

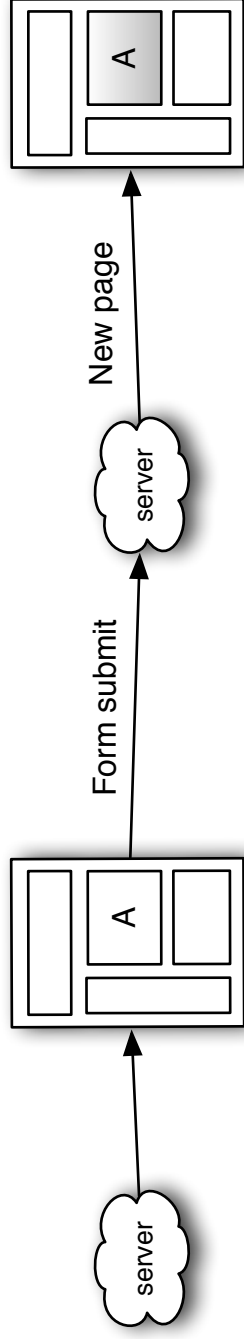
Give up your jQuery! Give up your MVC!

Hunter Blanks. Engineer, Monetate.
2010-11-13

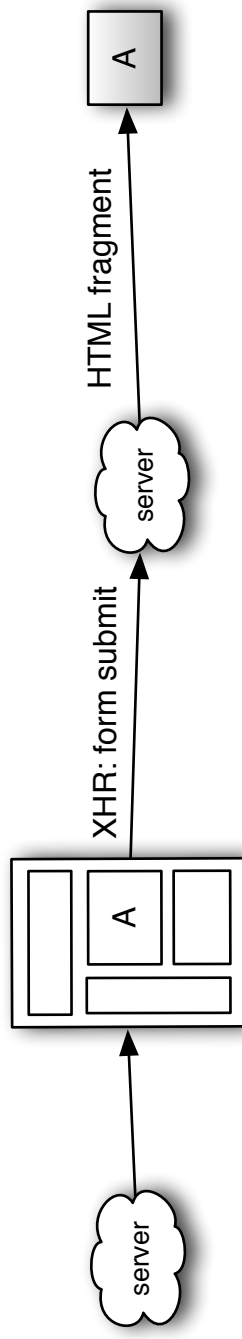
Circa... we did things like...

powered by...

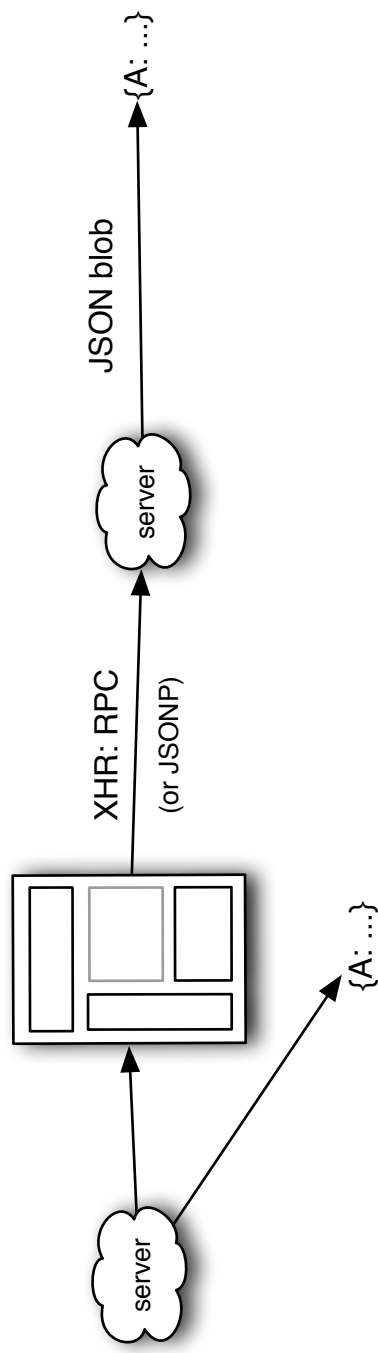
1999



2004



?



Doing this...

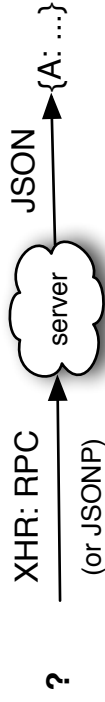
the wins are...

and the pitfalls are...



Greater interactivity
Modular templates
Simpler JS

Frequent DOM inspection
Frequent DOM event registration/dereg.
Views depend on server, but also on JS.
Controllers have business & view logic.
State is mixed between server & browser.



State is mostly in browser.
Server only has business logic.
Views are entirely in JS.
Limited/shallow DOM inspection
UI code has type/compile checks.
Easy to make event buses

State must be managed in the browser.
You're gonna need a build system.
Search engines won't parse your pages.
Extra HTTP requests

So, when is it a good idea to stay in 2004?

