

The background of the slide is a blurred aerial photograph of a city, showing a dense concentration of buildings with various colored roofs and some green areas. The image is slightly out of focus, creating a soft, layered effect.

@MIKAELBREVIK

SKALERING AV FRONTEND- BYGGESYSTEM ETTER BEHOV

MAN TRENGER INGEN TASK MANAGER/
BYGGESYSTEM/GULP/GRUNT/BROCCOLI,
NÅR MAN HAR MAKE/SCRIPTS/WHATNOT

Mange bloggposter i løpet av de siste årene

DETER SANT. ■ ■ ■

...HELT TIL DET IKKE
ER SANT

SÅ VI MÅ KUNNE ENDRE
MENING ENKELT

HVORDAN VI SKAL BYGGE, ER IKKE
ET VALG VI MÅ TA MED EN GANG.

VI LAGER ET PRODUKT, IKKE
ET BYGGESENTER.

LA OSS HELLER ANGRIPE DET SOM
VI ANGRIPER VANLIG UTVIKLING

- 1. UTSETTE VALG TIL VI VET MER**
- 2. TILLATE UTVIDELSER/SKALERING**
- 3. TENKE PÅ GRENSESNITT (API)**

HVORDAN GJØRE DET?



STEG #1

INGEN BYGGING

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Test</title>
6   </head>
7   <body>
8
9     <script src="//fb.me/react-with-addons-0.14.0.js"></script>
10    <script src="//fb.me/react-dom-0.14.0.js"></script>
11    <script src="assets/js/app.js"></script>
12  </body>
13 </html>
```

1. VELDIG ENKELT

2. FORUTSETTER INGenting

DET FUNKER™

TIL DU MÅ HA SKIKKELIG
STRUKTUR OG ARKITEKTUR



STEG #2

KOMMANDOER

~/scaling-frontend

→ npm install browserify -g

```
~/scaling-frontend
`-> browserify assets/js/app.js -o dist/bundle.js █
```

```
1  {
2    "scripts": {
3      "build": "browserify assets/js/app.js -o dist/bundle.js"
4    }
5  }
```

```
1  {
2    "name": "scaling-frontend",
3    "scripts": {
4      "build": "browserify assets/js/app.js -o dist/bundle.js"
5    },
6    "devDependencies": {
7      "browserify": "^12.0.1"
8    }
9 }
```

```
~/scaling-frontend  
↳ npm run build
```

```
> scaling-frontend@ build /Users/mikaelbrevik/scaling-frontend  
> browserify assets/js/app.js -o dist/bundle.js
```

```
1  {
2    "name": "scaling-frontend",
3    "scripts": {
4      "build": "browserify -t babelify assets/js/app.js -o dist/bundle.js"
5    },
6    "devDependencies": {
7      "browserify": "^12.0.1"
8    }
9 }
```

- 1. FORBEDRER KODEKVALITET**
- 2. ENKELT Å INTRODUSERE**
- 3. TILDELS ENKELT Å UTVIDE**
- 4. INTRODUSERER ET API**

DET FUNKER™

TIL DU FÅR ØKT KRAV TIL MANIPULERING
AV KODEN SOM EN DEL AV BYGGET



STEG #3

SCRIPTS

```
1 ~ {  
2     "name": "scaling-frontend",  
3     "scripts": {  
4         "build": "browserify -t [babelify envify brfs] assets/js/app.js -o dist/bundle.js",  
5         "postbuild": "sed -i.old '1s;^;/* My library v2.0.0 */;' dist/bundle.js"  
6     },  
7     "devDependencies": {  
8         "babelify": "^7.2.0",  
9         "brfs": "^1.4.1",  
10        "browserify": "^12.0.1",  
11        "envify": "^3.4.0"  
12    }  
13 }
```

MANGE TYR TIL BASH/POWERSHELL,
MEN DET ER IKKE KRYSS-PLATFORM

...OG HVORFOR DET OM VI HAR ALT
HAR TILGANG PÅ NODE?

```
1 var b = browserify({ entries: './js/app.js' });
2 b.transform(envify);
3 b.transform(babelify);
4 b.transform(brfs);
5
6 var header = `/* My library v2.0.0 */\n`;
7 b.bundle(function (err, code) {
8   fs.writeFileSync('./dist/bundle.js', header + code.toString());
9 });
```

```
1 var b = browserify({ entries: './js/app.js' });
2 b.transform(envify);
3 b.transform(babelify);
4 b.transform(brfs);
5
6 var pckg = require('./package.json');
7
8 var header = `/* My library ${pckg.version} */\n`;
9 b.bundle(function (err, code) {
10   fs.writeFileSync('./dist/bundle.js', header + code.toString());
11 });
```

