



Über den Tellerand blicken: Oder wie funktioniert eigentlich dieses Vue?

Marc Teufel



Unsere Agenda

- Kurzes Recap: Was ist eigentlich dieses Vue ? ✓
- Erste Vue-Anwendungen schreiben ✓
- Geschmacksrichtungen
 - Verschiedene Builds von Vue
 - Scaffolding mit Vite
- Komponenten und Hierarchien
- Kommunikation und Datenfluss

Verschiedene
Geschmacks-
richtungen





Verschiedene Geschmackrichtungen oder Builds

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">
  <title>JS Bin</title>
  <script src="https://unpkg.com/vue@latest/dist/vue.global.js"></script>
</head>
<body>
  <div id="clock"> {{ time }}</div>
</body>
</html>
```

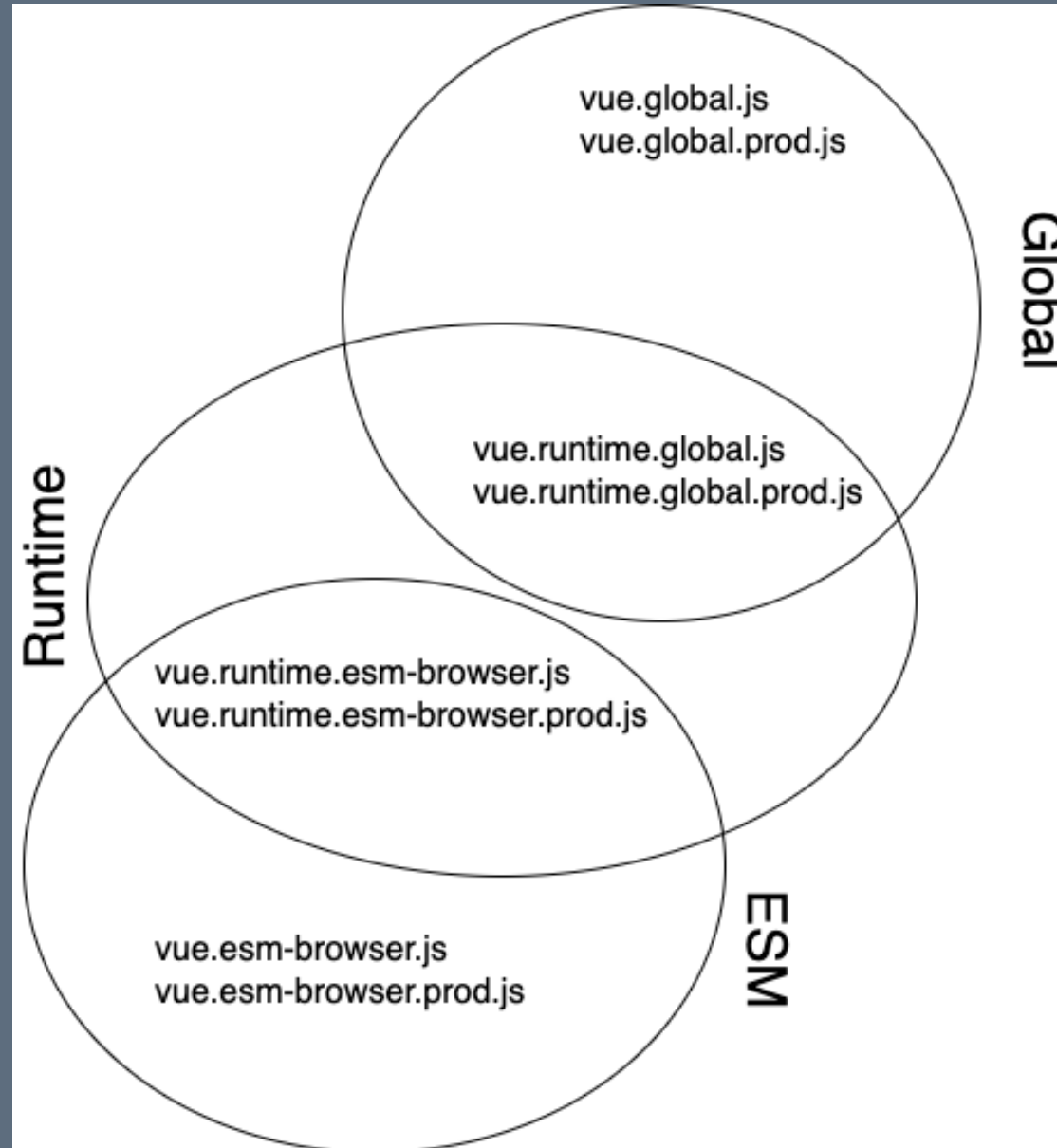


