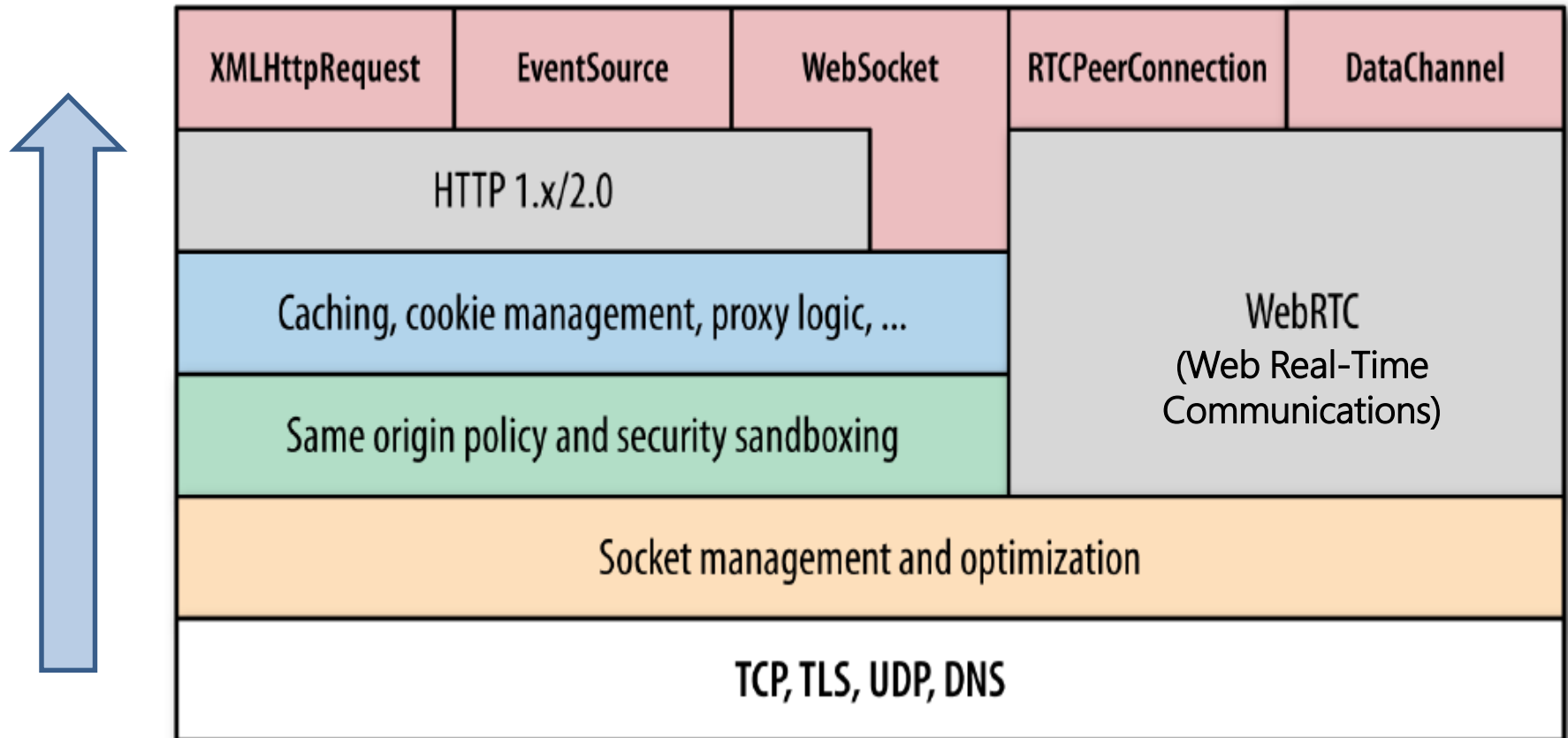


Modul 9:
**Web APIs (XHR, SSE,
WebSockets)**



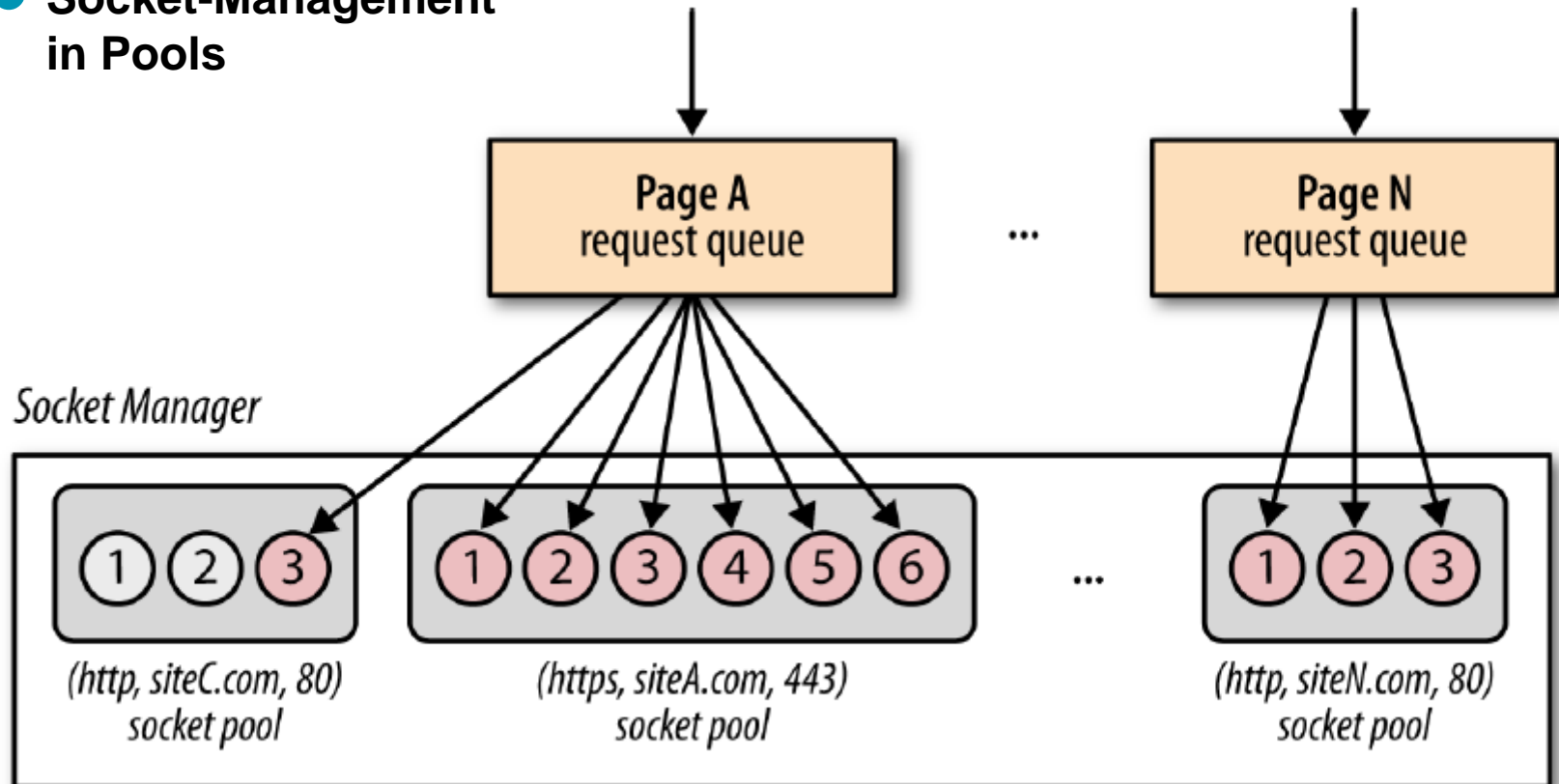
Browser Networking APIs, Protocols, and Services - Einordnung



Copyright © 2013 [Ilya Grigorik](#). Published by O'Reilly Media, Inc. Licensed under [CC BY-NC-ND 4.0](#)

Verbindungsmanagement und Optimierung

- **Socket-Management in Pools**





Verbindungsmanagement und Optimierung

- **Netzsicherheit + Sandboxing**
 - TLS-Aushandlung
 - Same-Origin Policy
 - Resource-Management
- **Verbindungsstatus (Authentifizierung, Cookies) über mehrere Tabs managen.**
Beispiel: Ausloggen auf einem Tab bewirkt ausloggen auf allen anderen offenen Tabs
- **Client-State-Caching**
 - Automatisches Cachen von Ressourcen.
 - Automatisches Auffrischung abgelaufener Ressourcen.
 - Größe des Caches automatisch managen.



XMLHttpRequest (XHR)

- **Programmierschnittstelle für JavaScript.**
- **Setzt auf HTTP und XML (und auch anderen Formaten) auf.**
- **Grundbaustein der [Ajax](#)-Technologie, um interaktive Webseiten zu programmieren.**
(Ajax = Asynchronous JavaScript and XML)
- **Asynchrone Hintergrund-Kommunikation mit dem Server unter voller Kontrolle von JavaScript.**
Ziel: Transaktionen im Hintergrund.
- **Die Hauptarbeit bei XHR – i.e. die Einbettung in JavaScript - macht der Browser.**
