

O'REILLY®

Velocity

CONFERENCE

BUILD RESILIENT SYSTEMS AT SCALE

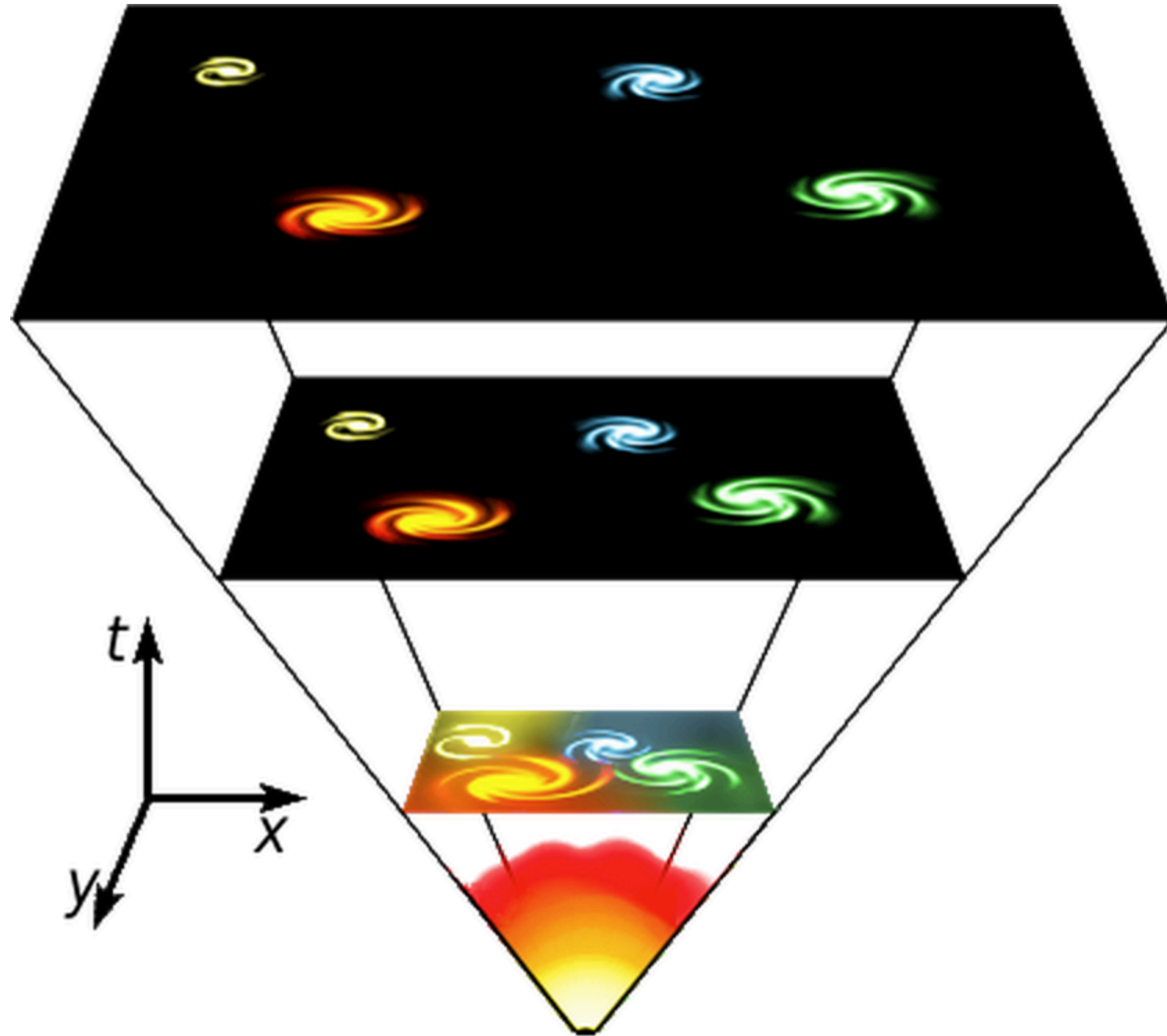
velocity.oreilly.com.cn

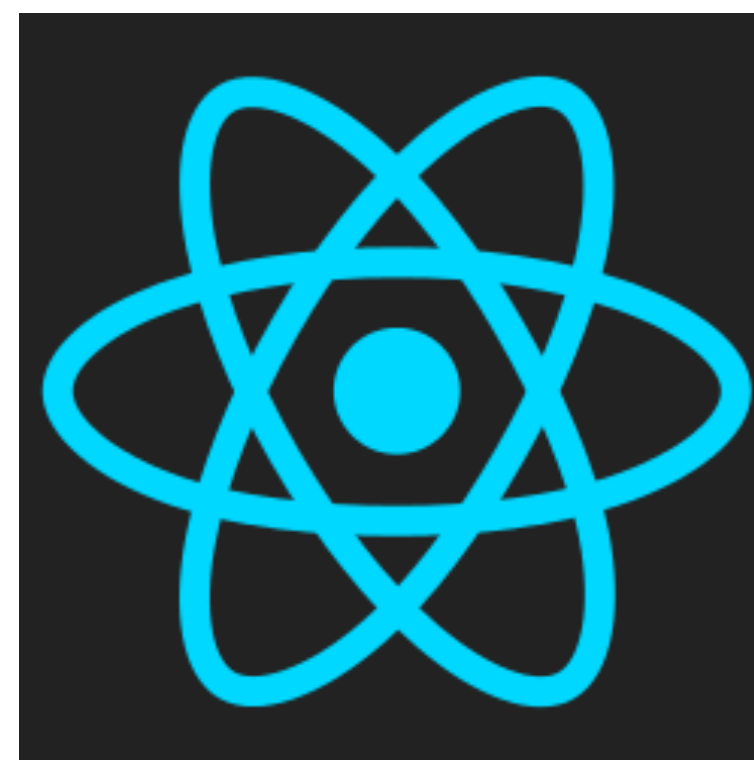
#velocityconf

Framework & Performance

框架和性能

Lingyan Zhu **YAHOO!**





Many more...



- Brought JavaScript to Server Side

So Many Choices, Which one do I Choose?

- What's important for you,
and your developers?

What are my application's main use cases?