```
{
  "imports": {
    "vue": "https://sfc.vuejs.org/vue.runtime.esm-browser.js",
    "luxon": "https://moment.github.io/luxon/es6/luxon.js"
  }
}
```





Verschiedene Geschmackrichtungen oder Builds

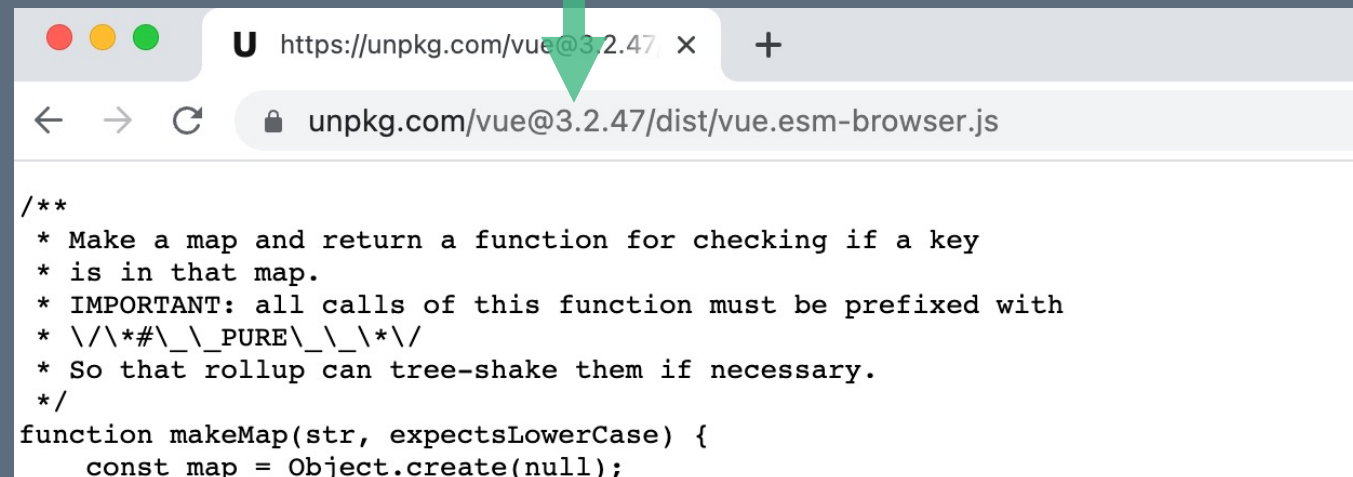




Verschiedene Geschmackrichtungen oder Builds

- <https://unpkg.com/vue@latest/dist/vue.global.js>
- <https://unpkg.com/vue@latest/dist/vue.runtime.global.js>
- <https://unpkg.com/vue@latest/dist/vue.runtime.esm-browser.js>
- <https://unpkg.com/vue@latest/dist/vue.esm-browser.js>

Links verweisen direkt auf die jeweils aktuelle Vue-Version in **dem** CDN unpkg.com:





It's
Workshop-
Time!



Empfehlungen zu Builds

- Die verschiedenen Builds sind nur interessant wenn Sie Vue irgendwo einbinden
 - Empfohlen wird auch immer die ESM-Variante zu verwenden, weil sich so gezielter einzelne Bestandteile von Vue „importieren“ lassen
