ACADEMIND

What are Services & APIs?

Services are just that: Services you can use in your projects / websites

- Packages like Axios or Express
- Google Maps API
- Google Analytics
- Stripe Payments

Expose “Actions“ or “Endpoints” you can access from inside your code

API (Application Programming Interface)

ACADEMIND

Web Services / APIs vs Websites

Websites

All about displaying ("rendering") HTML pages

Responses contain HTML code

Web Services / APIs

All about exchanging data and performing certain actions

Responses contain (JSON) data

ACADEMIND

Why Services / APIs?

As a (web) developer, you can't build all features on your own

Use paid or free third-party services / APIs to add certain features to your app

Or: Build a service / API that can be consumed by others

ACADEMIND

When Should You Build A Web Service?

Do you want to offer a service to others?

You might need a website for selling / advertising your service but not for the service itself

The service is exposed to customers via an API

Do you build your own website / app?

You might want to decouple (parts of) the frontend from the backend

Might simplify working in bigger teams & simplifies connection different clients

For mobile apps, you have no HTML, CSS etc. ➔ You build an Android / iOS frontend which can then be connected to the backend API

ACADEMIND

Build A JavaScript Package vs URL-based API

Could be talking to URL-based API

JavaScript Package	URL-based API
Expose certain functions, objects, methods etc.	Expose certain “actions” via “endpoints” (URLs + Http methods)
Build with help of frontend build tools & npm	Requests sent to different URLs trigger different actions & responses
Does not primarily / only require web dev skills	Requires web dev skills & we already built such APIs!

Introducing REST APIs

Representational **S**tate **T**ransfer

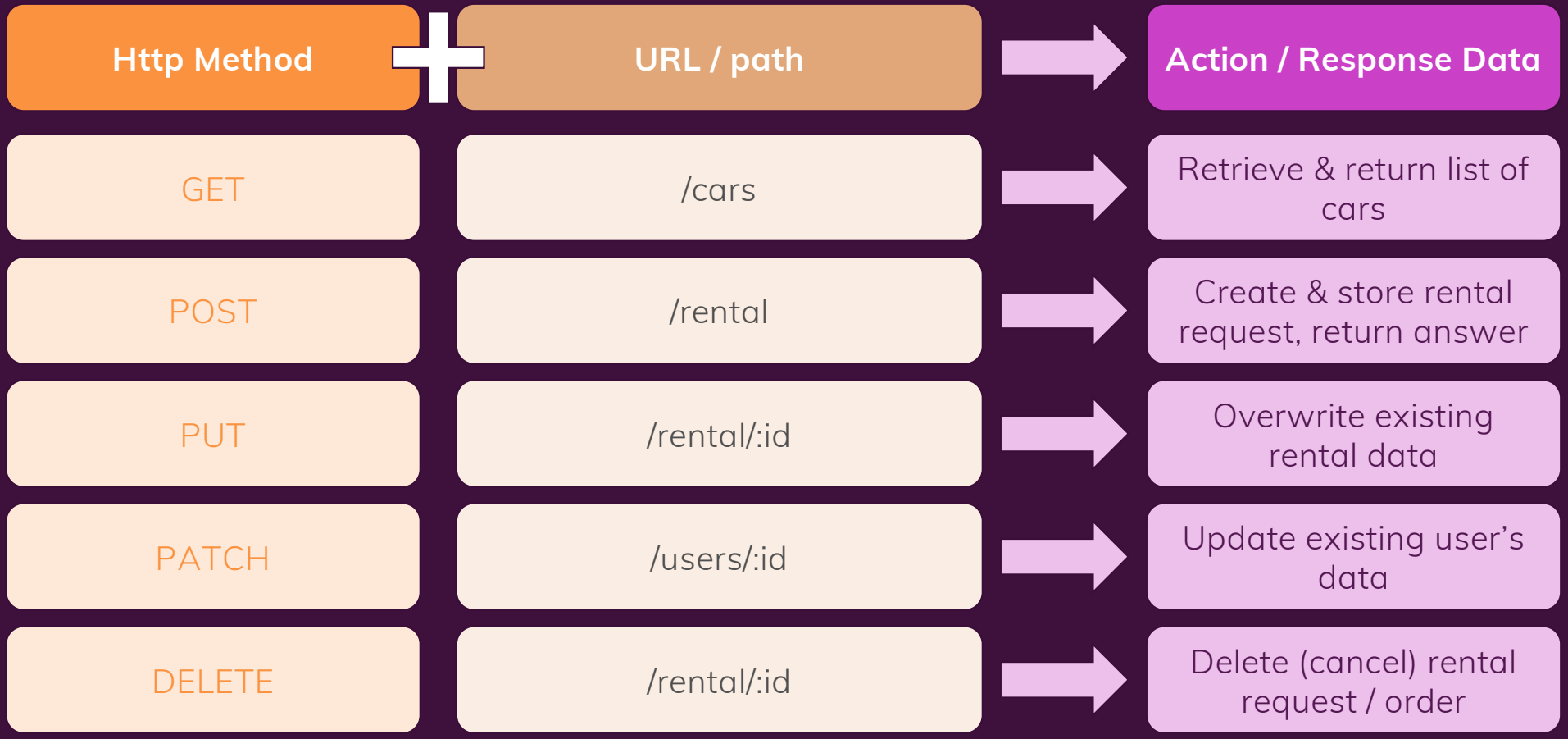
The most common / popular form of building URL-based Services & APIs

Core Idea: API “Endpoints” are URL + Http method combinations

GET /cars → Returns a list of available cars

ACADEMIND

REST & API Methods



ACADEMIND

The Backend Code Matters!

