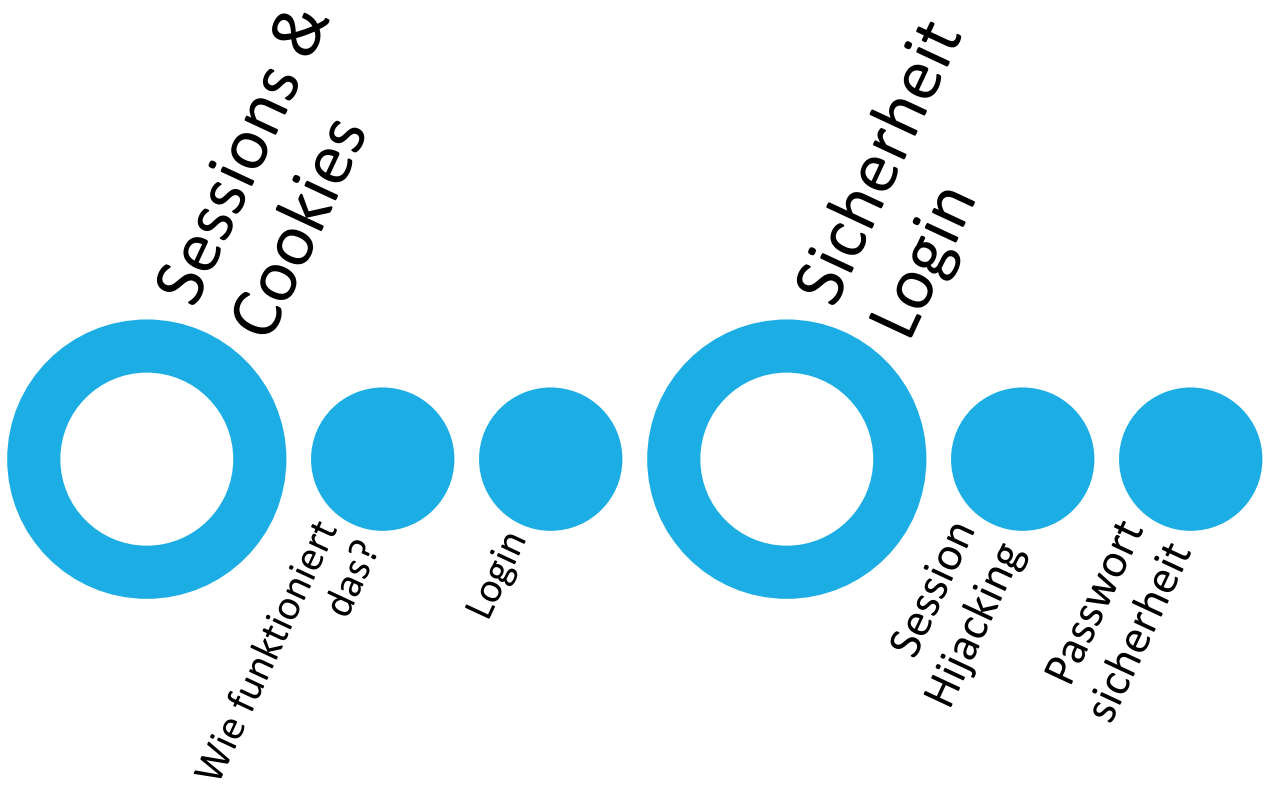
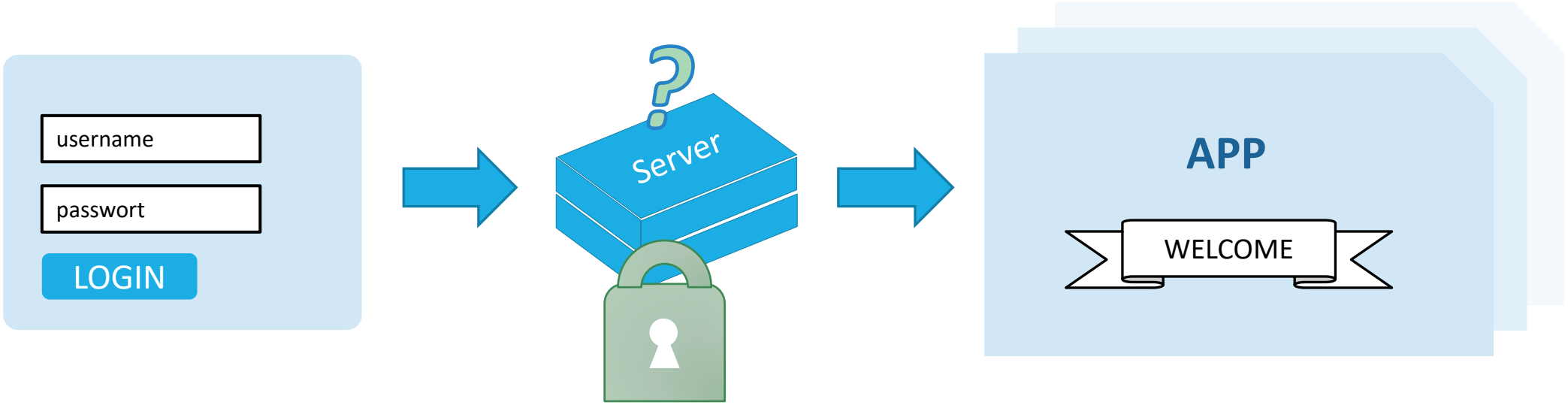
