Lesson: Authentication & Authorization

Wannes Fransen & Tom Eversdijk

UC Leuven

2021

Intro

Authentication

Authorization

Difference authentication & authorization

Authentication - AuthN

- Authentication is establishing the your identity
- *Verifies you are who you say you are*

Authorization - AuthZ

- Authorization is establishing your privilege
- *Decides if you have permission to access a resource*

Fun fact

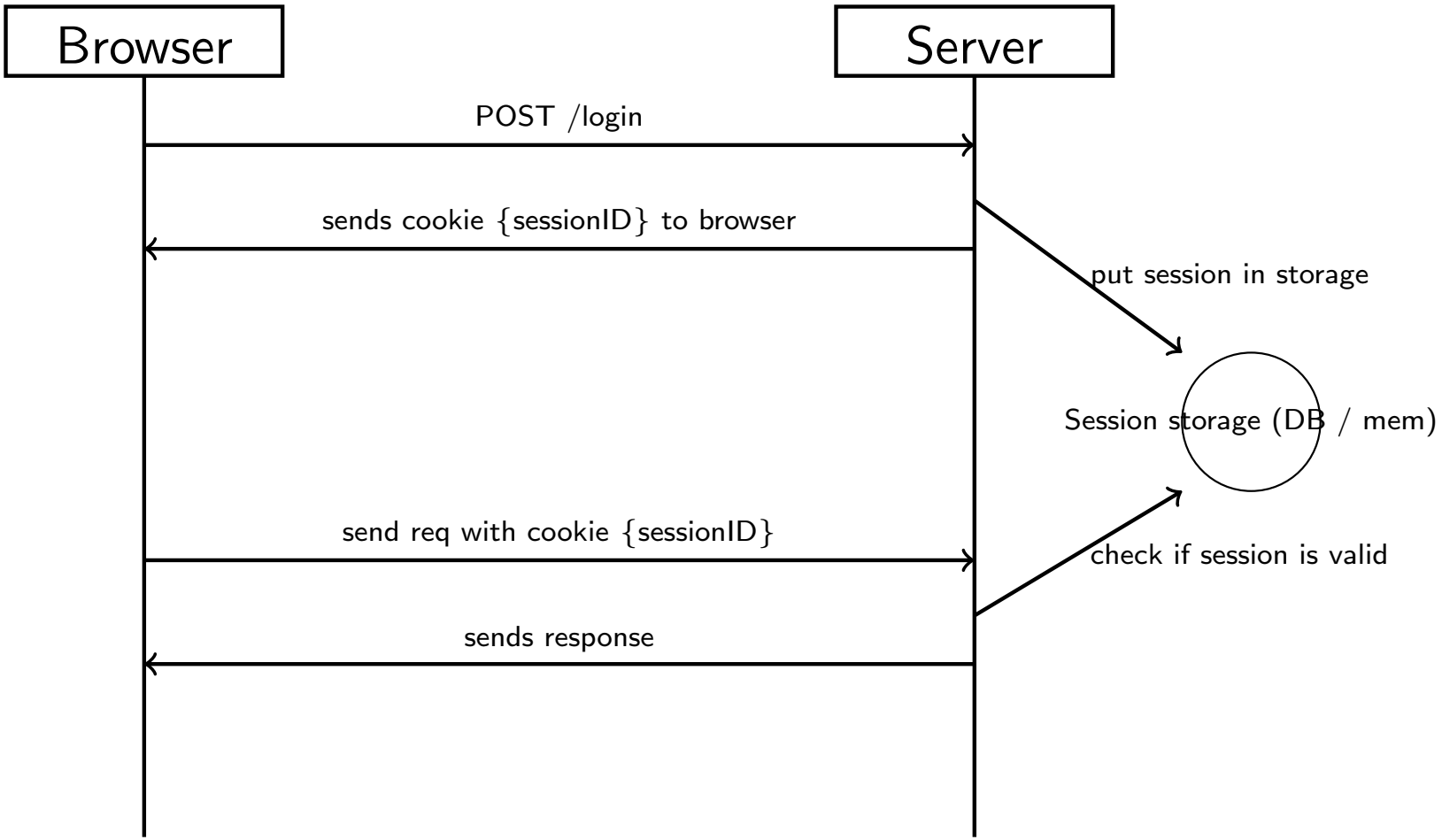
broken authentication / authorization is nr2 on the most critical security risks to web apps.