- How much data?
- Are there are lot of interactions?
- Do I need client side navigation?
- And so on...

Easy to Learn

Easy-to-follow
Code



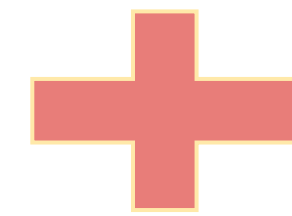
Great Docs
tutorials
scaffolding
videos
community support



Existing
Skill Set

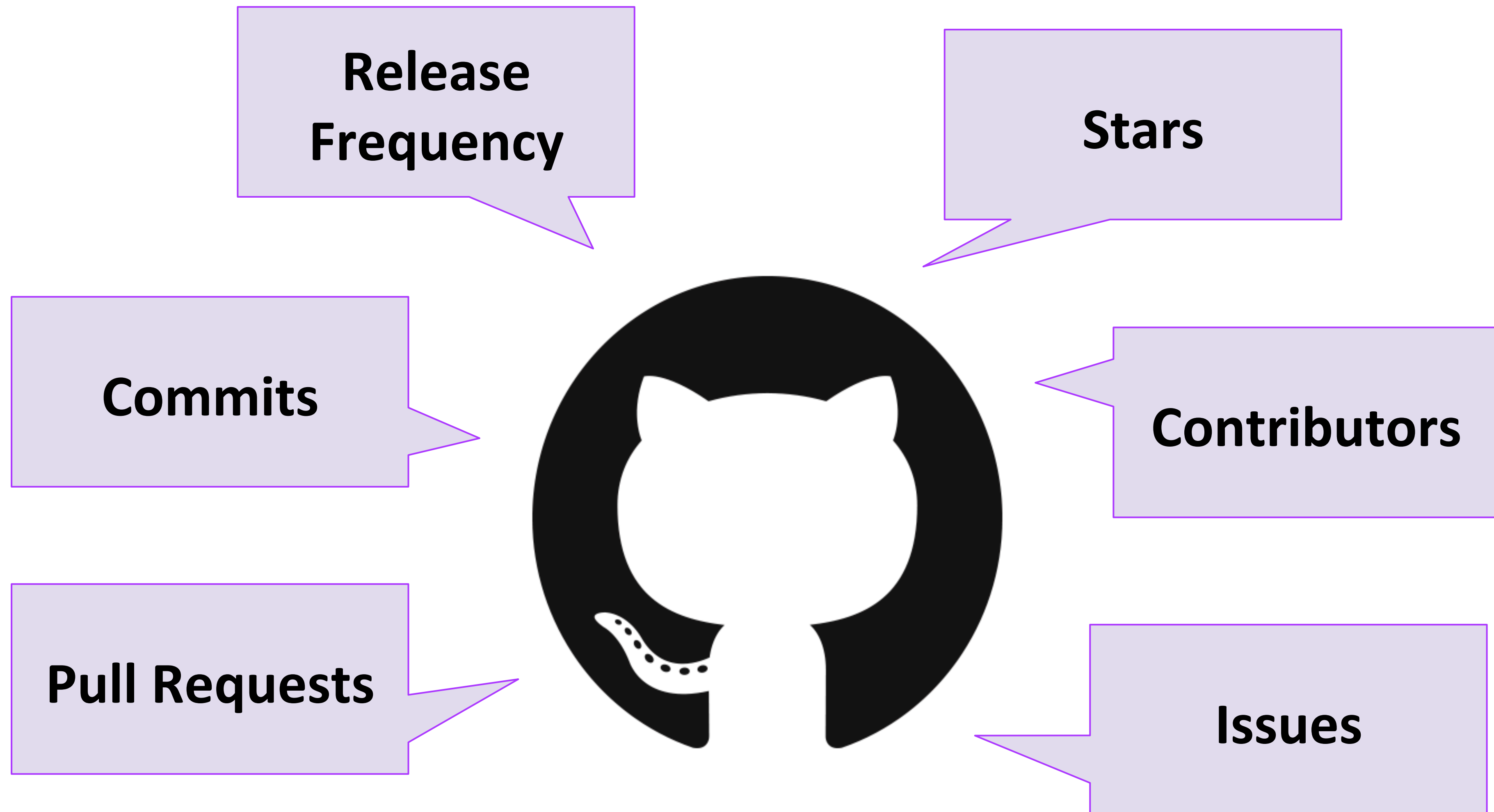
Easy to Debug

**Development
Environment**



**Real
Production
Environment**

How Active?



