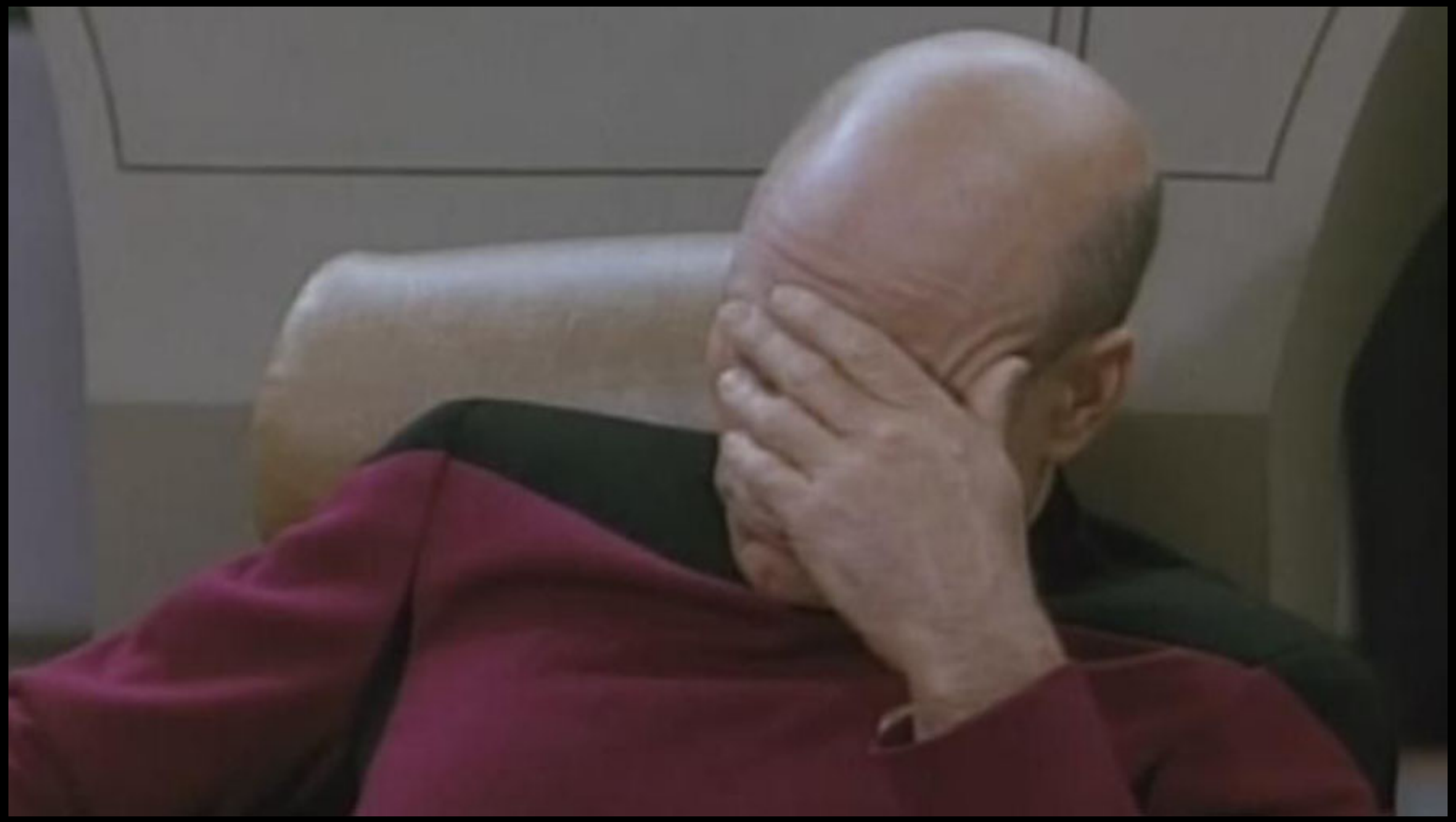
We read code more than we write.

