node.js in the browser using browserify

@elbuo8
Yamil Asusta
Developer Evangelist

Hola! Soy Yamil!

Originally from Puerto Rico

Devangelist @ SendGrid

@elbuo8

SendGrid

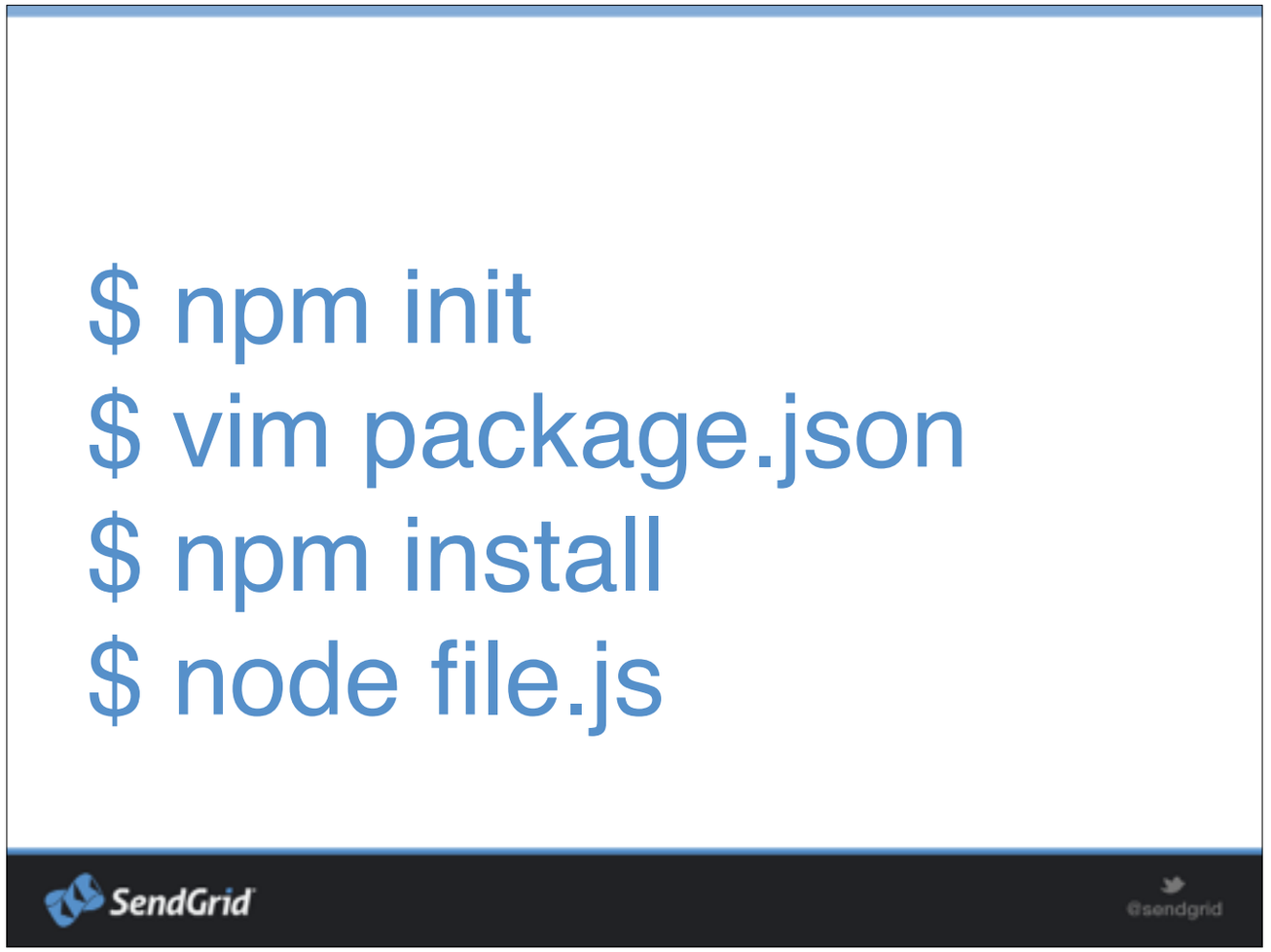
@sendgrid

My job is the make developers life easier. Regarding email or not. blah blah

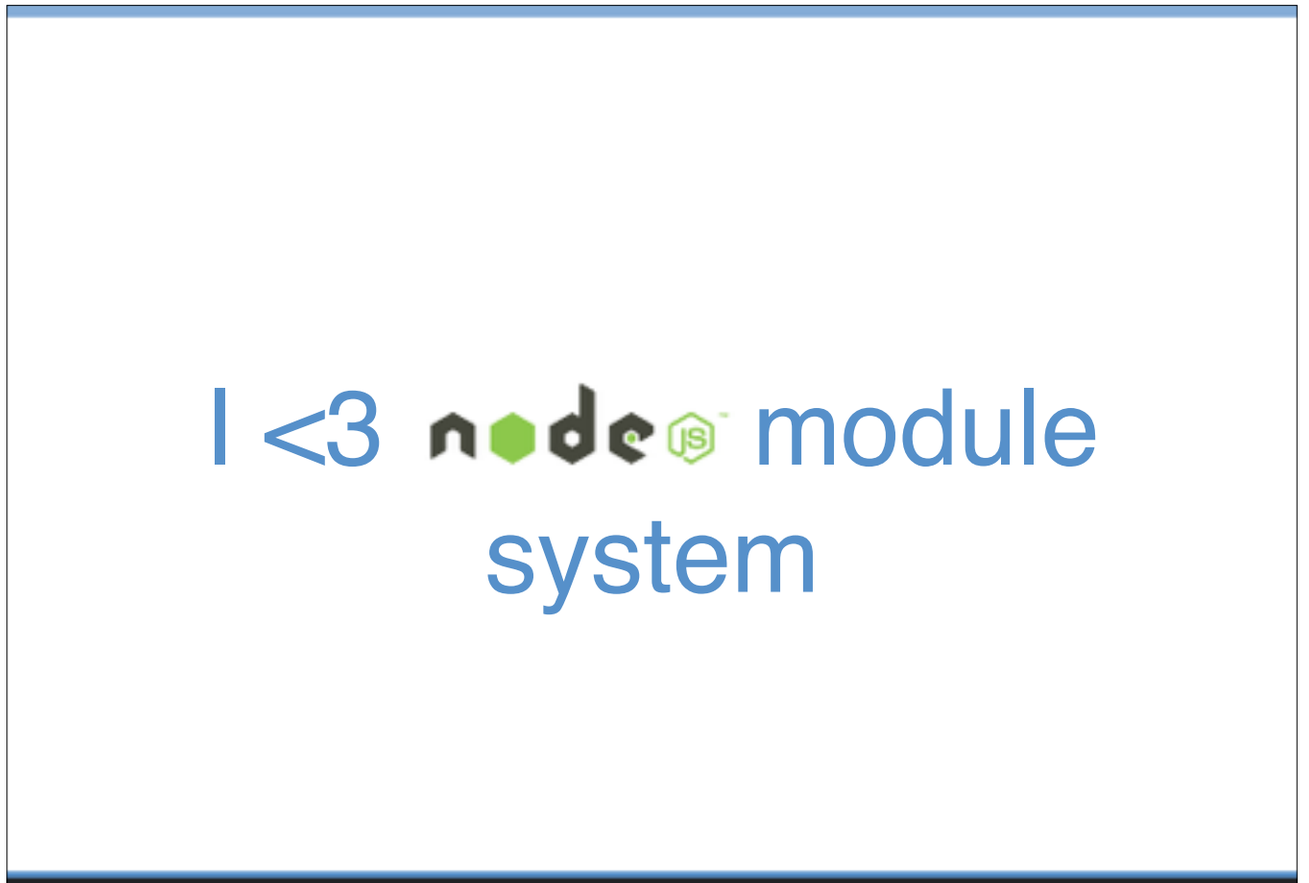
Shameless plug. Im not really a FE guy. I’m a JS guy, I can play where ever I want, but I have never focused on FE JS, I’ve mostly done Node. Therefore the tone for the talk will be mostly from a backend engineer perspective…because thats my specific case

What if I need to do Front End?

Every so often, I come across from requests to build something specific to the front end. Its still JS, but there are some differences coming from the node world. Native module system, http syntax and the different objects it has, for example.



