

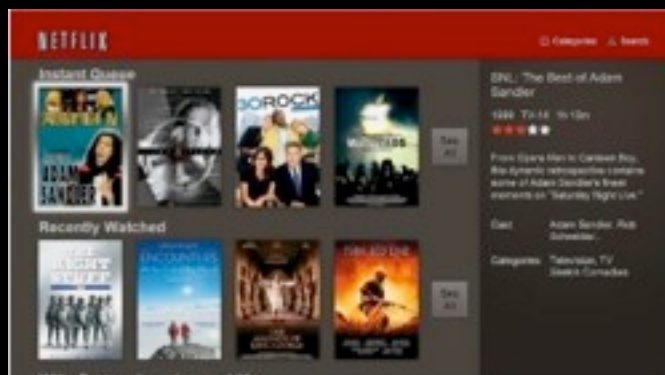
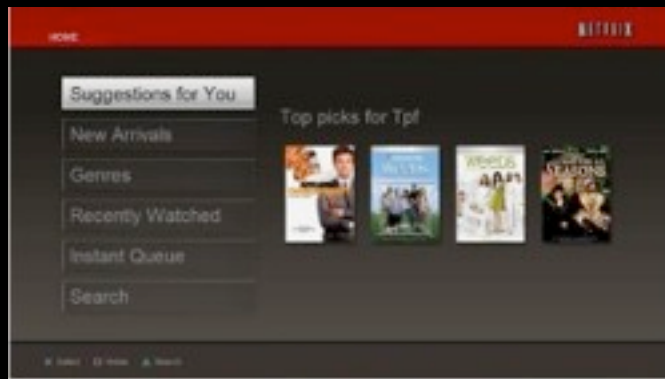


kicking up the dust with nodejs

refactoring paypal's tech stack to enable lean ux

bill scott (@billwscott)

sr. director, user interface engineering, paypal. O'Reilly Fluent webcast. May 15, 2013



at Netflix 90% or more of the “ui bits” were thrown away every year.

doesn't take too many tests to result in lots of throw away code.

followed build/test/learn



the epiphany

design for volatility



NIGHT

AND



DAY

paypal vs netflix

paypal circa 2011

roll your own.

disconnected delivery

experience. culture of

long shelf life. inward

focus. risk averse.



tangled up technology

big problem. technology and
processes not geared to
build/test/learn.

new dna @paypal

jan 2012

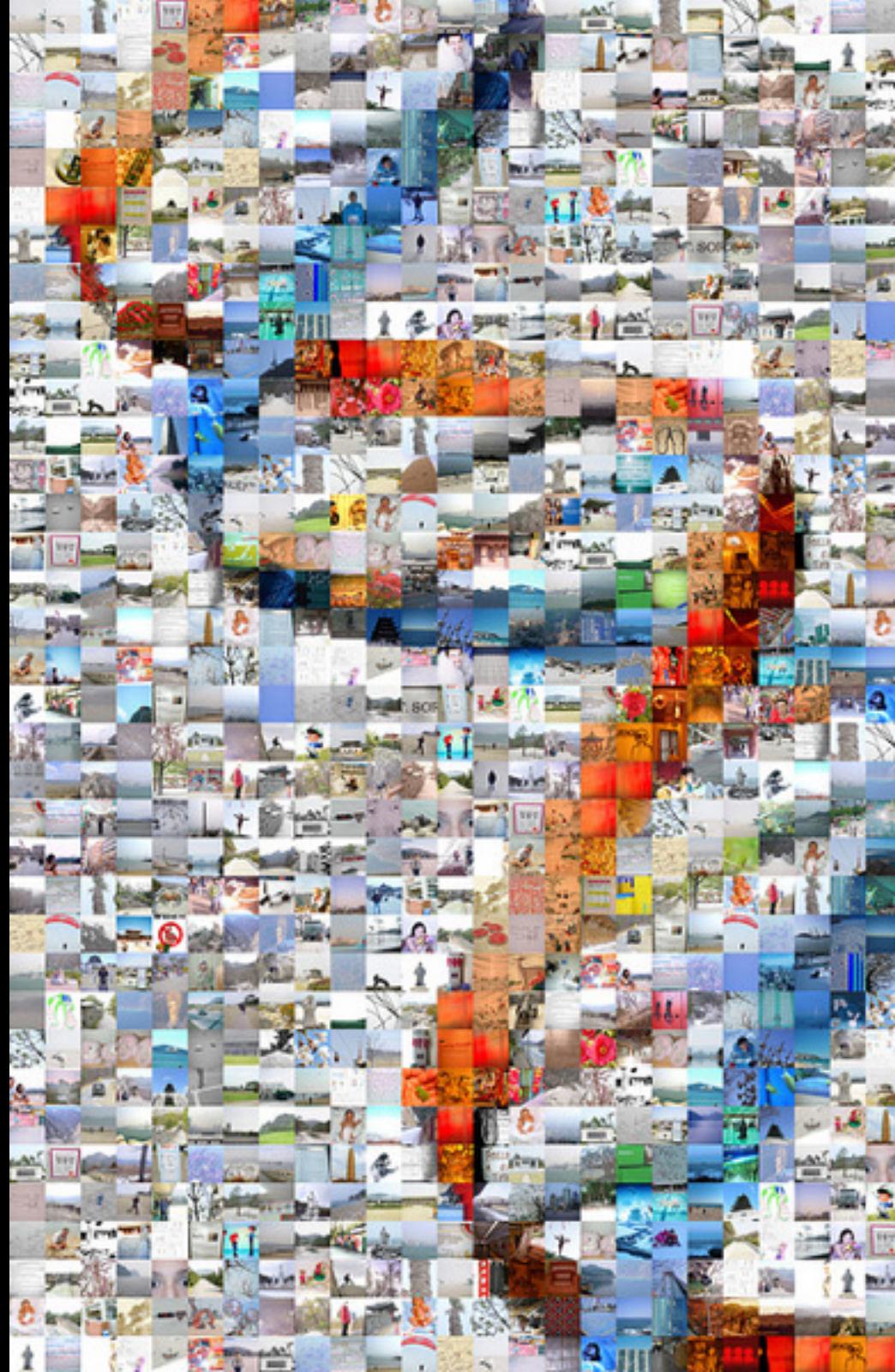
fleshed out ui layer that could
support rapid experimentation

