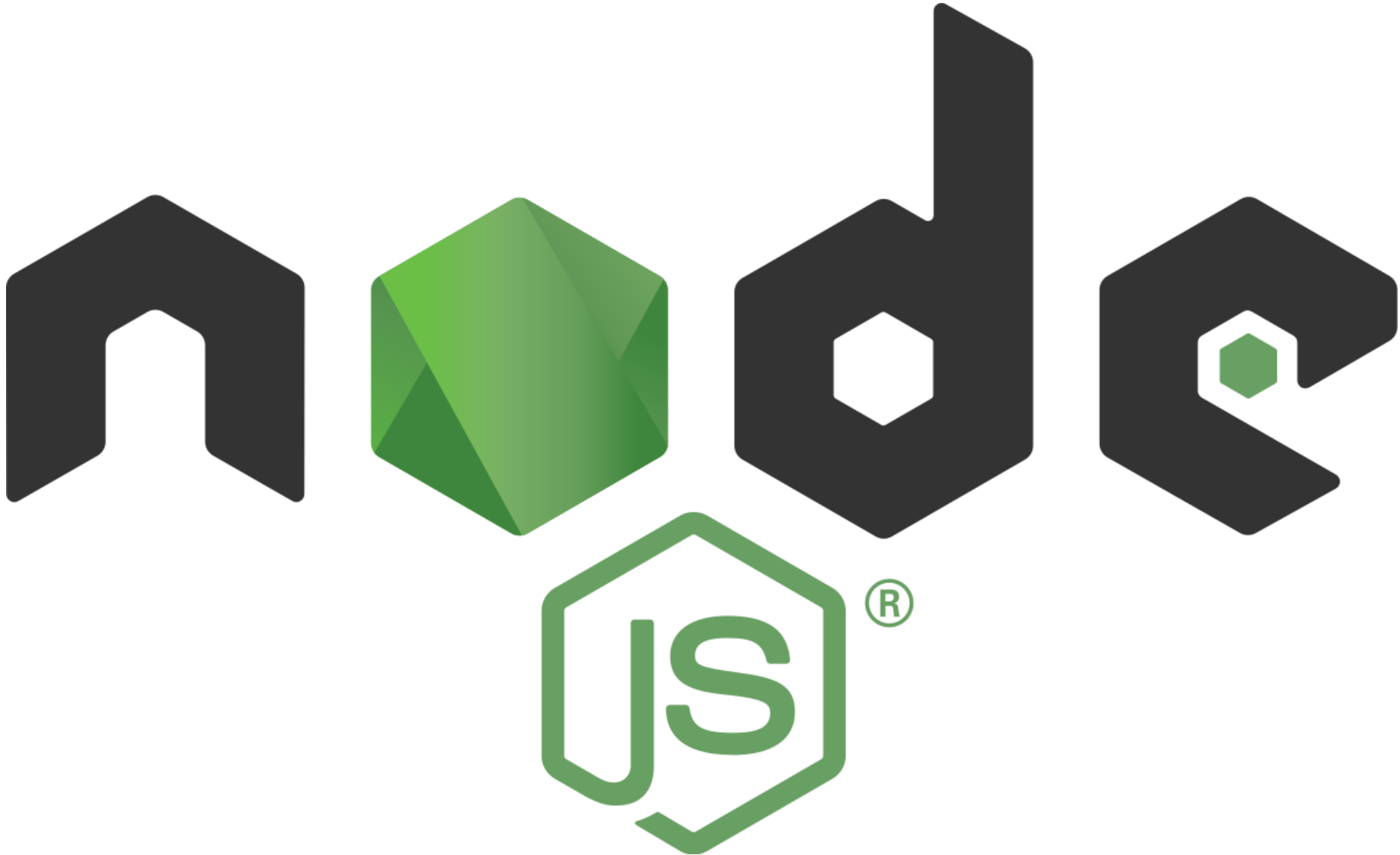
