

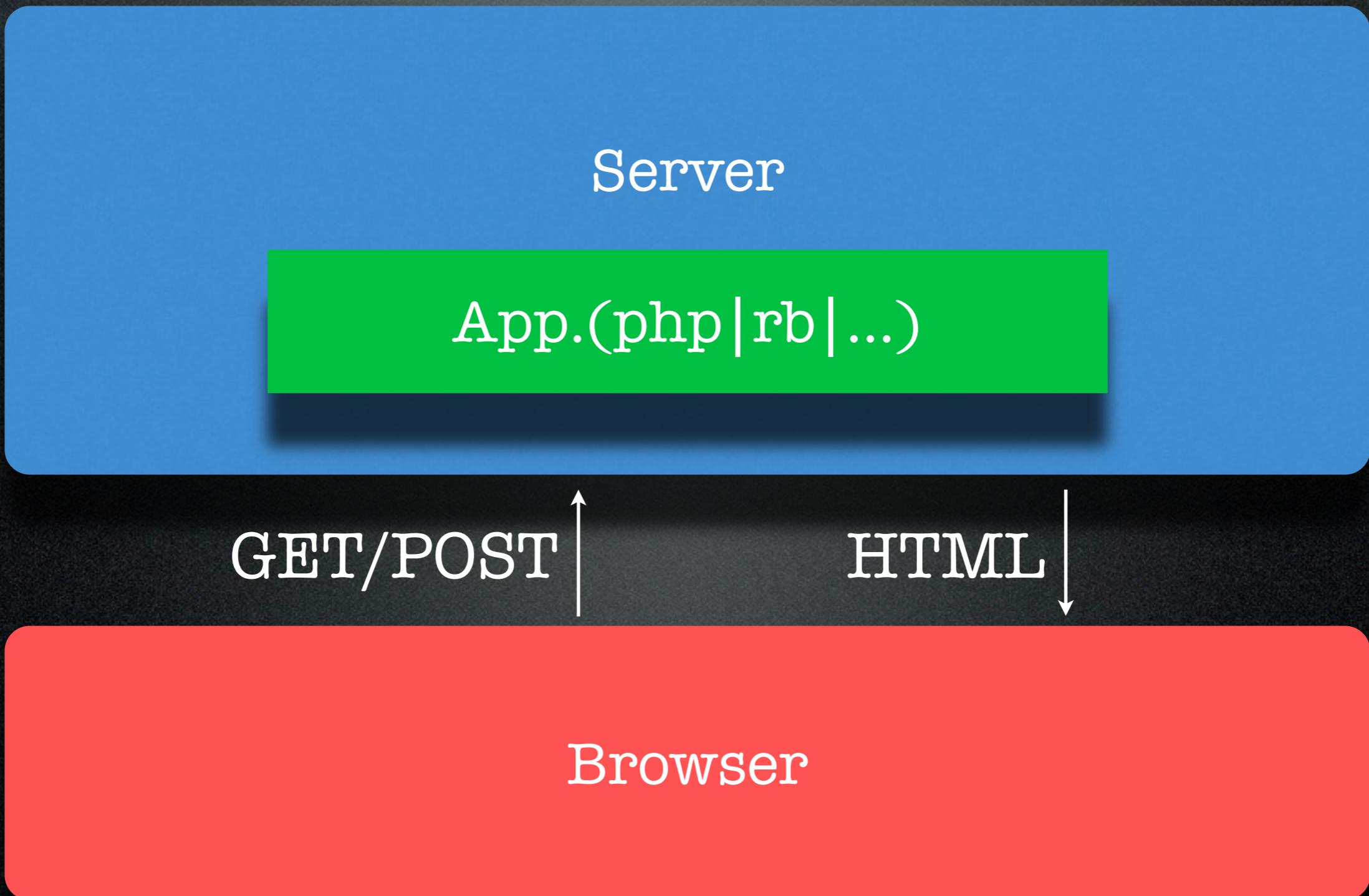
Views, anywhere!