Creating a
framework-less
server-less
node-modules-less web page

Did I start with a solution again ?

Did I start with a solution again ?



The Problem

The Problem

Create a landing page that shows dynamic content based on the roles in a JWT token.

The Problem

Create a landing page that shows dynamic content based on the roles in a JWT token.

The Problem

Create a landing page that shows dynamic content based on the roles in a JWT token.

The Problem

Create a landing page that shows dynamic content based on the roles in a JWT token.

Performant / Reliable

Templating

The Problem

Create a **landing page** that shows **dynamic content based on the roles** in a JWT token.

Performant / Reliable

Templating

The Problem

Create a landing page that shows dynamic content based on the roles in a JWT token.

Performant / Reliable

Templating

JWT decoder / deserialiser

The Problem

Create a **landing page** that shows **dynamic content based on the roles** in a **JWT token.**

Performant / Reliable

Templating

JWT decoder / deserialiser

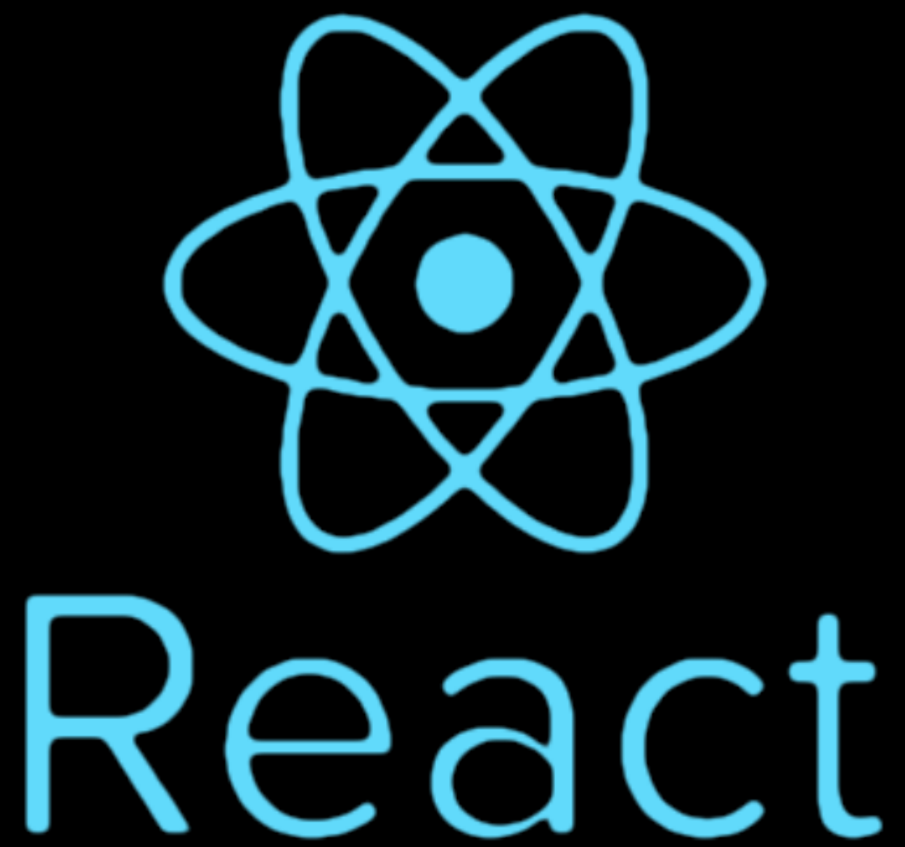
The Solution

Performant / Reliable

Templating

JWT decoder / deserialiser

The Solution



The Solution

React

Performant / Reliable

Templating

JWT decoder / deserialiser

Performant / Reliable

Templating

JWT decoder / deserialiser

The Problem

Create a **landing page** that shows **dynamic content based on the roles** in a JWT token.

Dynamic content is sensitive and should not be loaded on client side if user does not have relevant roles.

Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

The Problem

Create a **landing page** that shows **dynamic content based on the roles** in a JWT token.

