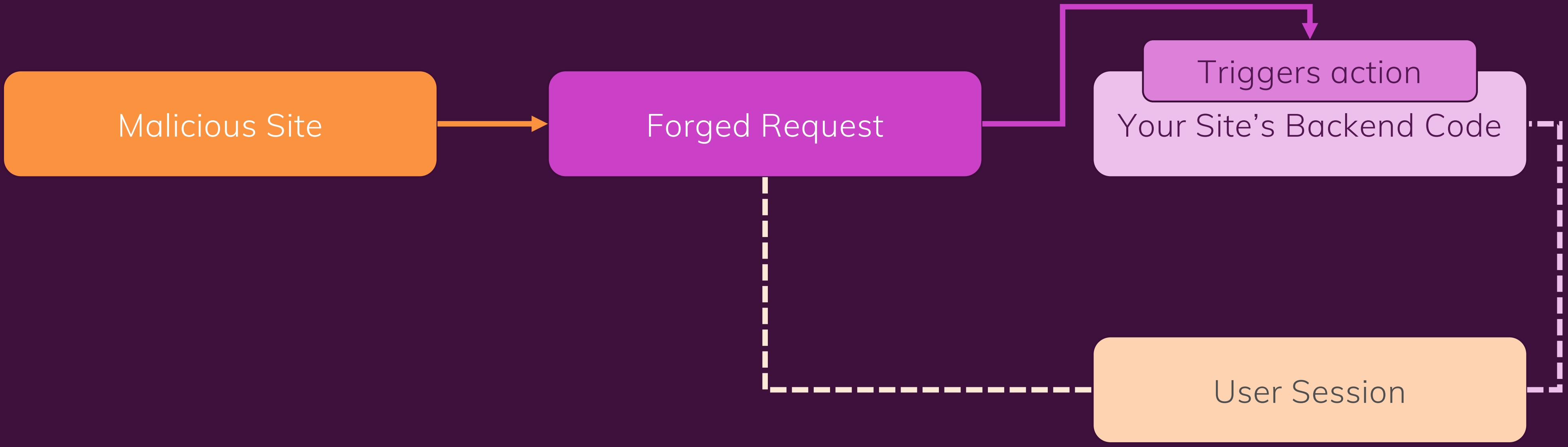
Security vs Authentication

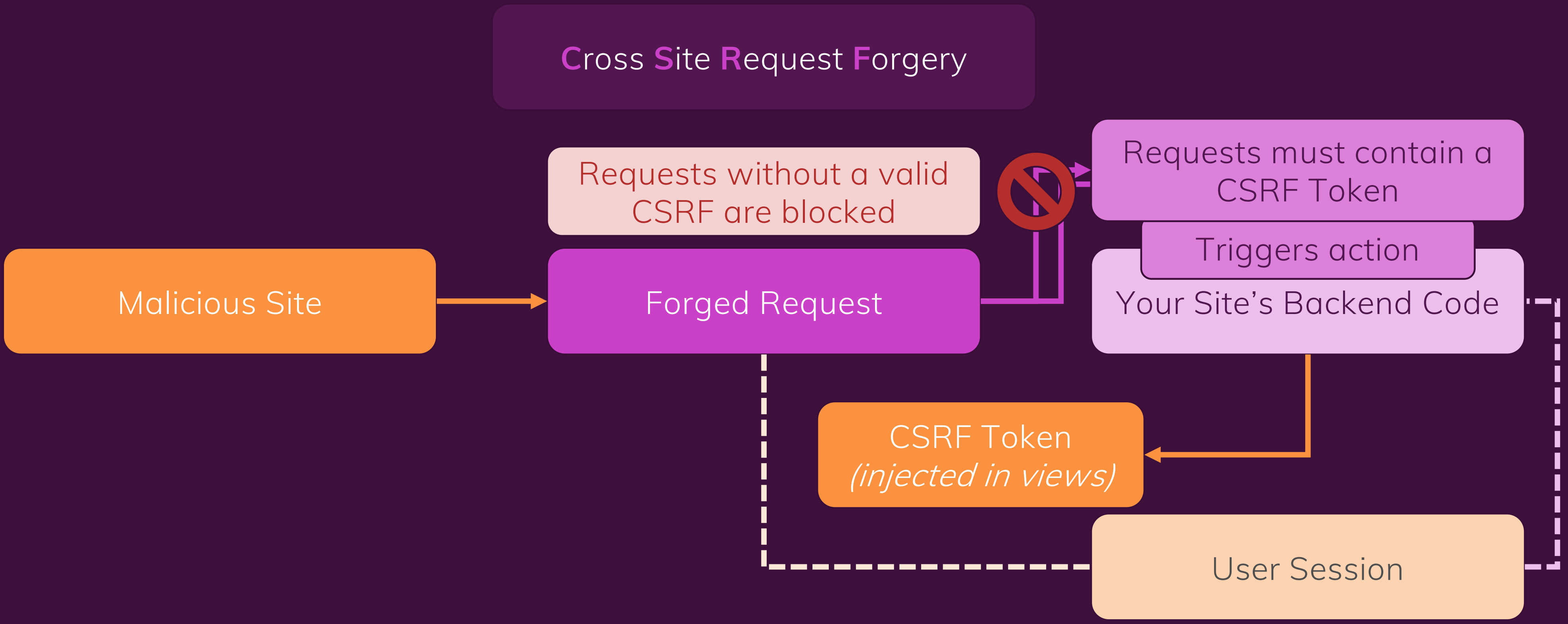
Authentication	Website Security
Control expected actions by your website visitors	Prevent unexpected (potentially malicious) actions by visitors / other people
Grant some visitors (e.g. logged in visitors) more privileges than others	Prevent exposing data or granting unwanted access to certain actions or your code

