

Comprendre les tours de passe-passe de Webpack

Talk @ Breizhcamp 2019 - Cédric Patchane





Cédric Patchane

 @CPatchane





The screenshot shows the Cozy Store interface. On the left, there's a sidebar with categories like Applications, Services, and Mes Applications. Under Services, 'Free' is listed under INTERNET. In the main area, the 'Cozy Banks' application is highlighted. It displays a current balance of 43 820,09 € across 7/8 accounts. A line graph tracks the balance from August to March. Below the graph, a table lists accounts with their numbers and names: N°3652983 (BNP Paribas), N°5545979 (Boursorama), and N°5652653 (Crédit Agricole). A mobile phone overlay shows a notification for 'COZYBANKS now' about a low balance of 34,89 €. The Cozy Store interface includes tabs for APPLICATIONS and PARAMÈTRES at the top right.



Synchronisez et partagez vos fichiers et photos



Importez vos factures automatiquement



Suivez vos dépenses du quotidien

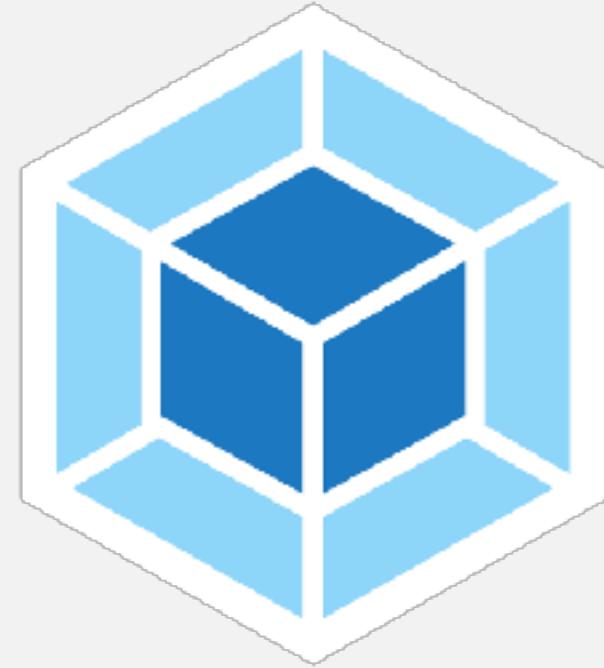


Prenez soin de votre vie privée

Qu'est ce qu'un *bundler* JS ?

Un outil qui regroupe tous vos **fichiers sources (modules)** et leurs **dépendances** pour en faire un ou plusieurs **paquets (bundles)** Javascript

Quelques *bundlers* JS : Webpack, Rollup, Parcel, Microbundle, ...



webpack



- *Static Module bundler* open-source (**MIT**), basé sur NodeJS
- Sortie en 2012, popularisé en 2015 car se différencie rapidement des *tasks runners* de l'époque
- Dernière version stable majeure : Webpack 4
- **Agnostique**, n'est pas spécifique à un langage Web
- Dispose de très nombreux **plugins et loaders**



webpack

- Utilisation :

CLI

```
webpack app/index.js -o bundle.js --mode=production
```

- L'univers des possibles 😱 :

Zéro config

Serveur de développement

Exécution de tâches

Build CSS

Offline (PWA)

Gestion de pages HTML

Hot Reload

Transpilation

Minification

Analyse de code

Optimisation

Gestion d'assets

Linting

etc...