Dynamic content is sensitive and **should not be loaded on client side** if user does not have relevant roles.

Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

The Solution

React

with SSR

Performant / Reliable

Templating

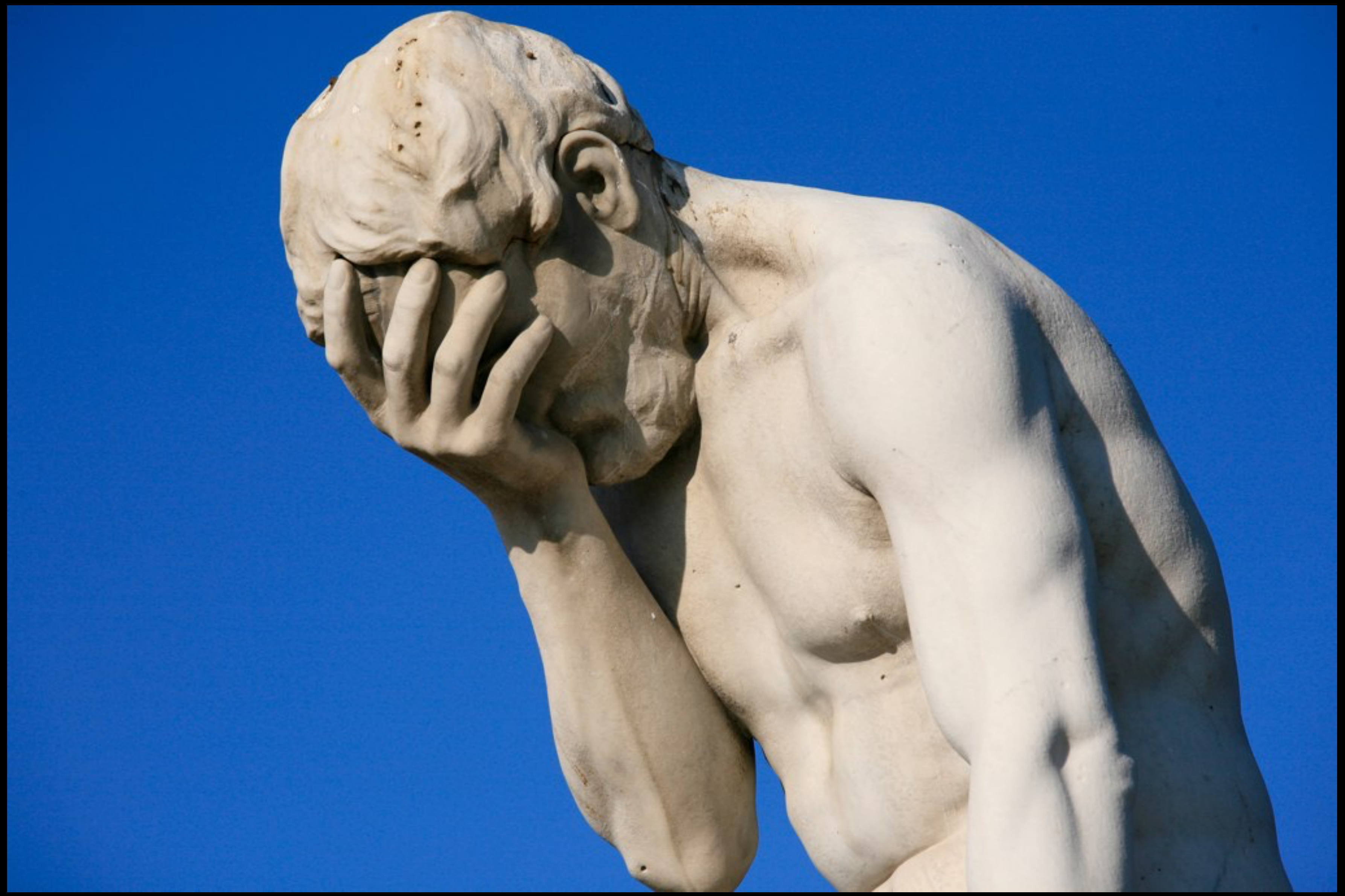
JWT decoder / deserialiser

Server Side Rendering

The Solution

React with SSR

💡Page does not have dynamic state



Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

The Solution

The Solution

spring
boot

Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

Model

View

Controller

Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

The Solution

spring
boot

✗ Model

View

Controller

Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

The Solution

spring boot

- ✗ Model
- ✅ View
- Controller