Stability

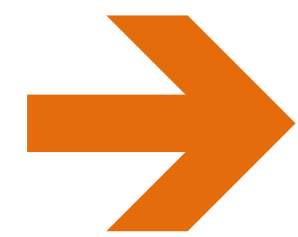
***ReadWrite:** ‘You don't buy into the premise that the Web must be broken to ensure forward progress?’*

***Yehuda Katz:** ‘No. We believe it's possible to **achieve stability without stagnation.**’*

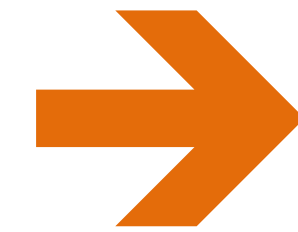
Break backward compatibility => bump the major version

Is it Performant?

**Pick most
important
use cases**



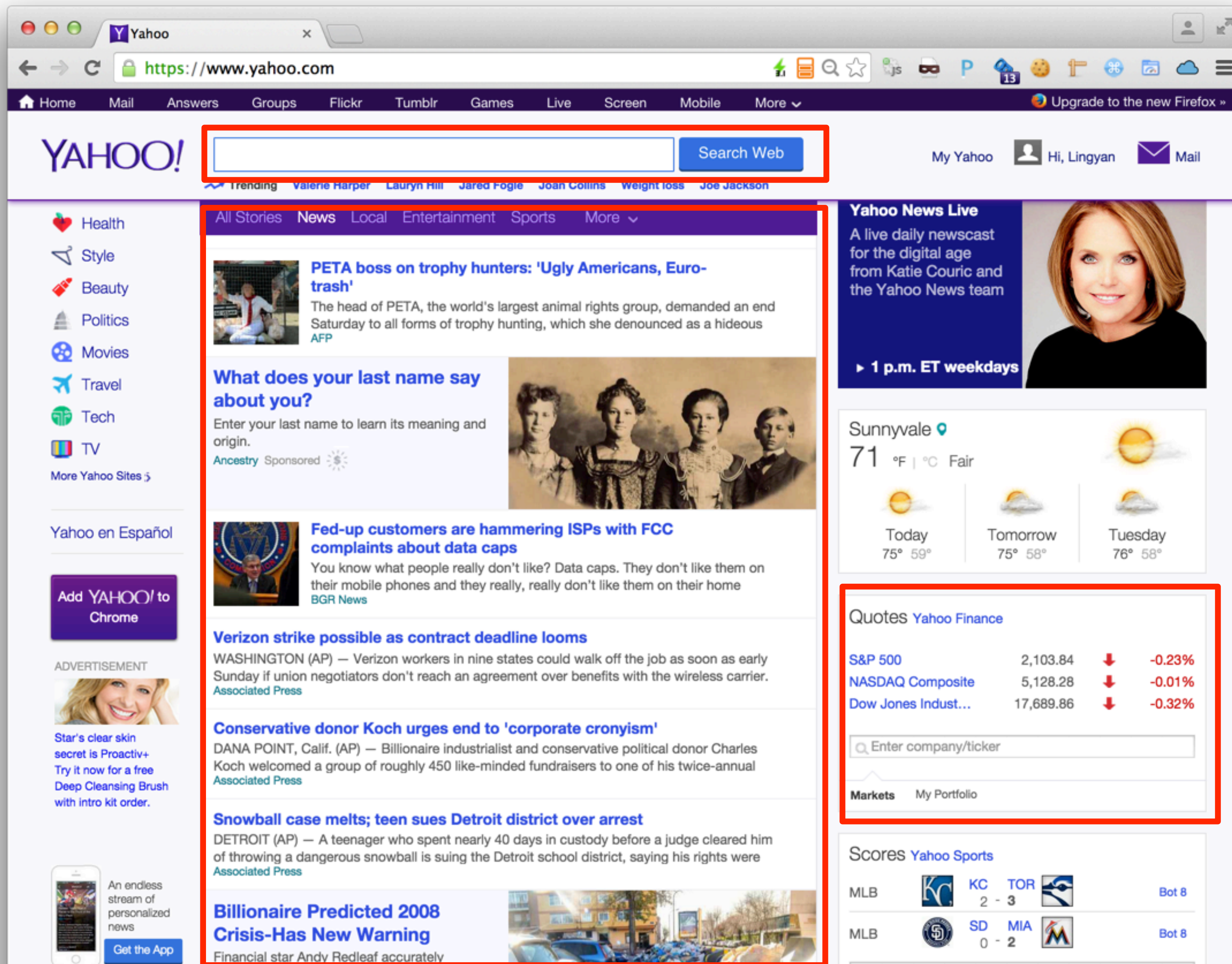
**Quickly
prototype**



Benchmark!