@sylvinus

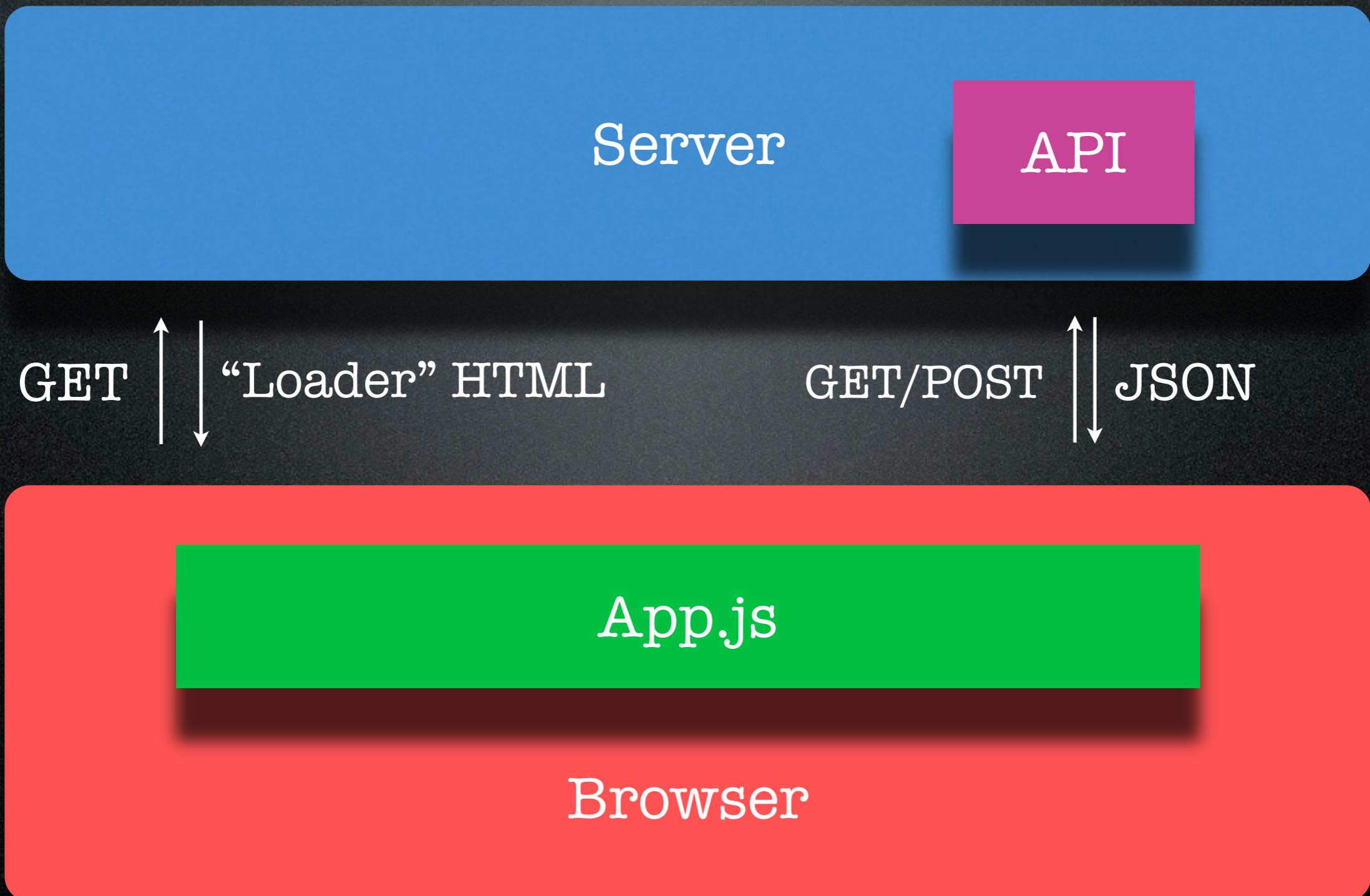
Paris Node Meetup, 8th June 2011

(I never thought I'd be
the opening act for God,
but here we are! ;-)

