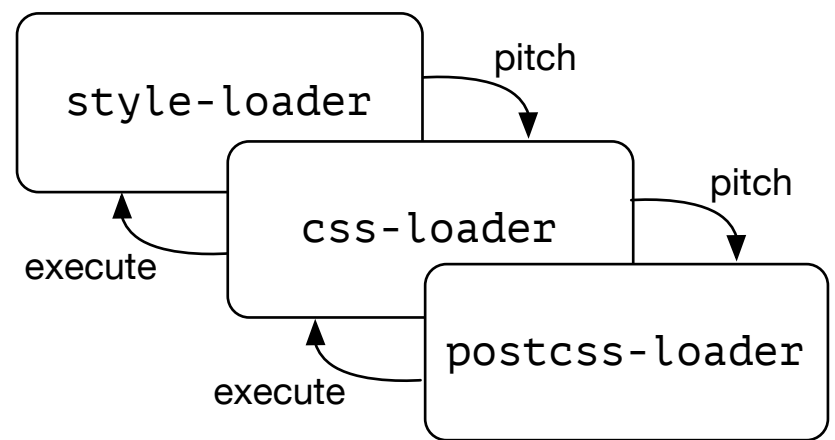


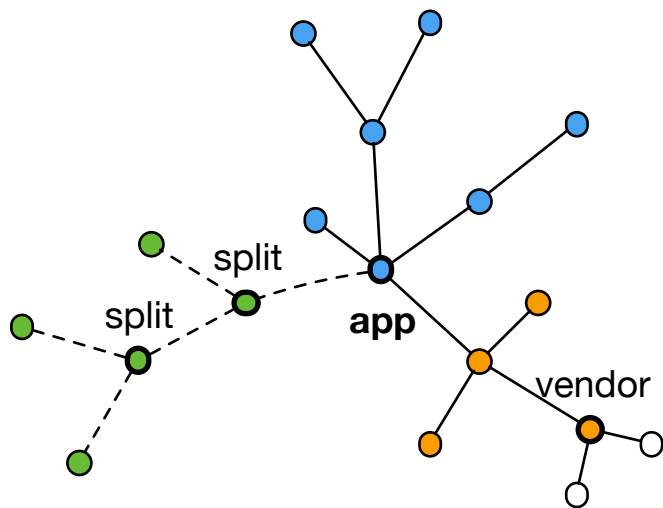
## Code splitting

```
import(
  /* webpackChunkName: "name" */
  './module'
).then(
  module => {...}
).catch(
  error => {...}
);
```