```
"scripts": {  
  "build": "node makeBundle.js"  
},
```

```
~/scaling-frontend  
~> npm run build
```

```
> scaling-frontend@ build /Users/mikaelbrevik/scaling-frontend  
> node makeBundle.js
```

1. MYE ENKLERE Å UTVIDE
2. FUNKER PÅ ALLE PLATTFORMER
3. SAMME API
(ABSTRAHERT BORT IMPLEMENTASJON)
4. KRAFTEN TIL PROGRAMMERING

DET FUNKER™

TIL DU HAR MANGE SCRIPTS
DU VIL STRUKTURERE



STEG #4

HÅNDTERE FLERE
SCRIPTS

HAR DU FLERE BYGGESCRIPTS
DU VIL STRUKTURERE?

HAR DU FLERE BYGGESCRIPTS DU VIL STRUKTURERE?
**DA GIR EN TASK-
MANAGER VERDI**

HVA ER EGENTLIG GULP/GRUNT/BROCCOLI? OPPGAVE-HÅNDTERING + REDUSERING AV BOILERPLATE

```
✓ └─ scaling-frontend
    ├ node_modules
    └ tasks
        └ jshint.js
        └ makeBundle.js
        └ packageBundle.js
{} package.json
```

```
1  {
2      "name": "scaling-frontend",
3      "scripts": {
4          "build": "node tasks/makeBundle.js",
5          "package": "node tasks/packageBundle.js",
6          "jshint": "node tasks/jshint.js"
7      },
8      "devDependencies": {←
14  }
```

```
1 var b = browserify({ entries: './js/app.js' });
2 b.transform(envify);
3 b.transform(babelify);
4 b.transform(brfs);
5
6 var pckg = require('./package.json');
7
8 var header = `/* My library ${pckg.version} */\n`;
9 b.bundle(function (err, code) {
10   fs.writeFileSync('./dist/bundle.js', header + code.toString());
11 });
```

```
1 var gulp = require('gulp');
2
3 gulp.task(js);
4
5 function js () {
6   var b = browserify({ entries: './js/app.js' });
7   // ...
8 }
```

```
1  {
2    "name": "scaling-frontend",
3    "scripts": {
4      "build": "gulp js"
5    },
6  >  "devDependencies": {  
13 }
```

```
~/scaling-frontend  
→ npm run build
```

```
> scaling-frontend@ build /Users/mikaelbrevik/scaling-frontend  
> gulp js
```

MER UTTRYKKSFULL HÅNDTERING AV OPPGAVER

FOR STRUKTURERING AV BYGGEKODE PÅ LIK LINJE SOM ANNEN KODE

```
1 var gulp = require('gulp');
2 var js = require('./tasks/js');
3 var jshint = require('./tasks/jshint');
4 var scss = require('./tasks/scss');
5
6 ▼ gulp.task('build',
7   ▼ gulp.parallel( // Non dependent tasks – run in parallel
8     gulp.series(jshint, js), // dependent task – run in series
9     scss
10   ));
```

REDUSERER BOILERPLATE

```
1 function createBundle () {  
2     var b = browserify({ entries: entryPoint });  
3     b.transform(envify);  
4     b.transform(babelify);  
5     b.transform(brfs);  
6  
7     return b.bundle()  
8         .pipe(source(pack.name))  
9         .pipe(derequire());  
10 }
```

```
1 gulp.task('build:js', gulp.parallel(js, minify));  
2  
3 var headerMessage = `/* My library ${pckg.version} */\n`;  
4  
5 function js () {  
6     return createBundle()  
7         .pipe(header(headerMessage))  
8         .pipe(gulp.dest(outputFolder));  
9 }  
10  
11 function minify () {  
12     return createBundle()  
13         .pipe(uglify())  
14         .pipe(header(headerMessage))  
15         .pipe(gulp.dest(outputFolder));  
16 }
```

- 1. KOMMER MED HJELPENDE KODE**
- 2. ENKLERE OPTIMALISERING MED
PARALLELE OPPGAVER**
- 3. TILGANG PÅ STORT ØKOSYSTEM**

DET FUNKER™



1. INGEN BYGGING

2. KOMMANDOER

3. SCRIPTS

4. TASK-MANAGER

1. UTSETTER VALG TIL VI VET HVA VI TRENGER

2. UTVIKLERE PÅ TEAMET TRENGER IKKE Å FORHOLDE SEG TIL IMPLEMENTASJONEN

3. UTVIDER OG REFAKTORERING PÅ EN SKIKKELIG MÅTE, UTEN Å BREKKE KODE.

**4. FREMFOR Å BRUKE TID PÅ Å SPEKULERE
I BYGGING, HAR VI BYGD ET PRODUKT.**

LES MER I BLOGGFORMAT:

[HTTP://OPEN.BEKK.NO/SCALING-FRONTEND-BUILD-STEPS-BY-NECESSITY](http://open.bekk.no/scaling-frontend-build-steps-by-necessity)

The background of the slide is a blurred aerial photograph of a city, showing a dense concentration of buildings with various colored roofs and some green areas. The image is slightly out of focus, creating a soft, layered effect.

@MIKAELBREVIK

SKALERING AV FRONTEND- BYGGESYSTEM ETTER BEHOV