march 2012

david Marcus becomes president
of PayPal

april 2012

formed lean ux team to reinvent
checkout experience



Introducing our new payment buttons

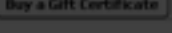
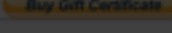
They're bold and colorful. Designed to catch the eye of customers – and increase sales.

How to get them

- If PayPal hosts your buttons, no problem. They'll be updated automatically in late October.
- If you host your own buttons, replace them with the new ones. Log in, click the Merchant Services tab, then go to the Create Buttons section. Customers will know you're a business that currently uses PayPal. This will make them feel safer when they buy – and keep them coming back.

Will this affect my operations?

No. There won't be any downtime, and there aren't any HTML changes required.

Buttons		
Name	Previous button	New button
Buy Now		
Add to Cart		
Donate		
Subscribe		
Gift Certificate		

Three little helpers...

Shopping
on Us
Pay Later*

Finish your shopping with us

[Learn About Holiday Exclusives](#)

See Terms & Conditions
*Pay Later is subject to credit approval

Buy into being safer

No matter where you shop, we'll keep your financial information private and protected.

[More about buying](#)

Sell in fewer steps

Gone are the days of waiting to get paid. Now you can request a secure payment in a few clicks.

[More about selling](#)

Transfer money to friends

Money is easier between friends when it's so simple to send them money.

[More about transferring](#)

in the midst of transformation

Sandbox

[Home](#)[Test Accounts](#)[Test Email](#)[API Credentials](#)[Test Tools](#)

Additional resources

[Documentation](#)[PayPal Developer Network](#)[Customer Support](#)

PayPal Sandbox

API Credentials

You must have credentials to test APIs for Website Payments Pro in the Sandbox. In most cases, you will use API signatures and not down

The test accounts identified below are enabled for API access.

Note: These credentials will not work outside the paypal.com to go live.

