Modular JavaScript at NETFLIX

@semmypurewal

techblog.netflix.com

jobs.netflix.com

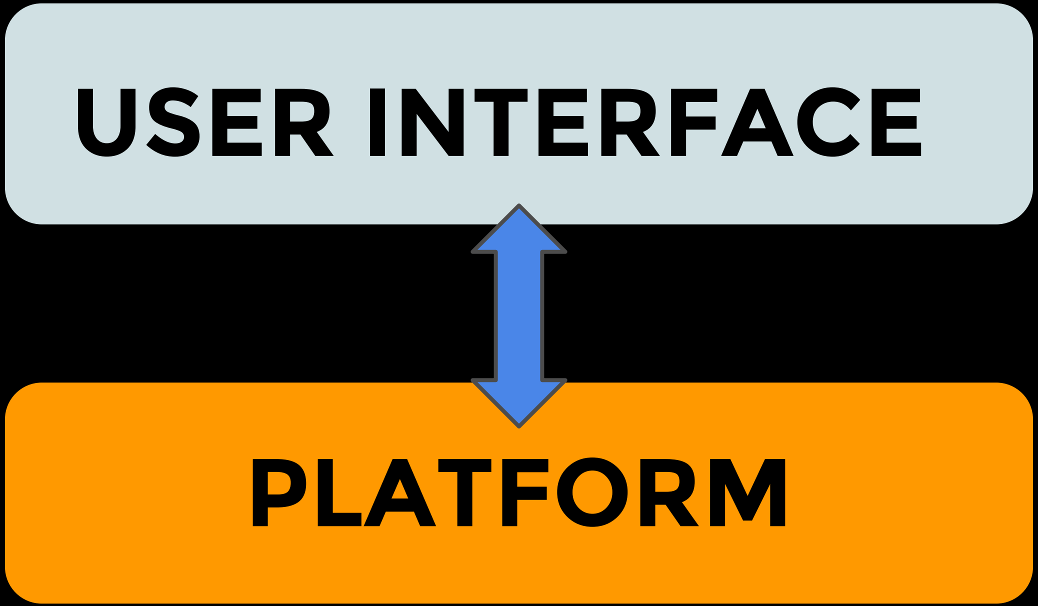
@netflixOSS

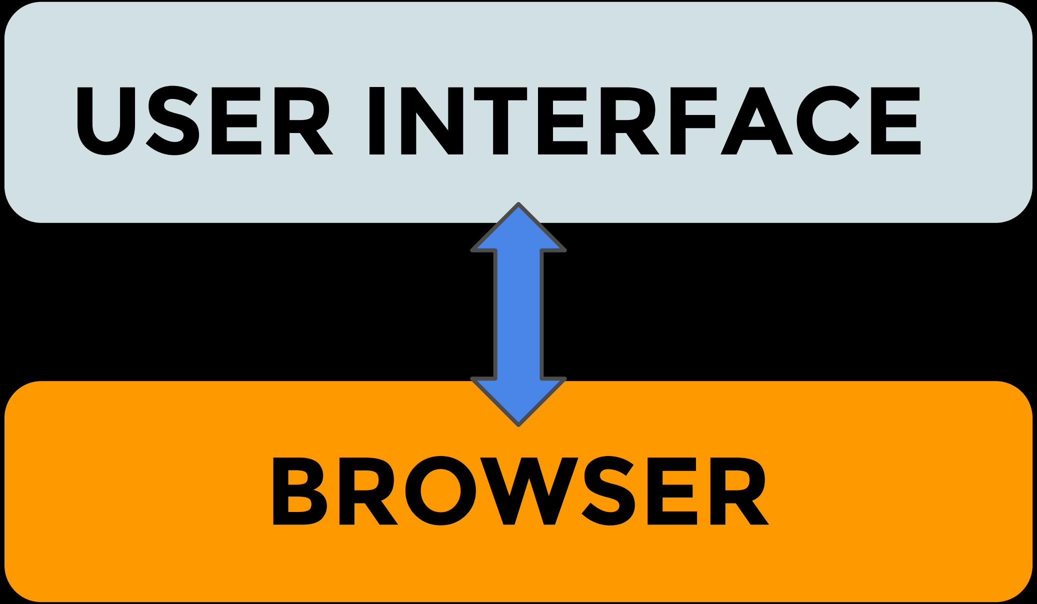
@netflixUIE