Don't be lazy. It will save you in the long run 😊

Yahoo! Front Page's Story



Big Scale

Super Fast
Page Serving

Reliable
Page Serving

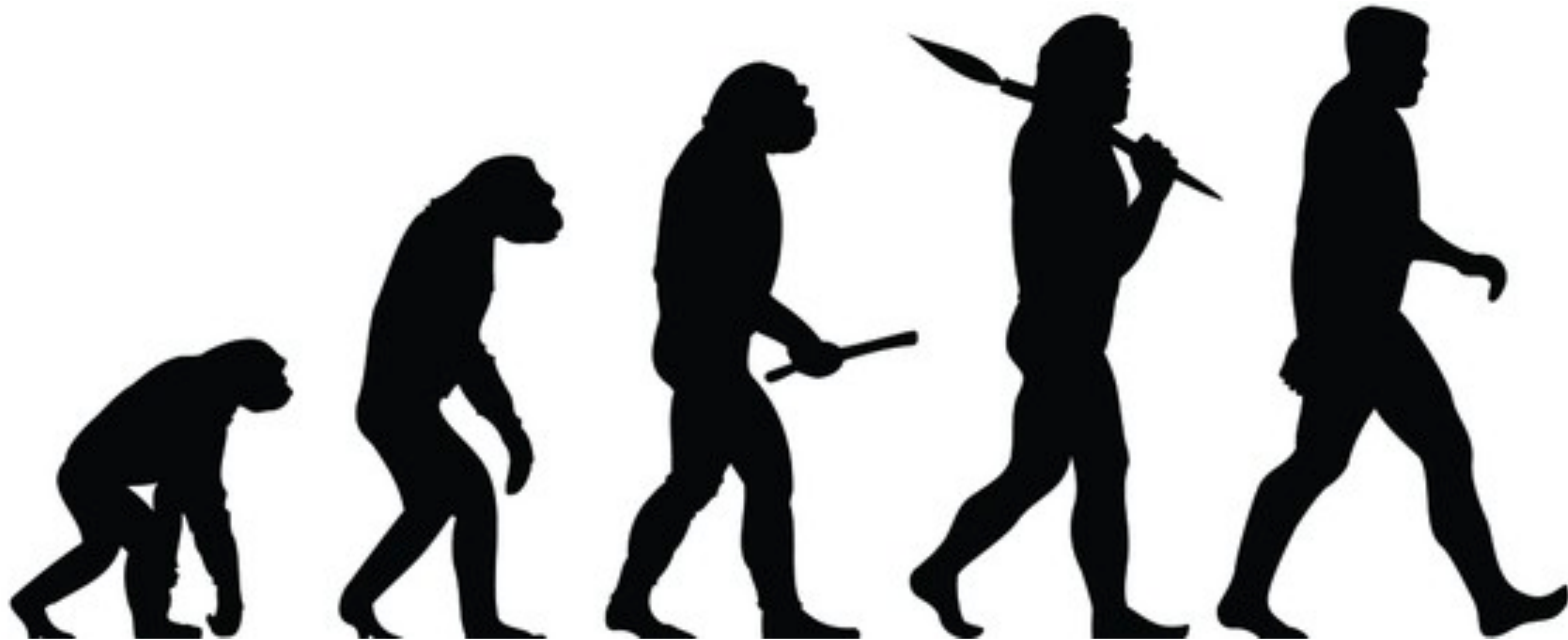
Fast Interaction

Composable
Content & Data

Live

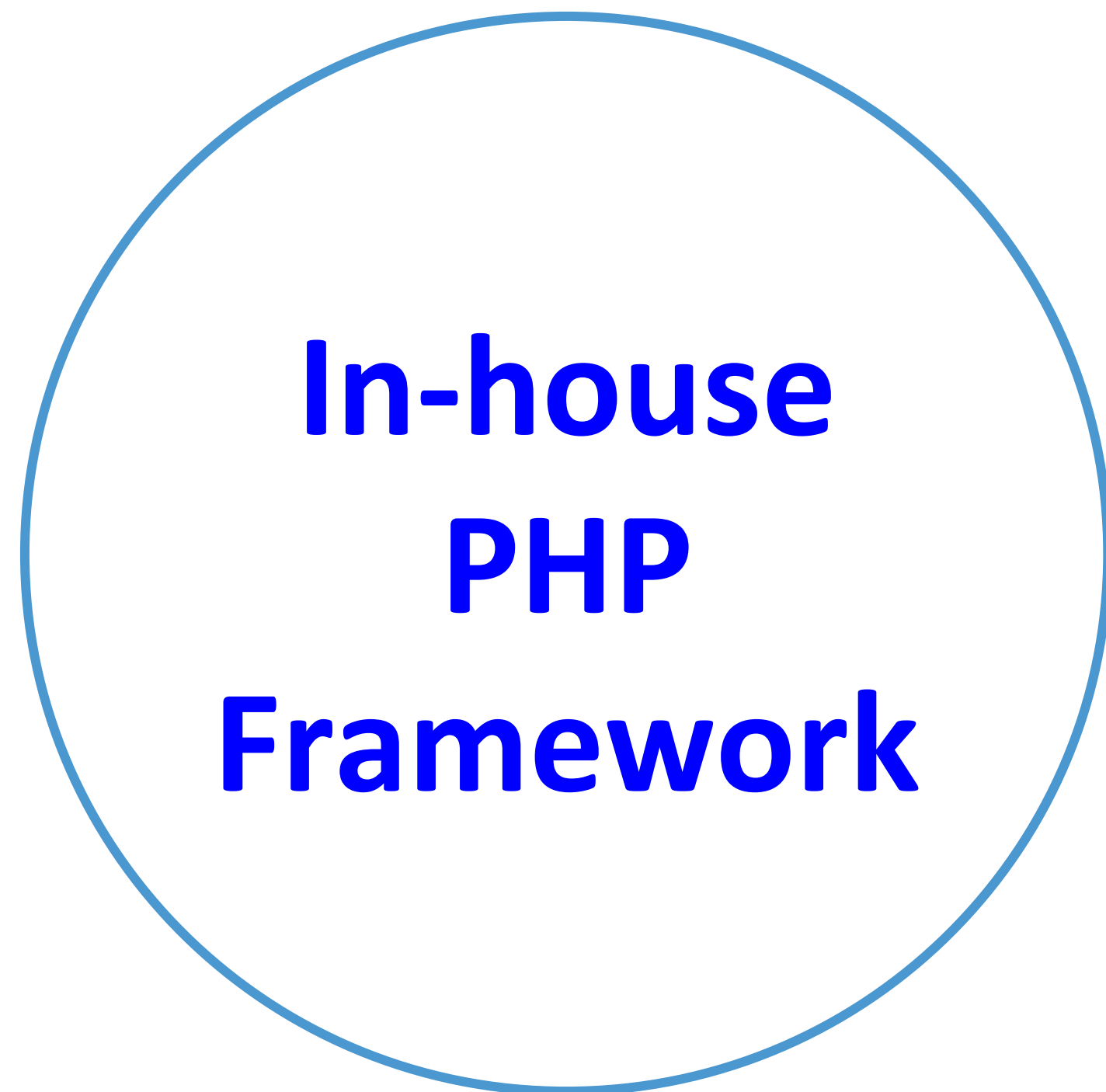
What would you pick for
a use case like this?

It's all about evolution...



Start with PHP / YUI

Server



Client



YUI abstracts away browser differences

AJAX. No MVC on the client. No data sharing between modules.