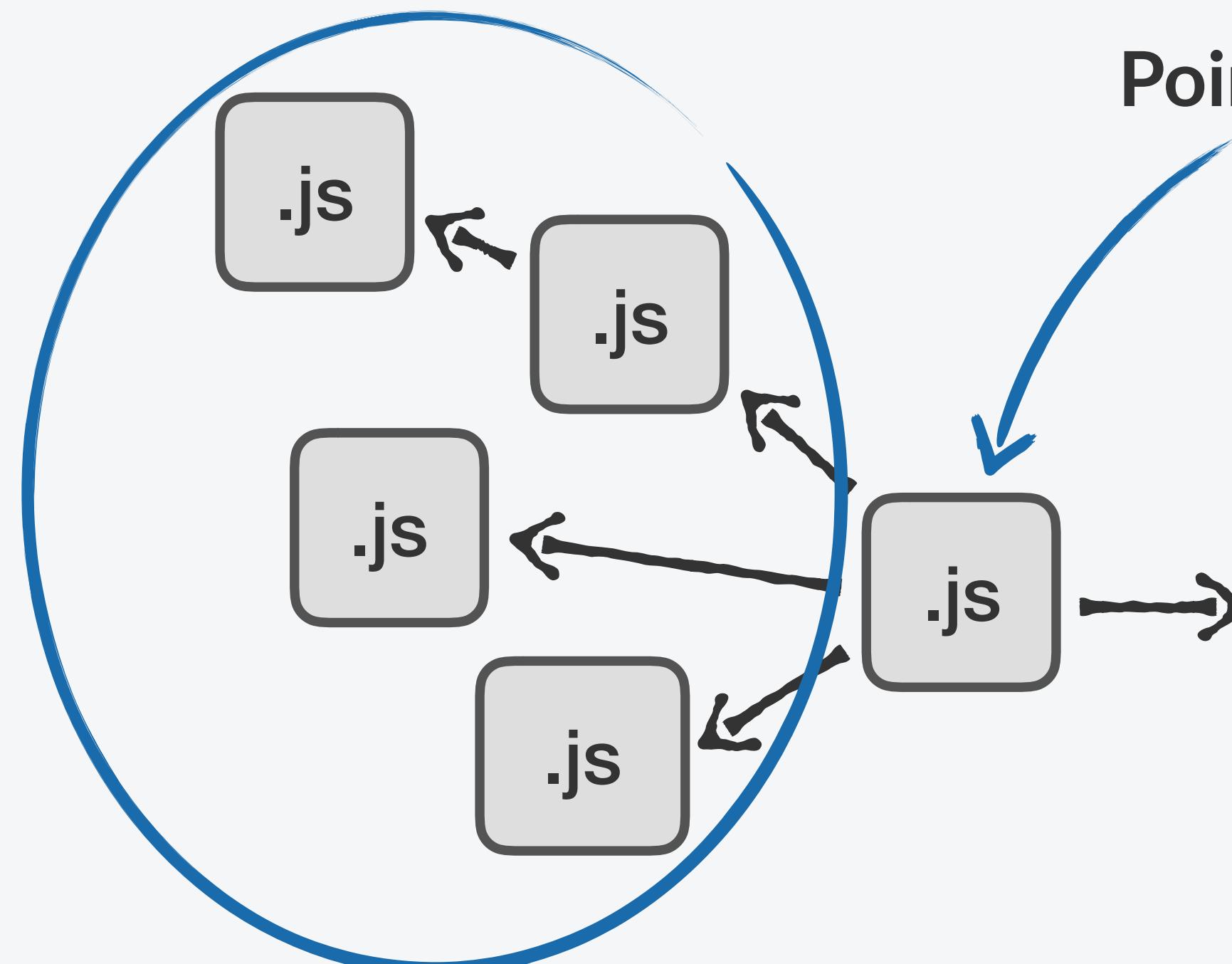
```
module.exports = {
  mode: 'production',
  entry: './app/index.js',
  output: { filename: 'bundle.js' },
  module: {
    rules: [ ... ]
  },
  plugins: [ ... ],
  ...
}
```