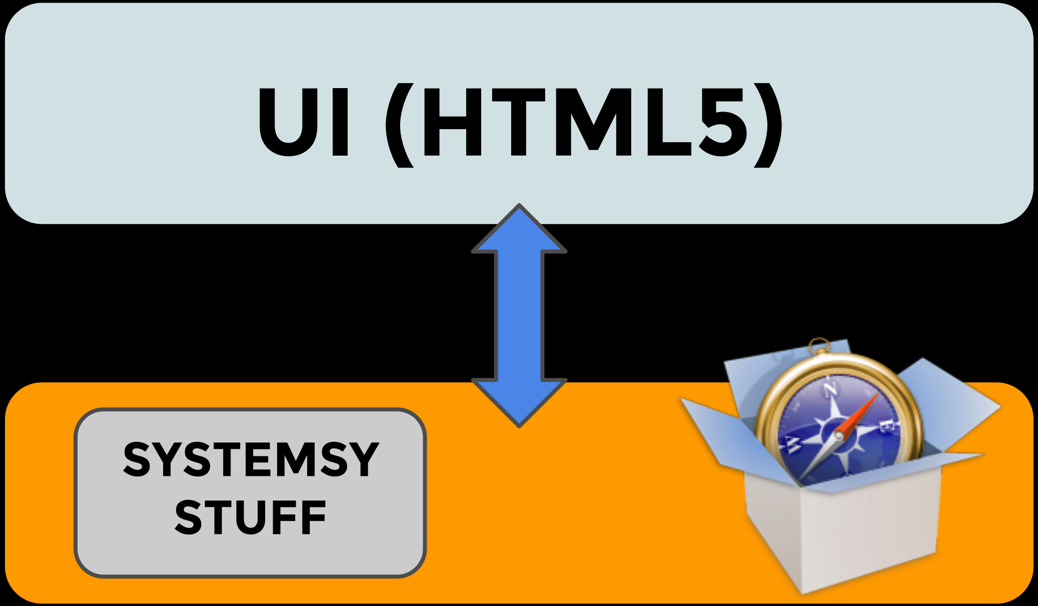
netflix.github.io

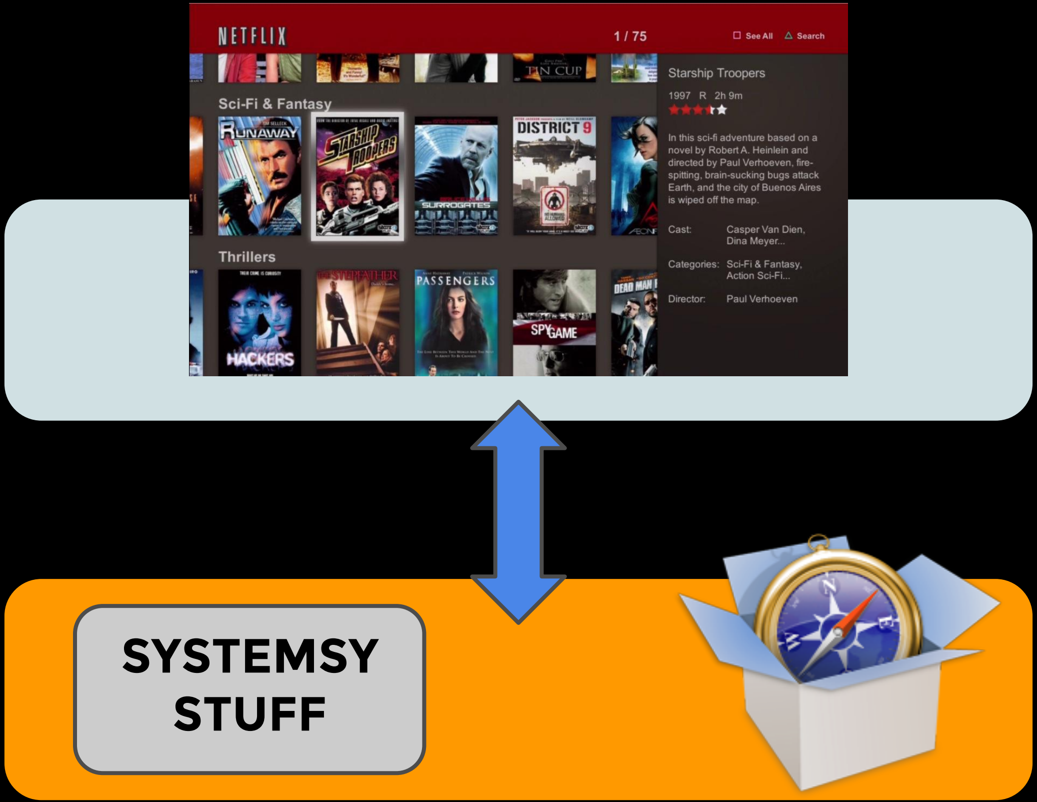
lets talk about devices

How can we create a consistent, updateable user-experience across all devices?