 - ESM sorgt für einen modernen Programmierstil
-
- Wenn Sie mit Scaffolding-Tools arbeiten, dann nutzen Sie automatisch die optimierten Runtime-Versionen von Vue
 - **Wenn Sie neue Projekte mit Vue aufbauen, ist dies auch der empfohlene Weg!**



Vite

- Next Generation Front End Tooling
- <https://vitejs.dev/>





Warum ist ESM so wichtig?

- Standard Modulsystem, nativ (nahezu) von allen modernen Browsern unterstützt!
- Davor proprietäre Modulsysteme und require-Befehl:
 - AMD
 - CommonJS
 - UMD

Kernproblem proprietärer Modulsysteme

Egal welches Format sie einsetzen, egal ob AMD, UMD oder CommonJS - alle diese Modulsysteme bleiben proprietär, sind zueinander nicht kompatibel und können nicht direkt vom Browser geladen und ausgeführt werden!

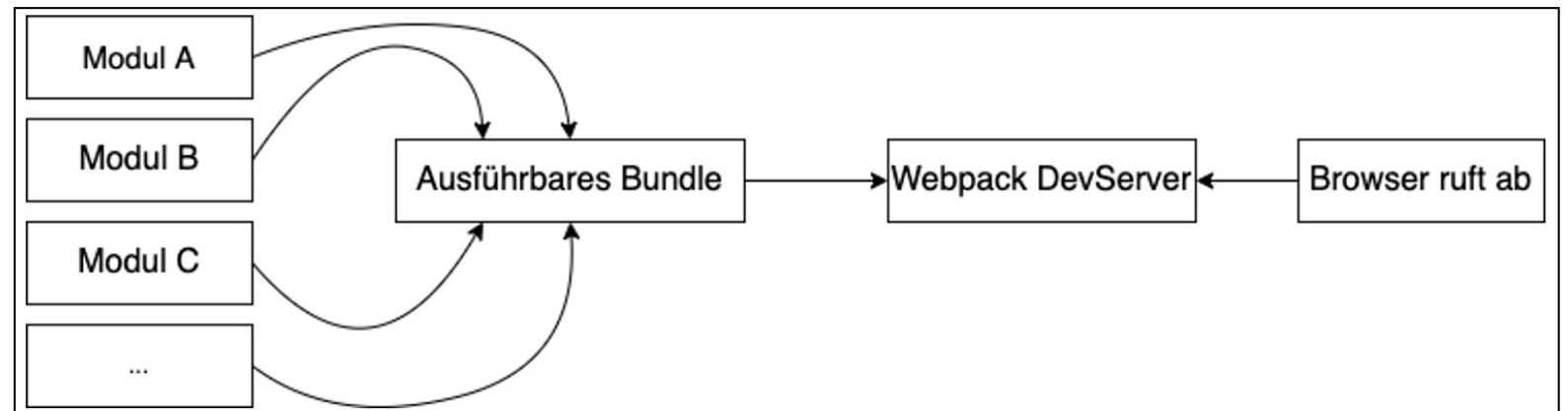
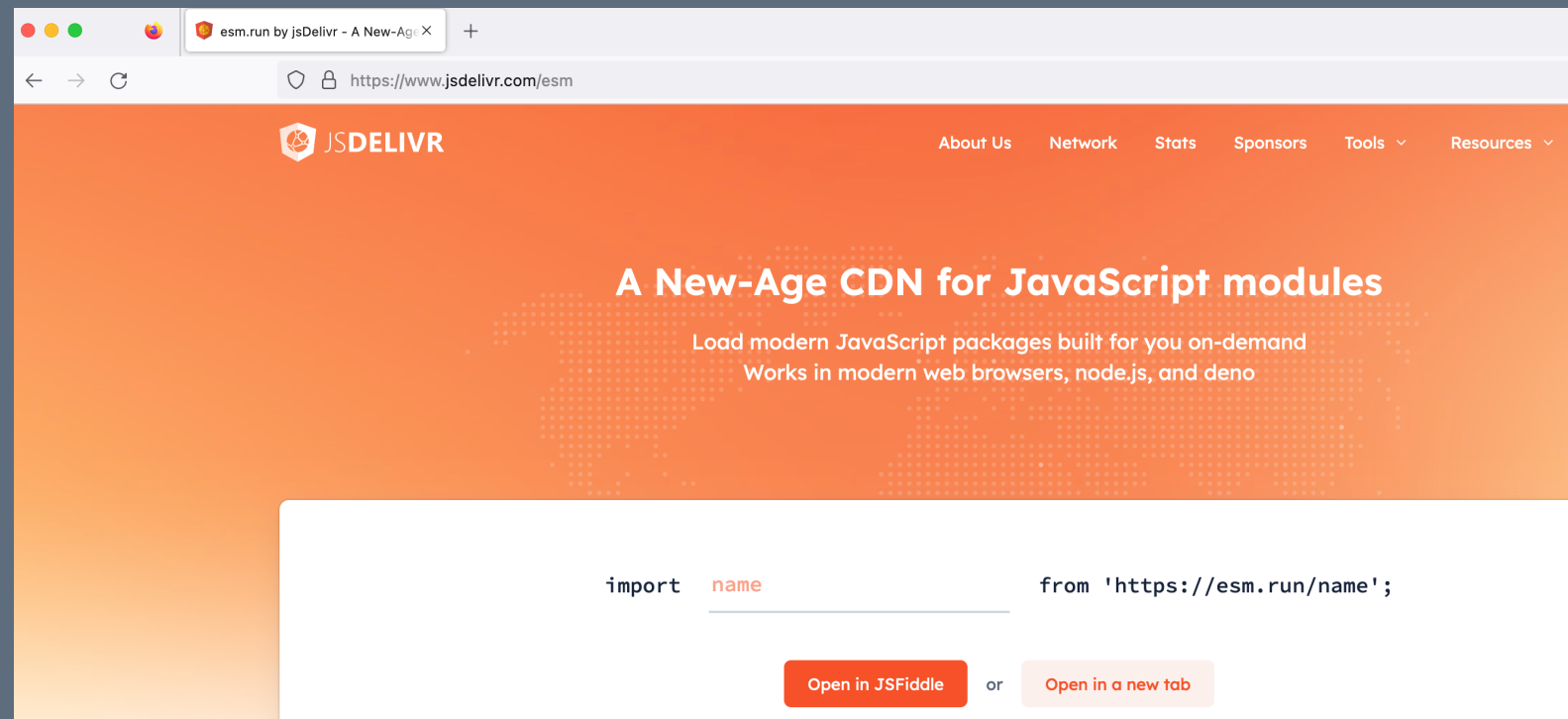


