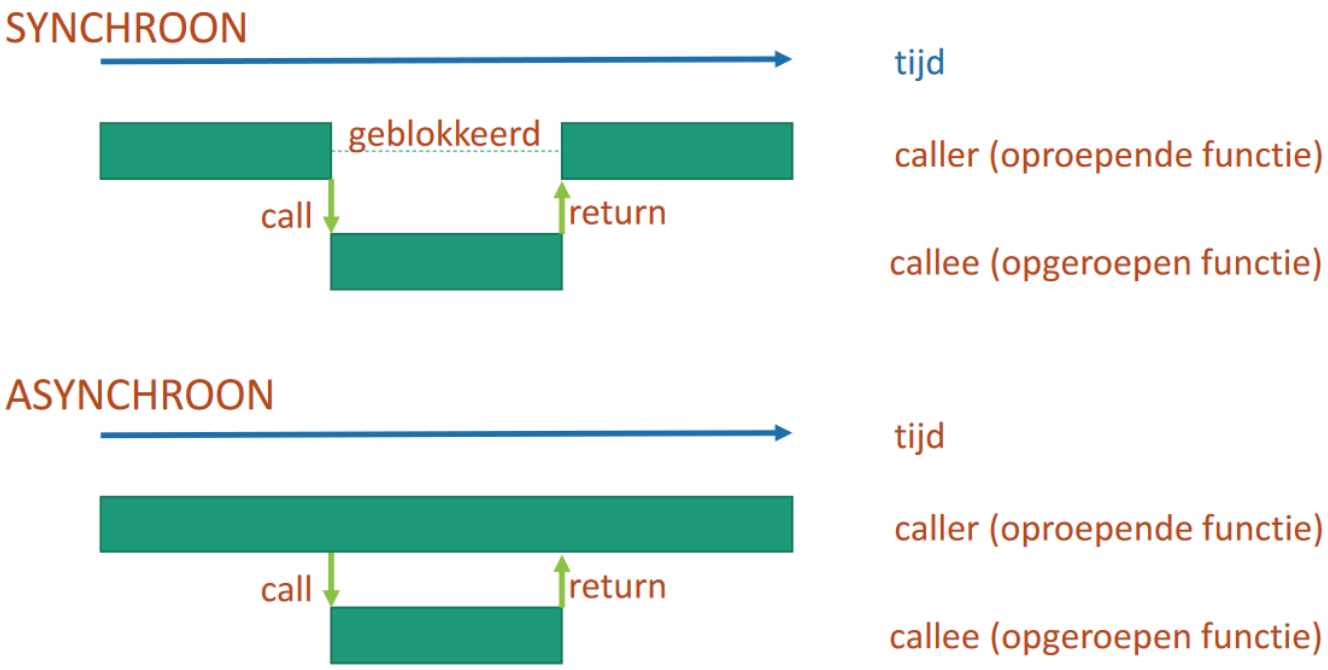
JavaScript - asynchroon programmeren

Voorstelling



Synchroon vs. asynchroon

```
// Synchroon
result = query('SELECT * FROM posts WHERE id = 1');
do_something_with(result);
// Asynchroon
query('SELECT * FROM posts WHERE id = 1', do_something_with);
```

Callback hell

- Veel functies met callbacks die genest zijn

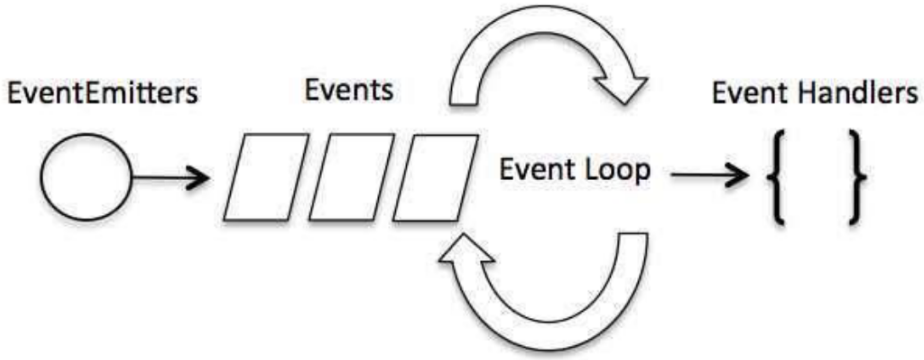
Promises

- Oplossing voor callback hell
- Promise = object toekomstig resultaat asynchrone functie
  - Drie toestanden: nog niet afgerond, vervuld, fout opgetreden
- Asynchrone functie
  - Functie keert volledig terug
  - Caller wacht niet op volledig uitvoeren functionaliteit

- Promise laat toe om callback of foutafhandeling toe te voegen

```
let p = new Promise((resolve, reject) => {
    let kans = Math.floor(Math.random() * 2);
    console.log(kans);
    if (kans === 0) resolve("Gelukt");
    else reject("Fout");
});
p.then((tekst) => console.log(tekst)).catch((fout) => console.log(fout));
```

Asynchroon gerealiseerd



Asynchrone functies

- Gebruik Promises vereenvoudigen
- Vanaf ECMA 7
- Het resultaat van een asynchrone functie is een Promise
- Er kan gebruik gemaakt worden van een await om op een bepaalde waarde te wachten