USER INTERFACE
PLATFORM


USER INTERFACE
BROWSER


UI (HTML5)
SYSTEMSY STUFF



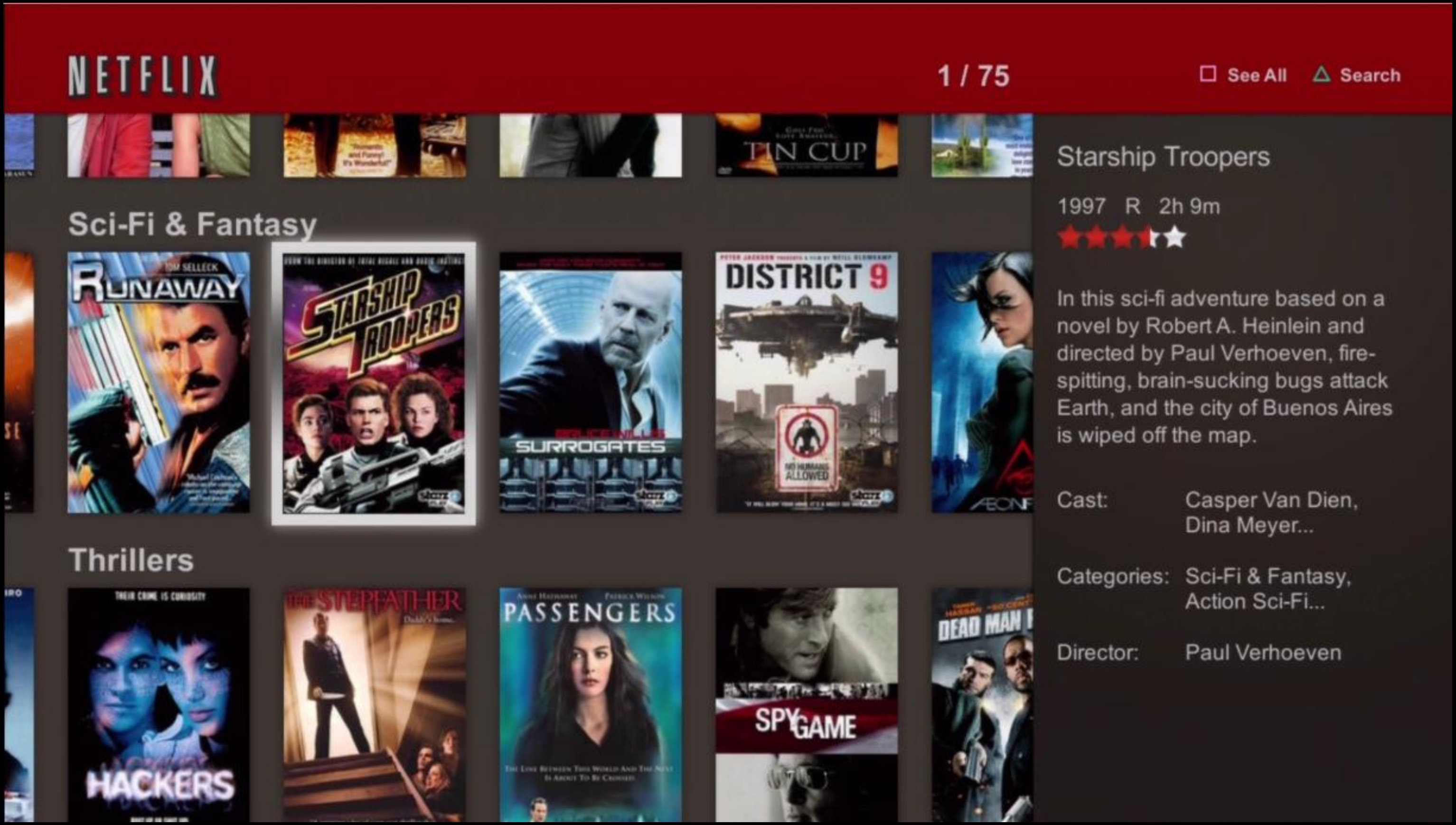
Starship Troopers

1997 R 2h 9m

In this sci-fi adventure based on a novel by Robert A. Heinlein and directed by Paul Verhoeven, fire-spitting, brain-sucking bugs attack Earth, and the city of Buenos Aires is wiped off the map.

Cast:	Casper Van Dien, Dina Meyer...
Categories:	Sci-Fi & Fantasy, Action Sci-Fi...
Director:	Paul Verhoeven

SYSTEMSY STUFF



Starship Troopers

1997 R 2h 9m

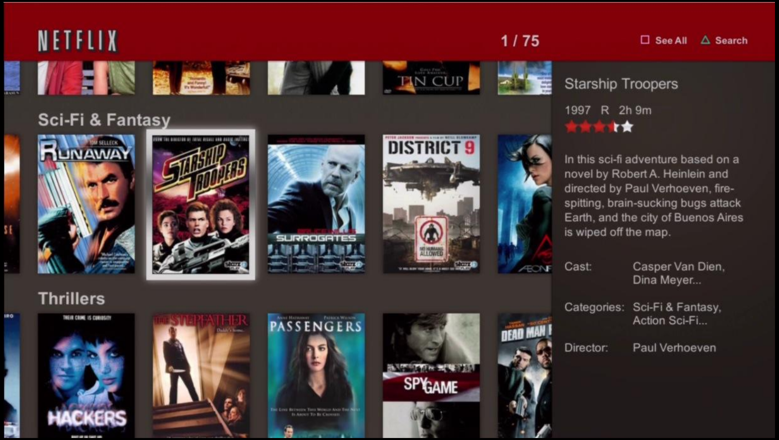
In this sci-fi adventure based on a novel by Robert A. Heinlein and directed by Paul Verhoeven, fire-spitting, brain-sucking bugs attack Earth, and the city of Buenos Aires is wiped off the map.