Test Account

Test Account: test_131206115

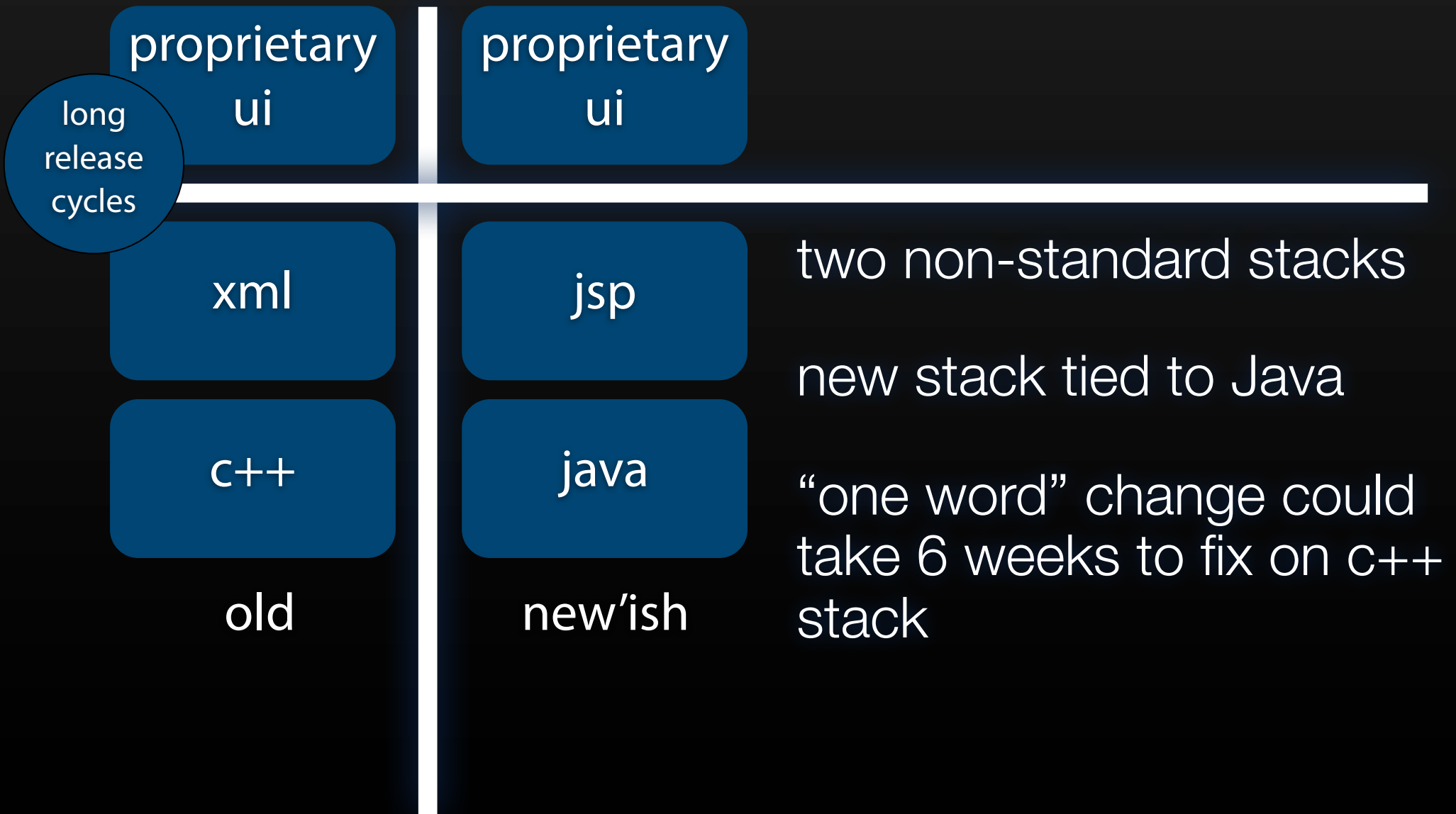
[Documentation](#)[Applications](#)[Dashboard](#)[Support](#)