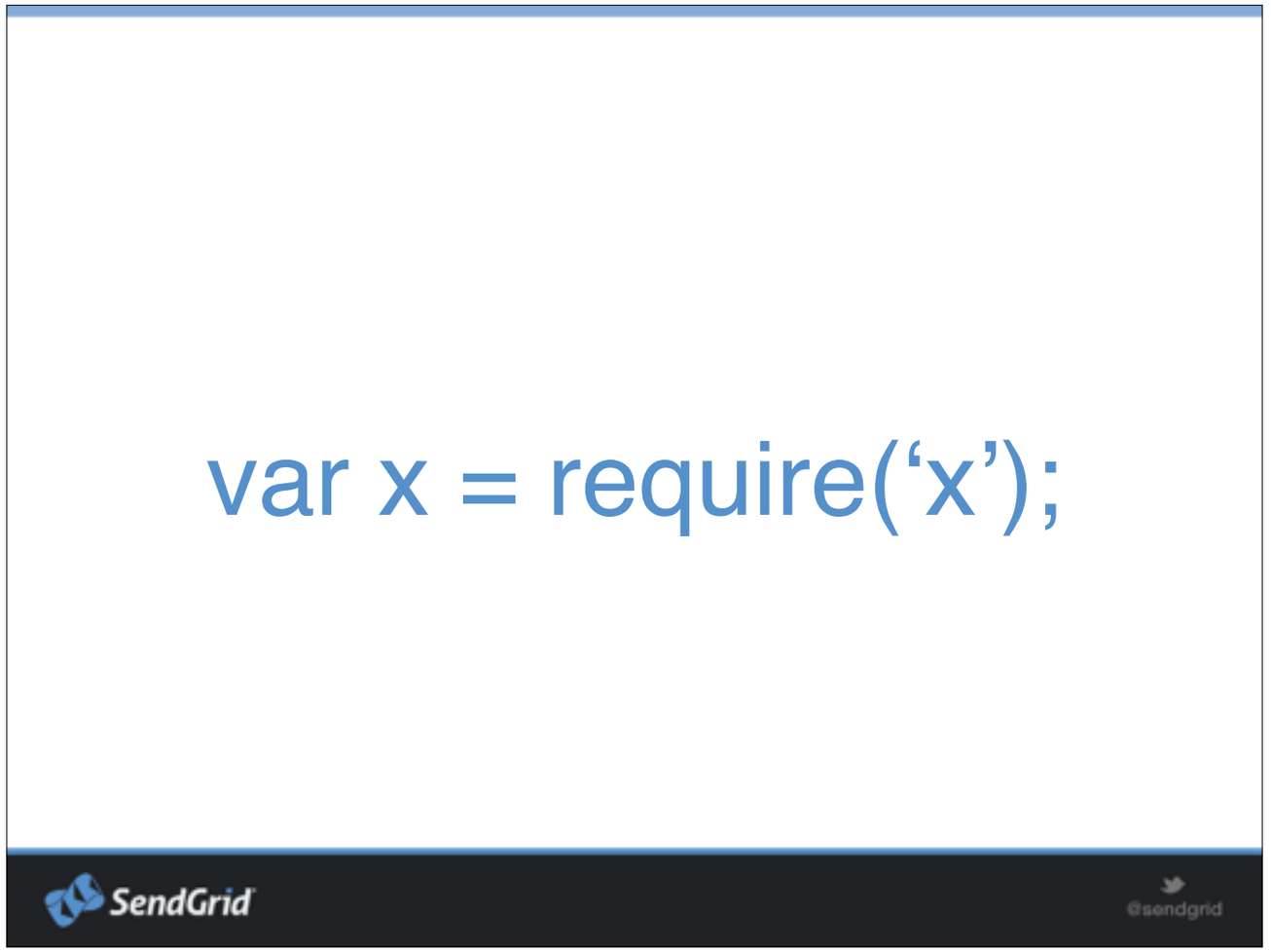
What do I want to do? I want to keep this same workflow, or change it as little as possible. Hence I still want to have my normal package manager, dependency management, my same standard module format. Basically, I don’t want to learn Bower, Require.js etc



Simply requiring what module you need and it will resolve all of its dependencies, versioning. Its simple and clean.