Intro

Authentication

Authorization

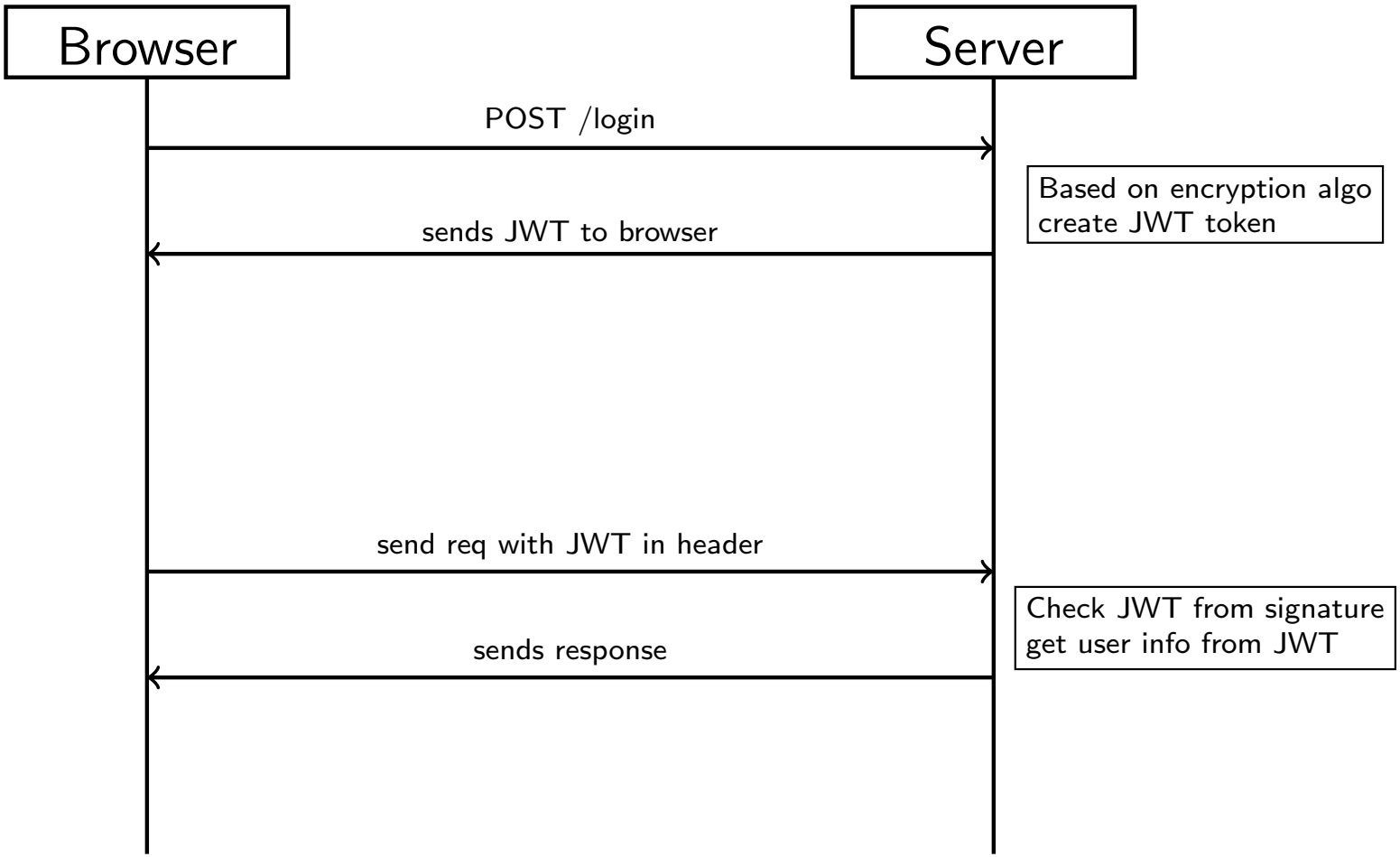
Recap: Session based



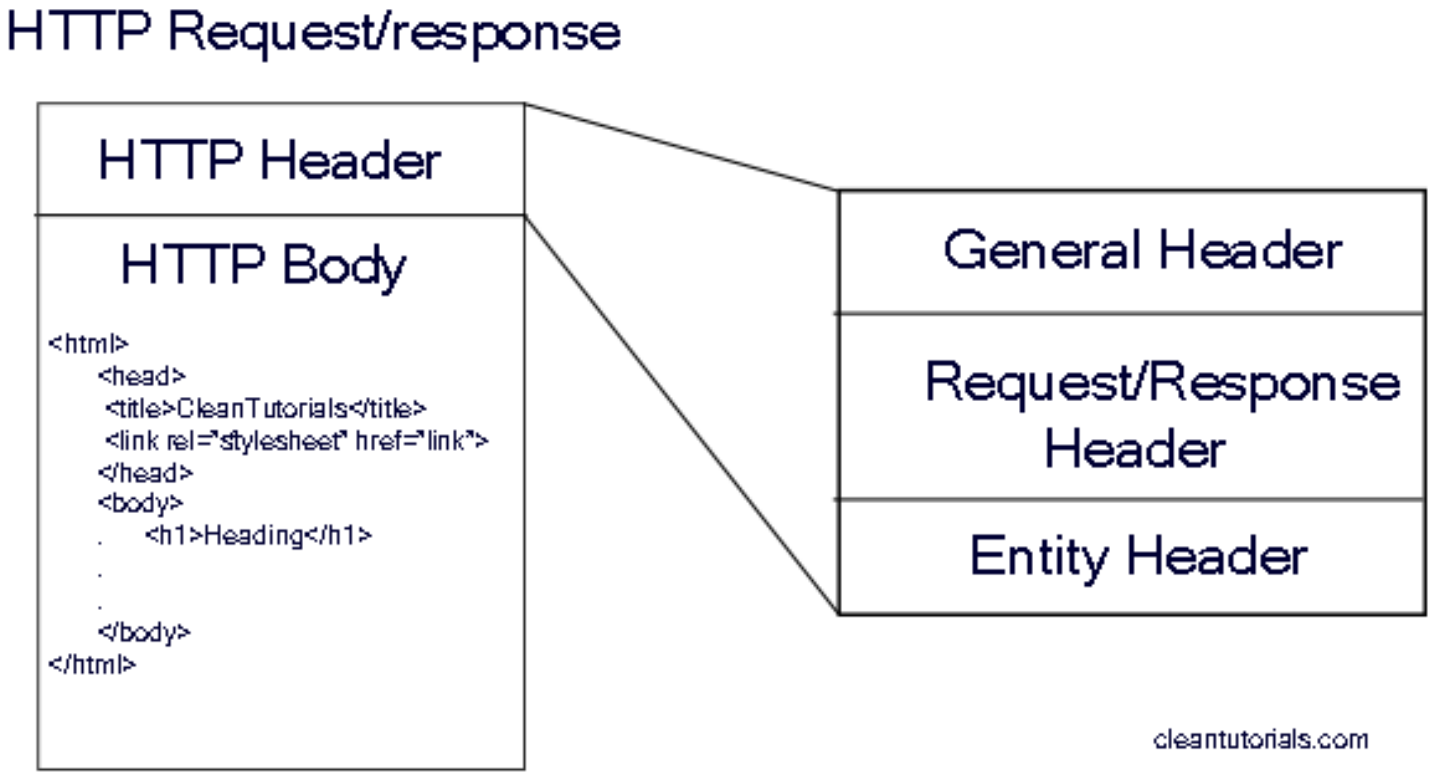
Session based characteristics

- Easy to implement stateful user sessions
- Can revoke at any time
- Can store any data alongside your session ID
- Can be a bottleneck in case of large volumes / data / traffic
- Serialization can be a problem (when storing objects)
- Session storage is a bottleneck in distributed contexts

