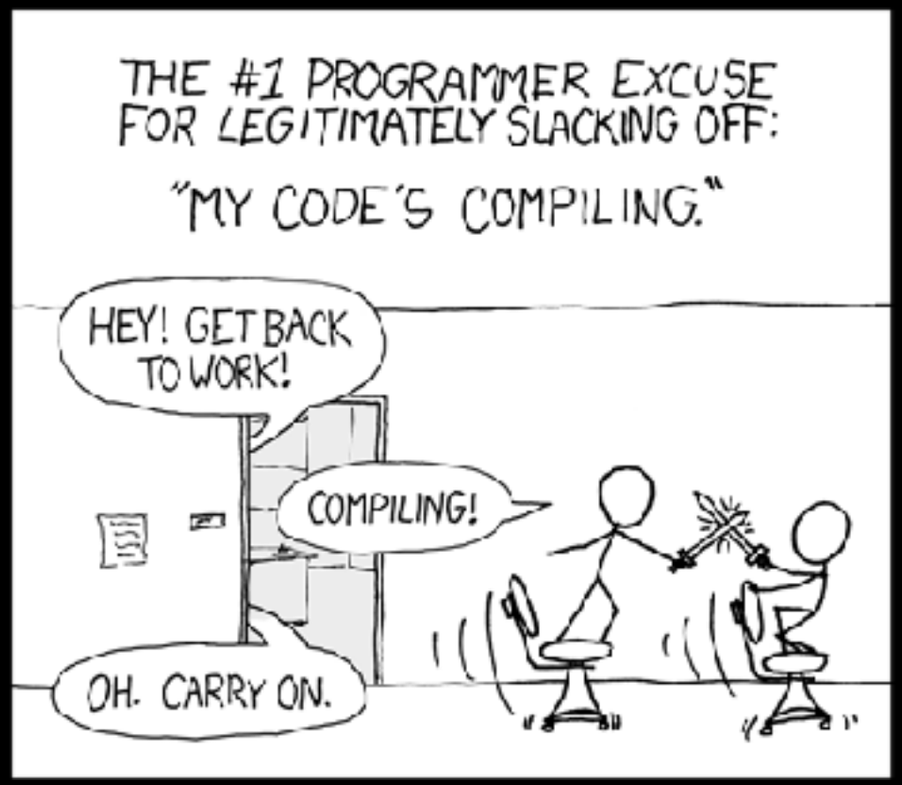


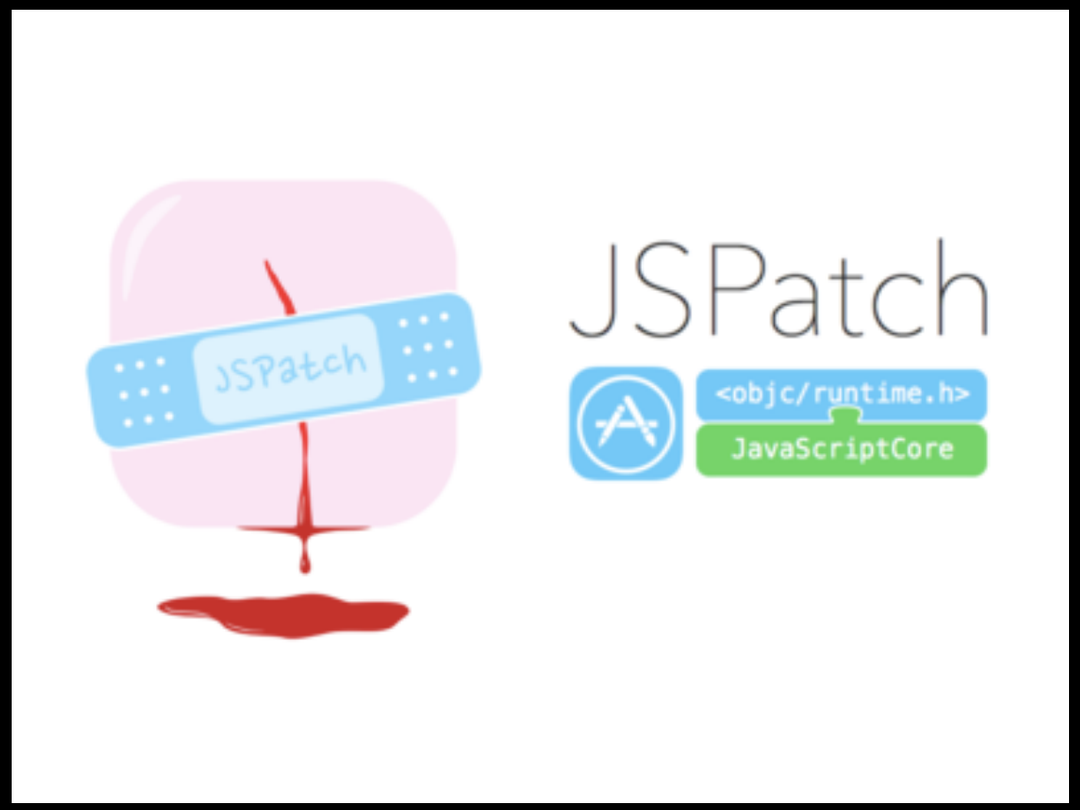


Video

Why hot swappable?