Construire du JS pour le navigateur

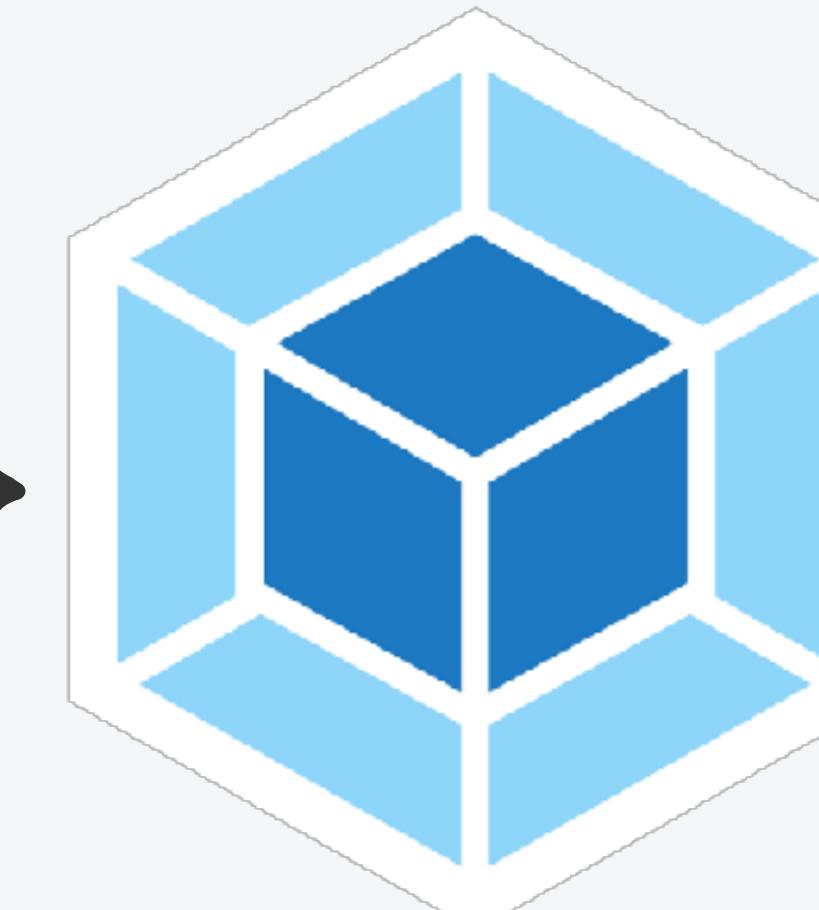
Revenons à l'objectif premier de Webpack

Construire du JS pour le navigateur

Graphe de dépendances



Point d'entrée



Modules

Bundle(s)

Tout est module : les loaders

Tout est module : les loaders

- Module Node, indépendant de Webpack en lui même
- Indique à Webpack comment “charger” un fichier
- Peut être enchaîné avec un autre loader
- Prend la forme d'une fonction qui retourne du code JS au format *String* ou *Buffer* :

```
// loader.js
export default function myLoader(source) {
  return `export default ${ JSON.stringify(_transform_(source)) }`
}
```

Tout est module : les loaders

`json-loader`

```
// json_loader.js
export default function jsonLoader(source) {
  return `export default ${ JSON.stringify(JSON.parse(source)) }`
}
```

N.B. : Intégré dans Webpack depuis la version 2

Tout est module : les loaders

`babel-loader`

```
// babel_loader.js
export default function babelLoader(source) {
  // transpiled JS content
  return getBabelTransformedCode(source)
}
```

