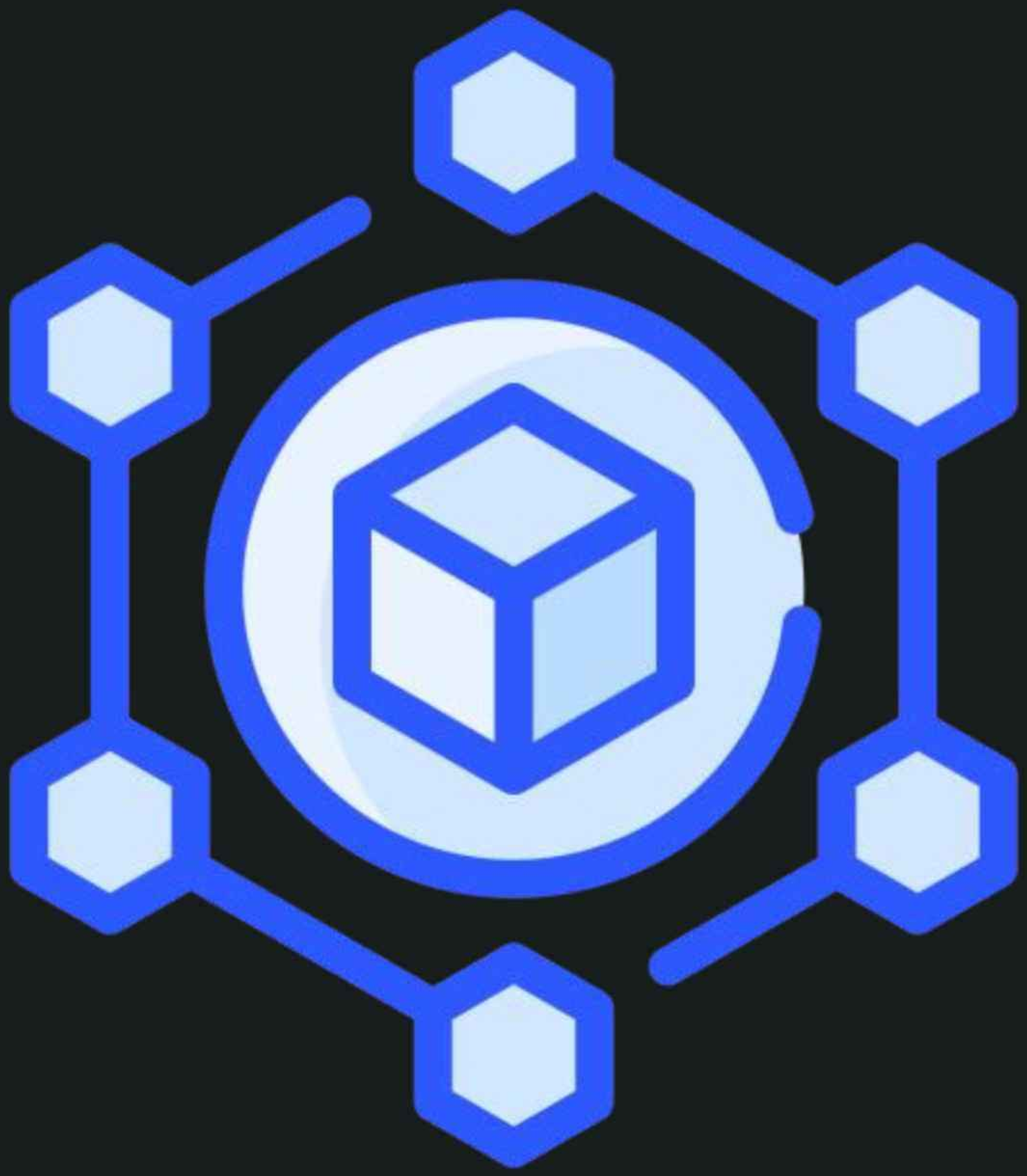
Gagan Saini
@gagan-saini-gs

Github: Gagan-Saini-GS



JSON Web Token

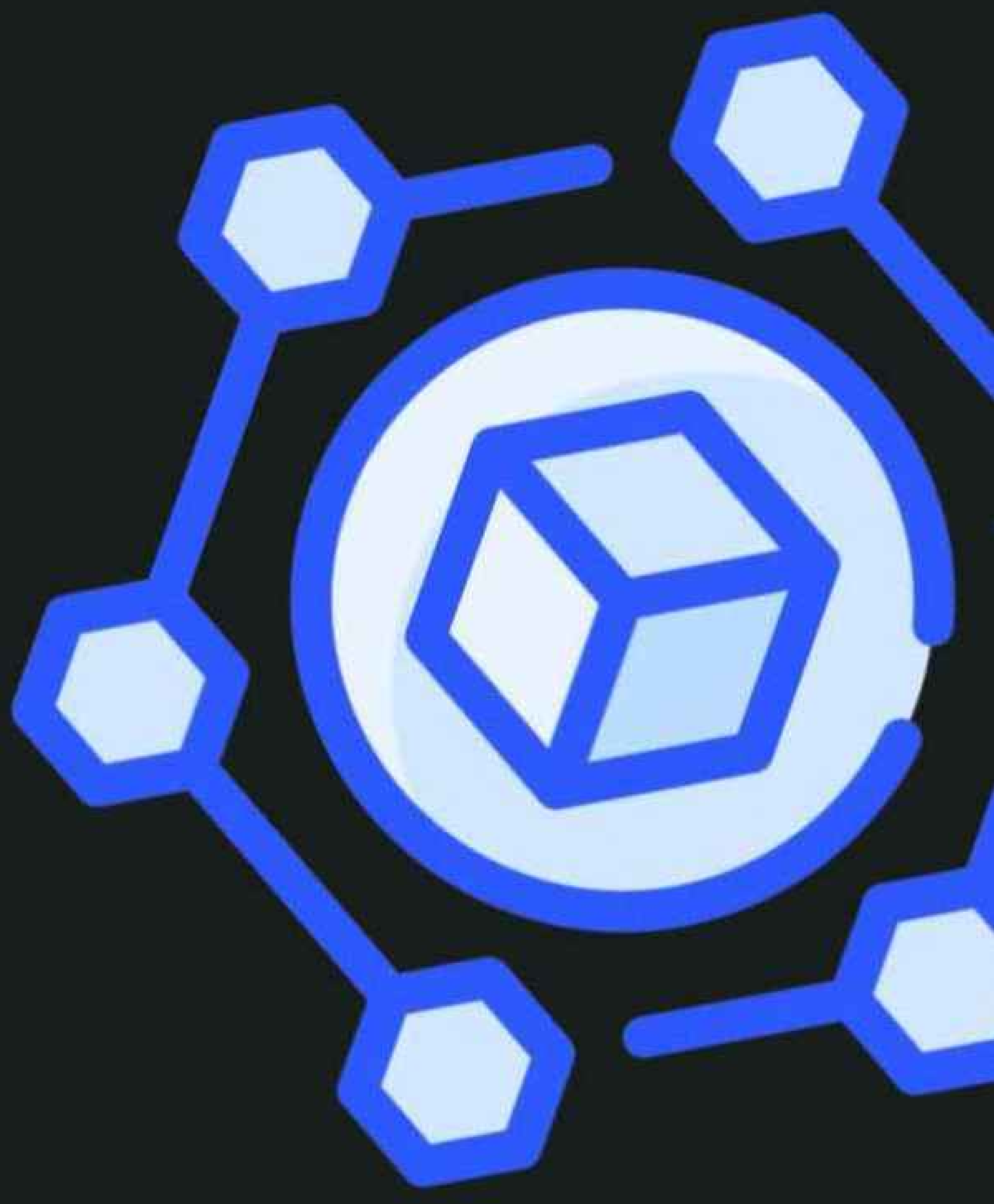
Gagan Saini
@gagan-saini-gs

Github: Gagan-Saini-GS

JWT

JSON Web Token (JWT) is an **open standard (RFC 7519)** that defines a compact and secure way to **securely transmit** information between two parties.

It achieves this by **encoding essential information** in a JSON object and signing it with a **cryptographic algorithm**.



Why JWT Used?

Authentication

JWT contain the necessary info (claims) about the user, **eliminating** the need for server-side lookups.
This **simplifies authentication logic** and reduces server load.

Info Exchange

JWTs can securely transmit info between parties. Since the info is **encoded & signed**, you can be confident about its integrity and authenticity.

Stateless Sessions

Unlike traditional session-based authentication, JWTs **don't require** the server to store session data for each user.

This makes them **scalable and suitable** for distributed architectures.