No client side navigation

No code sharing between client and server

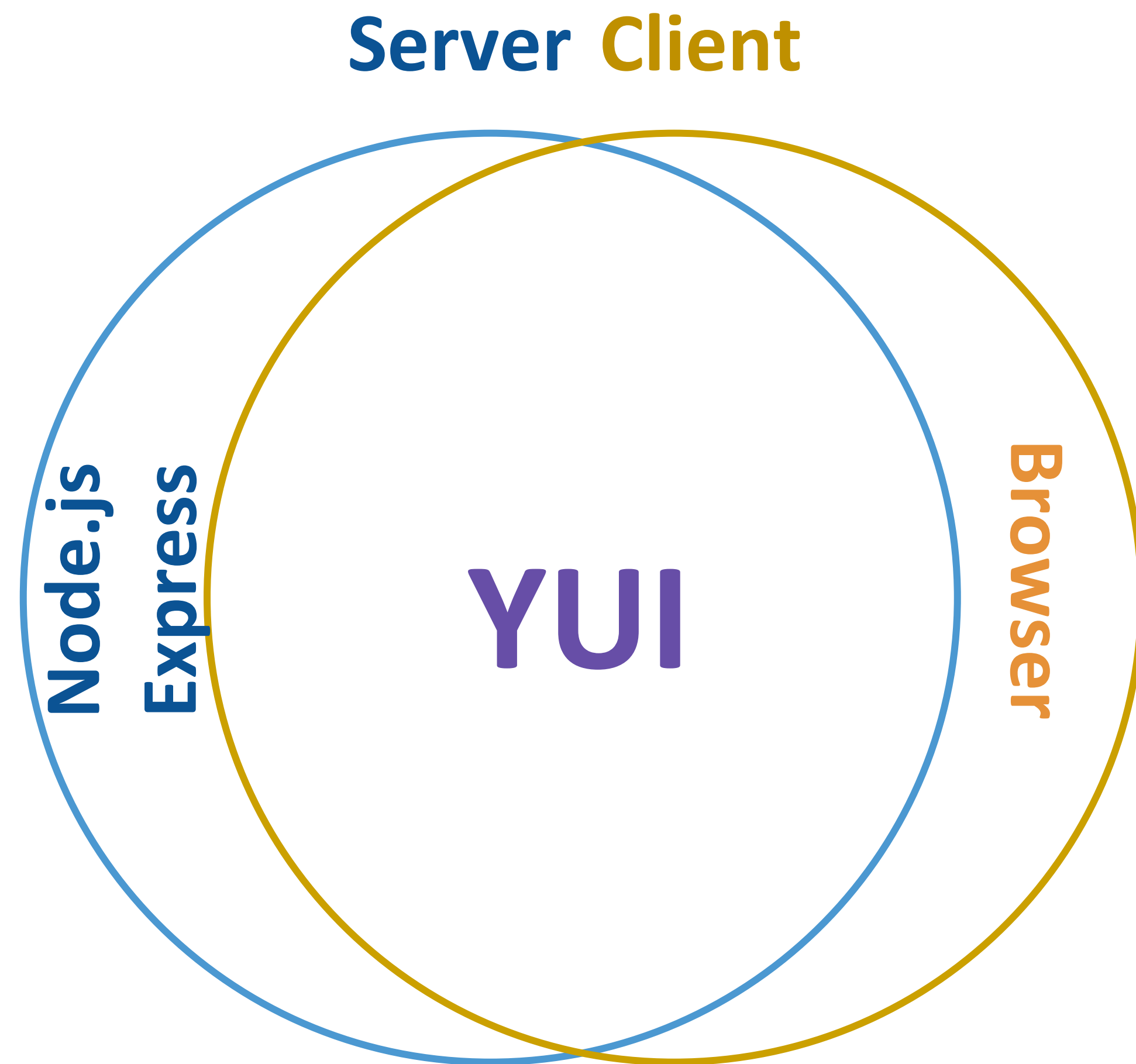
Constant context switch for developers



No More Context Switching!



First Attempt from Yahoo for Universal JS.



Device independent

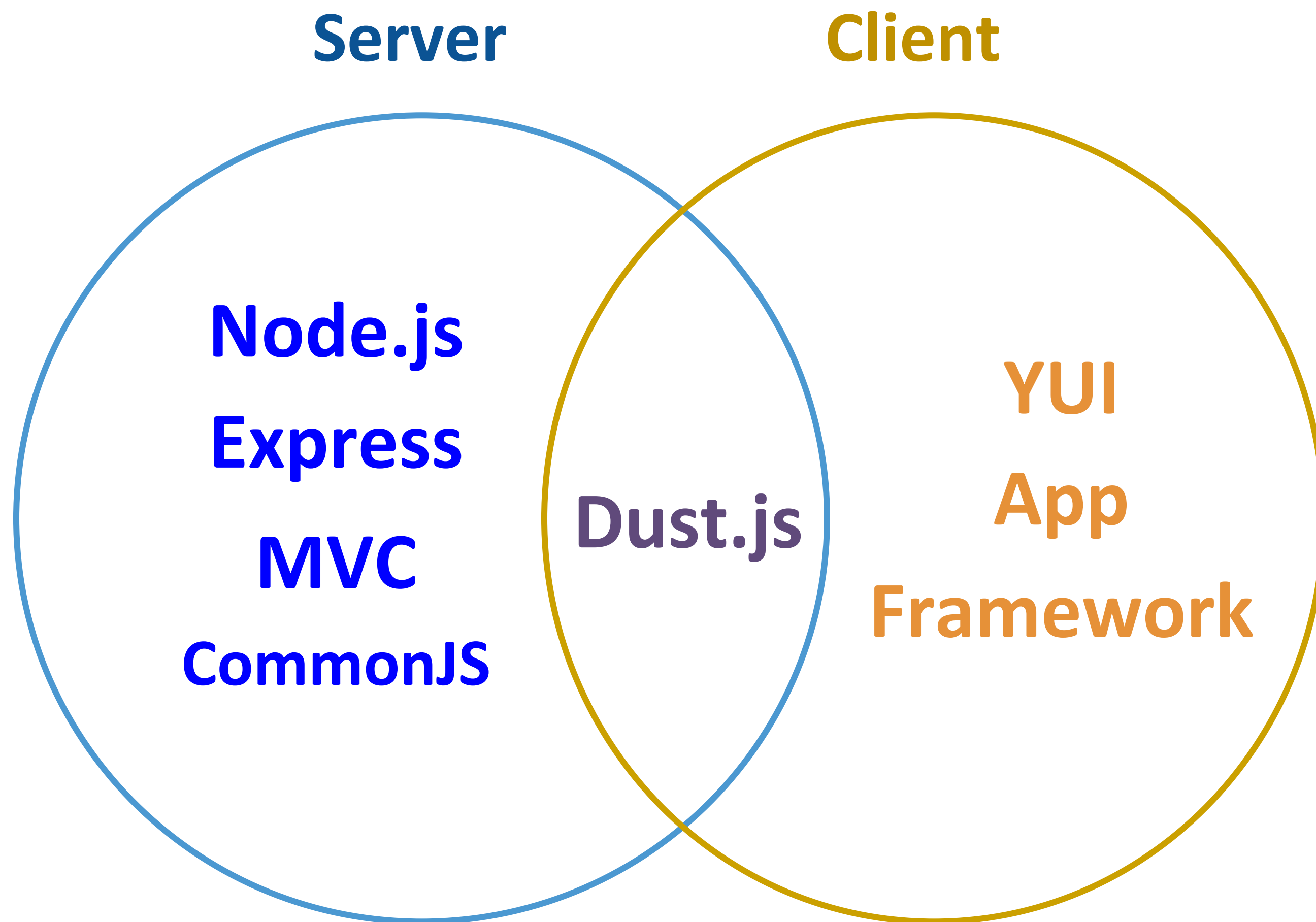
Use YUI on both server and client

Much more code sharing between client and server

Performance was not ideal. Was used by a few Yahoo products, not Front Page.