// REST APIs
// Native SDKs
// Built for developers

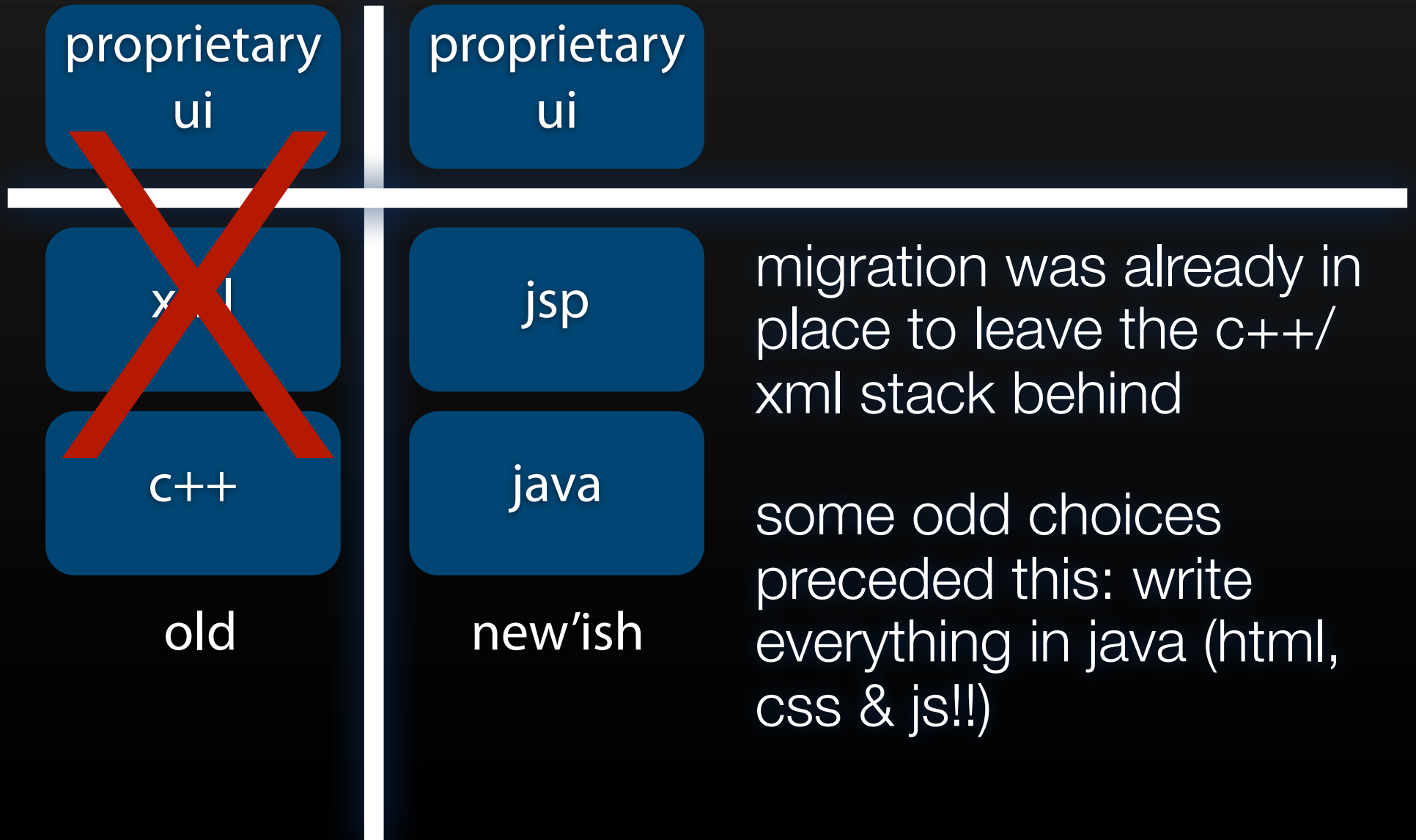
[Get Started](#)

[Sandbox?](#) Import your test accounts to continue testing. Learn [what's new](#).

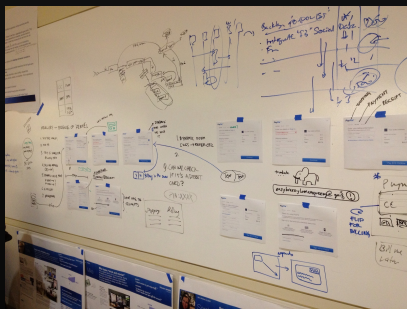
a tale of two stacks (c++ & java)



decision was to move to java



but we wanted to support lean ux enabling a living spec



product/design
team



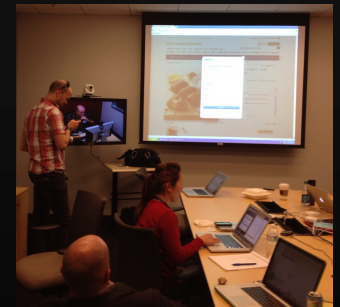
whiteboard
to code



user interface
engineers



code to usability



usability/customers

engineering stack requirements

- treat prototype & production the same
- allow rapid sketch to code life cycle
- allow quick changes during usability studies (RITE)
- support being the “living spec”

lean ux

designing products for build/
measure/learn (**lean startup**)