Cast:	Casper Van Dien, Dina Meyer...
Categories:	Sci-Fi & Fantasy, Action Sci-Fi...
Director:	Paul Verhoeven

WebKit-based HTML5 UI (McCarthy & Trott, OSSCON 2011)

WebKit-based HTML5 UI

performance on devices made innovation difficult



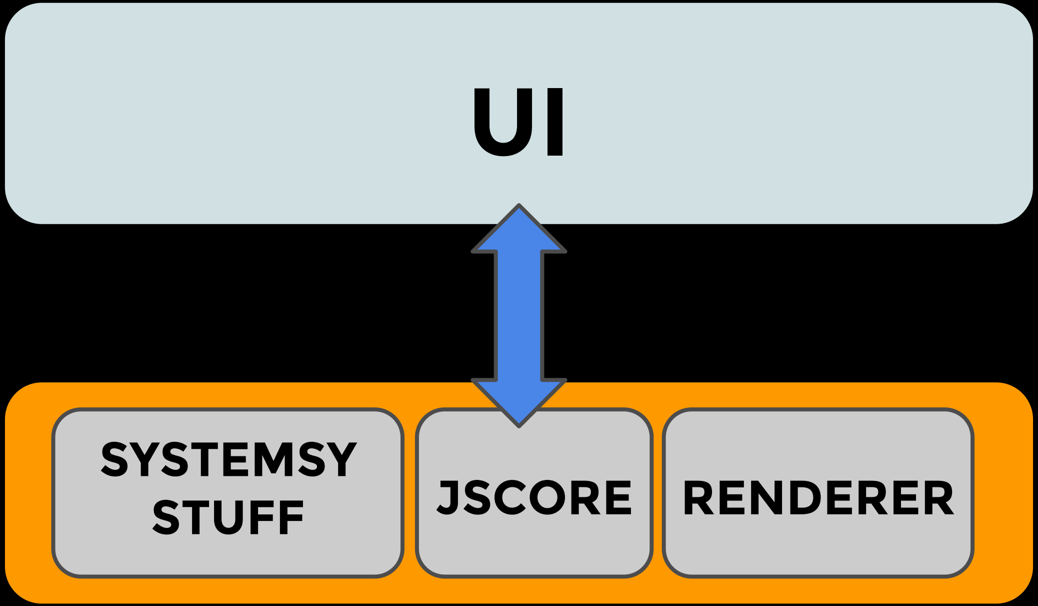
Starship Troopers

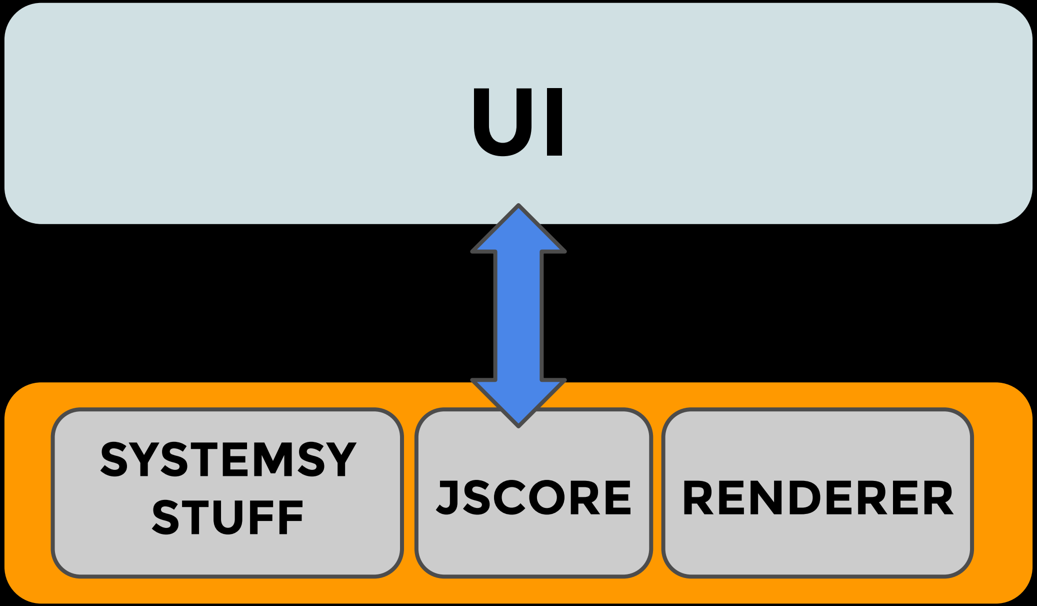
1997 R 2h 9m

In this sci-fi adventure based on a novel by Robert A. Heinlein and directed by Paul Verhoeven, fire-spitting, brain-sucking bugs attack Earth, and the city of Buenos Aires is wiped off the map.