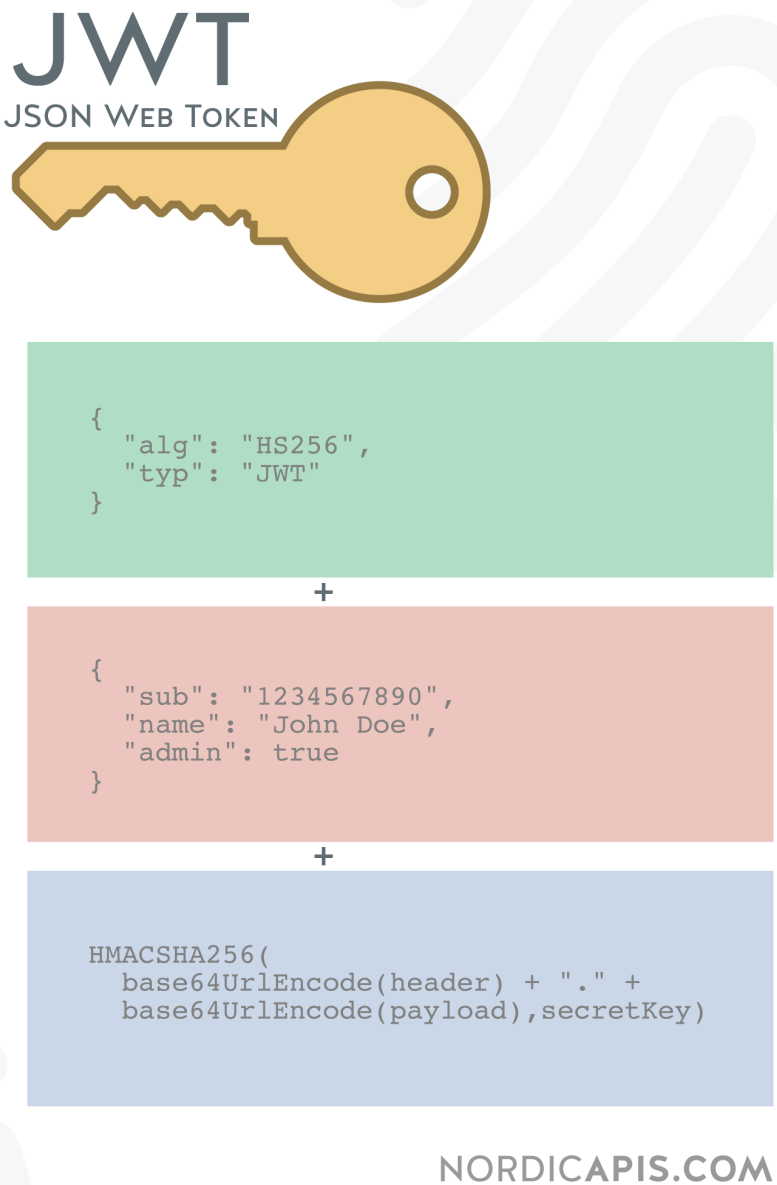
JWT based


Browser
Server
POST /login
Based on encryption algo
create JWT token
sends JWT to browser
send req with JWT in header
Check JWT from signature
get user info from JWT
sends response