=> Utilisation de Babel pour de la transpilation Javascript

Tout est module : les loaders

`file-loader`

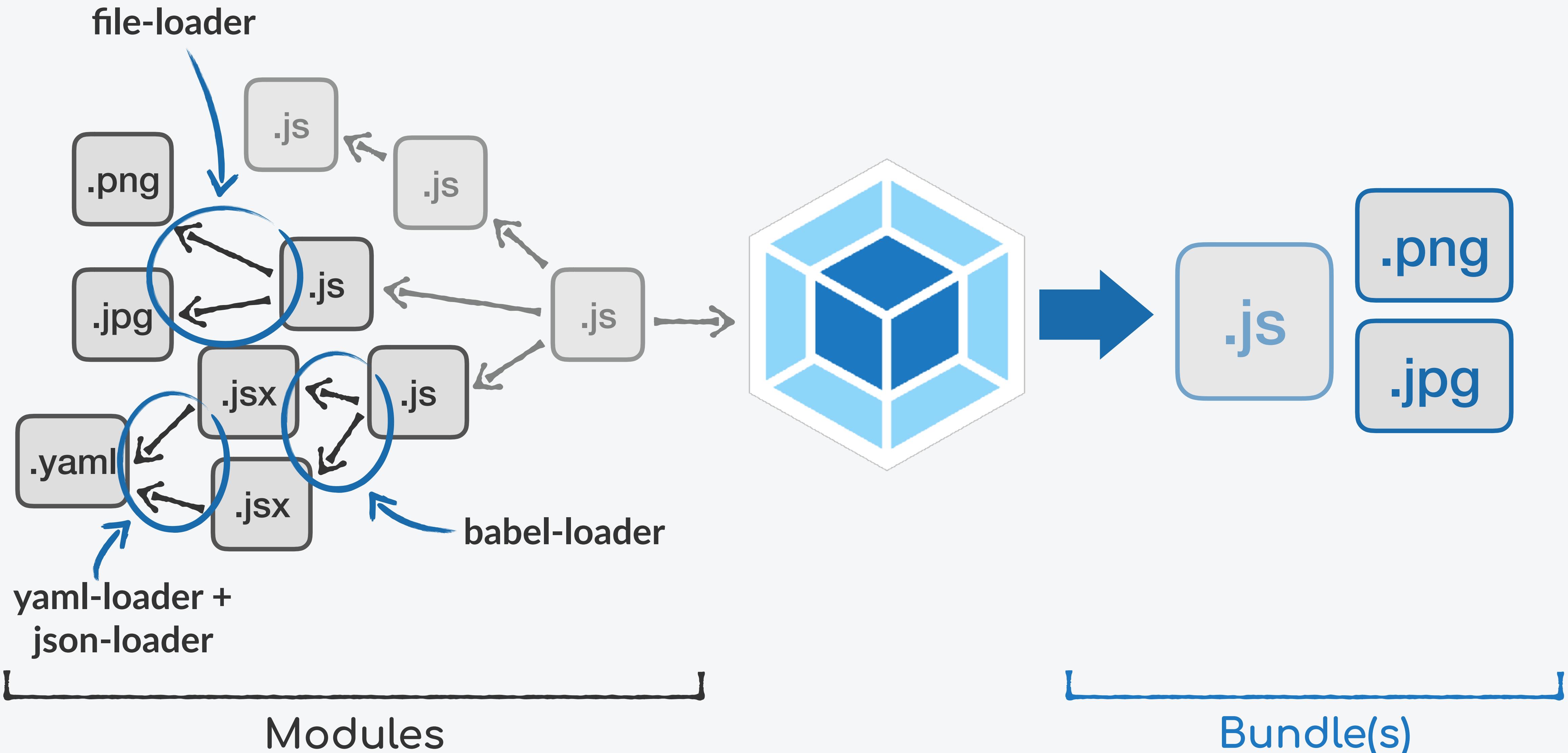
```
// file_loader.js
export default function fileLoader(source) {
  const outputPath = getOutputPath(...)
  this.emitFile(outputPath, source) // copy file to output folder
  // get output file path
  return `export default ${ outputPath }`
}
```

Tout est module : les loaders

Configuration via `webpack.config.js`

```
module.exports = {
  ...
  module: {
    rules: [
      { test: /\.png|jpg/, loader: 'file-loader' },
      { test: /\.yaml/, loaders: ['json-loader', 'yaml-loader'] }
    ]
  },
  ...
}
```

Tout est module : les loaders



Exécuteur de tâches : les plugins

Exécuteur de tâches : les plugins

- Module Node, indépendant de Webpack en lui même
- Executer des tâches en parallèle de la compilation Webpack
- Permet des fonctionnalités beaucoup plus avancées qu'un *loader*
- Prend la forme d'une classe JS utilisant sa méthode `apply(compiler)` pour agir lors d'évènements Webpack (via les *hooks*) :

```
export class SayByeWhenDone {  
    apply(compiler) {  
        compiler.hooks.done.tap('Say Bye Plugin', () => {  
            console.log('Finished! Good bye! 🙌🙌🙌')  
        })  
    }  
}
```

Exécuteur de tâches : **les plugins**

- Les *hooks* sont créés via *Tapable*, boîte à outils des *hooks* Webpack
- Chaque pièce maîtresse de Webpack “étend” *Tapable* et dispose de *hooks* à chaque étape de son fonctionnement
- Ces pièces sont elles-mêmes branchées par des plugins au sein de Webpack
- Un plugin peut ajouter de nouveaux *hooks* pour être utilisés par d'autres plugins ensuite
 - => **Chaque étape de fonctionnement de Webpack peut donc être écouté par un plugin pour y ajouter ou modifier une fonctionnalité**

“Everything is a plugin! Mastering webpack from the inside out” de Sean Larkin (Vidéo)

Exécuteur de tâches : les plugins

``mini-css-extract-plugin`` (plugin + loader)

Extraire le code CSS importé pour construire le code CSS destiné au navigateur

``html-webpack-plugin``

Gérer des fichiers HTML pour le(s) *bundle*(s)

``copy-webpack-plugin``

Copier simplement des fichiers sources dans le dossier de *build* final

``webapp-webpack-plugin``

Générer automatiquement les favicons et les injecter dans les fichiers HTML de `html-webpack-plugin`