Understanding CSRF Attacks

Cross Site Request Forgery



Protecting Against CSRF Attacks



Cross Site Request Forgery

Malicious Site

Forged Request

Requests without a valid CSRF are blocked

Requests must contain a CSRF Token

Triggers action

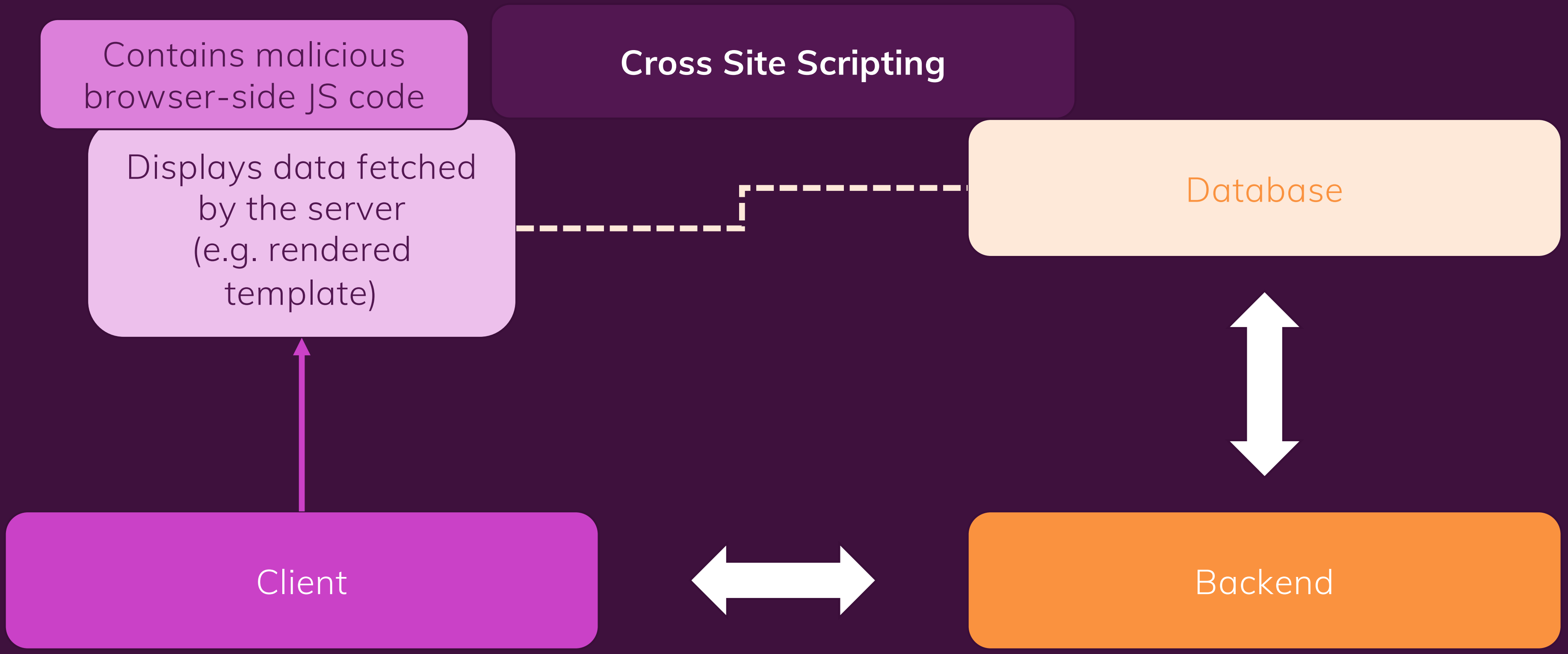