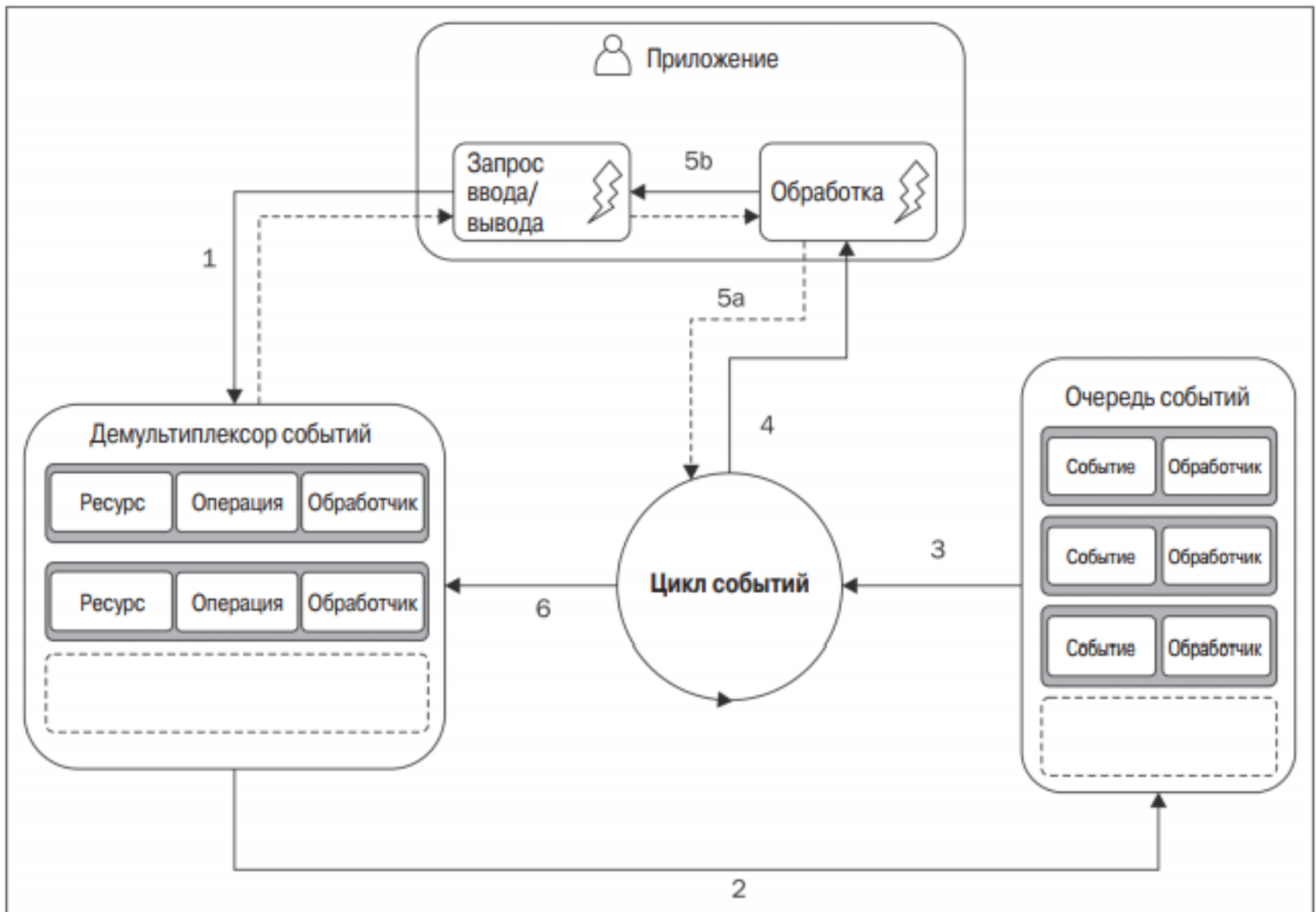