## Loader processing



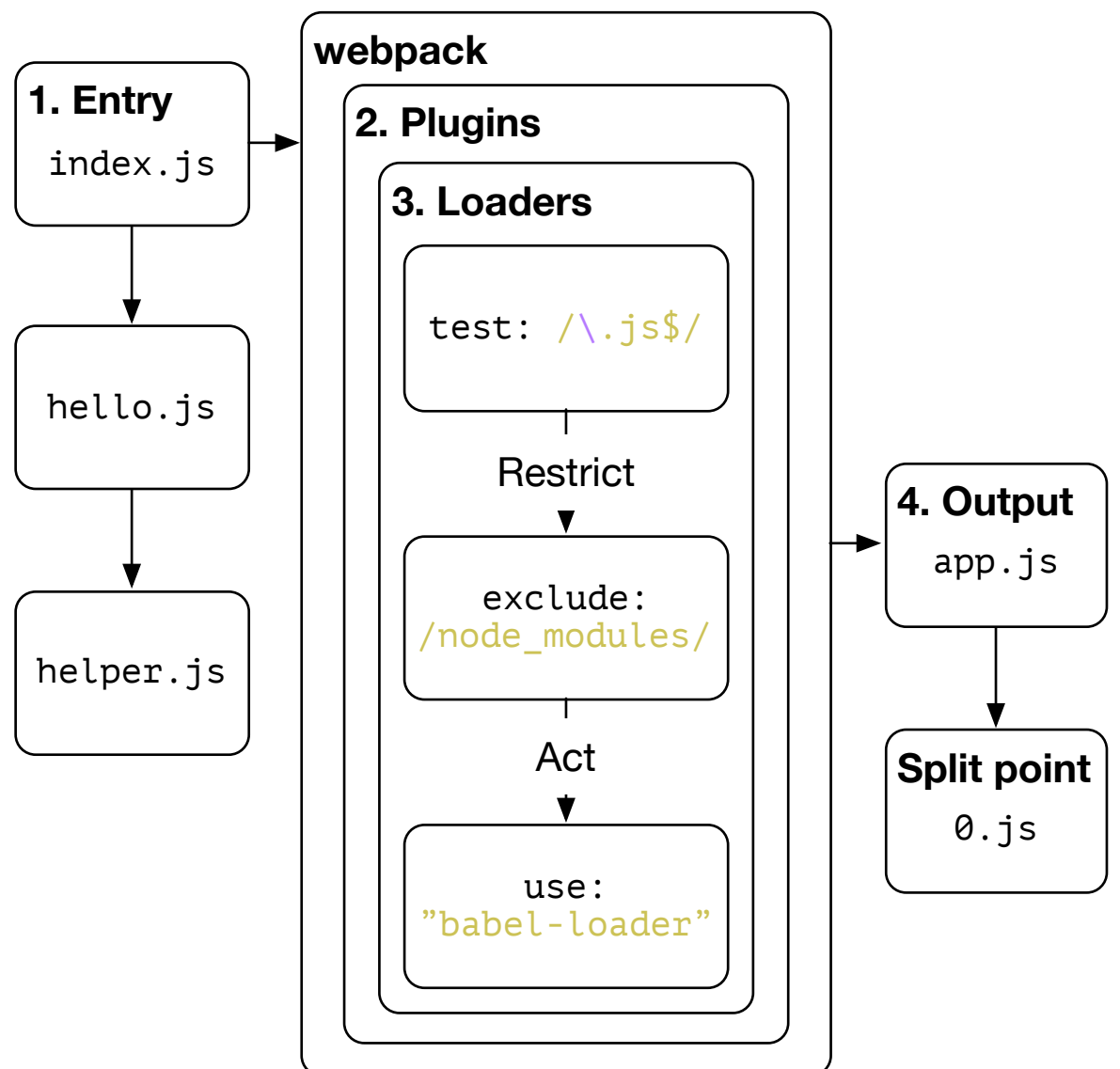
## Bundle and code splitting



## Configuration

```
module.exports = {
  // 1. Where to begin (input)
  entry: PATHS.app,
  // 4. Where to write (output)
  output: {
    path: PATHS.build,
    filename: "app.js",
  },
  // 3. How to process modules
  module: {
    rules: [
      {
        test: /\.js$/,
        use: "babel-loader",
        exclude: /node_modules/,
      },
    ],
  },
  // 2. Additional processing to perform
  plugins: [
    new webpack.DefinePlugin(...),
  ],
  // How to resolve modules and loaders
  resolve: {...},
  resolveLoader: {...},
};
```

## Process



## Bundle splitting

```
optimization: {
  splitChunks: {
    chunks: "initial",
  },
  runtimeChunk: {
    name: "manifest",
  },
},
```

```
module.exports = {
  // 1. Where to begin (input)
  entry: PATHS.app,
  // 4. Where to write (output)
  output: {
    path: PATHS.build,
    filename: "app.js",
  },
  // 3. How to process modules
  module: {
    rules: [
      {
        test: /\.js$/,
        use: "babel-loader",
        exclude: /node_modules/,
      },
    ],
  },
  // 2. Additional processing to perform
  plugins: [
    new webpack.DefinePlugin(...),
  ],
  // How to resolve modules and loaders
  resolve: {...},
  resolveLoader: {...},
};
```

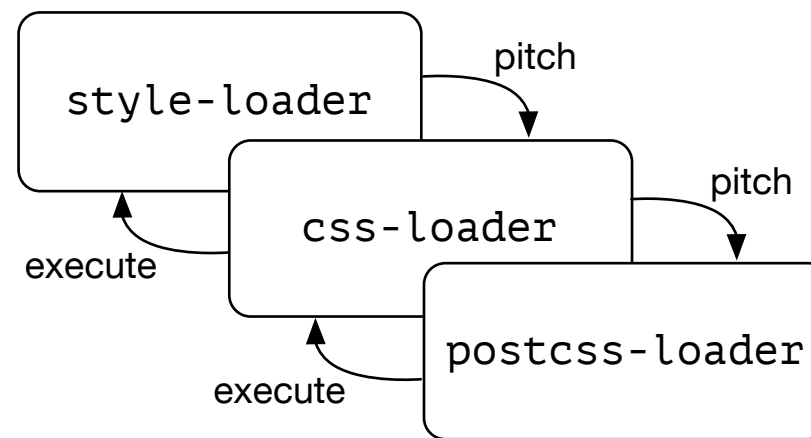
### Bundle splitting

```
optimization: {
  splitChunks: {
    chunks: "initial",
  },
  runtimeChunk: {
    name: "manifest",
  },
},
```