1. Low use cost: no reinstall and no reboot
2. Low development cost: incremental build, runtime debug
3. Low maintenance cost: zero downtime


THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S COMPILING."
HEY! GET BACK TO WORK!
COMPILING!
OH. CARRY ON.

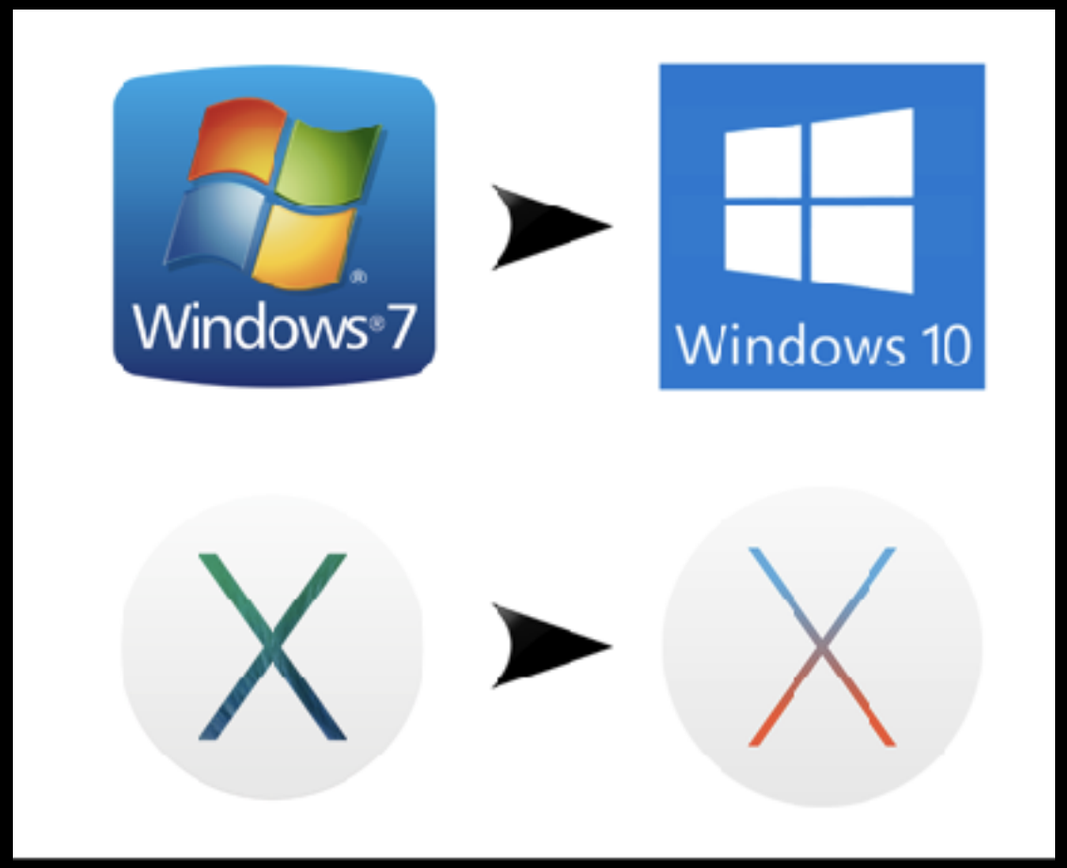

JSPatch
JSPatch
<objc/runtime.h>
JavaScriptCore


ERLANG

How to

make your code HOT swappable?