requires 3 rules to be followed
at all times

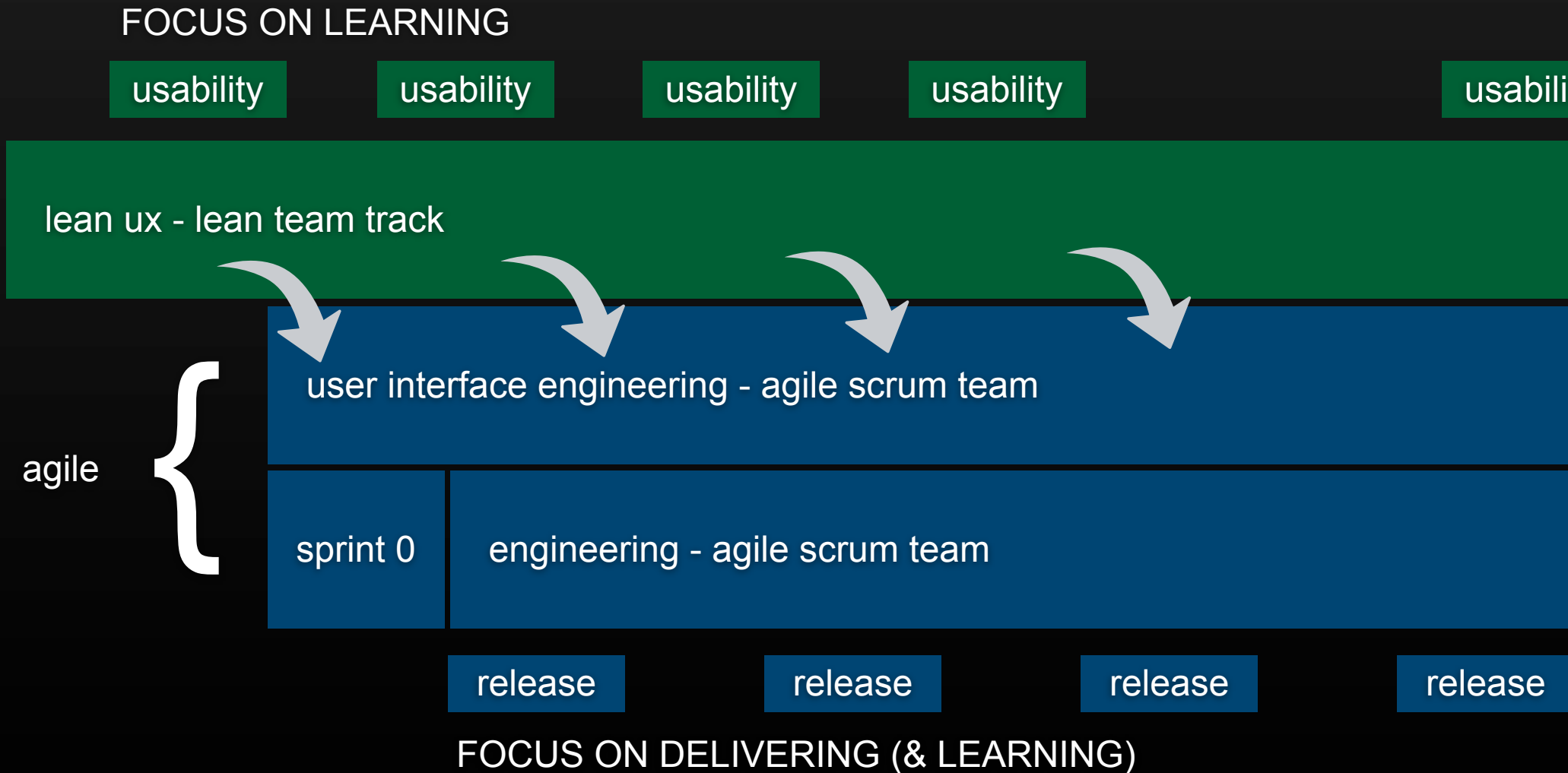
get to & maintain a **shared
understanding**

form **deep collaboration**
across disciplines

keep **continuous customer
feedback** flowing



free to iterate and drive agile



lean engineering

engineering
for learning

old stack not designed for learning

prototyping
was hard

proprietary
ui

“ui bits”
could only
live here

jsp

this new stack was not
conducive to prototyping

java

followed an “enterprise
application” model. ui
gets built into the “app”

ajax/components all
done server-side
(subclass java controller)



step 1

set the ui bits free

separate the ui bits

templates = JS
{dust}

code = JS
(backbone)

style = CSS
(less)

images

re-engineered the user interface stack so that the only artifacts are:

- javascript
- css
- images

ditched the server-side mentality to creating UIs

- no more server-side only templates
- no more server-side components
- no more server-side managing the ui

use javascript templating

templates = JS
{dust}

code = JS
(backbone)

style = CSS
(less)

images

templates get converted
to javascript

<p>Hello {name}</p>

{dust}

javascript
executed
to generate ui

JavaScript

compiles to...

we use dust.js

ui bits now just natural web artifacts

templates = JS
{dust}

code = JS
(backbone)

style = CSS
(less)

images

server-side language independent

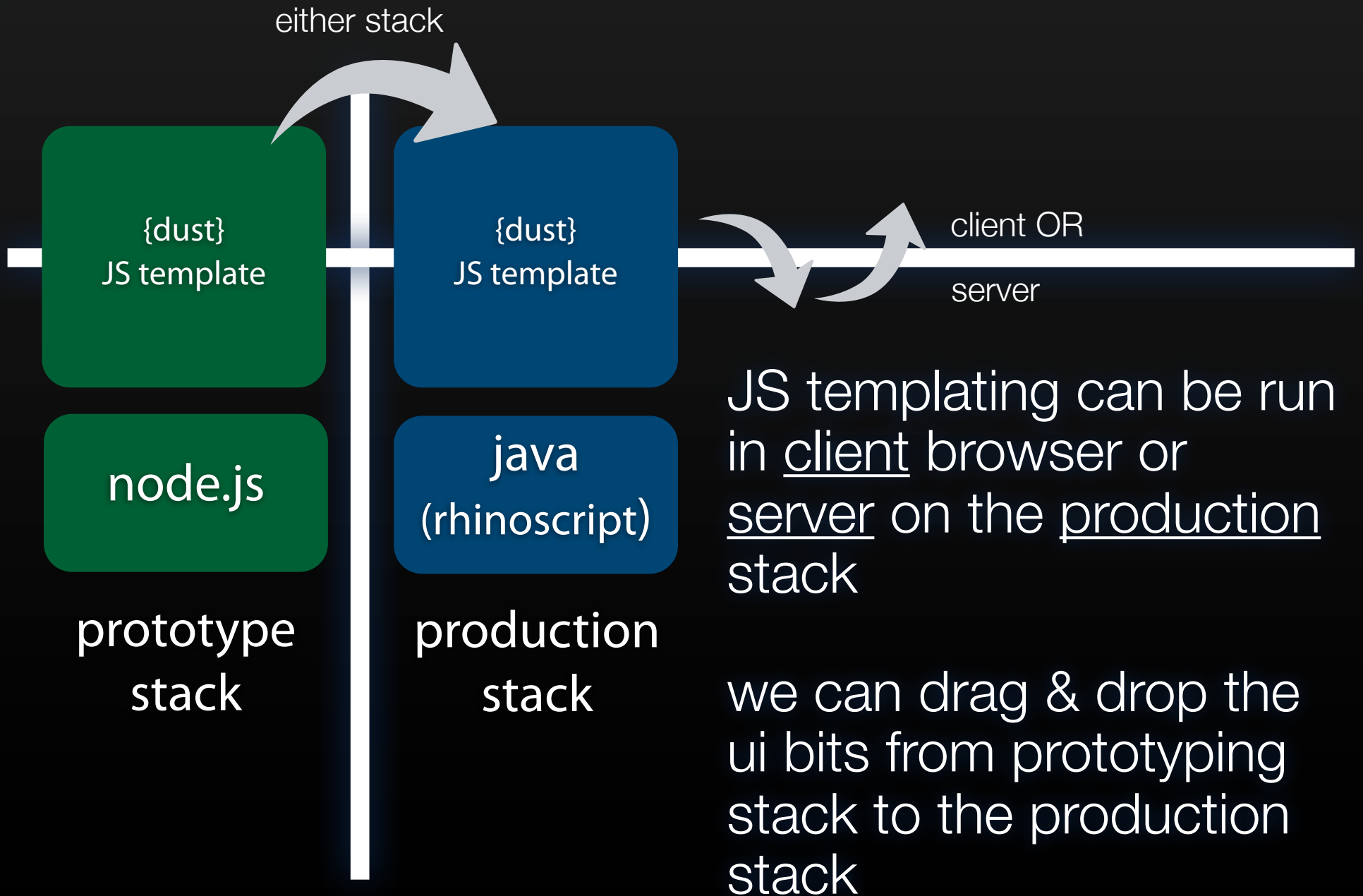
server/client agnostic

CDN ready

cacheable

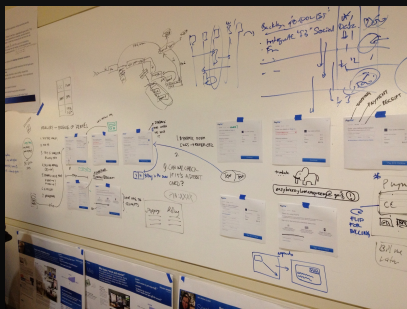
rapid to create

portable in all directions

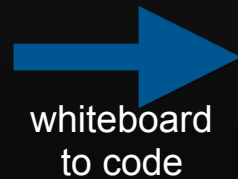


started using nodejs for proto stack

enabled rapid development (coupled with dustjs)



product/design
team



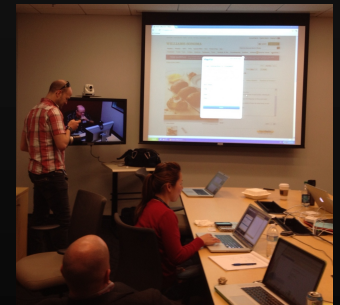
whiteboard
to code



user interface
engineers



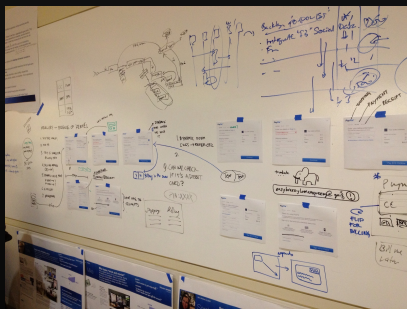
code to usability



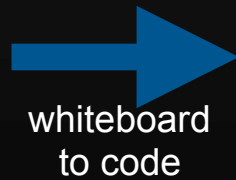
usability/customers

started using nodejs for proto stack

enabled rapid development (coupled with dustjs)



product/design
team



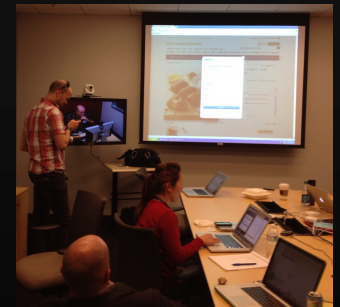
whiteboard
to code



user interface
engineers



code to usability



usability/customers

{dust}

backbone

less

images

node.js (set up as prototype stack)

free to turn sketch to code

forcing function.

brought about a close collaboration between engineering
and design

created a bridge for shared understanding

required a lot of confidence and transparency



free to test frequently with users

* DO NOT ERASE *					
	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
Co-located in ASTUTE	(all day) - Design - Content - Iterative Coding	(all day) - Design - Content - Iterative Coding	Feedback + Iterative Coding	9am Usability Session	1pm - 2:30pm Usability Review
		Hand-off	4pm Usability Delivery		3pm What to build next week
INVOLVED	UED UIG PO	* DO NOT ERASE *	* DO NOT ERASE *		



step 2

embrace open source

use open source religiously

Bootstrap, from Twitter express



BACKBONE.JS

UNDERSCORE.JS



{dust}



BOWER

work in open source model

internal github revolutionized
our internal development

rapidly replaced centralized
platform teams

innovation democratized

every developer encouraged
to experiment and generate repos
to share as well as to fork/pull request



give back to open source

we have a string of projects that will be open sourced

- node bootstrap (similar to yeoman)

- contributions to bootstrap (for accessibility)

- and more...