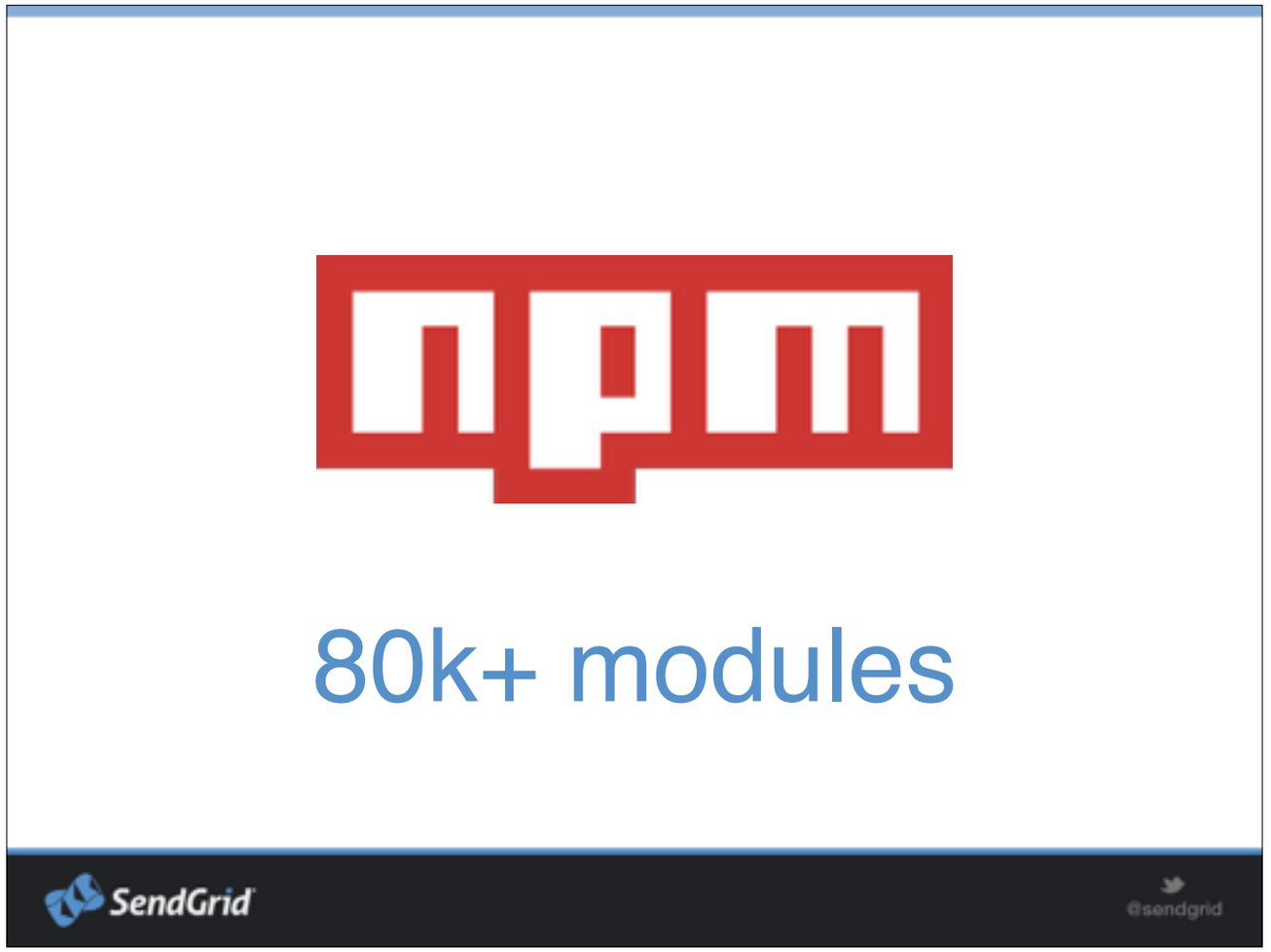
And its fairly simple. package json contains semantic versions of each package you are using. several script areas and meta data which is super useful to automate unit testing, deployments etc. You can use it as a pipeline if you want.

Each module also contains its own package.json. Hence there arent problems with using different versions of a module since each one is self contained. Its great.