Basic principles

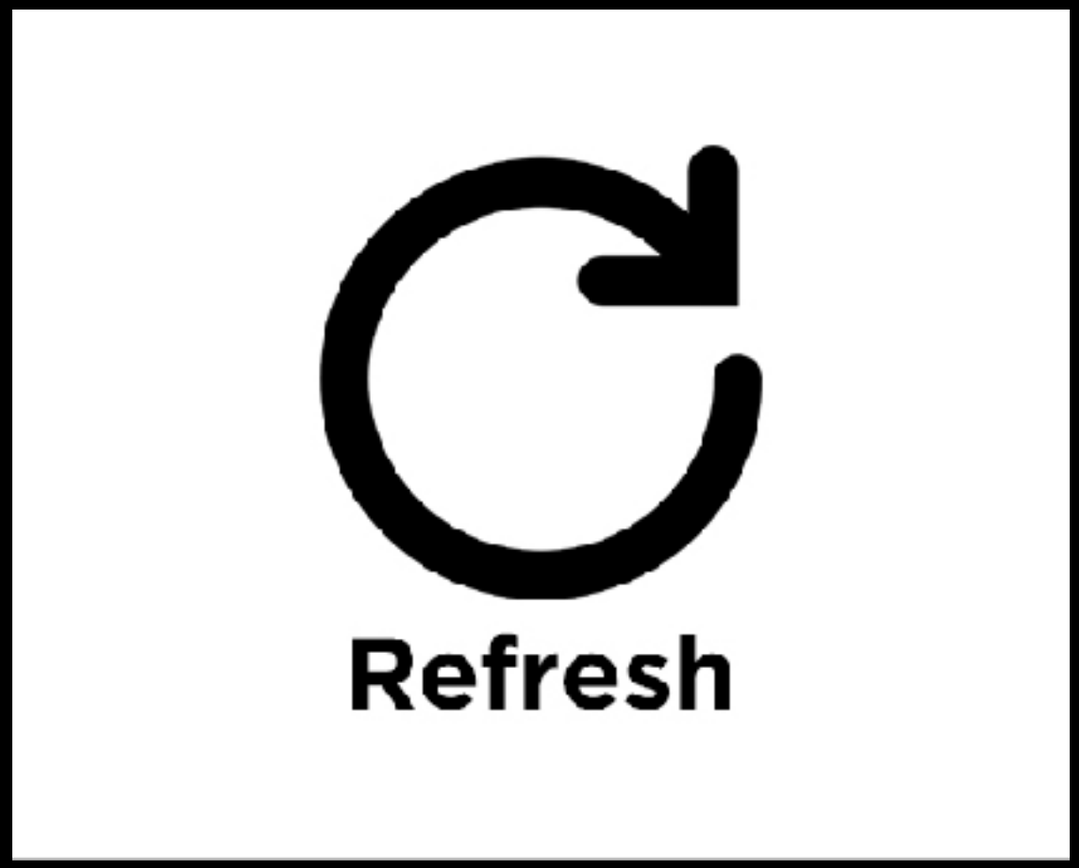


Full build and reboot are required. 😂

Basic principles

In app, full build and reboot. 😂
In OS, no rebuild no reboot. 😀

Basic principles



In webpage, full build and reboot are required. 😂
In browser, no rebuild no reboot. 😀
In OS, no rebuild no reboot. 😀

Basic principles

In webpage, no full rebuild no reboot. 😀
In browser, no rebuild no reboot. 😀
In OS, no rebuild no reboot. 😀

Add-ons Manager - Mozilla Firefox

Add-ons Manager

Firefox | about:addons

Search

Get Add-ons
Extensions
Appearance
Plugins
Languages
Services

Search all add-ons

Check to see if your plugins are up to date

OpenH264 Video Codec provided by Cisco System...
This plugin is automatically installed by Mozill... More
Preferences | Always Activate

DivX® Web Player
DivX Web Player version 1.4.0.233 More
Ask to Activate

iTunes Application Detector
This plug-in detects the presence of iTunes when opening iT... More
Ask to Activate

QuickTime Plug-in 7.6.6
The Videos 3.10.1 plugin handles video and audio streams. More
Ask to Activate

VLC Multimedia Plugin (compatible Videos 3.10.1)
The Videos 3.10.1 plugin handles video and audio streams. More
Ask to Activate

Windows Media Player Plug-in 10 (compatible; Videos)
The Videos 3.10.1 plugin handles video and audio streams. More
Ask to Activate

Atlassian Marketplace for Bitb

ladjzero

安全 | https://code.dianrong.com/plugins/servlet/upm/marketplace/atlassian

Bitbucket Projects Repositories People Find a repository...

Atlassian Marketplace for Bitbucket

Discover powerful add-ons compatible with your Bitbucket version via the Atlassian Marketplace.

Search the Marketplace | By Atlassian | All categories | All paid & free

Bitbucket Server Protect Unmerged Hook
Atlassian • Unsupported
REPOSITORY HOOKS
(3)
1,105 installations
Free
Request

A pre-receive hook that stops any push that deletes a branch involved in an active pull request.

Atlassian Universal Plugin Manager
Atlassian • Atlassian supported
ADMIN TOOLS
(138)
Preinstalled
Free

Use the UPM to manage installed add-ons, search for and install new add-ons, upgrade an application or add-on, enter license information and more.