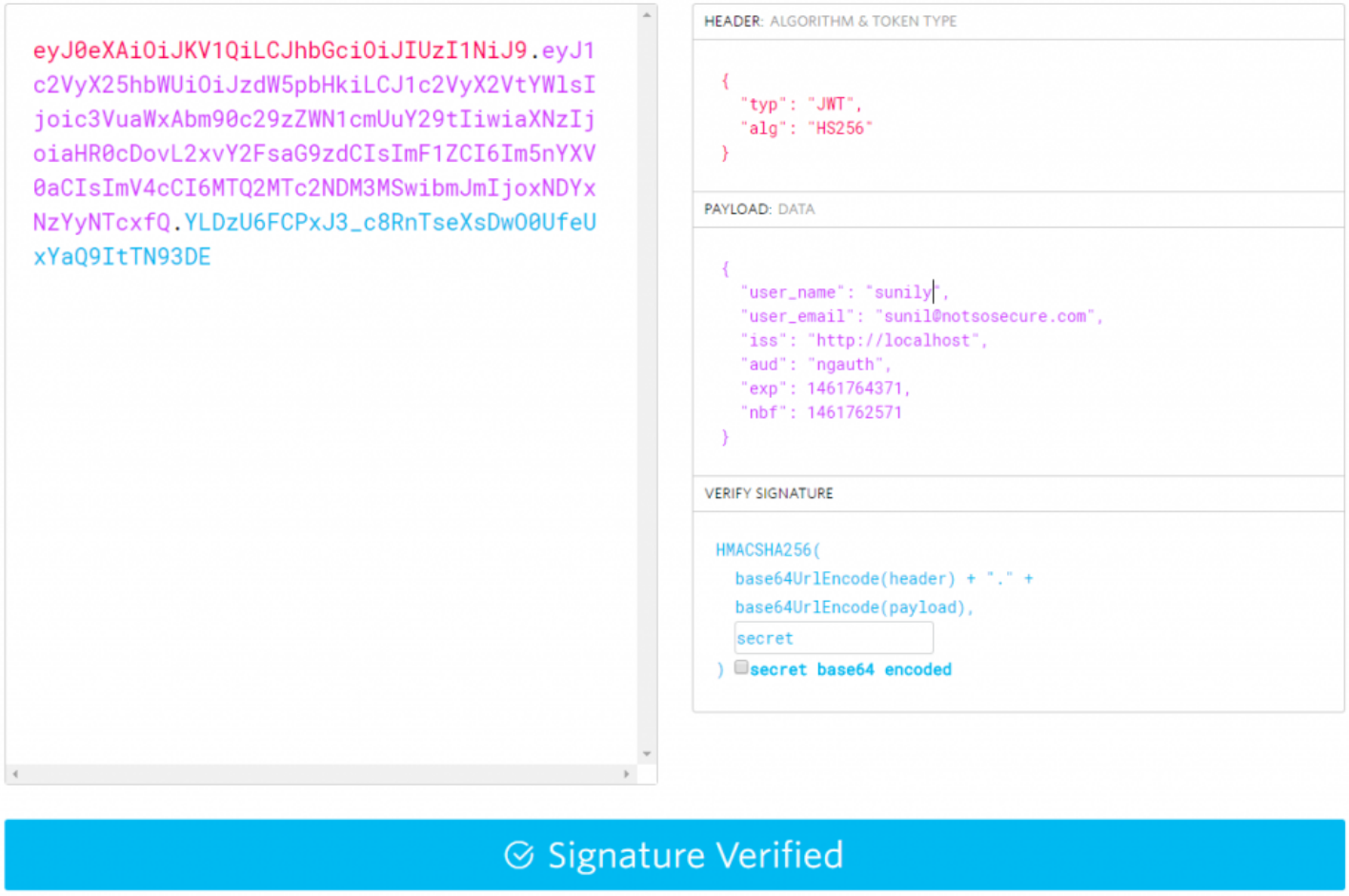
JWT: Where is it exactly?



JWT: What is it exactly?



JWT: What is it exactly?



JWT based characteristics

- no central bottleneck anymore
- allows horizontal scaling
- built-in expiration functionality
- no way to revoke the JWT token
- self-contained, but keep it short

What to use?

- Both are viable, depends on your needs
- We'll go with JWT (Guardian library) as this can be re-used in Distributed Applications

Intro

Authentication

Authorization

Usages

- Roles (Superuser, forum maintainer, normal user)
- User-specific resources (You manage your own resources)
- permissions

Libraries

- Different libraries have their own implementation
- Same as a mindset, different ways to achieve the same result

extra references

- Bodyguard can be used to implement Authorization in Elixir