

PHP & MySQL

DYNAMISCHE WEBSITES ERSTELLEN

TERRY HARKER, TERRY.HARKER@BYTEKULTUR.NET

Kompetenz-Ziel

Eine datenbankbasierte Anwendung in PHP so programmieren, dass bei einem SQL-Injection-Angriff keine Datenverarbeitung ausgeführt wird.

Kenntnisse und Fertigkeiten

- Gefahrenquellen für Injections in der Applikation erkennen
- MySQLi Prepared Statements anwenden
- Risiken kennen und Relevanz einschätzen

SQL-Injection: Risiko #1 in Datenbank-Applikationen

WIE PHP ALS «AUTOR» DES SQL STATEMENTS AUSGENUTZT WIRD

HI, THIS IS
YOUR SON'S SCHOOL.
WE'RE HAVING SOME
COMPUTER TROUBLE.



OH, DEAR - DID HE
BREAK SOMETHING?

IN A WAY -



DID YOU REALLY
NAME YOUR SON
Robert'); DROP
TABLE Students;-- ?



OH, YES. LITTLE
BOBBY TABLES,
WE CALL HIM.

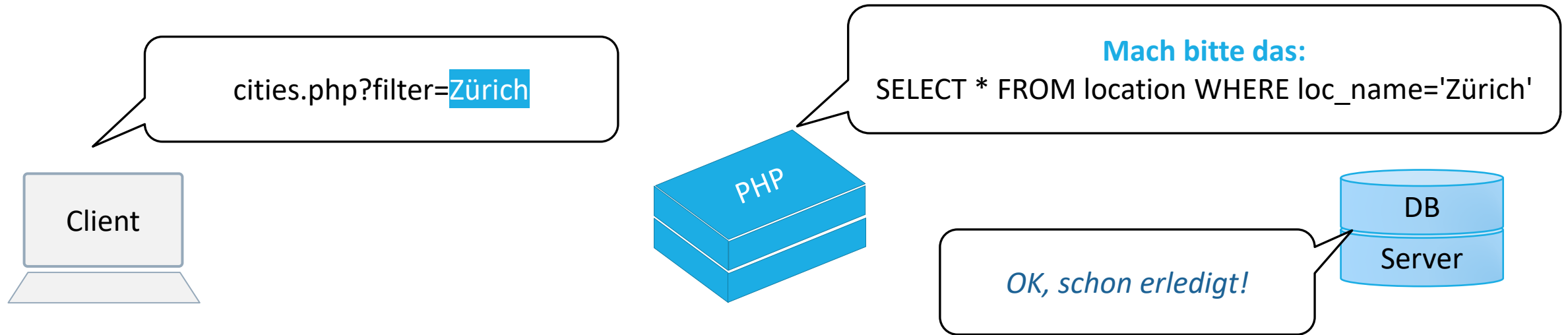
WELL, WE'VE LOST THIS
YEAR'S STUDENT RECORDS.
I HOPE YOU'RE HAPPY.



AND I HOPE
YOU'VE LEARNED
TO SANITIZE YOUR
DATABASE INPUTS.

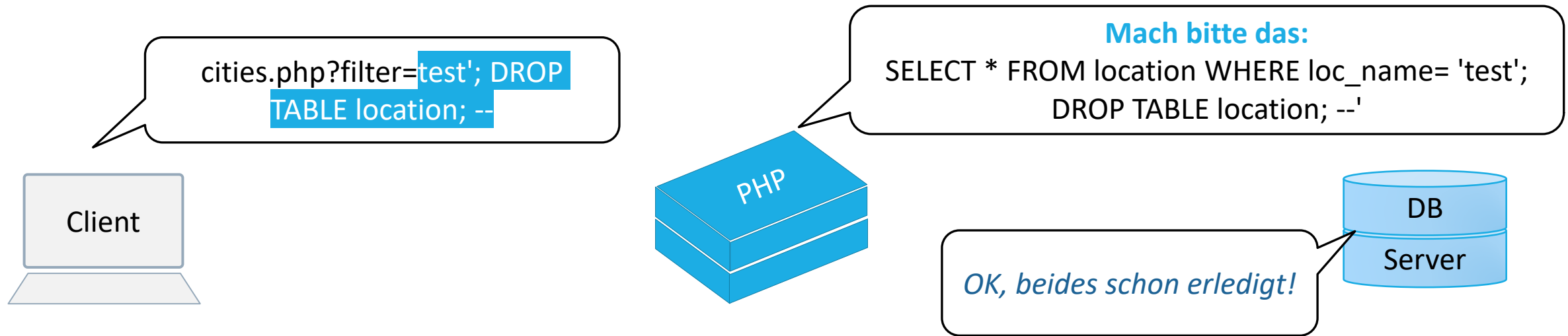
Normales SQL Statement

Kommunikation im Team Web-Technologien



Normales SQL Statement

Kommunikation im Team Web-Technologien



Problem:

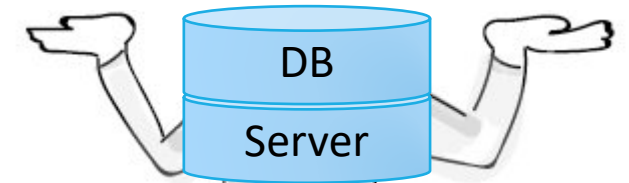
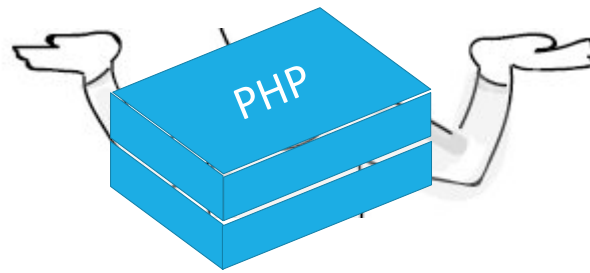
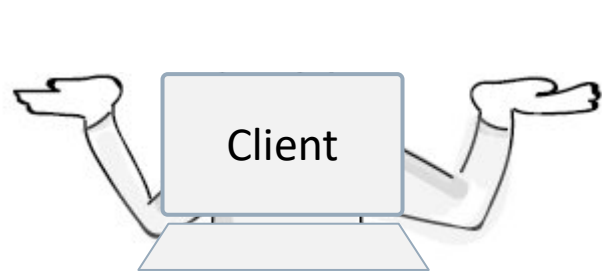
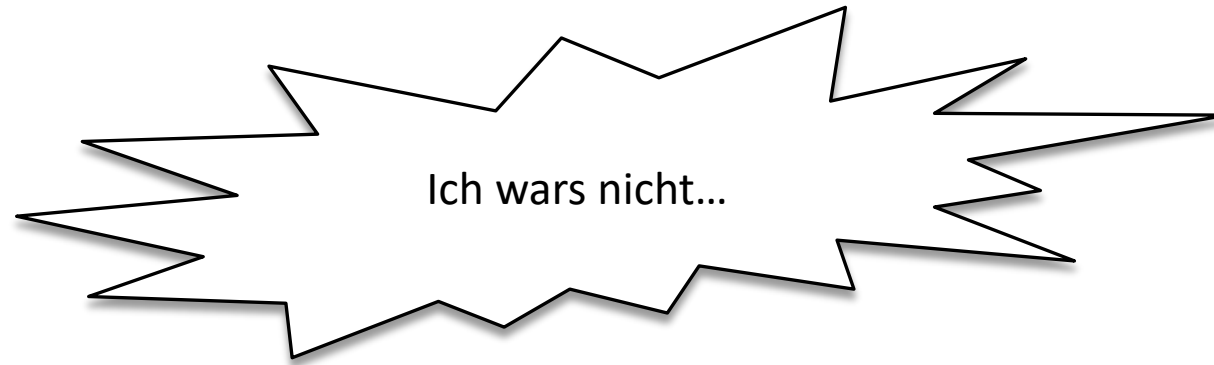
- User Input könnte SQL Code enthalten

Problem:

- Validierung nur hier möglich
- Aber: PHP versteht kein SQL

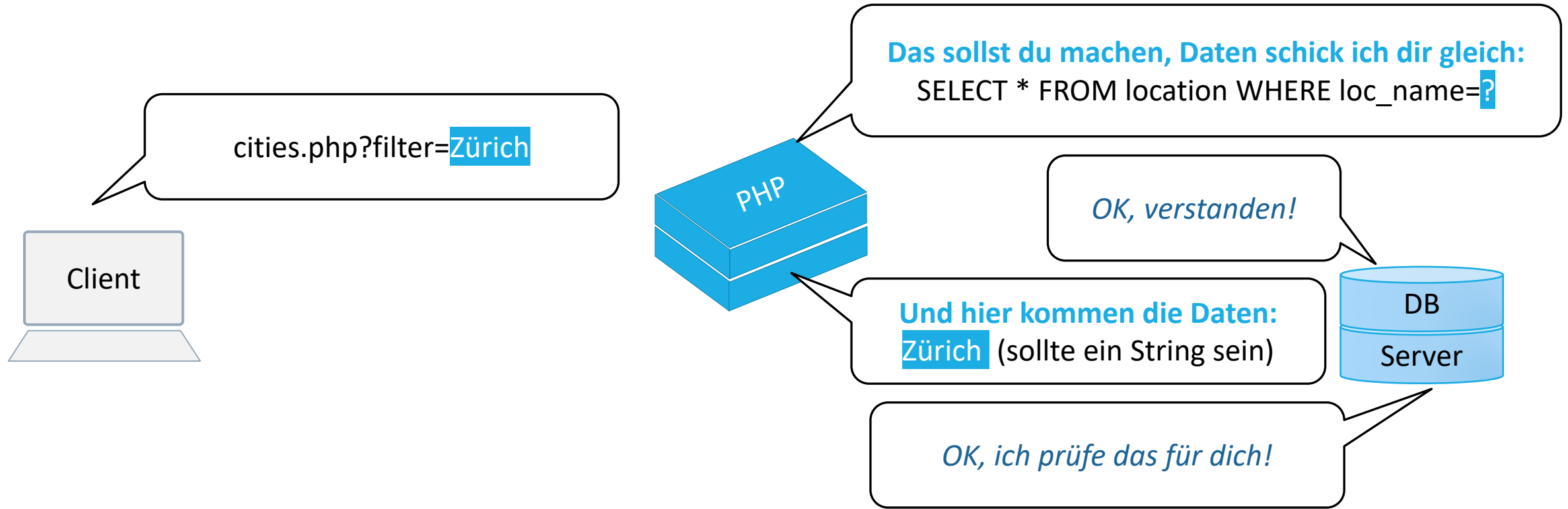
Problem:

Keine Möglichkeit, User Input zu erkennen (oder zu validieren)



Vorbereitetes SQL Statement

Intelligente Kommunikation im Team Web-Technologien



LÖSUNG:

TRENNUNG VON DATEN & LOGIK bis zum MySQL Server aufrecht erhalten.