- E.g. yui node module is 23MB in size
- Run-time slowness as well

Improve Server Side First...



Only share templates between client and server

Much better server side performance

Introduced client side navigation for front page on smartphone, for sub-second content consumption

Different MVC on client and server

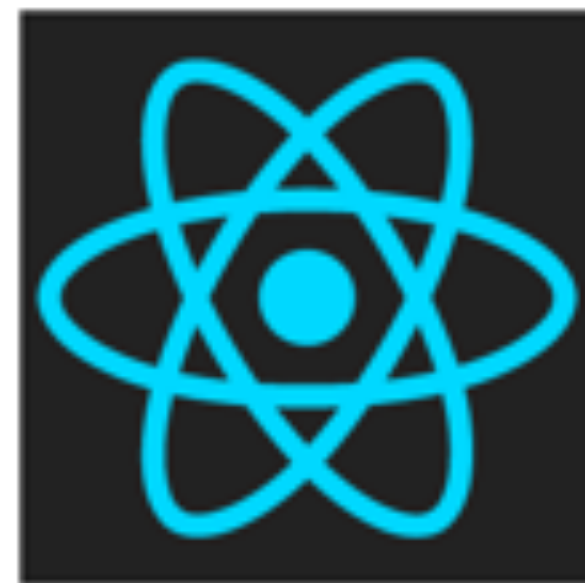
Client side performance still not ideal

Now it's time to tackle client side performance!

Stop penalizing modern browsers!