Your Site's Backend Code

CSRF Token *(injected in views)*

User Session

ACADE MIND

Understanding XSS Attacks



Cross Site Scripting

Contains malicious browser-side JS code

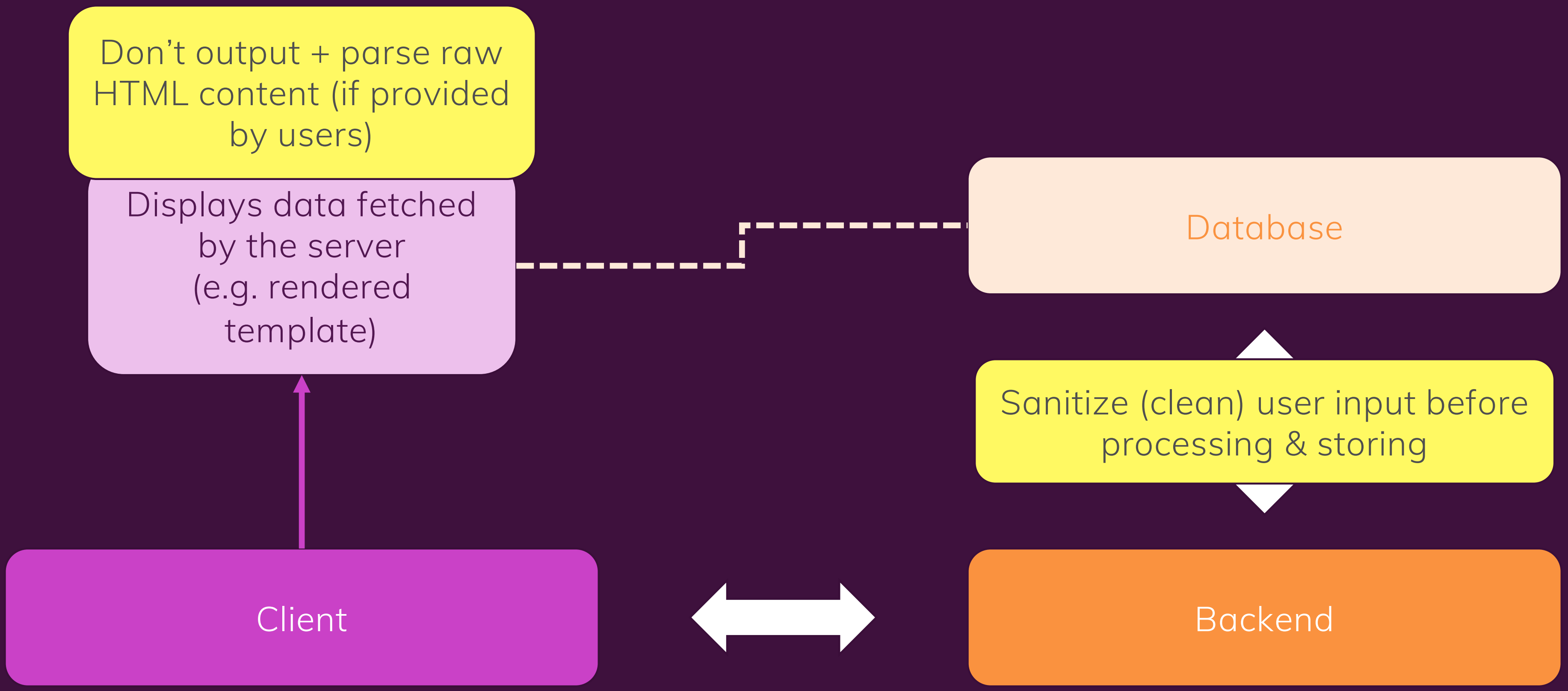
Displays data fetched by the server (e.g. rendered template)