Old server-side apps



Current client-side apps



We should NOT be happy

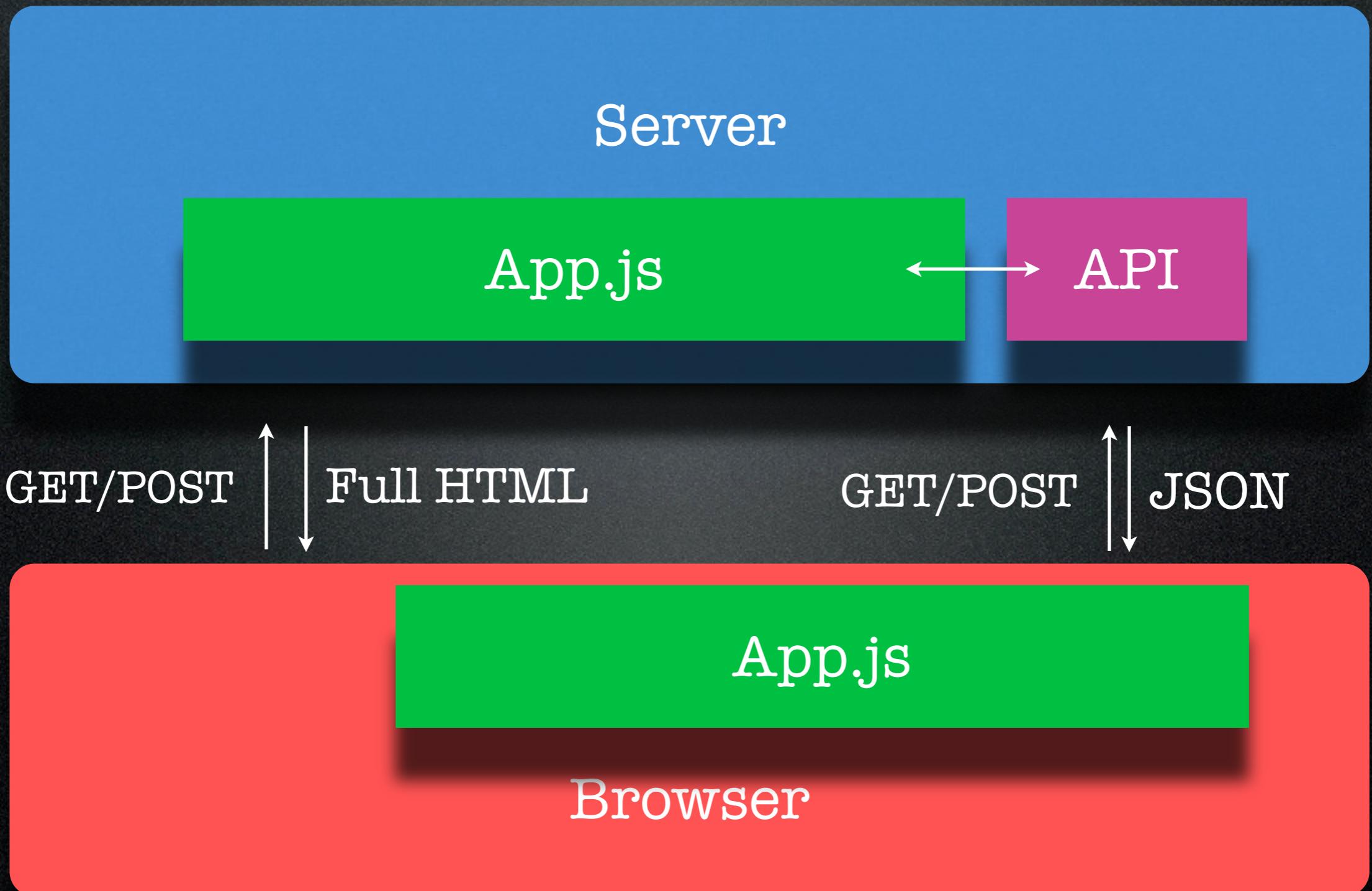
- Google & other crawlers?
- Browser history? (twitter.com#wtf)
- Accessibility?
- Mobile devices?
- Set-top boxes? CEHTML?
- Whatever is next?

Plain old HTML is
your friend.

Why Server-side JS?

- JS is cool
- Client or Server, only one language
- Client or Server, only one **app!**

Updated design



Basic idea

- One common app core
- Two adapters : Nodejs & Browsers
- As much as possible in the shared core!
- This is nothing really new (cf Gmail, again)
- BUT Nodejs makes it so much easier

Demo

History?

- HTML5 pushState()
- Server answers to GET /index.html
- Client listens to “/index.html” event
- If !HTML5, 2 possible fallbacks:
 - #bang
 - HTML-only mode

Benefits

- Only one codebase
- Serve HTML-only versions of your JS app for Crawlers, Odd devices, IE6, ...
- (Respect the way the web was built)
- Feel better about pushing advanced client-side features

The View Class

View.render(callback)

- Executed on server & client
- Plug APIs, Templates, ...
- callback(error, html)

View.enhance()

- Executed on client only
- Bind event listeners, too much jQuery plugins

View.transitionTo(newView)

- Executed on client only
- Plug some fancy animations!

Should you go this way?

- Check the requirements of your app
- Check your love for JS :)
- 29th June: get all of this (& more) built-in with the Joshfire framework!

Thanks!
Questions?