Event Loop



ШАБЛОН REACTOR



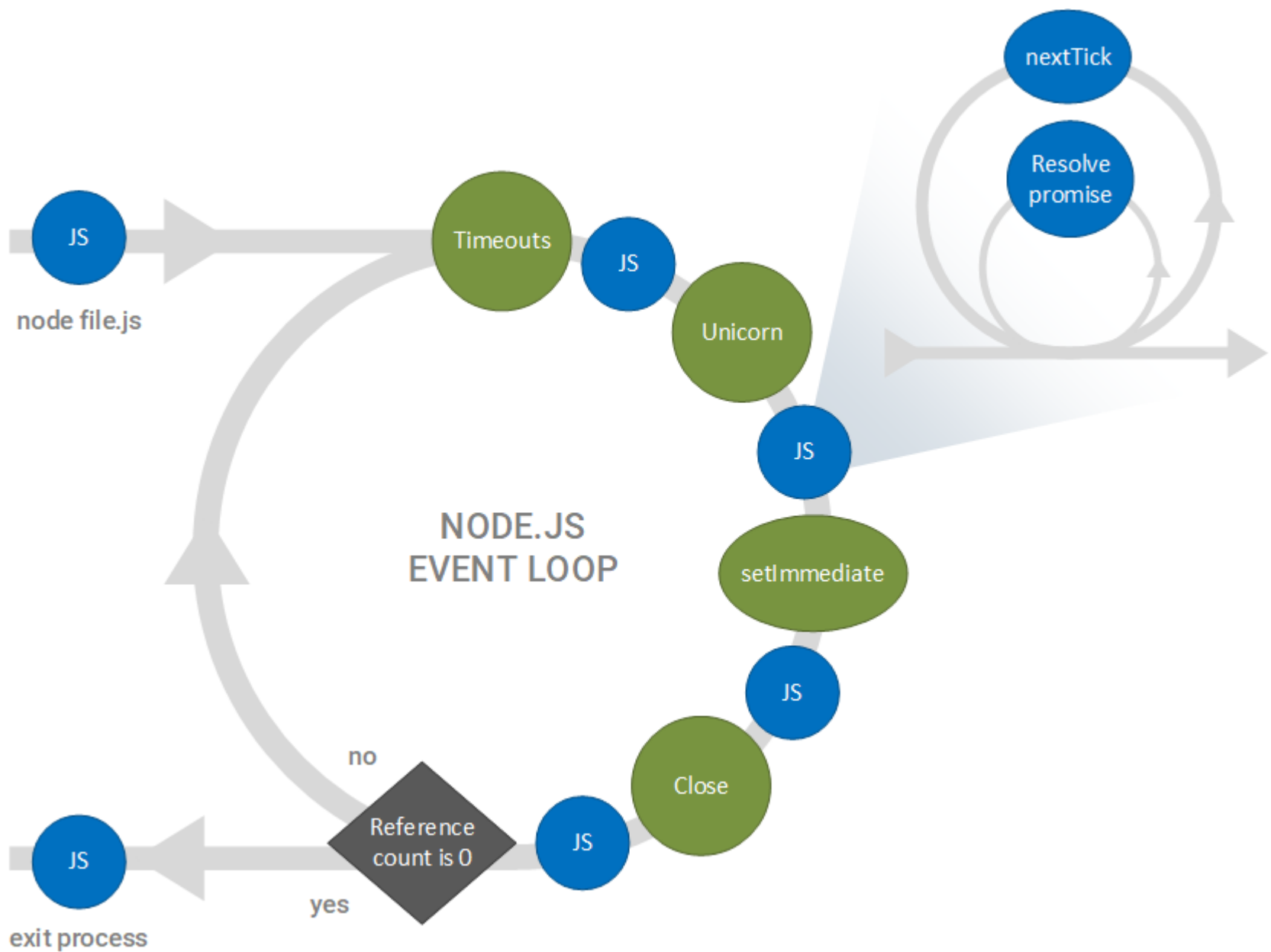
НЕПРЕДСКАЗУЕМАЯ ФУНКЦИЯ

Одной из наиболее опасных ситуаций является наличие программного интерфейса, который в одних условиях работает синхронно, а в других – переходит в асинхронный режим:

```
const fs = require('fs');
const cache = {};
function inconsistentRead(filename, callback) {
    if(cache[filename]) { //вызывается синхронно
        callback(cache[filename]);
    } else { //асинхронная функция
        fs.readFile(filename, 'utf8', (err, data) => {
            cache[filename] = data;
            callback(data);
        });
    }
}
```

ВЫСВОБОЖДЕНИЕ ЗАЛГО

```
function createFileReader(filename) {  
    const listeners = [];  
    inconsistentRead(filename, value => {  
        listeners.forEach(listener => listener(value));  
    });  
    return { onDataReady: listener => listeners.push(listener) };  
}  
  
const reader1 = createFileReader('data.txt');  
reader1.onDataReady(data => {  
    console.log('First call data: ' + data);  
    const reader2 = createFileReader('data.txt');  
    reader2.onDataReady( data => {  
        console.log('Second call data: ' + data);  
    });  
});
```



ПОЛЕЗНЫЕ ССЫЛКИ

<https://dzone.com/articles/introduction-to-nodejs-3?fromrel=true>

- An Introduction to Node.js (Part 1) (англ)

<https://blog.risingstack.com/node-js-at-scale-understanding-node-js-event-loop/>

- Understanding the Node.js Event Loop (англ)