Structure of JWT

A JWT is composed of three parts, each Base64Url encoded.

Header

Contains info about the token type (JWT) and the **signing algorithm used**.

Payload

This is the most crucial part, containing claims **(pieces of information)** about the user or the application.

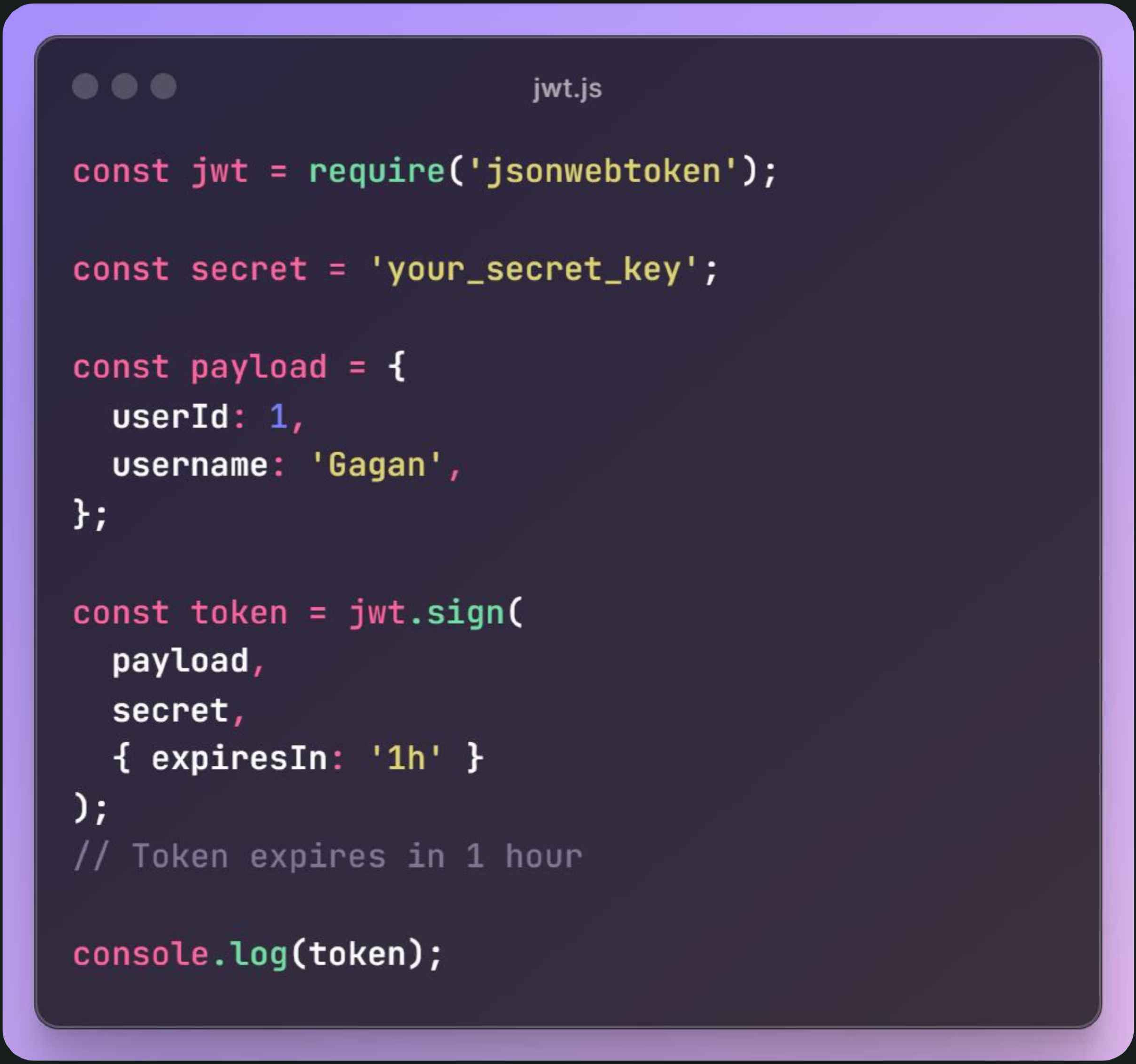
These claims can be anything you want to transmit **securely**.

Signature

This part is generated by signing the **encoded header** and **payload using a secret key**.

It ensures the integrity of the claims, **verifies** that the **token hasn't been tampered** with.

