The Dojo Build System

Tobias von Klipstein - uxebu Ltd.
klipstein@uxebu.com

What?

- Process for
  - compressing your JS files
  - grouping several JS files into one
  - interning external non-JS files (HTML Templates)
  - resolving of css @import statements

→ Speed up your app

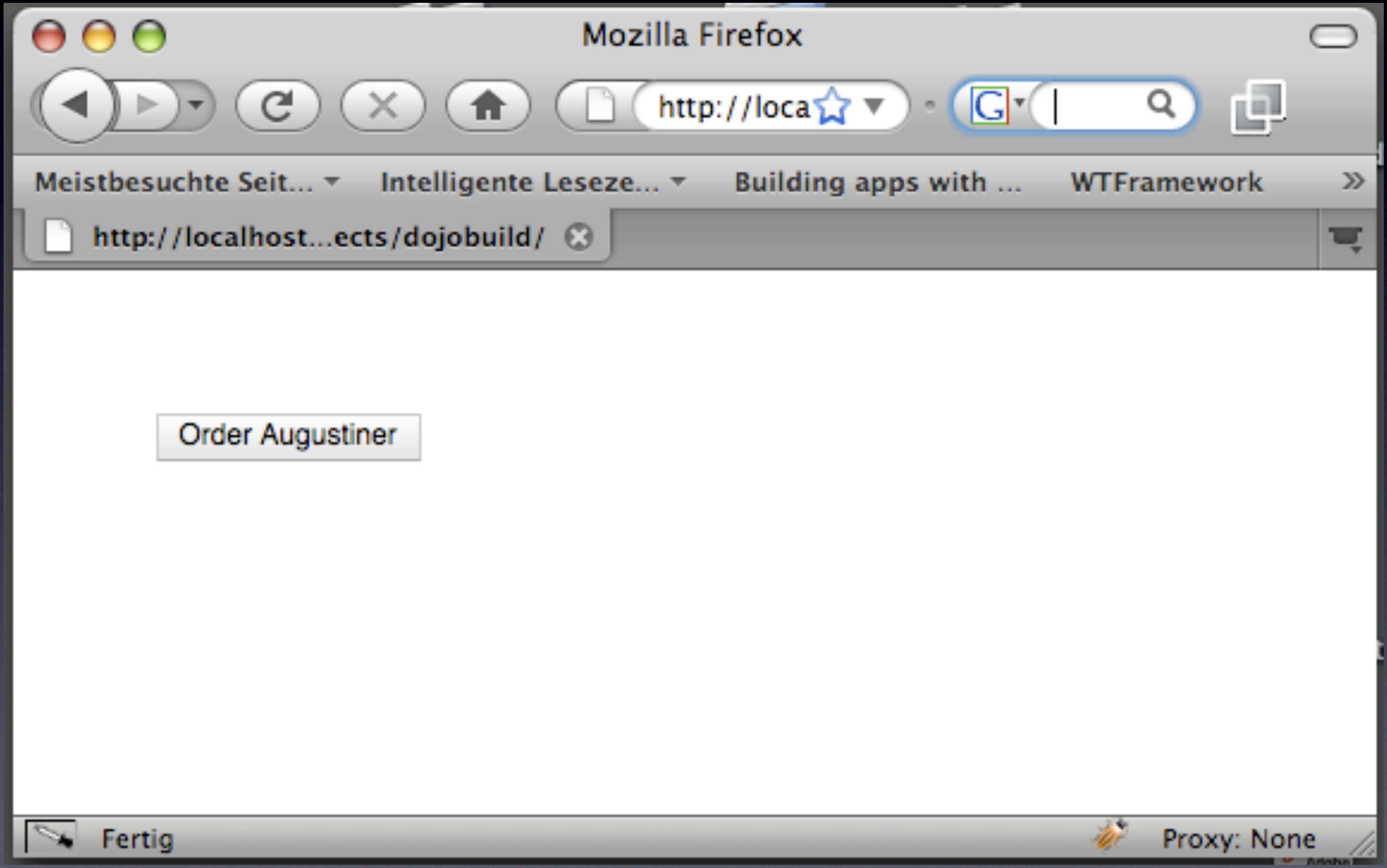
We need?