Support Tools Plugin
Atlassian • Atlassian supported
ADMIN TOOLS
(62)
Preinstalled
Free

The Support Tools Plugin provides self-help resources such as Hercules, which lets you scan your log files for known problems.

Atlassian Plugin SDK - Windows
(1)

Basic principles

1. No full rebuild
2. No reboot
3. Less impact if they are required

Case: OSGi framework

OSGi is known as Open Services Gateway initiative

- defines the bundle which can be remotely installed, started, stopped, updated and uninstalled
- implements life cycle to management for bundles
- implements services layer to connect bundles with each other

So we need

1. a component model to isolate swappable components from the software system

Hot swapping steps

- Detect changes
- Load the new
- Swap out the old
- Free the old

Case: nginx hot reload

When nginx hot-reloads the new configuration, the master process checks and applies the new

1. and then starts new worker processes to service new clients
2. and closes old worker processes gracefully
3. old worker processes are shut down once all clients are serviced

So we need

1. a component model to isolate swappable components from the software system

2. a runtime to manage the swapping steps

Let’s code

Using Node.js

- *main( )* invokes *print( )* every second
- edit and save print.js
- to make what *main( )* prints change

main-1.js

```
const print = require('./print')

function main() {
  setInterval(print, 1000)
}

main()
```

main-2.js

```
function main() {
  setInterval(() => {
    const print = require('./print')
    print()
  }, 1000)
}

main()
```

main-3.js

```
const fs = require('fs')

function main() {
  setInterval(() => {
    fs.readFile('./print.js', (err, data) => {
      const module = { exports: null }
      eval(data.toString())
      module.exports()
    })
  }, 1000)
}

main()
```

main-4.js

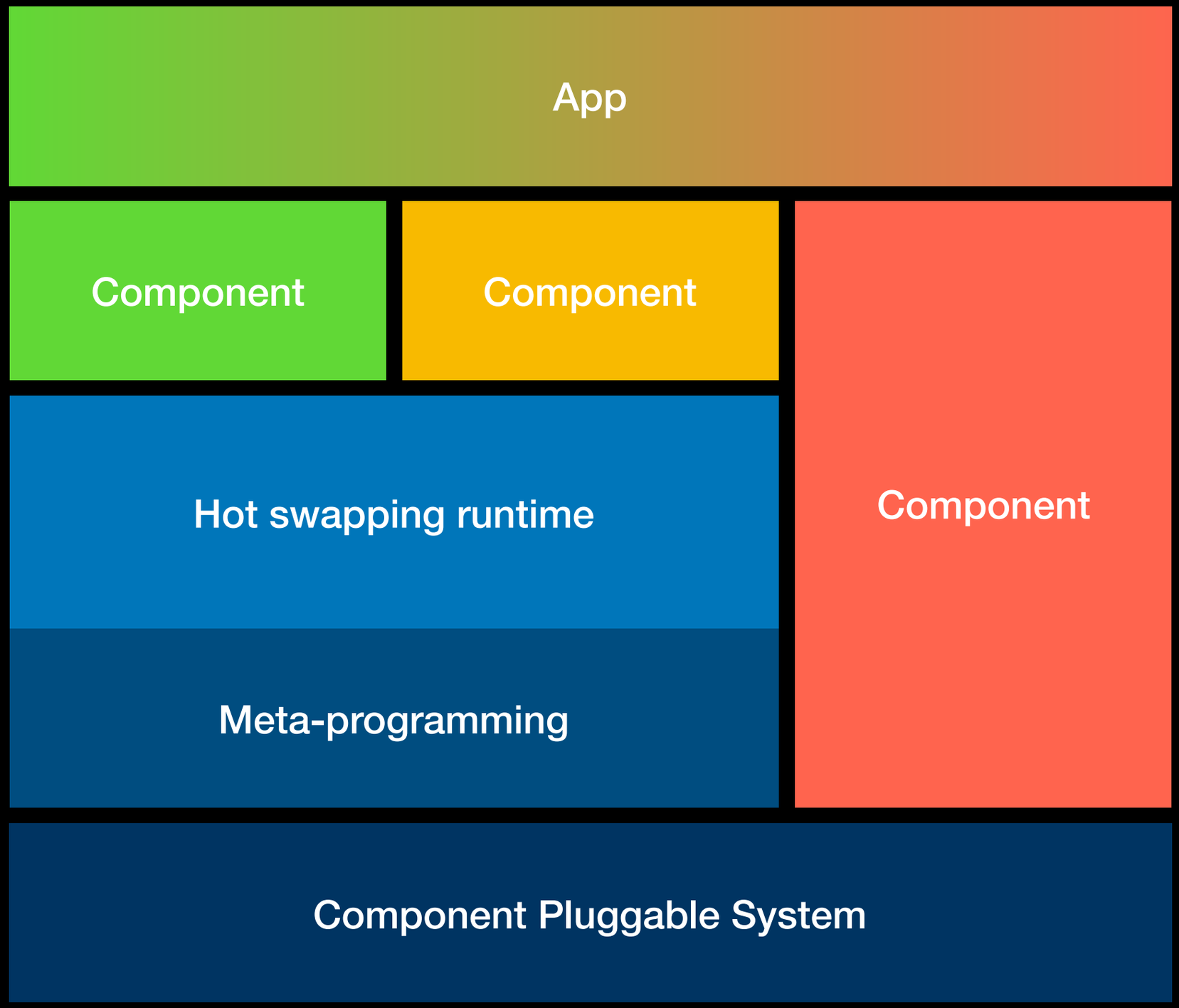
```
function main() {
  setInterval(() => {
    delete require.cache[require.resolve('./print')]
    const print = require('./print')
    print()
  }, 1000)
}

main()
```