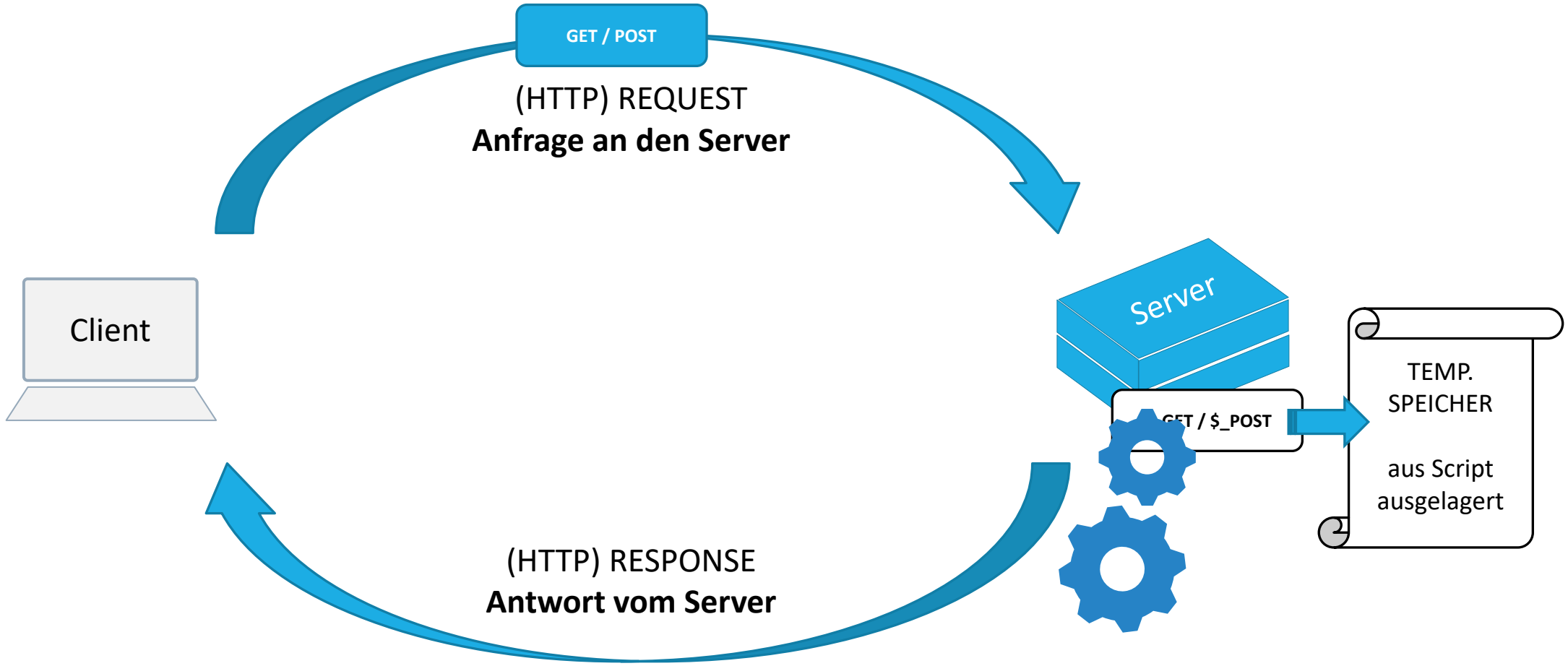
Programm diese Woche