Abbildung 2.7: Klassische Build Tools wie Webpack erzeugen aus proprietären Modulen ausführbare Bundles



Was macht ESM so flexibel?

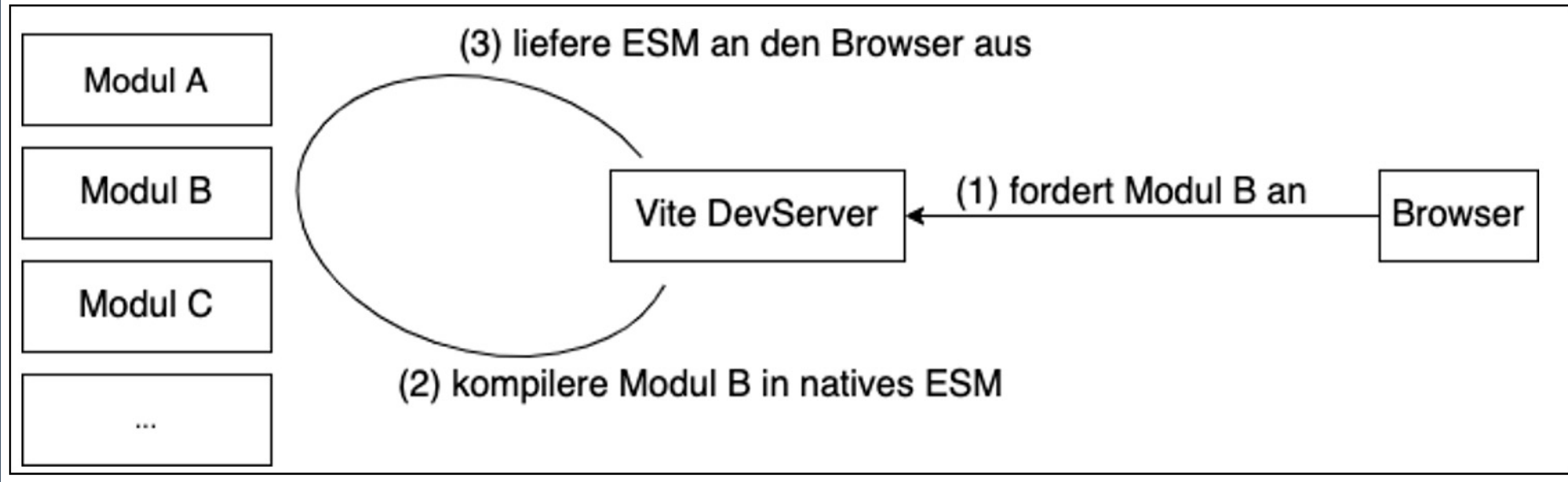
- Mit ESM können über den import-Befehl Code nativ im Browser importiert werden
- über HTTP-Protokoll
- Dynamisches Nachladen kleiner Code-Fragmente bis hin zu vielen Bibliotheken mit und ohne Import Maps sind möglich
- Der Umwandlungsprozess in ein Modul entfällt, wenn es bereits in ESM programmiert
- Alternativ konvertiert Vite es nach ESM
- CDN → <https://www.jsdelivr.com/esm>





Wie funktioniert Vite?

Im Unterschied zu Bundlern wie Webpack oder Parcel erzeugt Vite für jede Quellcode-Datei natives ESM und keine Bundles mehr. Diese nativen ES-Module sind außerdem über einen HTTP-Request vom Vite-eigenen Entwicklungsserver (DevServer) durch den Browser abrufbar.





It's
Workshop-
Time!