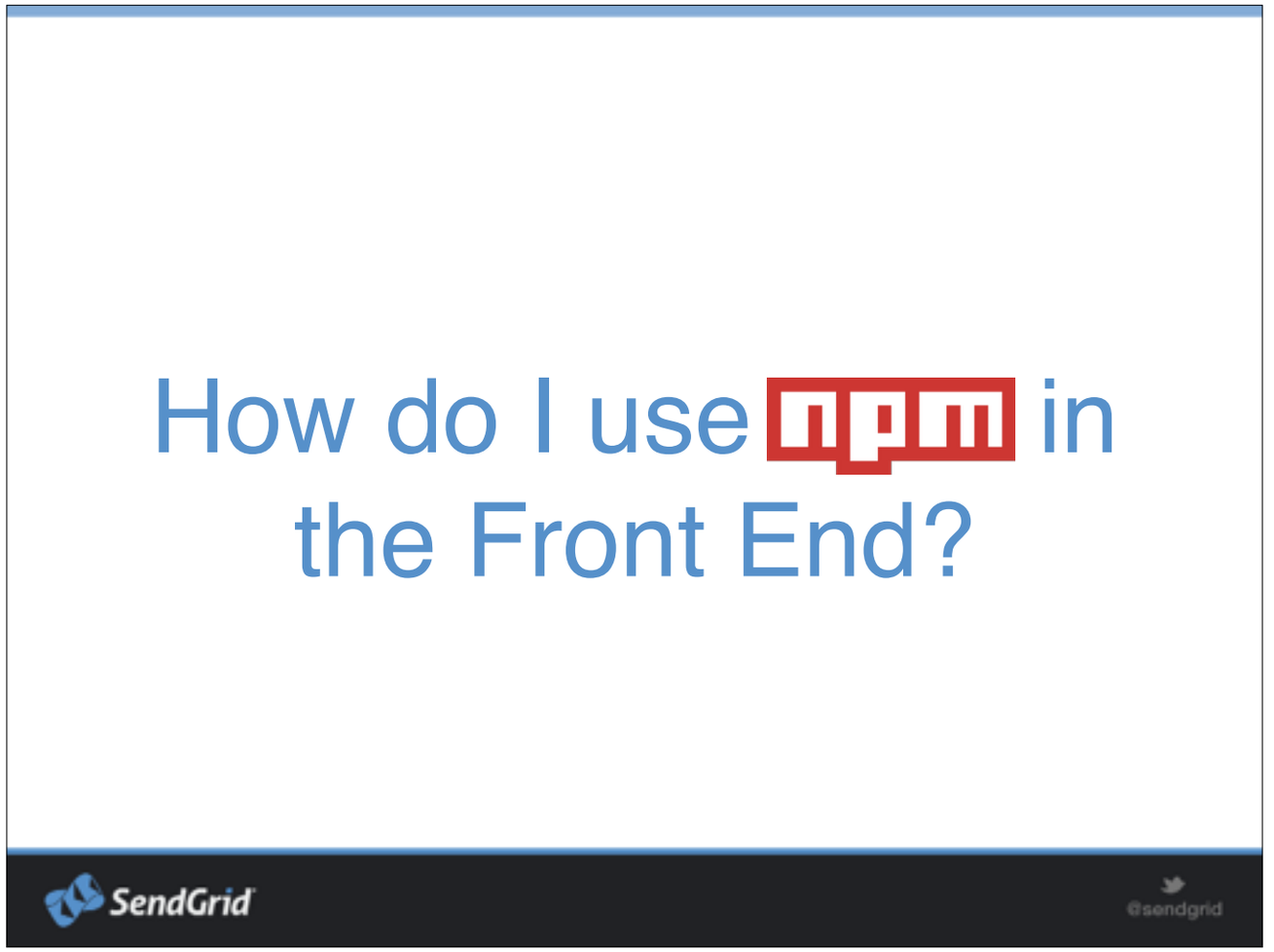


This is how you define a module in node. I can simply do this, and x will be ready to use in that file. No awkward defines and requires. I can also require local modules relative to the file.

But the real reason I want to keep my workflow…



My brain is wired to use NPM all the time. I’m familiar with it, I’ve been writing node style for a while. I always check that npm doesn't have the solution to my current problem. The community is huge, there are 5x more packages in npm than bower.