Login



Datenverfügbarkeit in Server-Client Kommunikation



CLIENT	SERVER

Labor Übung

Zeit: 10min

Beobachte die Cookies mit einem Cookie Manger

Nutze (installiere) dafür ein Cookie Browser-AddOn (z.B. Chrome: EditThisCookie)

- wann sind sie da, wann nicht mehr?
- Woran erkennt man, ob eine Session existiert?

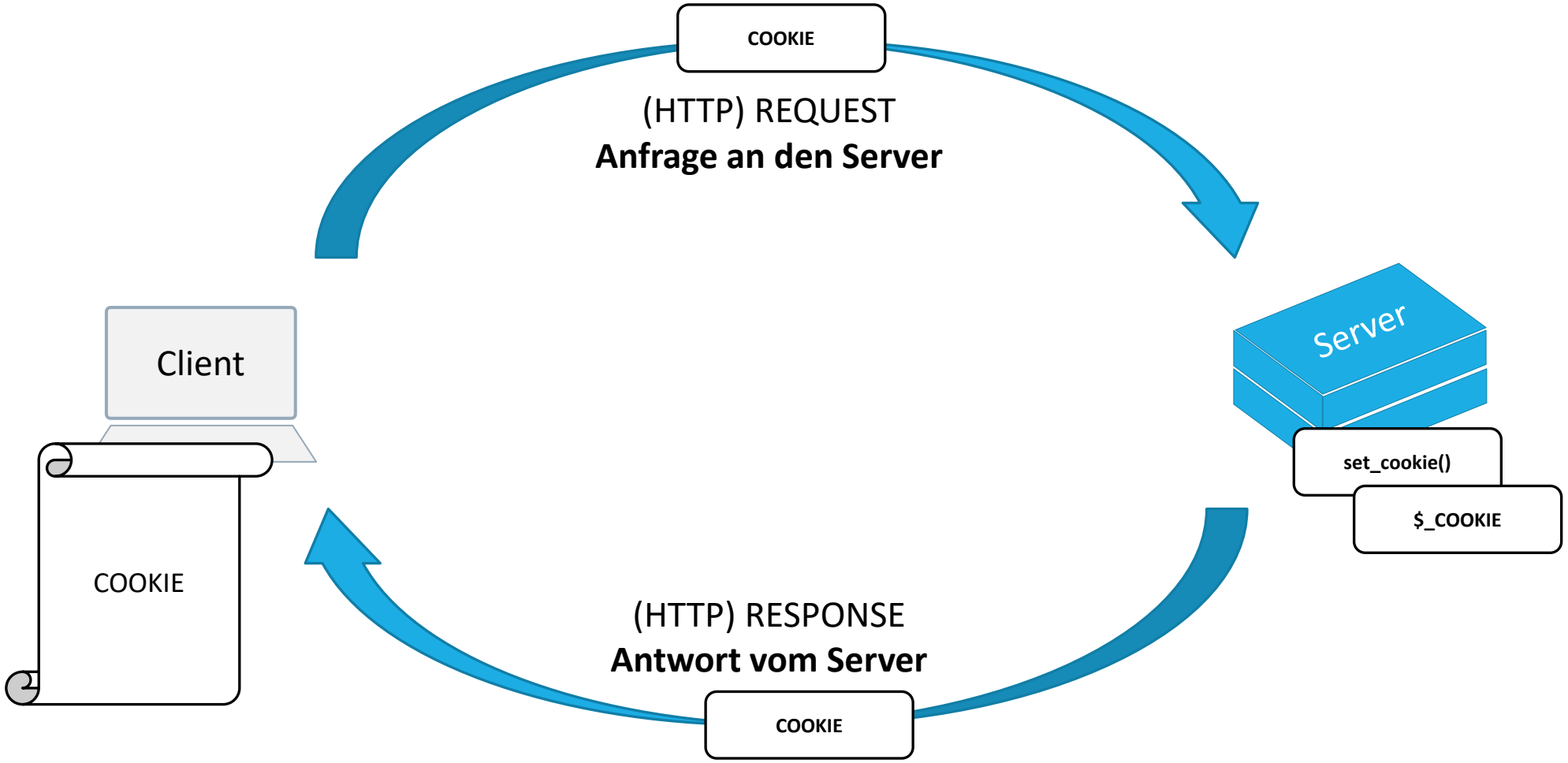
Arbeit: in Kleingruppen

Kompetenz

«einen sicheren, passwortgeschützten Bereich mit Login aufbauen unter Berücksichtigung der spezifischen Gefahren»