- **Mit XHR ist es möglich, einen Request an den Server zu schicken, ohne die Seite komplett neu laden zu müssen.**
Hierdurch können neue Infos vom Server abgerufen und in die vorhandene Seite eingebaut werden.



Server-Sent Events (SSE)

- **Teil von HTML5**
- **Protokoll, um Textdaten (Events) über HTTP in Realtime an den Client zu streamen.**
- **Der Server pushed Daten asynchron an den Client, ohne vorher vom client angepollt werden zu müssen.**
- **Mögliche Anwendungen:**
 - **Monitor für laufende Statistikdaten, z.B. Herzschlag+Blutdruck bei Patienten-Monitoring im Krankenhaus.**
 - **Anwendung für Meßdatenüberwachung**
 - **Chatanwendung**



WebSocket

- **Bidirektionales Streaming von Text und binären Daten zwischen Client und Server.**
- **Hierzu bauen WebSockets eine persistente TCP-Verbindung zwischen Client und Server auf.**
- **Durch WebSockets wird ein leistungsfähiger Rückkanal von Server zu Client auf einfache Weise möglich.**
- **Durch WebSockets wird kein Anwendungsprotokoll vorgegeben. Dieses muss zwischen den Anwendungen vordefiniert sein bzw. bidirektional ausgehandelt werden.**
- **Gute geeignet für:**
 - Implementierung von Spielen,
 - kundenspezifische Applikationen.
- **Spezifiziert in RFC6455, The WebSocket Protocol, 12/2011**

Kommunikationsfluss von XHR, SSE und WebSocket