I dont see myself porting everything. That would take forever and might as well just learn require and other stuff and do it the right way since the beginning.

Browserify is baller. Its done by Substack (James Halliday). And is does something really well, port Node code to the browser. How does it do it?

It injects node's **require** in the browser.

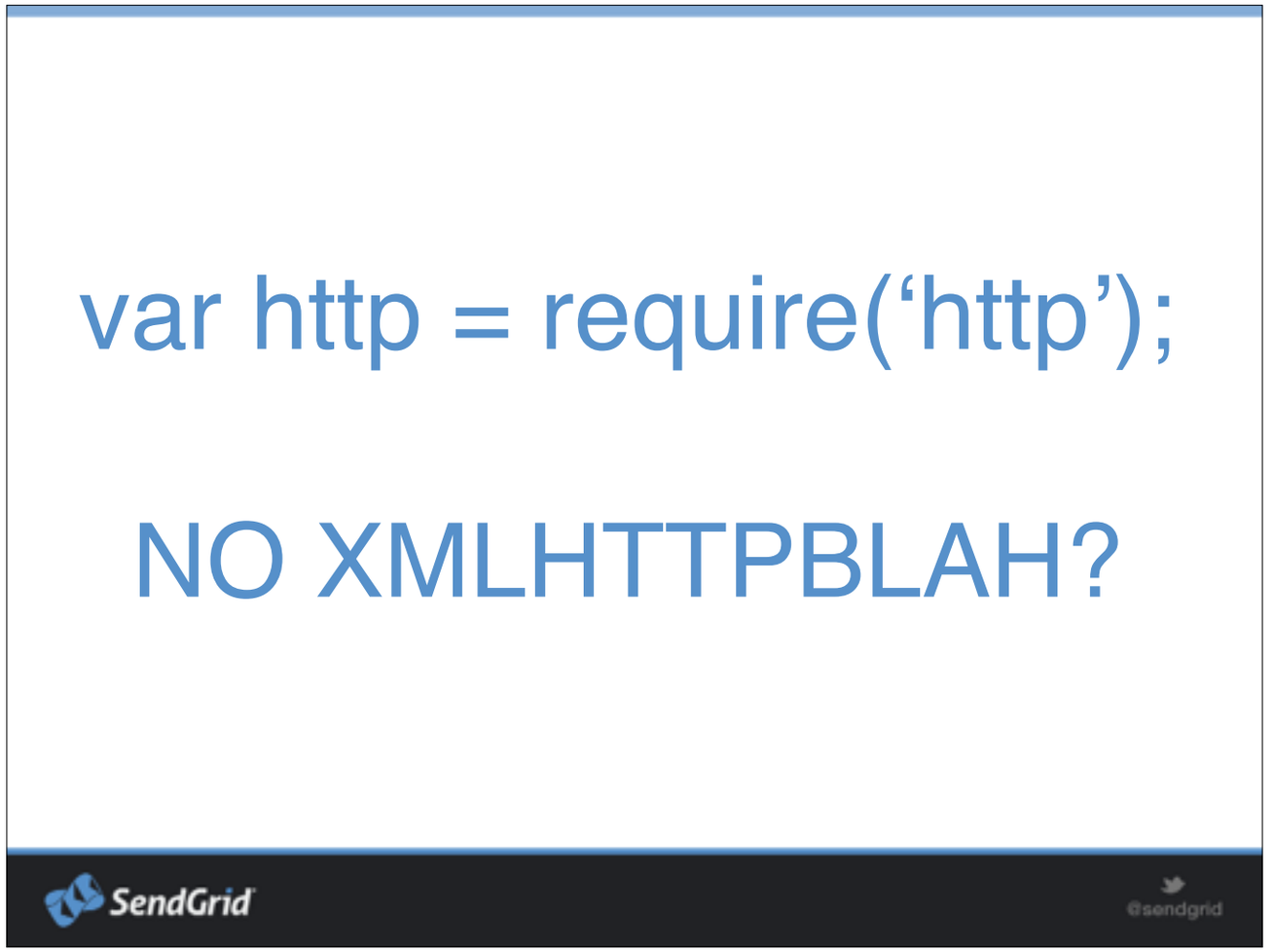
Therefore it uses commonjs type of requires. These ones I'm used to instead of the AMD modules that require.js uses. In other words, node's module system now works seamlessly