Http methods + URLs / paths are used to target specific endpoints

↓

The backend code (routes) defines which endpoints are provided & what happens upon incoming requests

You could write code that deletes a user upon GET /cars → That wouldn't make sense and wouldn't make for a good API / service

ACADEMIND

Example API

A Simple “Daily Quotes” (Web) Service

GET /quote

Retrieve and return a (random) quote

Quote data will be supplied directly be the service owner
(through a different API or with direct database queries)

ACADEMIND

REST APIs Are "Stateless"

State?

Data that's mapped to a specific client & persisted across requests (!) on the server

REST APIs don't store client-mapped data on the server

Every incoming request is treated standalone

Example API

A Simple Todos Management (Web) Service

GET /todos	Retrieve and return a list of todos
POST /todos	Create and store a new todo (return success response)
PATCH /todos/:id	Update an existing todo (e.g. change text)
DELETE /todos/:id	Delete an existing todo

ACADEMIND

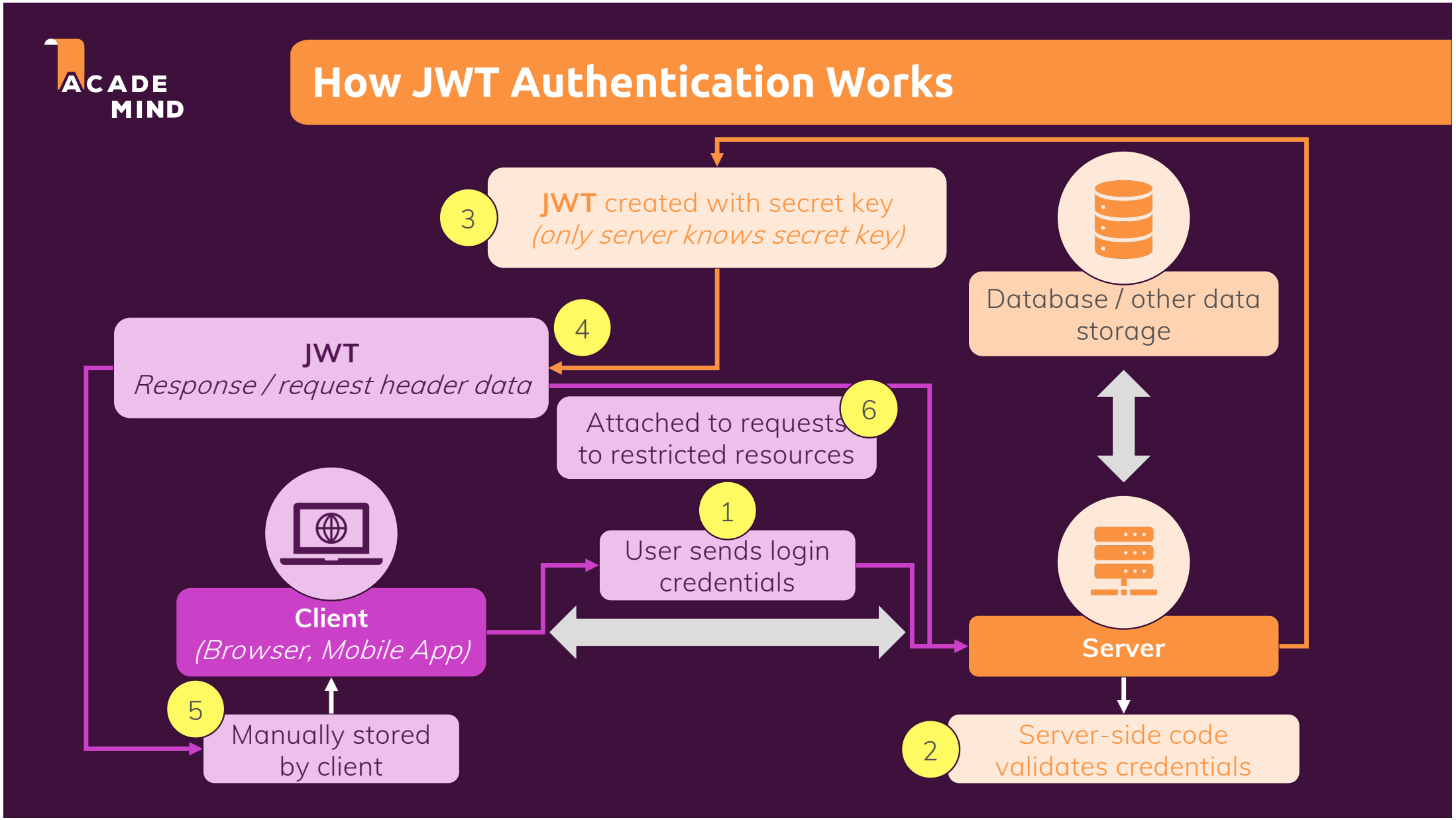
(REST) API Authentication

No Sessions → No session-based authentication

Solution: Provide a “self-authenticating” data package to the client

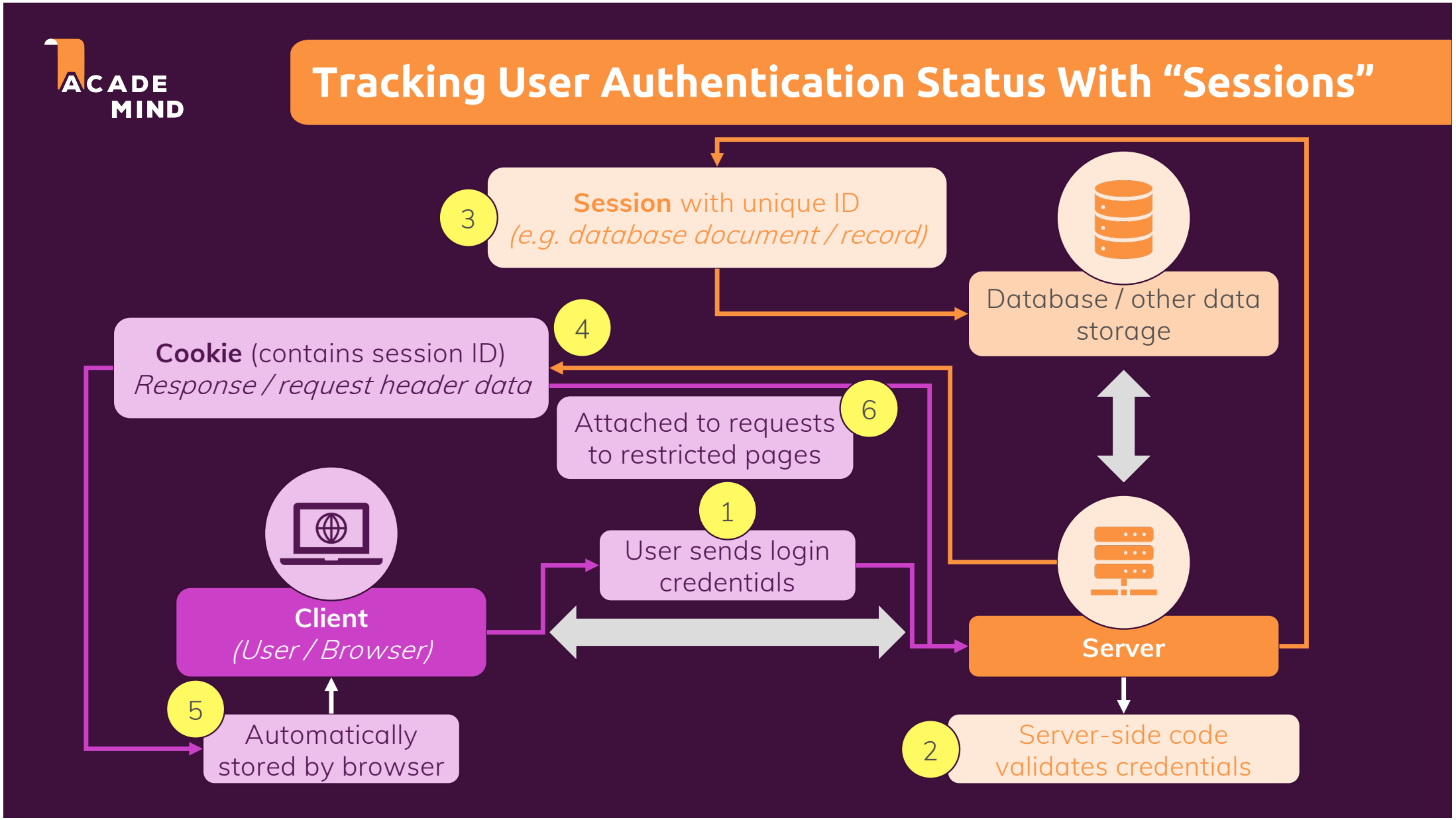
↓

JSON Web Tokens (JWT)


ACADEMIND
How JWT Authentication Works
3
JWT created with secret key
(only server knows secret key)
Database / other data storage
4
JWT
Response / request header data
6
Attached to requests to restricted resources
1
User sends login credentials
Client
(Browser, Mobile App)
Server
5
Manually stored by client
2
Server-side code validates credentials

ACADEMIND

Tracking User Authentication Status With "Sessions"



ACADEMIND

REST API Alternatives