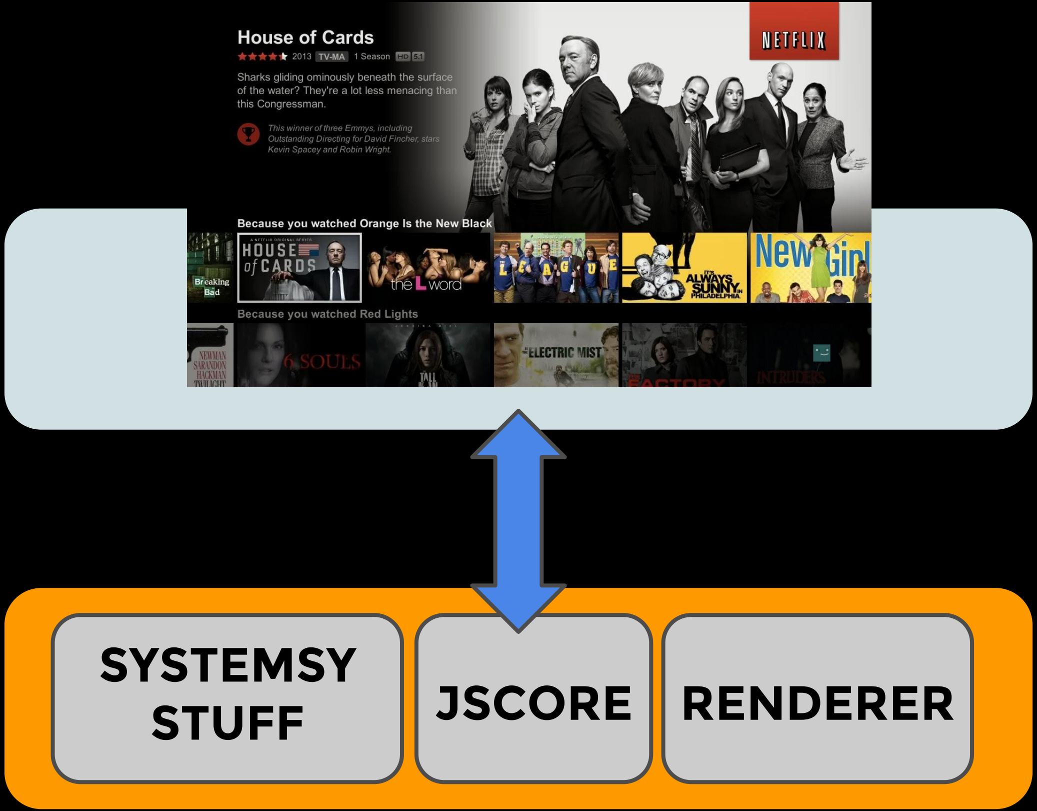
Cast:	Casper Van Dien, Dina Meyer...
Categories:	Sci-Fi & Fantasy, Action Sci-Fi...
Director:	Paul Verhoeven

Do we really need the entire DOM and all of its baggage?


UI
SYSTEMSY STUFF
JSCORE
RENDERER



(kinda like Node.js, but with a high-performance renderer)


House of Cards

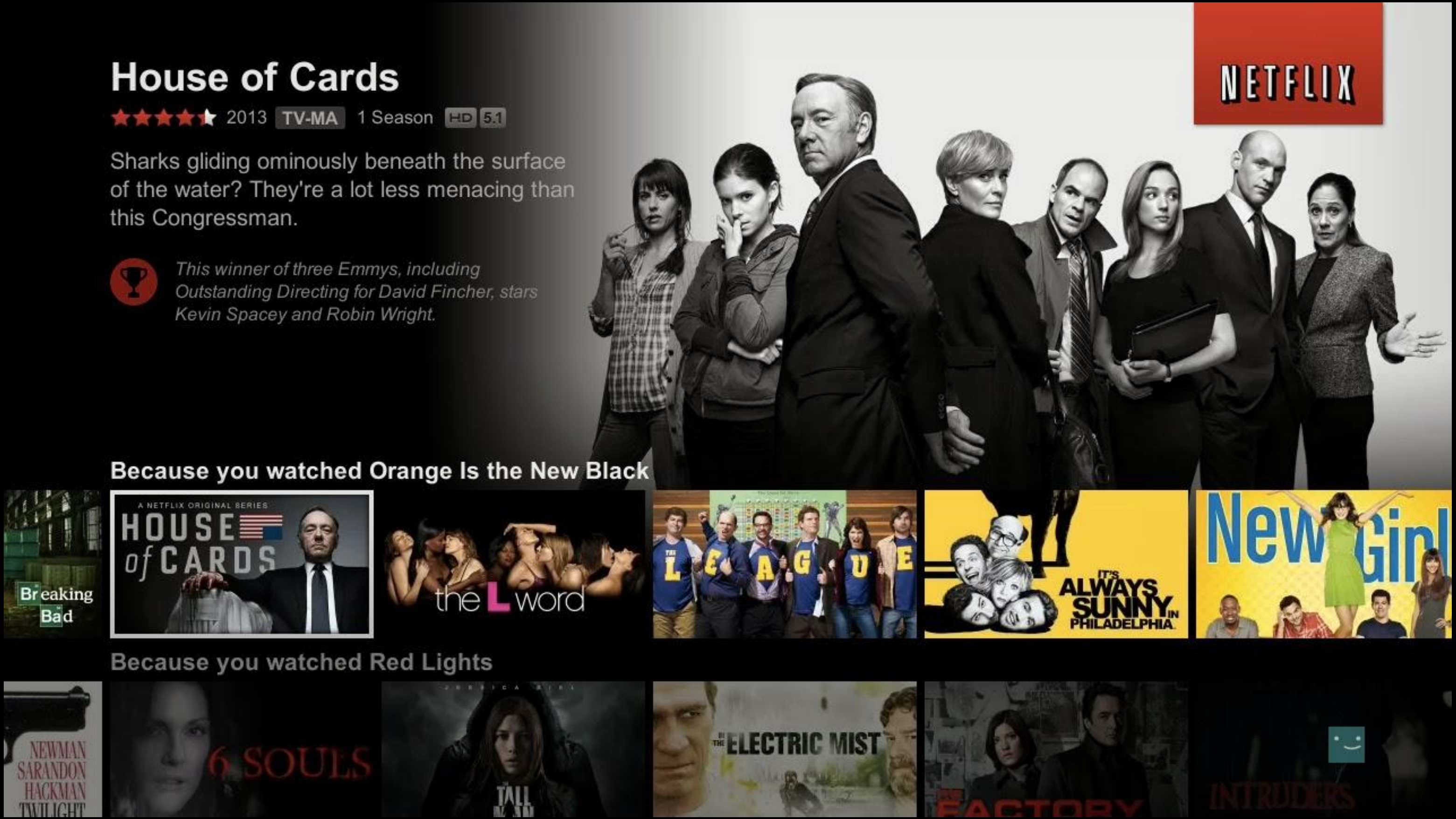
2013 TV-MA 1 Season HD 5.1

Sharks gliding ominously beneath the surface of the water? They're a lot less menacing than this Congressman.