The Solution

spring boot

Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

- ✗ Model
- ☑ View
- ✗ Controller

Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

The Solution

RAILS

- ✗ Model
- ✓ View
- ✗ Controller

Performant / Reliable

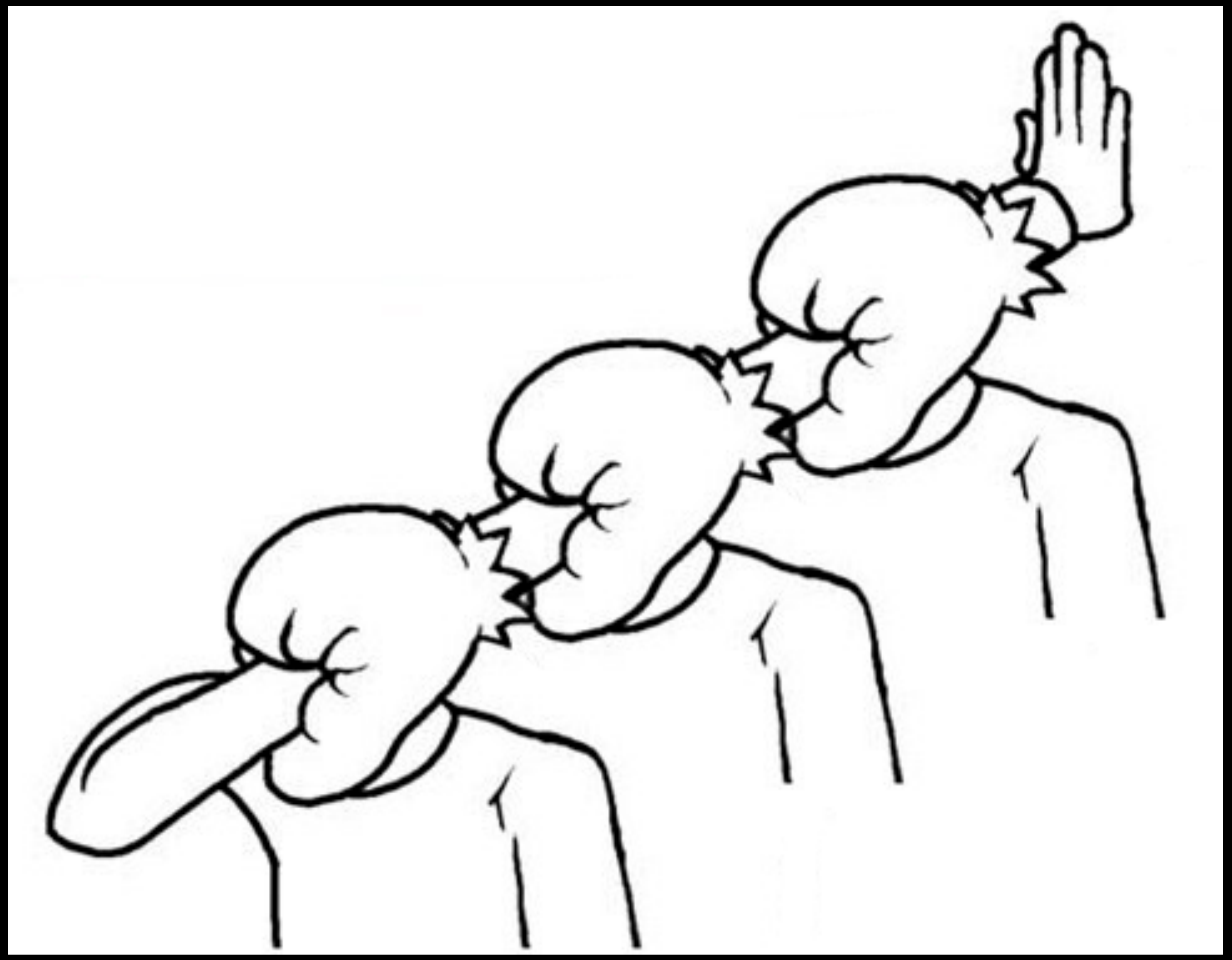
Templating

JWT decoder / deserialiser

Server Side Rendering

The Solution

RAILS



The Solution

Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

The Solution

Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

The Solution

JWT Decoder

1 line of code

Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

The Solution

Performant / Reliable

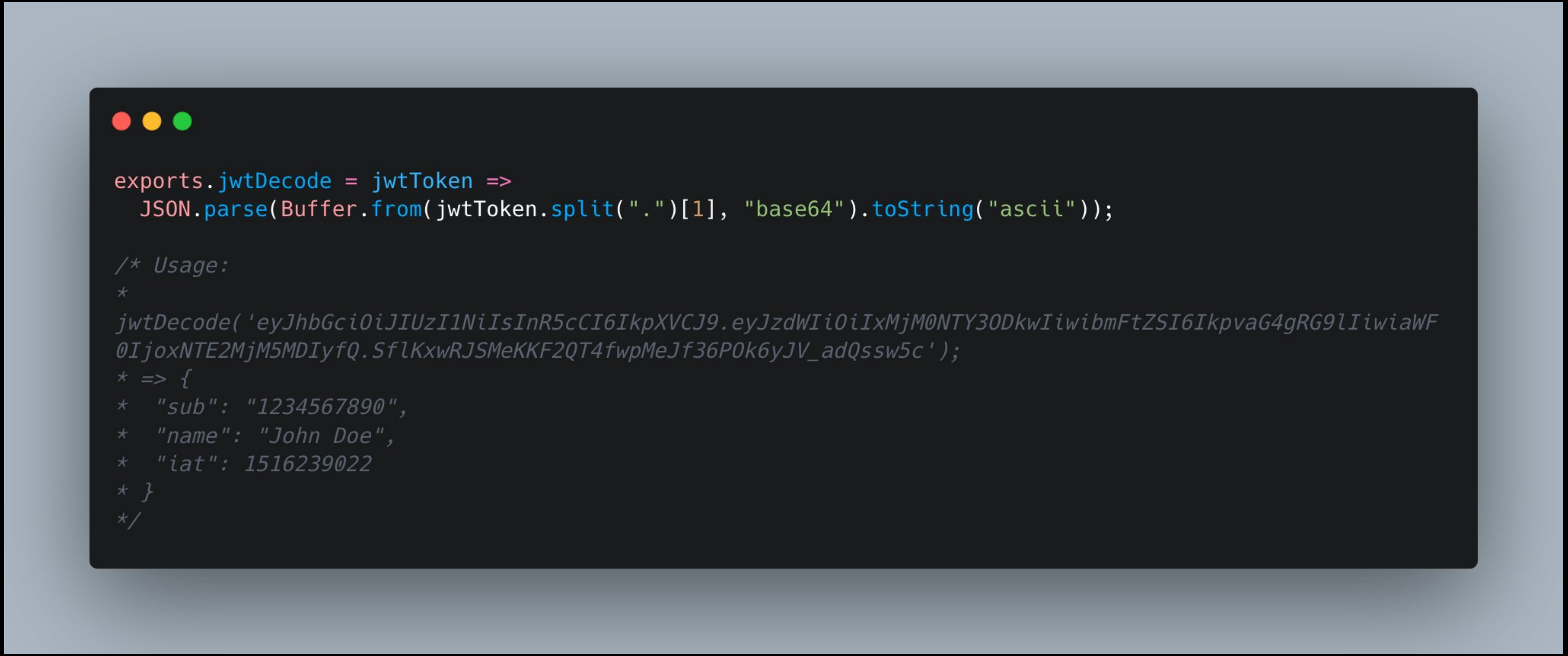
Templating

JWT decoder / deserialiser

Server Side Rendering

JWT Decoder

1 line of code



