



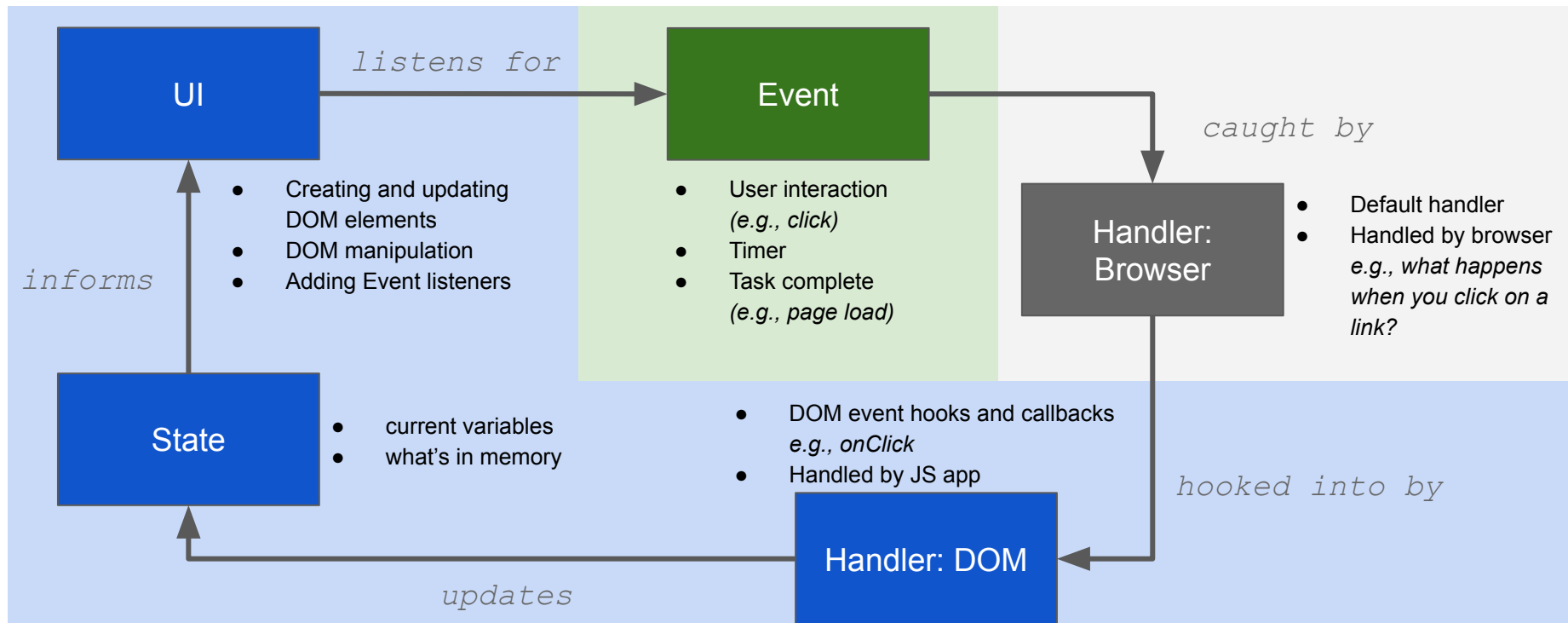
Web APIs, Event Loop and State

Web Development
Lesson 4.1



Event loop

Handling user interaction



Event loop

Who has control?

Our application:

- Javascript
- HTML
- CSS

- User
- Timer
- External data sources
- Browser

Browser

State

Three levels of state

01

In memory

- Variables in Javascript
 - Data attributes on elements
1. *Very volatile*
 2. *Quick to access*
 3. *Won't persist over a page load*
 4. *Short term*

02

Browser API

- localStorage
 - sessionStorage
 - indexedDB
 - Cookies
 - Cache API
1. *Persist over page load*
 2. *Stuck in browser*

03

Remote

- Server-side file storage
 - MySQL
 - MongoDB
1. *Over internet (not stuck in browser)*
 2. *"Slow" to access*
 3. *Long term*

State

Uses for each level

01

In memory

Application State:

- variables
- Data attributes
- React Context
- Redux

02

Browser API

Application State:

- localStorage
- indexedDB

Session State:

- sessionStorage
- Cookies

Resource State:

- Cache API

03

Remote

Session State:

- Database

Resource State:

- Files