Metaprogramming is writing programs that operate on other programs.

http://cs.lmu.edu/~ray/notes/metaprogramming/

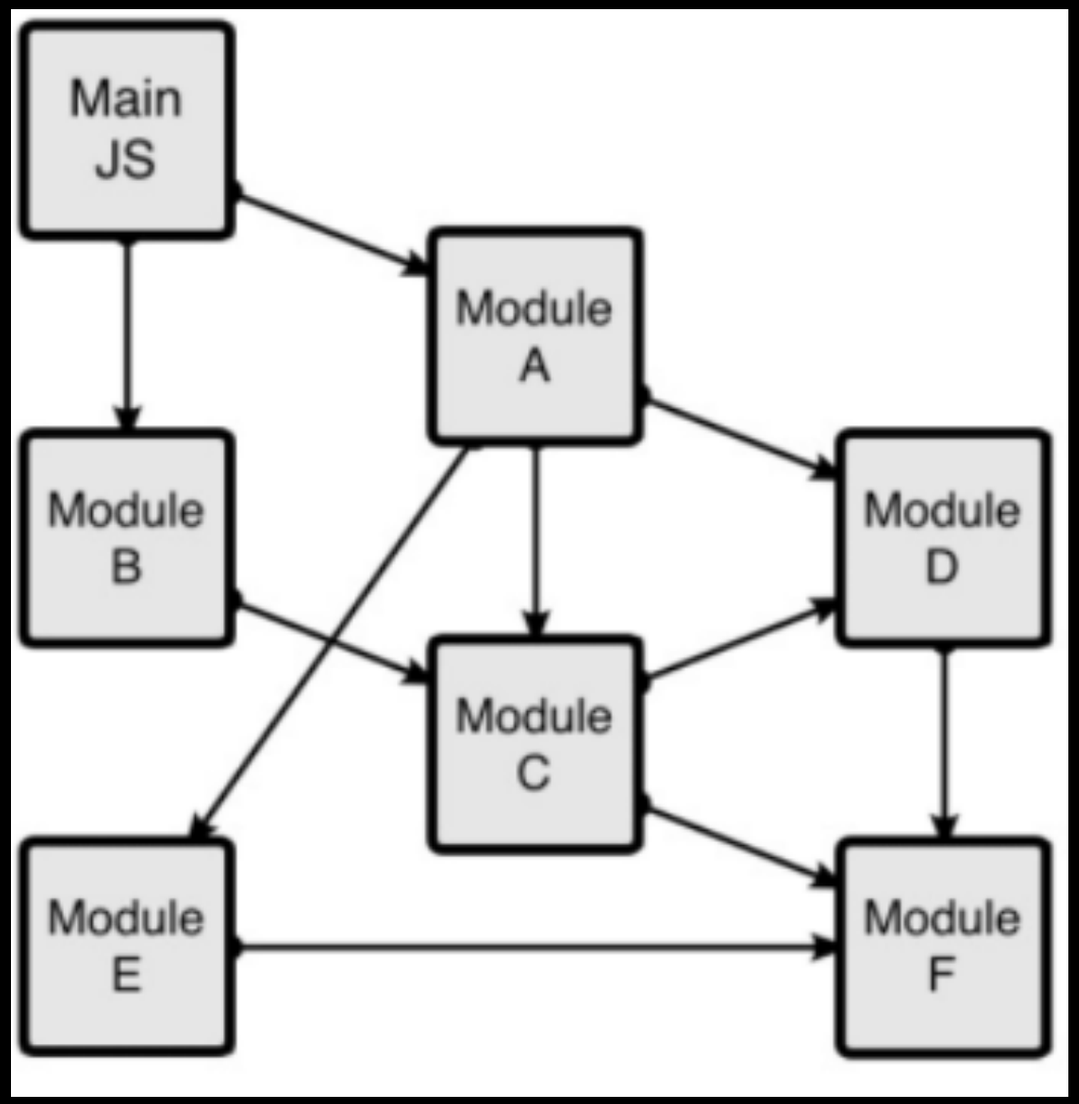
So we need

1. a component model to isolate swappable components from the software system
2. a runtime to manage the swapping steps
3. meta-programming capability to link dynamically