``svg-sprite-loader`` (plugin + loader)

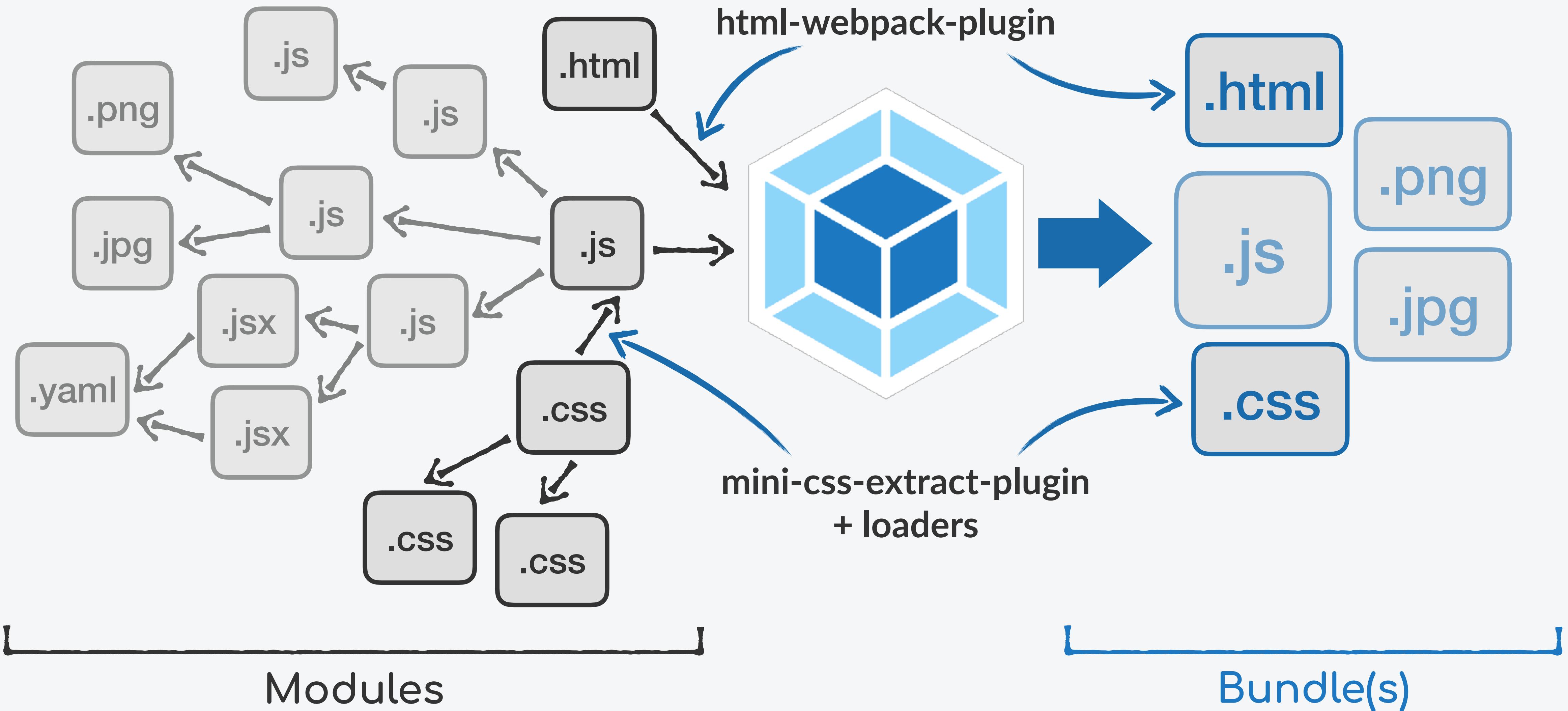
Extraire tous les fichiers SVG importés dans un seul sprite SVG