- Sessions und Cookies für temporäre Datenspeicherung nutzen
- Passworte sicher ablegen
- Gefahren durch Sessions und Cookies kennen
- Geeignete Massnahmen anwenden, um Gefahren durch User Input zu minimieren

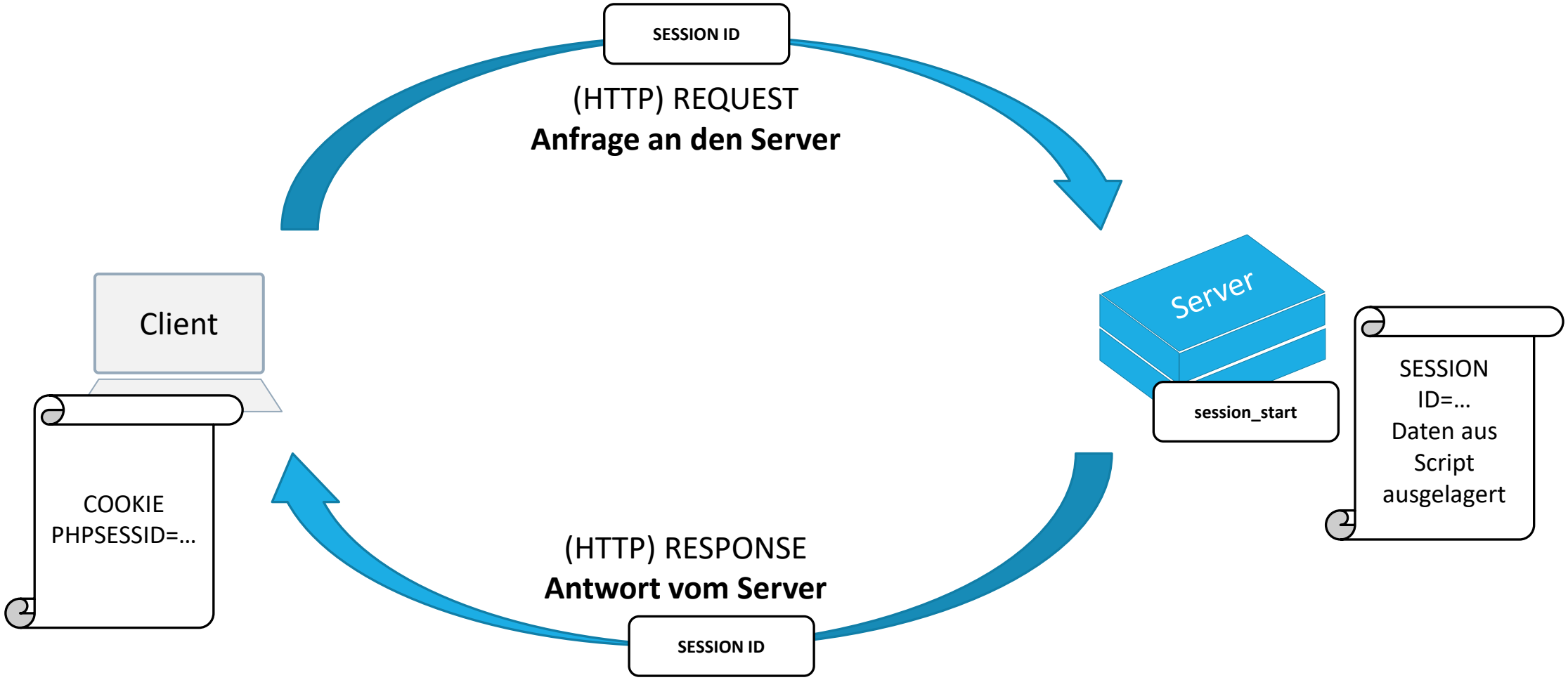
Cookie & Server-Client Kommunikation



CLIENT: Cookie

SERVER: Cookie verwalten / auslesen

Session & Server-Client Kommunikation



CLIENT: Session (ID) Cookie	SERVER: Session-Daten

Session Hijacking

entführen einer Session ID – eigentlich Session ID Hijacking

Ein Hacker…

1. liest Session ID (mittels Spyware oder Sniffing)
2. legt auf seinem Computer ein Cookie mit der Session ID an
3. ruft Applikation über diesen Computer auf (Request mit Sesssion ID) und gibt sich als das Opfer aus