This winner of three Emmys, including Outstanding Directing for David Fincher, stars Kevin Spacey and Robin Wright.

Because you watched Orange Is the New Black

Because you watched Red Lights


House of Cards

2013 TV-MA 1 Season HD 5.1

Sharks gliding ominously beneath the surface of the water? They're a lot less menacing than this Congressman.

This winner of three Emmys, including Outstanding Directing for David Fincher, stars Kevin Spacey and Robin Wright.

Because you watched Orange Is the New Black

Because you watched Red Lights

Device UI, evolved (Nel, Netflix Techblog 11/2013)

lets talk about that systemsy stuff

Video Decoding & Playback (naturally)

Networking

Logging

Crypto & Security

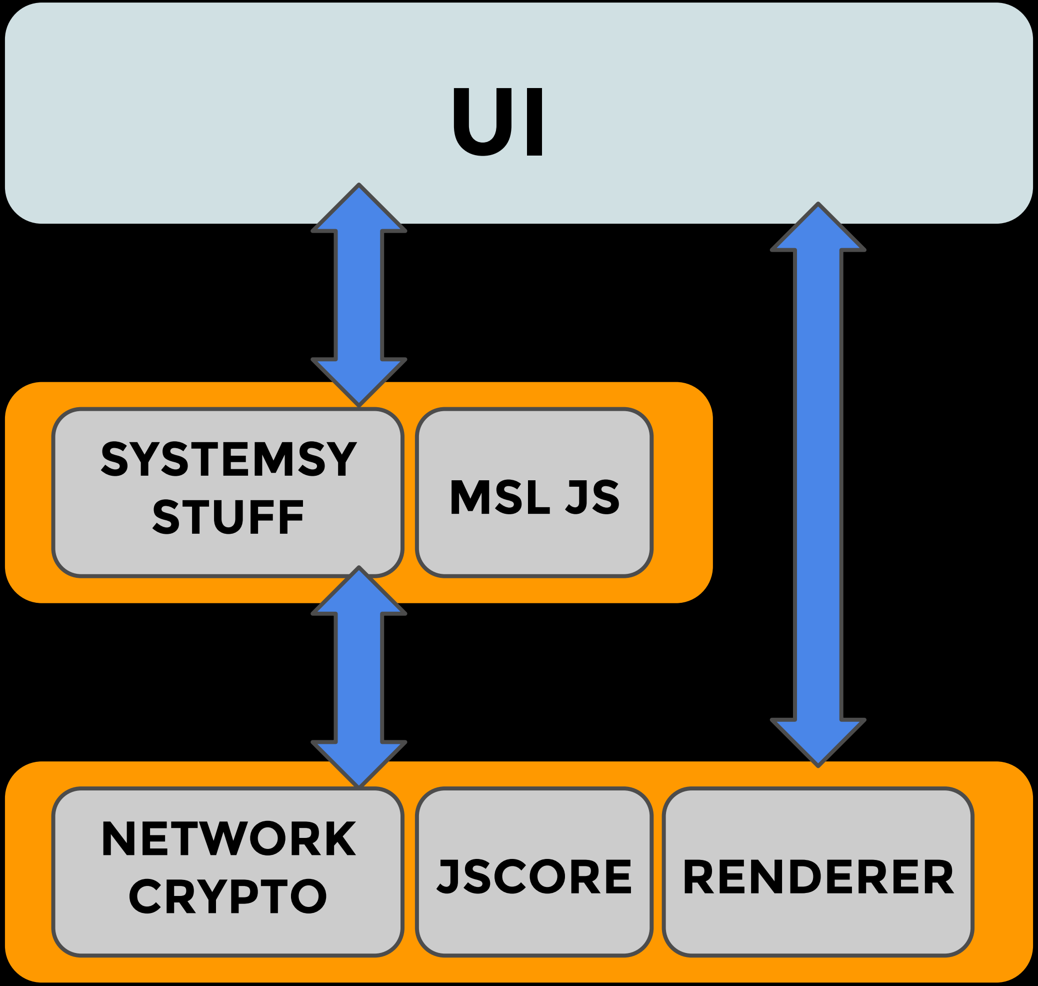
Content-Control and Caching

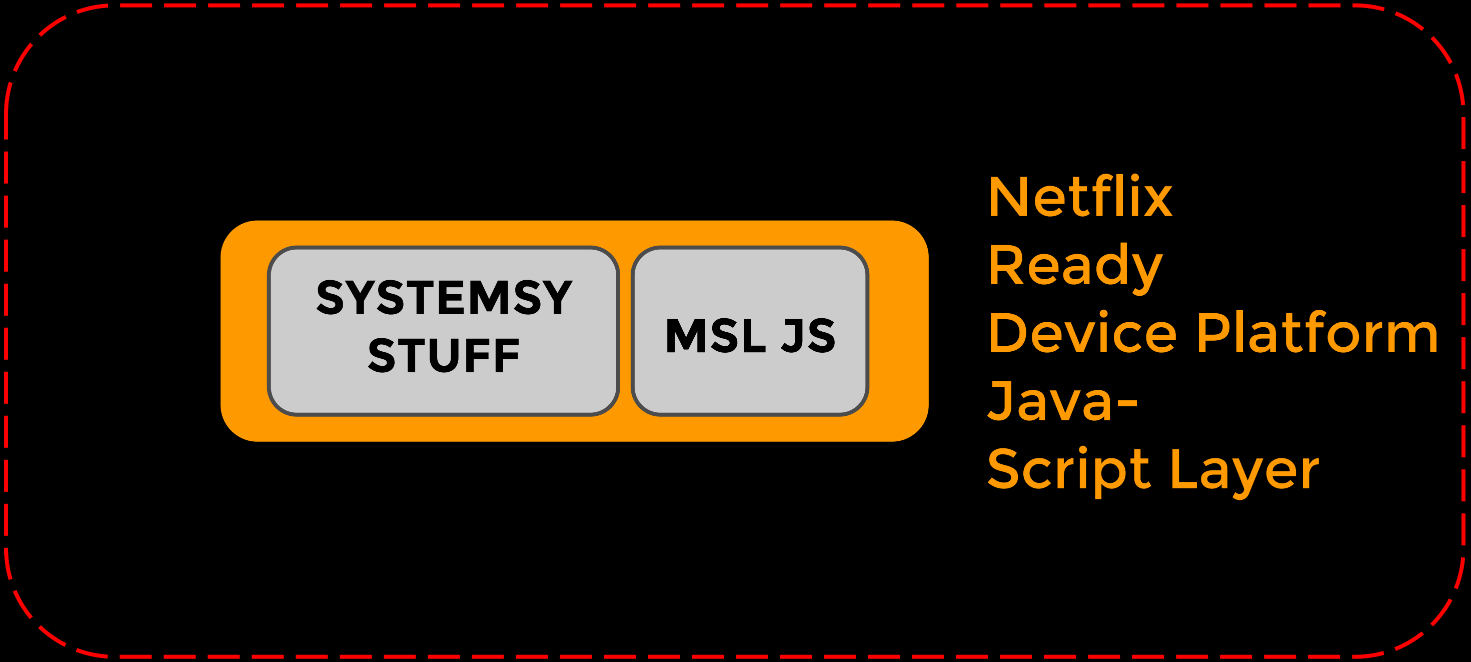
Adaptive Streaming

lets talk about that systemsy stuff

Video Decoding & Playback (naturally)
Networking
Logging
Crypto & Security
Content-Control and Caching
Adaptive Streaming

Can we move non-performance critical stuff to JS so it's updateable and we can experiment with it?


UI
SYSTEMSY STUFF
MSL JS
NETWORK CRYPTO
JSCORE
RENDERER


SYSTEMSY STUFF
MSL JS
Netflix Ready Device Platform Java-Script Layer

Translating a bunch of C++ into JavaScript? What could possibly go wrong?

Quick, what's wrong with this?

```
(function main () {
    var videoMgr,
        subtitleMgr
        audioMgr;

    //... rest of program contained here
}());
```

phew, dodged a bullet!

```
(function main () {
    var videoMgr,      // forgot a comma
        subtitleMgr, // so audioMgr was
        audioMgr;      // a global variable

    //... rest of program contained here
}());
```

but...did we really fix anything?

```
(function main () {
    var videoMgr,      // forgot a comma
        subtitleMgr, // so audioMgr was
        audioMgr;      // a global variable

    //... rest of program contained here
}());
```