App
Component
Component
Component
Hot swapping runtime
Meta-programming
Component Pluggable System

Case study

webpack HMR



Component model

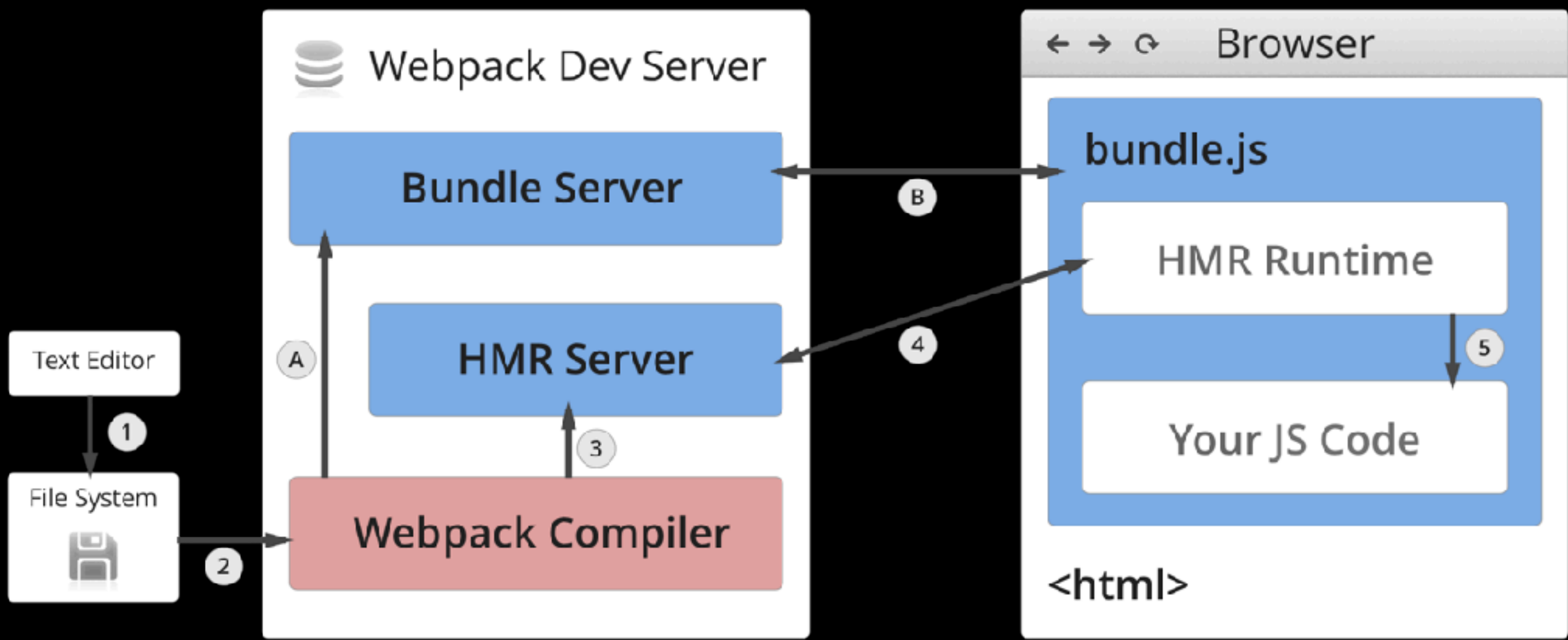
In webpack

- any file is a module
- code dependencies are abstracted as a module dependency graph with a single root
- require/import is reimplemented in order to manage modules' life cycle