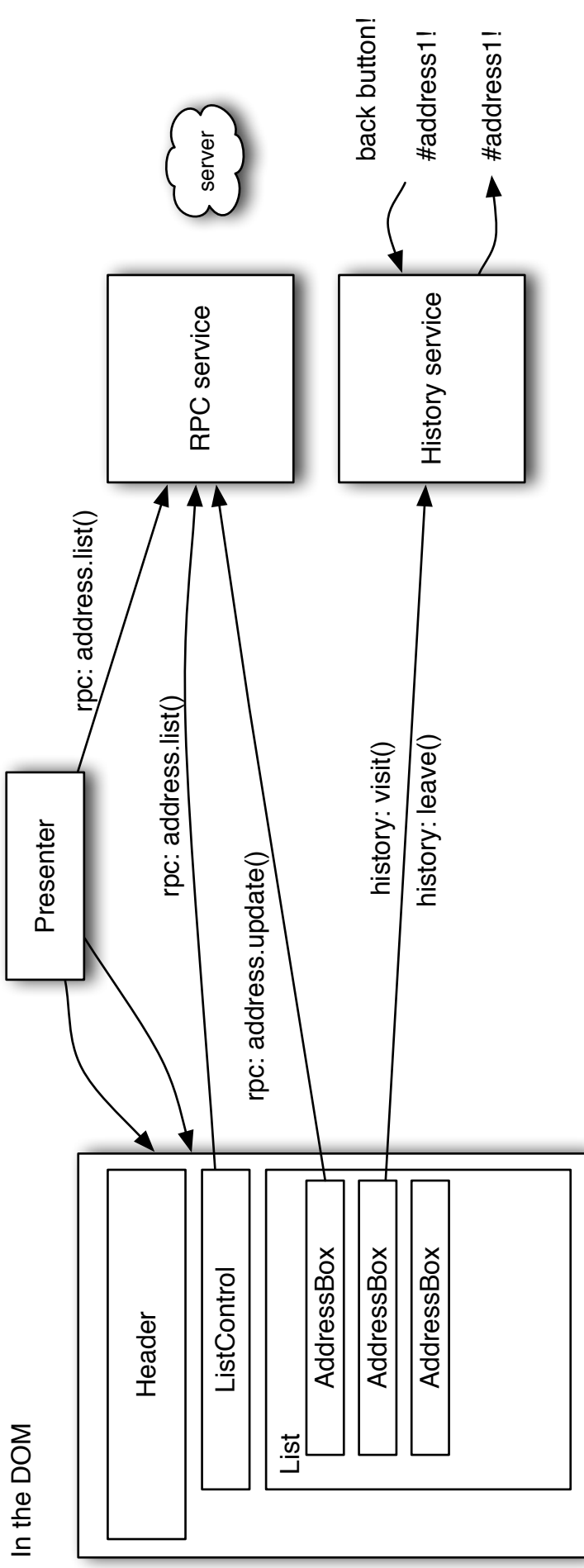
When is it a good idea to leave?

And if you do leave, how do you do it?

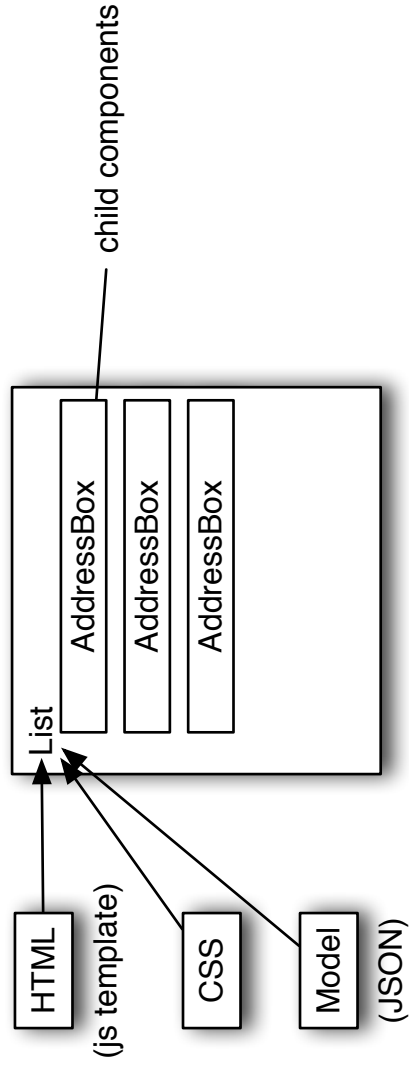
Google Web Toolkit, SproutCore, Cappuccino, RJS, Pyjamas, Closure Tools, ?

And if you do this, what might it look like in the JS?

supposing an ever-implemented address book application...



... and as a component



And if you do this, what might you want to take a look at?

Technologies:

- Google Web Toolkit
- Cappucino
- SproutCore
- Google Closure Library
- Pyjamas
- RJS?
- Other things?

Talks, books, and papers:

- Google Web Toolkit Best Practices (Ray Ryan, at Google IO)
- Closure: The Definitive Guide
- Other things?

Thank you!

hblanks@monetate.com (PS, we're hiring!)