- Dojo source release
- Java 1.5 (recommended) for running Rhino
- A Dojo application, we want to optimize

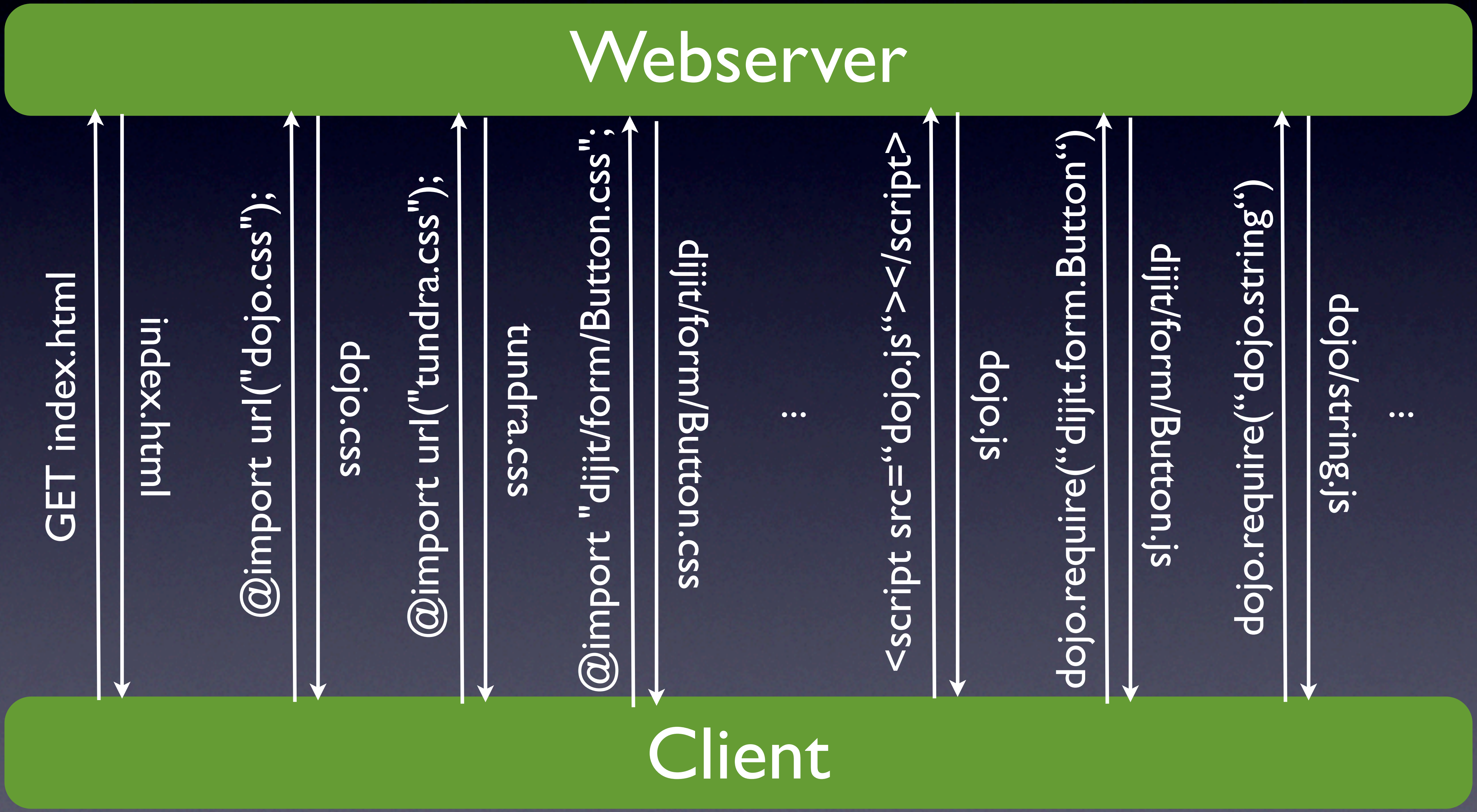
Dojo Source Release

- dojo (Base / Core)
- dijit
- dojox
- util (Rhino / Shrinksafe)