Steps management runtime

webpack injects HMR runtime

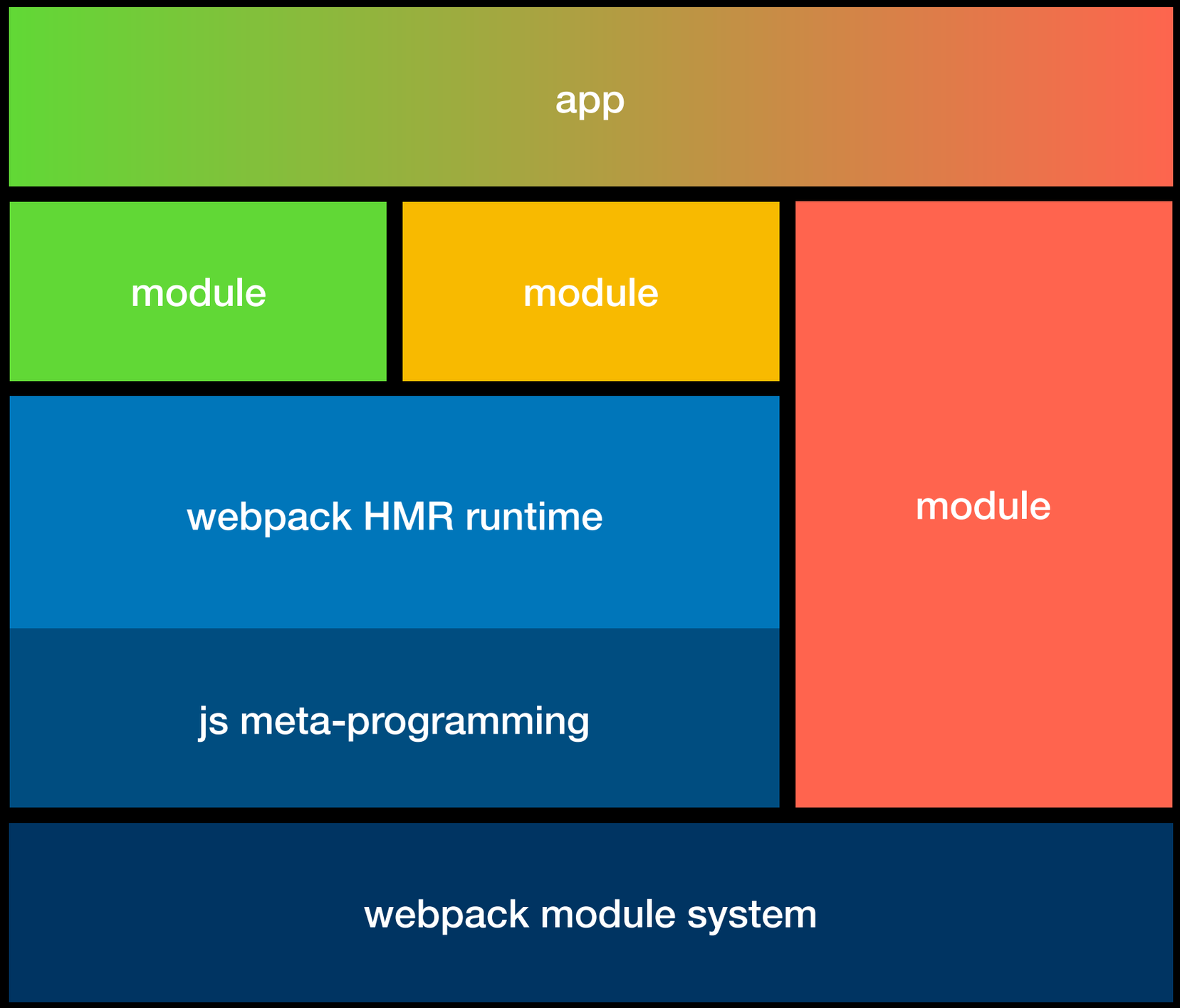
- via webpack-dev-server and HMR plugin
- to keep modules between browser and server in sync
- to check, download and apply the new modules
- and clear the old ones and their side effects


Webpack Dev Server
Bundle Server
HMR Server
Webpack Compiler
Browser
bundle.js
HMR Runtime
Your JS Code
<html>
Text Editor
File System
A
B
1
2
3
4
5

Meta-programming feature

JavaScript is pretty dynamic so that it

- can link new codes by injecting <script />
- can run plain source code using ‘eval’
- can modify any plain object easily, so that webpack can manage the module system easily it provides


app
module
module
module
webpack HMR runtime
js meta-programming
webpack module system

Questions

1. What is the so called component?
2. How to make it hot swappable?
    - How to transfer the state and handle side effects?
3. How to compose them?