Session Fixation

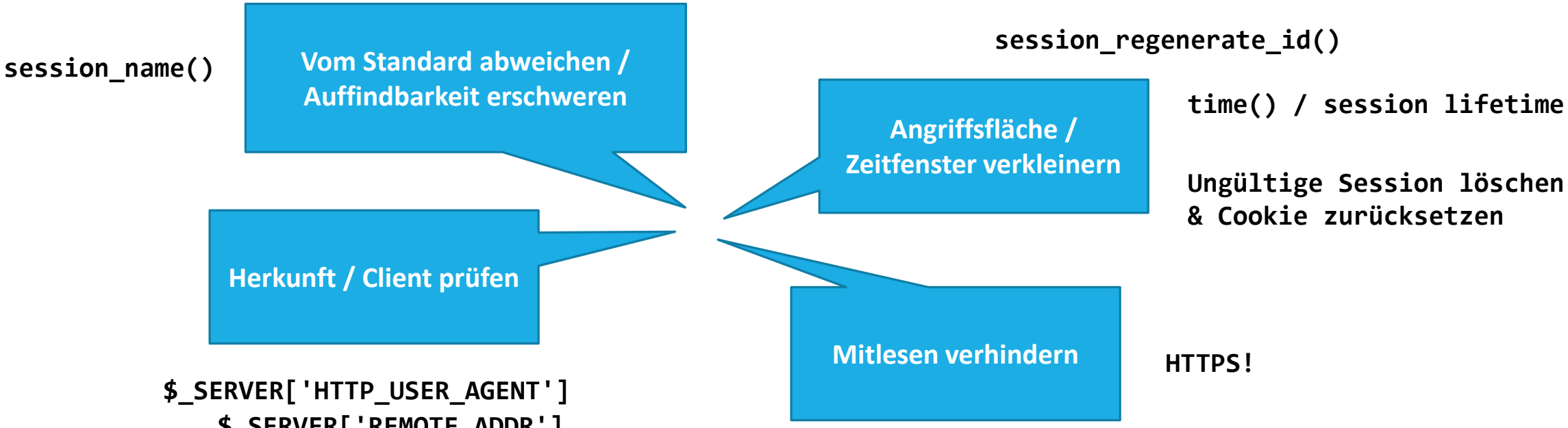
Vorpräparieren einer Session ID – Session ID unterjubeln

Ein Hacker…

1. legt auf seinem Computer ein Cookie mit einer Session ID an
2. Verführt den ahnungslosen Benutzer, sich über ein präpariertes Formular an der Applikation anzumelden (z.b. via Phishing E-Mail)
3. Benutzer authentifiziert die schon existierende Session-ID mit seinem Login, statt eine eigene Session ID zu generieren
4. Hacker hat Zugriff

Empfohlene Prinzipien & Technik

Diese Massnahmen sind als Bausteine zu verstehen. Die Kombination erschwert einen Angriff deutlich



Quelle: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/WebSec/WebSec.pdf

Passwort Sicherheit

Problem: Daten sind NIE 100% sicher

1. Userdaten **können geklaut werden** (leak)
2. Hacker nutzen Tabellen mit bekannten Passwort / Hashkombinationen sog. **Hash Tables / Rainbow Tables**
3. Nach einem Leak: Passwort-Hashes in den Tabellen finden, um ihr Klartext-Äquivalent zu erhalten

Massnahme: Password Hashing

Ziel: Passworte schützen, damit sie nicht gelesen werden können

1. Passworte werden **NIE im Klartext gespeichert**
2. Hash darf **keinen Hinweis auf Klartext-Äquivalent** geben
3. **Einweg-Hashing-Algorithmus** (nicht umkehrbar)
   -> vgl. Verschlüsselung (umkehrbar)
4. Hashing mit **Salt** (=unterschiedliches Ergebnis für jeden Hashvorgang)

Hashing vs. Encryption

HASHING

- Hashing = Verschleierung
- Nicht reversibel (kann nicht umgekehrt werden)
- Gleicher Algorithmus führt zu gleichem Hash
- Hash kann überprüft werden

Anwendungen: Prüfsummen (wurde Content verändert?) / Passworte

ENCRYPTION

- Encrypting = Verschlüsselung
- Reversibel (Originalzustand kann wiederhergestellt werden)
- Schlüssel beeinflusst Algorithmus
- Entschlüsselung nur mit bekanntem Schlüssel

Anwendungen: E-Mail, sensible Daten, Sichere übermittlung (HTTPS)