Our App


Mozilla Firefox
http://loca
Meistbesuchte Seit...
Intelligente Leseze...
Building apps with ...
WTFramework
http://localhost...ects/dojobuild/
Order Augustiner
Fertig
Proxy: None

Usual Page Load



Webserver

GET index.html

index.html

@import url("dojo.css");

dojo.css

@import url("tundra.css");

tundra.css

@import "dijit/form/Button.css";

dijit/form/Button.css

...

<script src=“dojo.js“></script>

dojo.js

dojo.require(“dijit.form.Button“)

dijit/form/Button.js

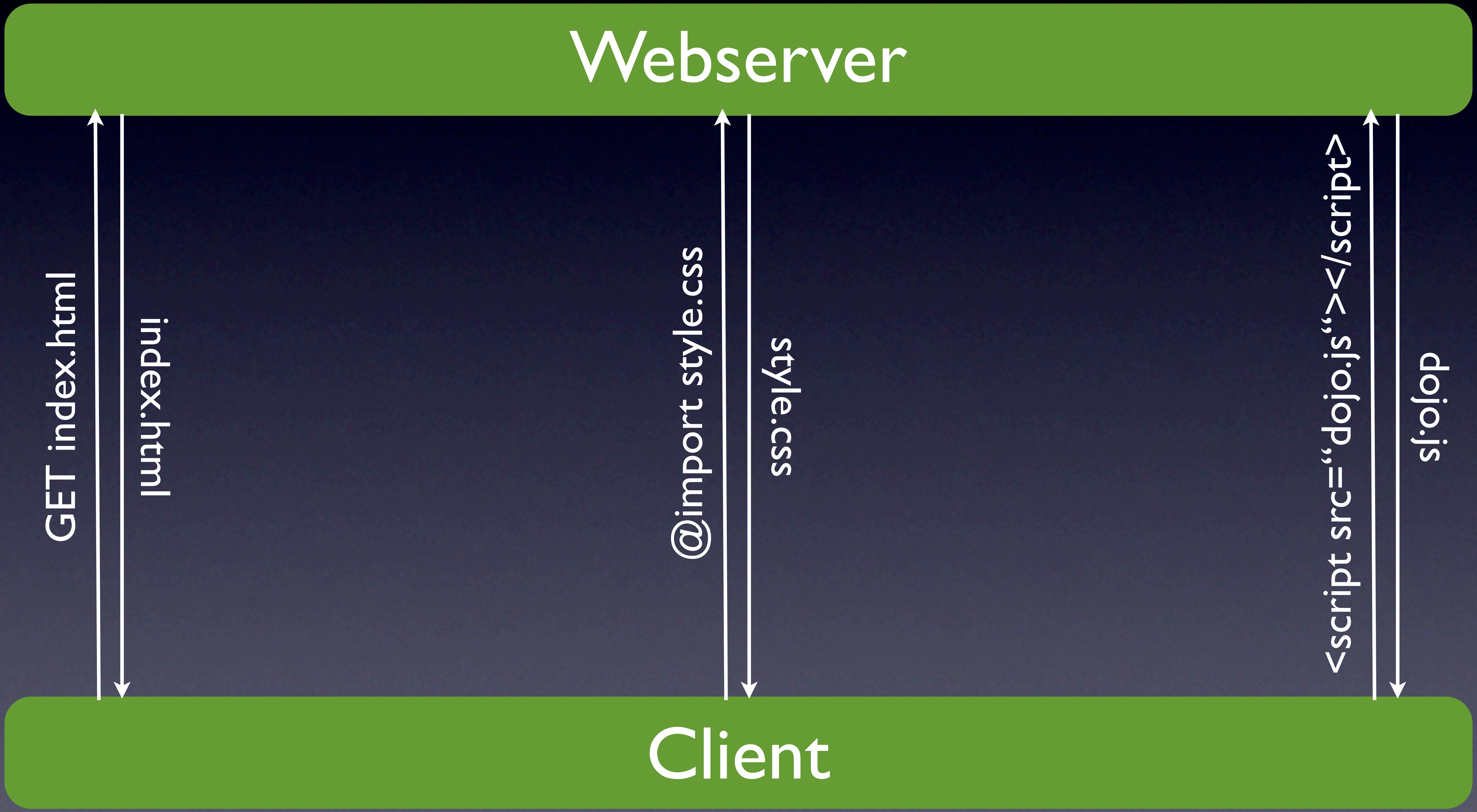
dojo.require(„dojo.string“)