Component stack

webapps

business domains

react components

webpack modules

Component model systems

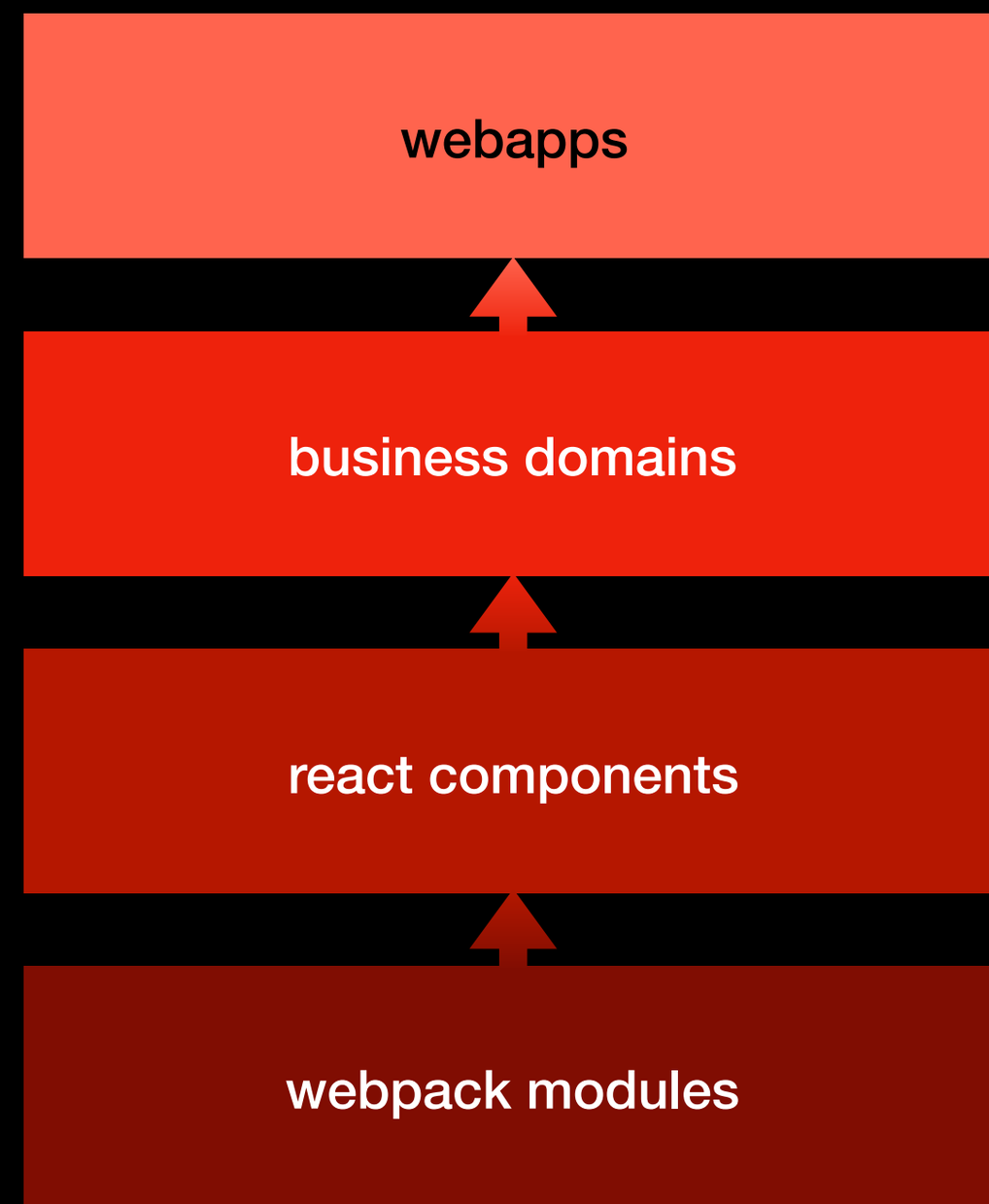
webapps	something like portal project
business domains	something more powerful than dva.js or mirror.js
react components	react
webpack modules	webpack module system

So we should first

- figure out the component stack against to the specified hot swapping (no rebuild, no reinstall, no reboot or something else)
- make clear which layer should be focused on
- design the component model of that layer

Runtime implementation

- If the lower layer is hot swappable. The upper layer could be hot swappable.
- If the lower layer is not hot swappable. Em, you have to reconstruct the stack.



Case: react-hot-loader

- webpack module is hot swappable
- webpack provides HMR API
- react-hot-loader hacks every react component it loads
- and implements HMR logic underneath
- so that the modified component re-renders magically

HMR API Demo

A HS component should be

- pure
- impure but side effects revocable
- stateless or state transferable
- compatible
- none of the above, but tolerable to system

Case: some real react app

Impure and side effects irrevocable

- log some info *onComponentDidMount*

State untransferable

- *var A = ( ) => { var key = randomKey(); return <B key={key} />}*

Incompatible

- *props.id* is required but it is optional before

State and side effects

Part 2 may cover this topic.

- They are very difficult and expensive to handle.
- They drive us to reboot our programs
- You can go into react-hot-loader and vue-loader for more details. 🙄

Compose them together

Part 2 may cover.

Please refer to component based programming.

Summary

To build a hot swappable architecture, we need

1. a component model which swappable components can be plugged in
2. a runtime to manage swapping things
3. meta-programming features underneath

Summary

To make the component model hot swappable, we need

1. hot swappable components below
2. component features like pure and stateless
3. reasonable compound mode

Thanks