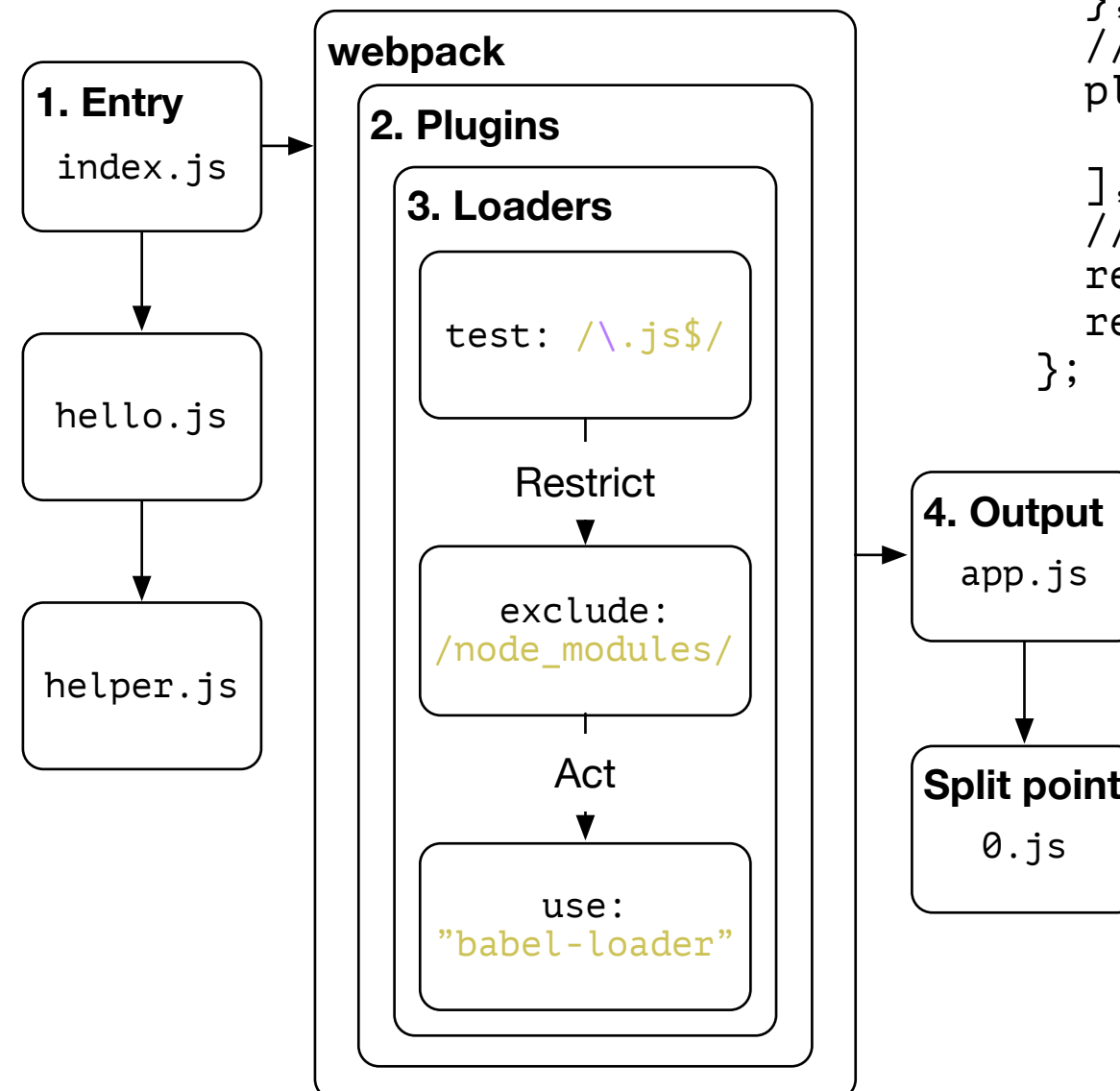
### Code splitting

```
import(
  /* webpackChunkName: "name" */
  "./module"
).then(
  module => {...}
).catch(
  error => {...}
);
```

### Loader processing



### Process



### Bundle and code splitting