dojo/string.js

...

Client

Optimized Page Load



Webserver

GET index.html

index.html

@import style.css

style.css

<script src=“dojo.js“></script>

dojo.js

Client

Build Profile (1)

- like ant‘s build.xml or Makefiles
- consists of
  - layers
  - modules
- util/buildscripts/profiles/<profile name>.profile.js

Build Profile (2)

The simplest build profile

¶

Build profile (3)

```
dependencies = {
    layers: [
        {
            //always "../[....]" or just "dojo.js"
            name: "../beer/layername.js",
            dependencies: [
                "dojo.string"
                "dijit.form.Button"
            ],
            //relative to the util/buildscripts
            //directory or an absolute path.
            copyrightFile: "myCopyright.txt"
        }
    ],
    prefixes: [
        //relative to the dojo/dojo.js file
        ["dijit", "../dijit",
        ["beer", "../beer", "../beer/copyright.txt"],
    ]
};
```

Doing a Build

- cd util/buildscripts
- build.sh [or .bat]
  profile=<profile name>
  action=release
  + [more options]

build.sh [options] (1)

- releaseName [default: dojo]
- releaseDir [default: ../../release/]
- version [default: 0.0.0.dev]
- profileFile (instead of profile option)
- localeList [default: de-de, en-gb, ...]

build.sh [options] (2)

- loader [default or xdomain]
- scopeDjConfig

build.sh [options] (3)

- cssOptimize (comments or comments.keepLines)
- optimize (shrinksafe / shrinksafe.keepLines / packer)
- layerOptimize [default: shrinksafe]
- mini [default: false]

Shrinksafe (1)

- removing whitespace
- removing comments
- replacing non-public symbols (argument / variable / function names) with shorter ones

Shrinksafe (2)

```
function MyClass(){
    this.foo = function(argument1, argument2)
{
        var addedArgs = parseInt(argument1)+
                        parseInt(argument2);
        return addedArgs;
    }
    var anonymousInnerFunction = function(){
        // do stuff here!
    }
}
function MyFunc(){
    // this is a top-level function
}
// we've got multiple lines of whitespace
here
```

→

```
function MyClass(){
this.foo=function(_1,_2){
var _3=parseInt(_1)+parseInt(_2);
return _3;
};
var _4=function(){
};
}
function MyFunc(){
}
```

Let‘s build

```
./build.sh profile=beer action=clean,release
```

What else you can do?

- Use a CDN (if just <5 modules are used)
- GZIP your content
- Loading your javascript files at the end and CSS upfront
- Creating a separate domain for your js files: js.example.com (xdomain)
- Determine often used modules (regularly)

Links

- docs.dojocampus.org/quickstart/custom-builds
- dojotoolkit.org/docs/shrinksafe
- dojocampus.org/content/2008/05/03/dojo-build-101-basics/
- dojocampus.org/content/2008/05/26/dojo-build-201-layers-and-css-optimizing-builds/
- download.dojotoolkit.org (source release)