Use Plain JavaScript
& Polyfills

Need faster alternative for YUI App Framework

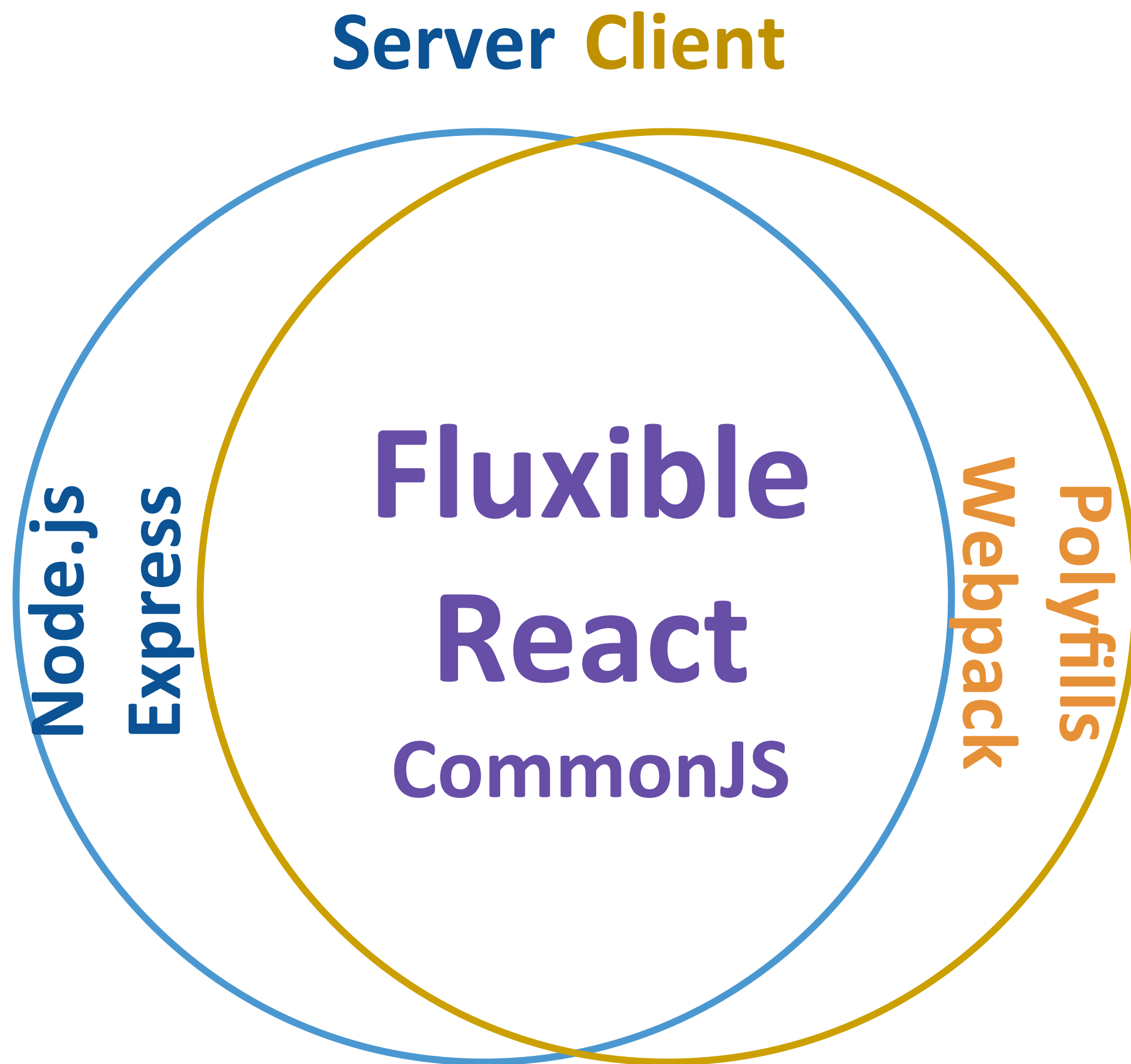


React



Flux

React & Universal Flux



Same MVC and routing logic on client and server

Better client side performance

Good server side performance, continue improving

Open source contributions:

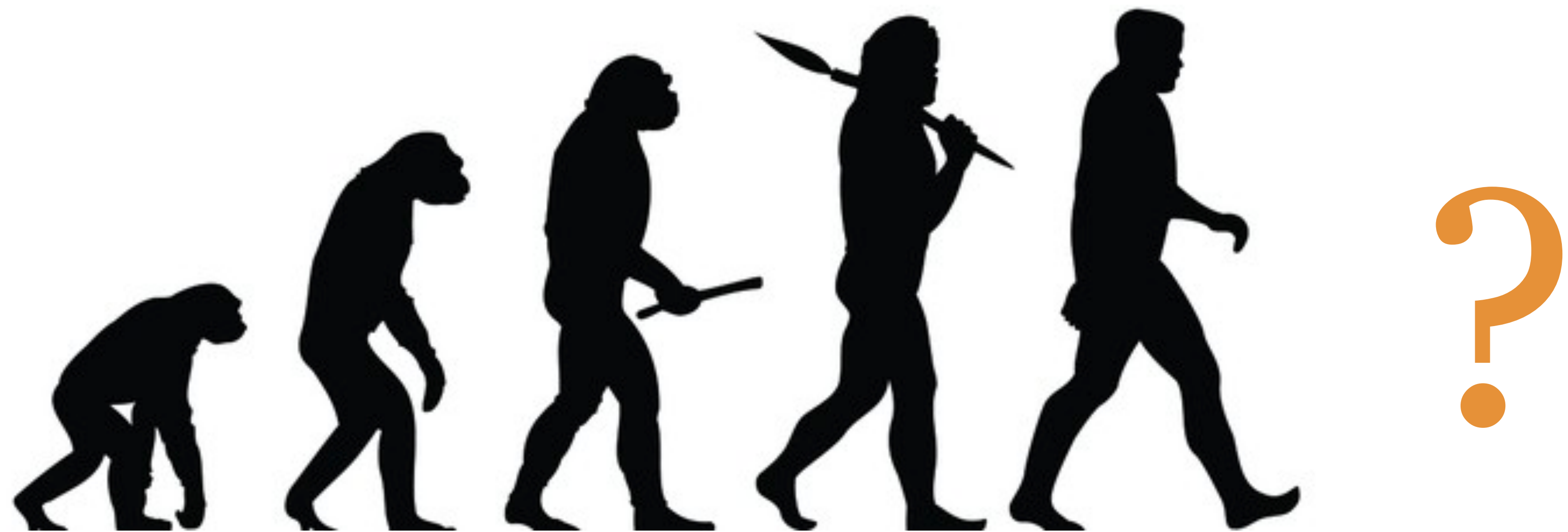
- <http://fluxible.io>
- <https://github.com/yahoo/react-i13n/>
- <https://github.com/yahoo/react-intl/>
(by our previous YUI team)

Current Status...

Already powering Yahoo sites in production:

- Taiwan Mobile Front Page: <https://tw.mobi.yahoo.com/>
- Sports Daily Fantasy App: <https://sports.yahoo.com/dailyfantasy>
- Beauty digital magazines: <https://www.yahoo.com/beauty>
- International magazines: E.g. <https://uk.celebrity.yahoo.com/>
- Search SDK App: <https://developer.yahoo.com/search-sdk/apps/>

Continue analyzing and tuning performance of this new stack to make it fully ready for US front page.



*“Don’t worry about what anybody else is going to do... **The best way to predict the future is to invent it.** Really smart people with reasonable funding can do just about anything that doesn’t violate too many of Newton’s Laws!”*

— Alan Kay in 1971, inventor of Smalltalk.

We can't predict the future,
inventing the future is not that easy...
can we ***prepare*** for it?

Do one thing and
do it well.

Break up systems into
small, straightforward
cooperating pieces.

Simplicity

Value developer
time over machine
time.

Small is beautiful.

Simplicity over
perfection.

Don't be Dogmatic...

Even the best software from smartest people has limitations.

Embrace Open Source

- Standing on the shoulders of giants

Thank you...

Lingyan Zhu

Front End Architect, **YAHOO!** Front Page



lingyan



@lingyanzhu