Exécuteur de tâches : les plugins

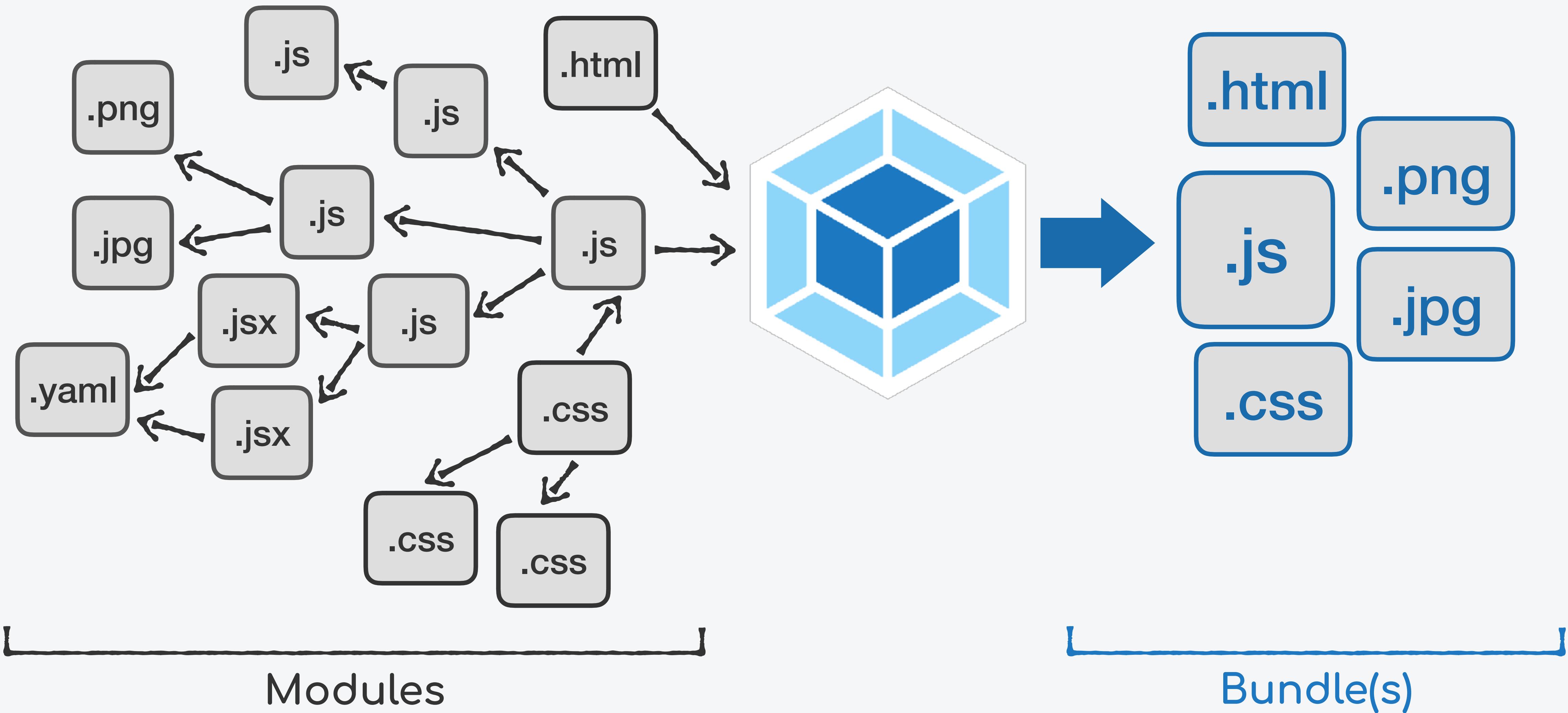
Configuration via `webpack.config.js`

```
module.exports = {
  module: {
    rules: [
      {
        test: /\.css$/,
        use: [ { loader: MiniCssExtractPlugin.loader }, 'css-loader' ]
      }
    ],
    plugins: [
      new CopyPlugin([ { from: './manifest.json' } ]),
      new MiniCssExtractPlugin({ filename: '[name].css' }),
      new HtmlWebpackPlugin({template: './src/index.html'})
    ]
  }
}
```

Exécuteur de tâches : les plugins



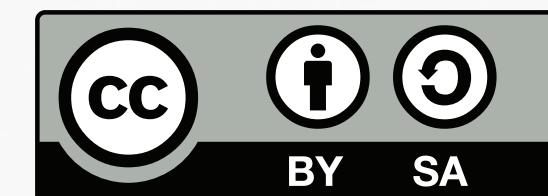
Build d'une app avec Webpack



Merci !

 github.com/CPatchane/talks

 [@CPatchane](https://twitter.com/CPatchane)



Sources images :

Logo Webpack : <https://github.com/webpack/media>

Image Fond : Photo by Leone Venter on [Unsplash](#)