Database

Client

Backend

Protecting Against XSS Attacks



Don’t output + parse raw HTML content (if provided by users)

Displays data fetched by the server (e.g. rendered template)

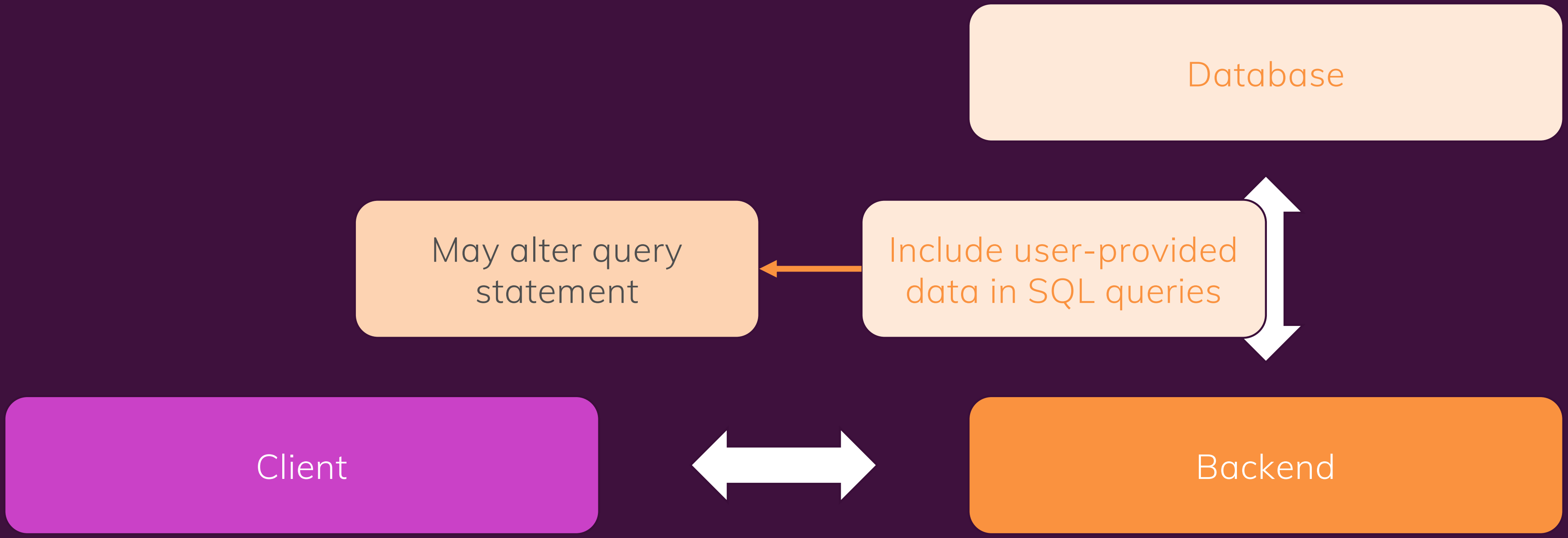
Database

Sanitize (clean) user input before processing & storing

Client

Backend

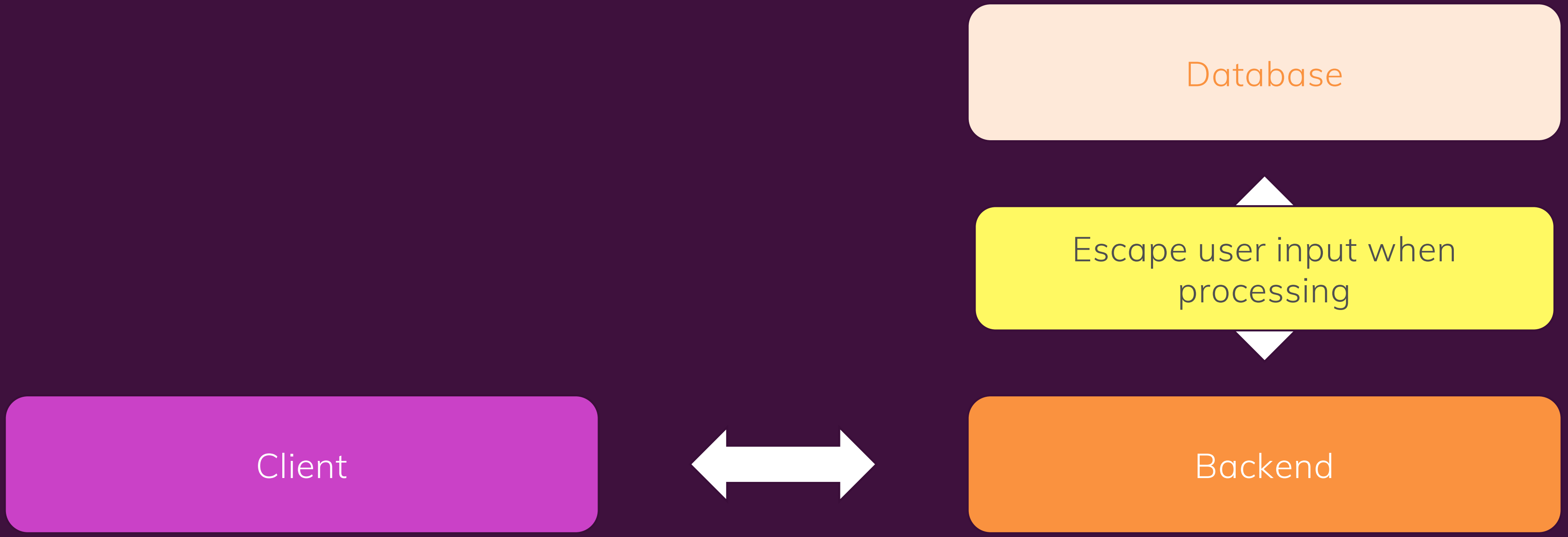
Understanding SQL Injection Attacks



Database
May alter query statement
Include user-provided data in SQL queries
Client
Backend

ACADEMIND

Protecting Against SQL Injection Attacks



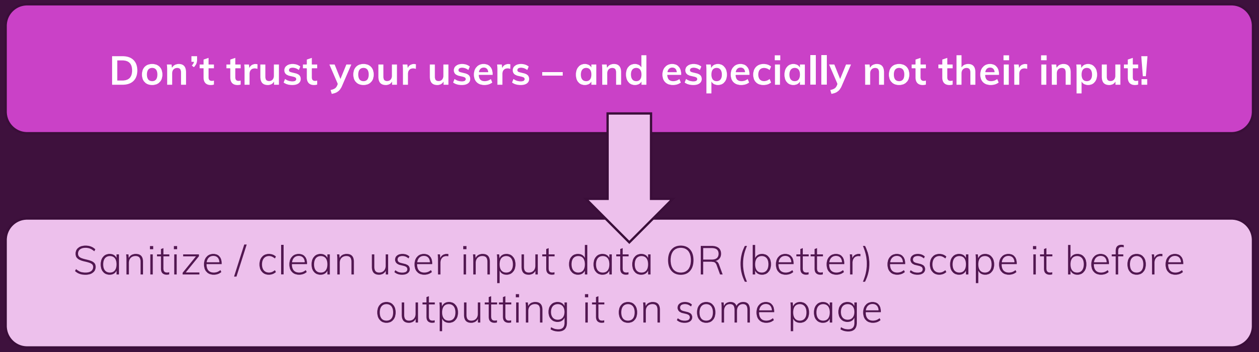
Database

Escape user input when processing

Client

Backend

Key Takeaways



Don’t trust your users – and especially not their input!

Sanitize / clean user input data OR (better) escape it before outputting it on some page

Only output unescaped (i.e. raw) input data in the browser if you really know what you’re doing

ACADE MIND

Don’t Expose Your Backend Code & Data

Be careful when serving folders (and their content) statically

All files that are served statically can be requested and viewed without issues

You want that for your CSS, Images and browser-side JS files but not for anything else!

Avoid sending raw error messages to visitors

Will very likely contain information (e.g. code snippets) you don’t want to expose

Set up custom error handling + messages instead