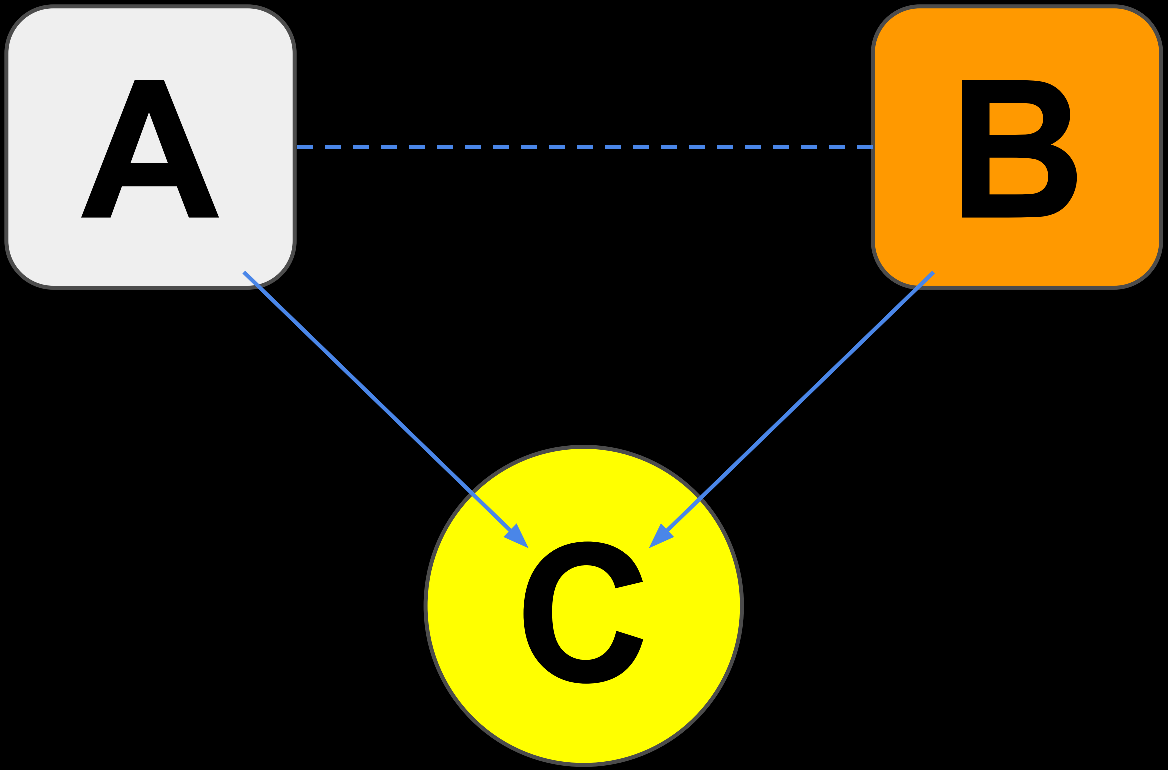
how long is this program?

```
(function main () {
    var videoMgr,       // forgot a comma
        subtitleMgr,    // so audioMgr was
        audioMgr;       // a global variable

    //... rest of program contained here
}());
```

The problem isn't global variables.

The problem is wide-scope.



action-at-a-distance makes reasoning hard

Our code suffered from lots of problems relating to wide-scope.

We used concatenation to build our final artifact...

(using CMake, as a bonus)

our artifact was 1 giant function

```
(function main () {
    var videoMgr,      // forgot a comma
        subtitleMgr, // so audioMgr was
        audioMgr;      // a global variable

    //... rest of program contained here
}());
```

we used lots of stateful singletons

```
// video_manager.js
window.videoManager = {
    play : function () { ... };
}
```

singletons are global objects that promote action-at-a-distance

we used namespaces

```
// in foo.js
window.videoManager.play();
```

not necessarily bad, unless your namespaced object stores state

we used privacy by convention

```
window.videoManager = {
    // public
    play : function () { /*...*/ },

    // private
    _calcOffset : function () { /*...*/ }
}
```

they are still available to more subsystems than are necessary

we didn't have unit-tests

```
function play () {
    window.logger.warn("doing random stuff");

    // start managers
    window.videoManager.play();
    window.audioManager.play();
    window.subtitleManager.play();
}
```

it's really hard to mock out global state

code sharing was impossible

"The problem with object-oriented languages is they've got all this implicit environment that they carry around with them. You wanted a banana but what you got was a gorilla holding the banana and the entire jungle."

-Joe Armstrong

these are all anti-patterns relating to wide-scope.

how does this happen?

Modern programming abstractions are designed around the idea that data and functionality should only be accessible by the constructs that absolutely require them.

In the past, JavaScript lacked "familiar" language primitives that support hiding data and functionality.

(hint: no classes!)

JavaScript Developers have evolved to use the module as the preferred approach to limiting scope.

JavaScript Developers have evolved to use the module as the preferred approach to limiting scope.

what's a module?

modules, CommonJS style

```
var videoMgr = {}

videoMgr.play = function play () {/*...*/}

function _calcOffset() { /*...*/ };

module.exports = videoMgr;
```

modules, CommonJS style

```
var videoMgr = require("./videoMgr.js");
```

How does modular programming relate to more "familiar" abstractions?

Modular Programming is a superset of class-based Object-Oriented Programming.

(less opinionated)

exporting, class-style

```
// constructor functions
var VideoManager = function () { /* ... */ }

// public functions
VideoManager.prototype = {
    play : function () { /* ... */ }
}

module.exports = VideoManager;
```

Modular Programming is a subset of procedural programming.

(more opinionated, but only slightly)

exporting, procedural-style

```
var _ = {};

// stateless procedures
_.each = function each (list, func) { /*...*/ };
_.reduce = function reduce (list, func) { /*...*/ };

// ...
module.exports = _;
```

Benefits

independent development, less team ownership

programming by contract

programming to an interface

tools (npm!)...

So what? How did this help us?

we started migrating...

1) Grunt -- moved from CMake, built exactly the same artifact

2) Browserify -- resolve "requires", shims some node

3) Jasmine -- unit tests

our first modules…

EventEmitter (roughly modeled after the Node.js API)

Mixin (a single function to do inheritance-type stuff)

over the next year...

All new features were implemented as CJS modules...

All singleton subsystems were refactored into instance-based subsystems (moved namespaced singletons to DI)...

Single "main" entry point to our code and initialization...

Two weeks ago removed the concatenation step altogether!

game changer!

Our code became leaner, more organized, and more testable...

We started sharing more code with other teams (built an internal NPM)...

We've moved from three-week cycles to daily deployment...

take-aways

1) get your infrastructure in place (browserify or webpack)

2) start small with by exporting the API of one or two modules

3) implement new features as modules

Questions?

@semmypurewal
@NetflixUIE
@NetflixOSS