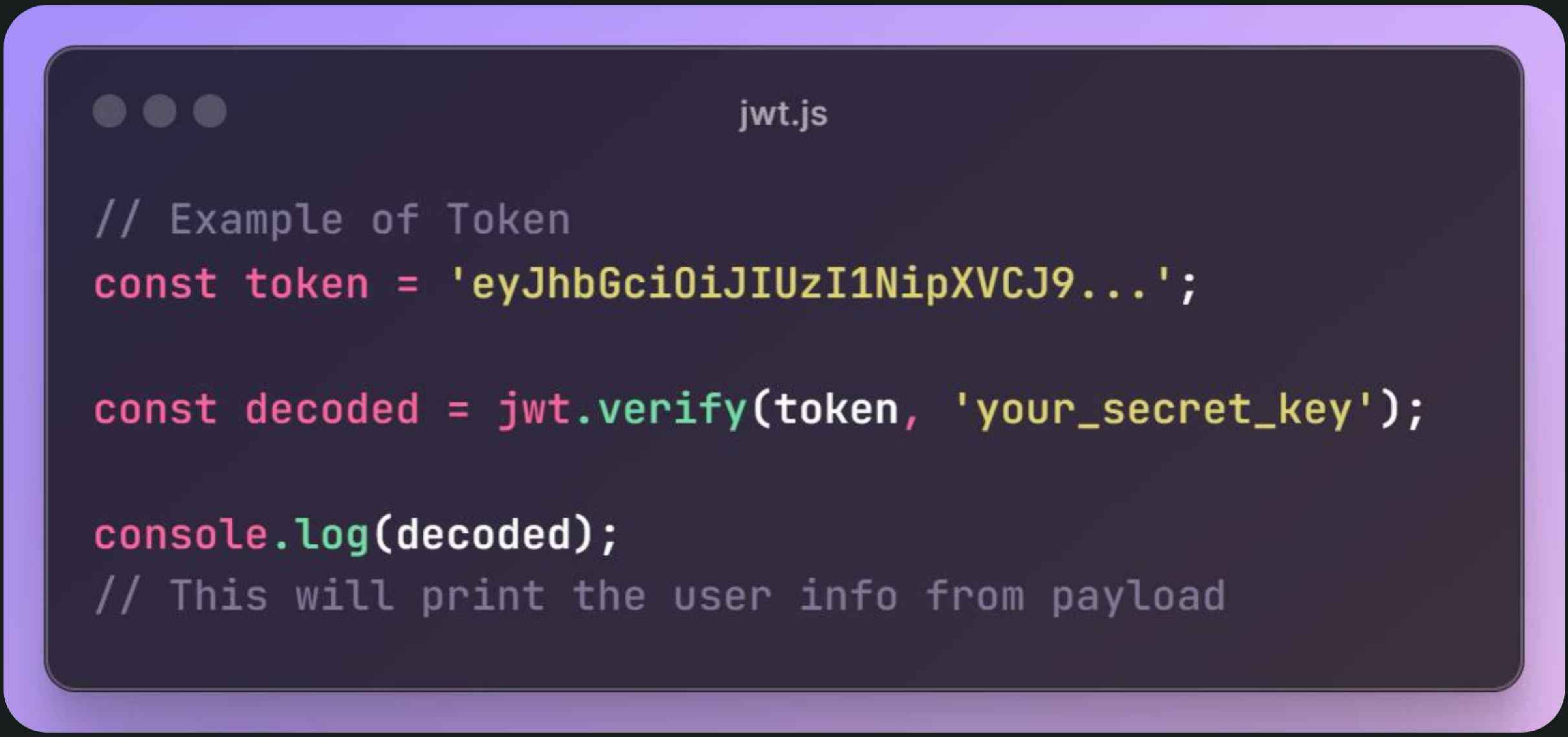
Server Side Code



jwt.js

```
const jwt = require('jsonwebtoken');

const secret = 'your_secret_key';

const payload = {
  userId: 1,
  username: 'Gagan',
};

const token = jwt.sign(
  payload,
  secret,
  { expiresIn: '1h' }
);
// Token expires in 1 hour

console.log(token);
```

Client Side Code



```
// Example of Token
const token = 'eyJhbGciOiJIUzI1NipXVCJ9...';

const decoded = jwt.verify(token, 'your_secret_key');

console.log(decoded);
// This will print the user info from payload
```