step 3

release the kraken

project kraken

ready nodejs for production

enable all of standard paypal
services

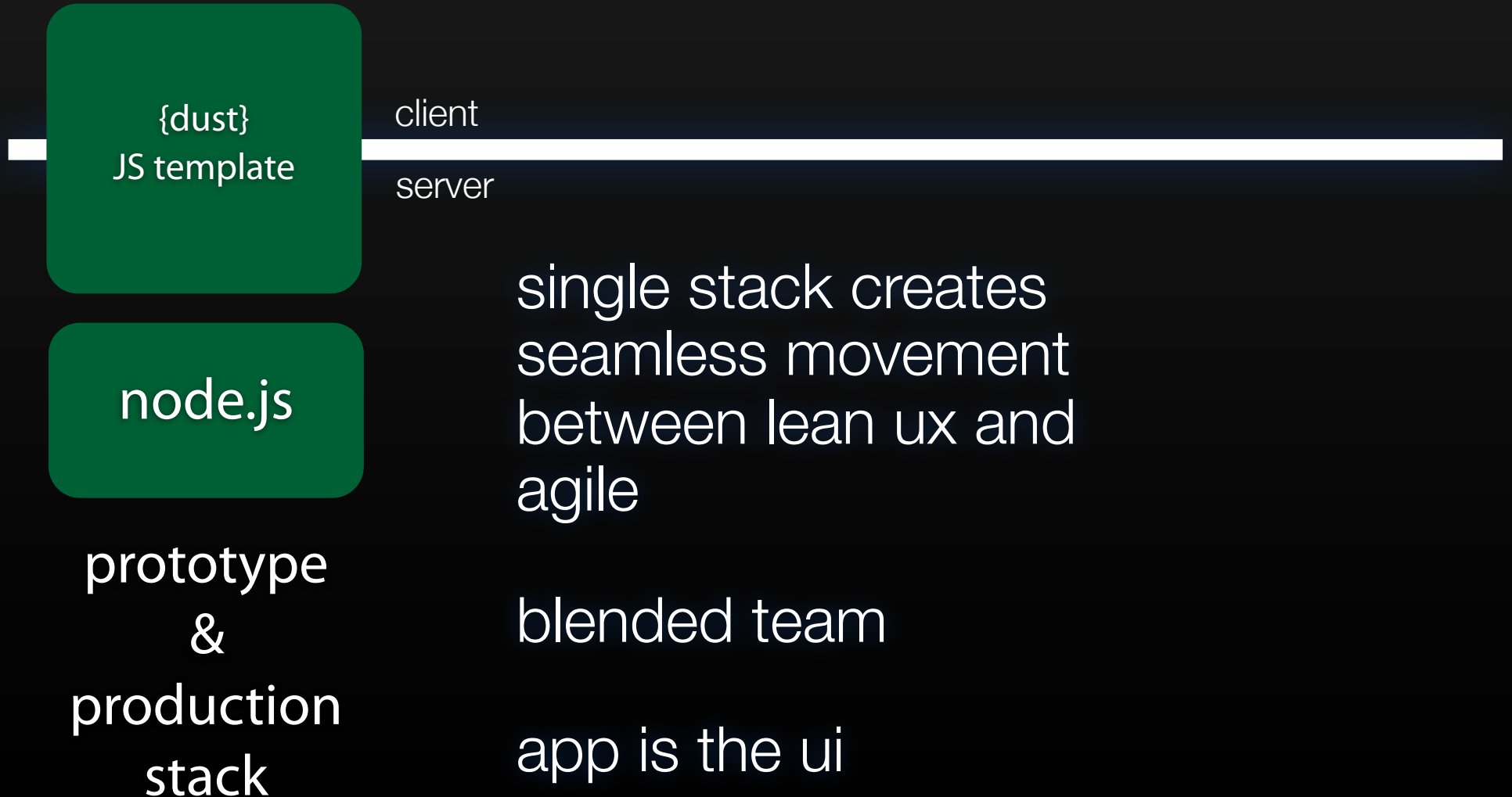
but do it in a friendly npm way

monitoring, logging, security,
content, local resolution, analytics,
authentication, dust rendering,
experimentation, packaging,
application framework, deployment,
session management, service access, etc.

mvc framework/scaffolding. hello world in 1 minute.



one stack to rule them all



introducing nodejs into **your** org

get the camel's nose under the tent

- start as an API proxy (shim) to really old services

- use as a rapid prototyping stack

move into the tent

- find a good proof of concept API or application to begin to scale

- do it in parallel and test till scales

do it with talent

- ensure best developers are on the initial work to set the standard

a few other key items

github democratizes coding & decentralizes teams

bower provides a clean way to componentize the ui

requirejs makes packaging simple and enforces modules

running internal npm service drives modularization

having a CLI for creating/extending apps is empowering

less cleans up css code

backbone provides standard client event modeling



the steps we took

set the ui free
embrace open source
release the kraken



follow me on twitter

@billwscott