It replaces node's built in modules with browser compatible ones

SendGrid

@sendgrid

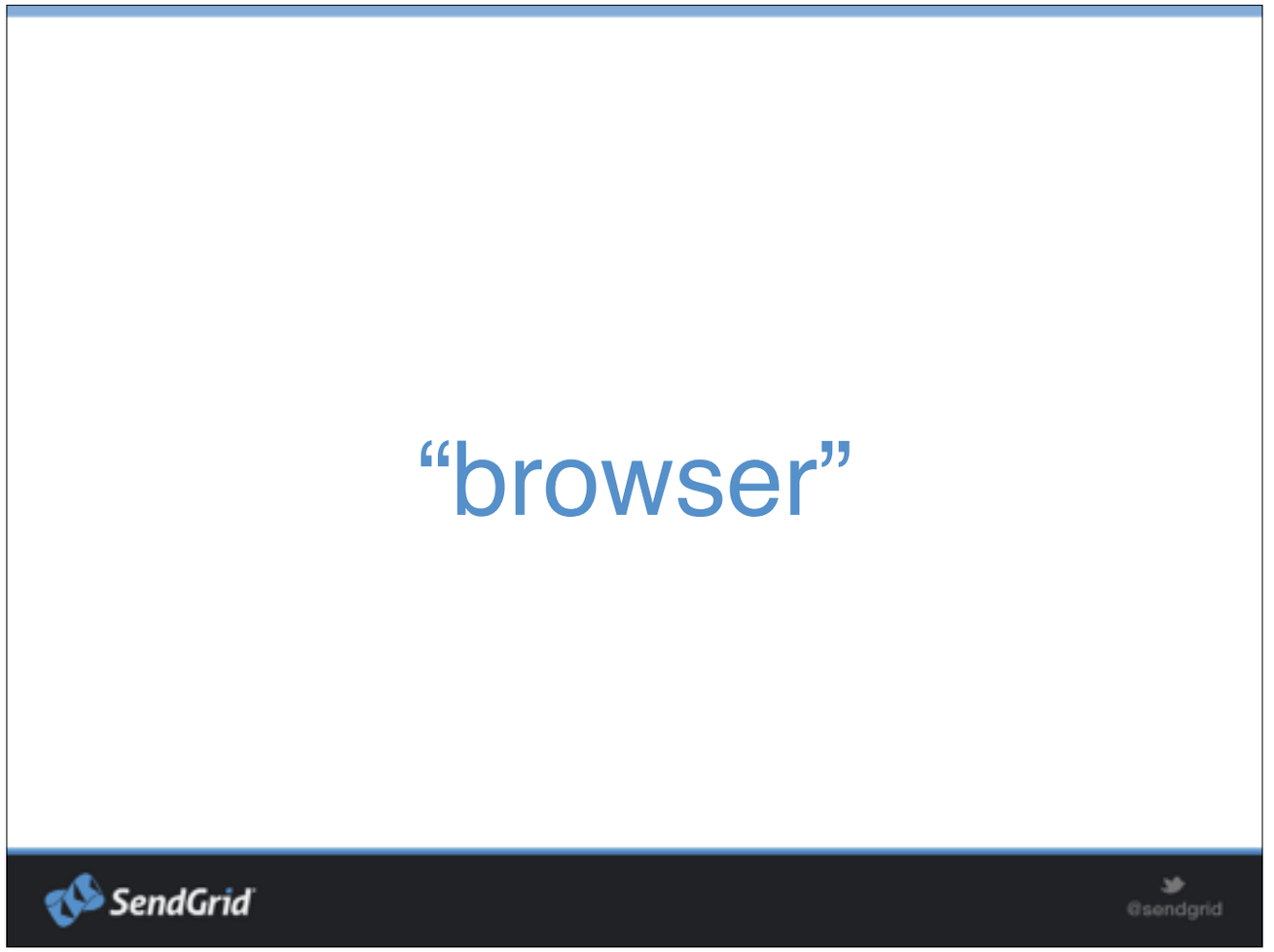
For example -> next slide



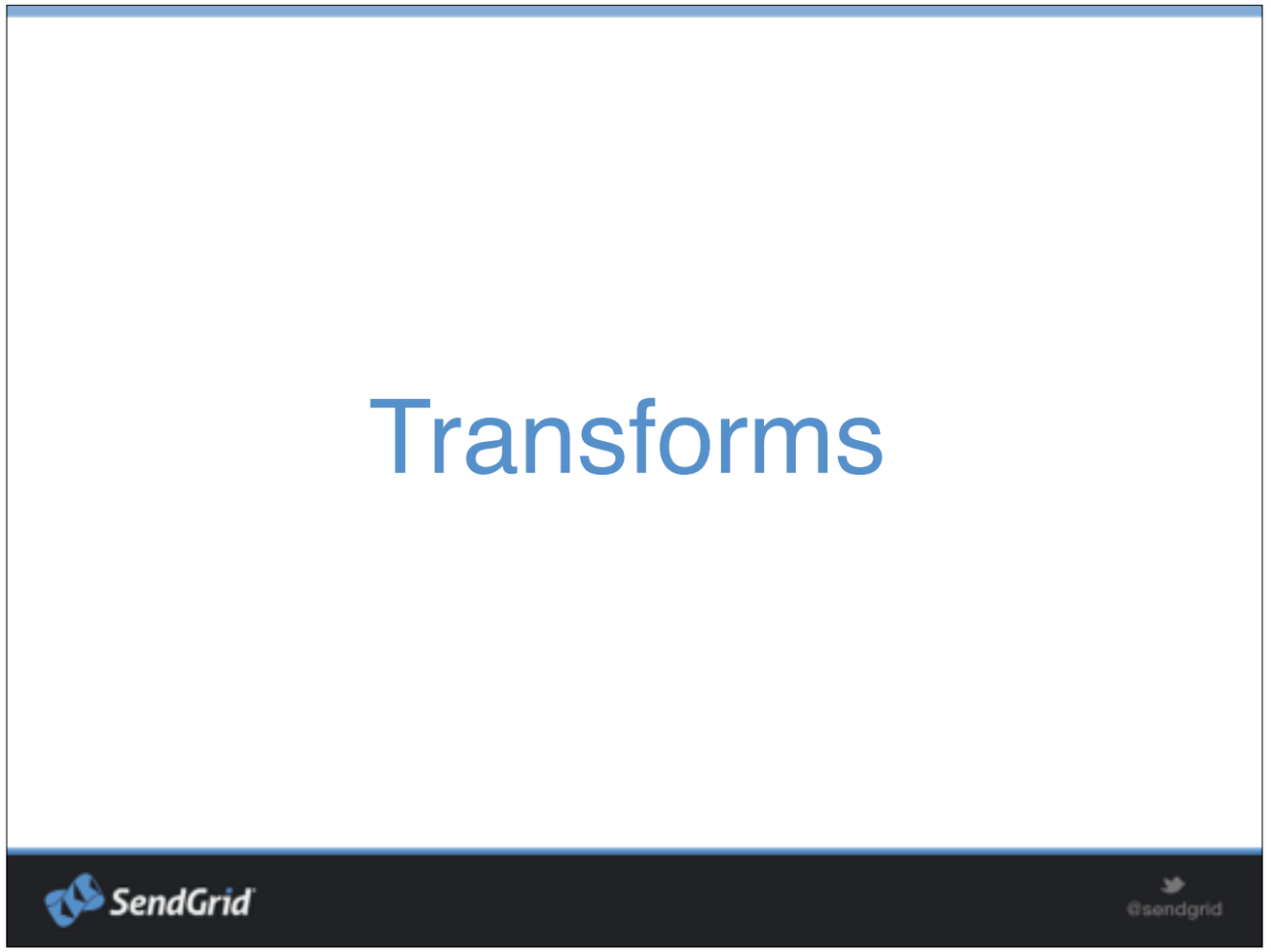
Browserify abstracts XMLHTTPObject with the regular syntax of its own http method. https is the same. And like this, there are countless other modules which work perfectly in Node land and have a equivalent in browser land

this are called shims

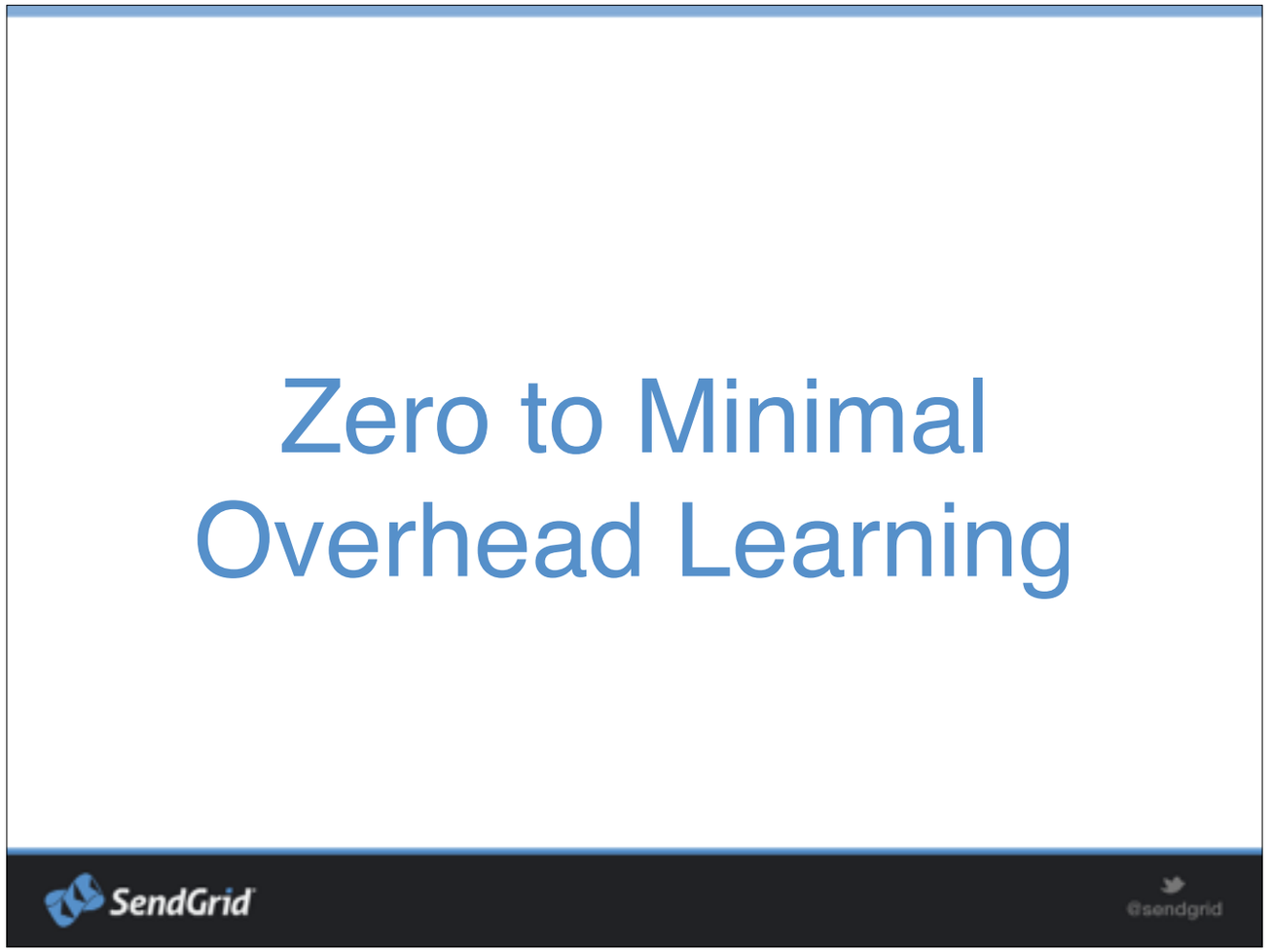

ReactionGIFS.me



Browserify will look for a **browser** field in package.json where it will perform substitutions for specific packages. you can have a node version, browserify and have a browser version of the module, seamless.



Write .coffee? No problem. Had AMD modules and want commonjs? No problem. Keep writing code like usual and it will get resolved and bundled all together. Think of it like Gulp injected into the browserify pipeline. There is a big list of transformations available that you can use



Lets recap a bit. I didn't have to learn a different module system. I didn't have to learn specific JS objects in the DOM/Browser for things I can still do in Node. I can keep writing the way and the only thing I had to inject into my workflow was browserify.