```
exports.jwtDecode = jwtToken =>
  JSON.parse(Buffer.from(jwtToken.split(".")[1], "base64").toString("ascii"));

/* Usage:
*
jwtDecode('eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF
0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c');
* => {
*   "sub": "1234567890",
*   "name": "John Doe",
*   "iat": 1516239022
* }
*/
```

Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

The Solution

JWT Decoder

1 line of code

Role-Component Map

Number of roles

Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

The Solution

JWT Decoder	1 line of code
Role-Component Map	Number of roles
HTML Partials	Static HTML

The Solution

Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

JWT Decoder	1 line of code
Role-Component Map	Number of roles
HTML Partials	Static HTML
Templating	6 lines of code

Performant / Reliable

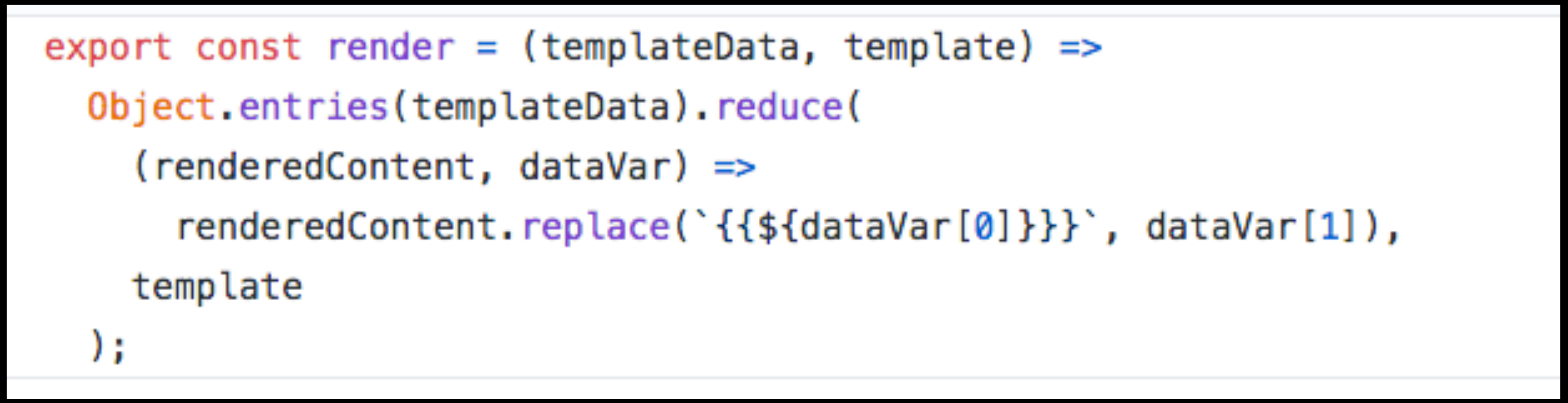
Templating

JWT decoder / deserialiser

Server Side Rendering

The Solution

JWT Decoder	1 line of code
Role-Component Map	Number of roles
HTML Partials	Static HTML
Templating	6 lines of code



```
export const render = (templateData, template) =>
  Object.entries(templateData).reduce(
    (renderedContent, dataVar) =>
      renderedContent.replace(`{{${dataVar[0]}}}`, dataVar[1]),
    template
  );
```

The Solution

JWT Decoder	1 line of code
Role-Component Map	Number of roles
HTML Partials	Static HTML
Templating	6 lines of code
Page	Static HTML

Performant / Reliable

Templating

JWT decoder / deserialiser

Server Side Rendering

But is this simple?

Well it’s simple in XYZ!

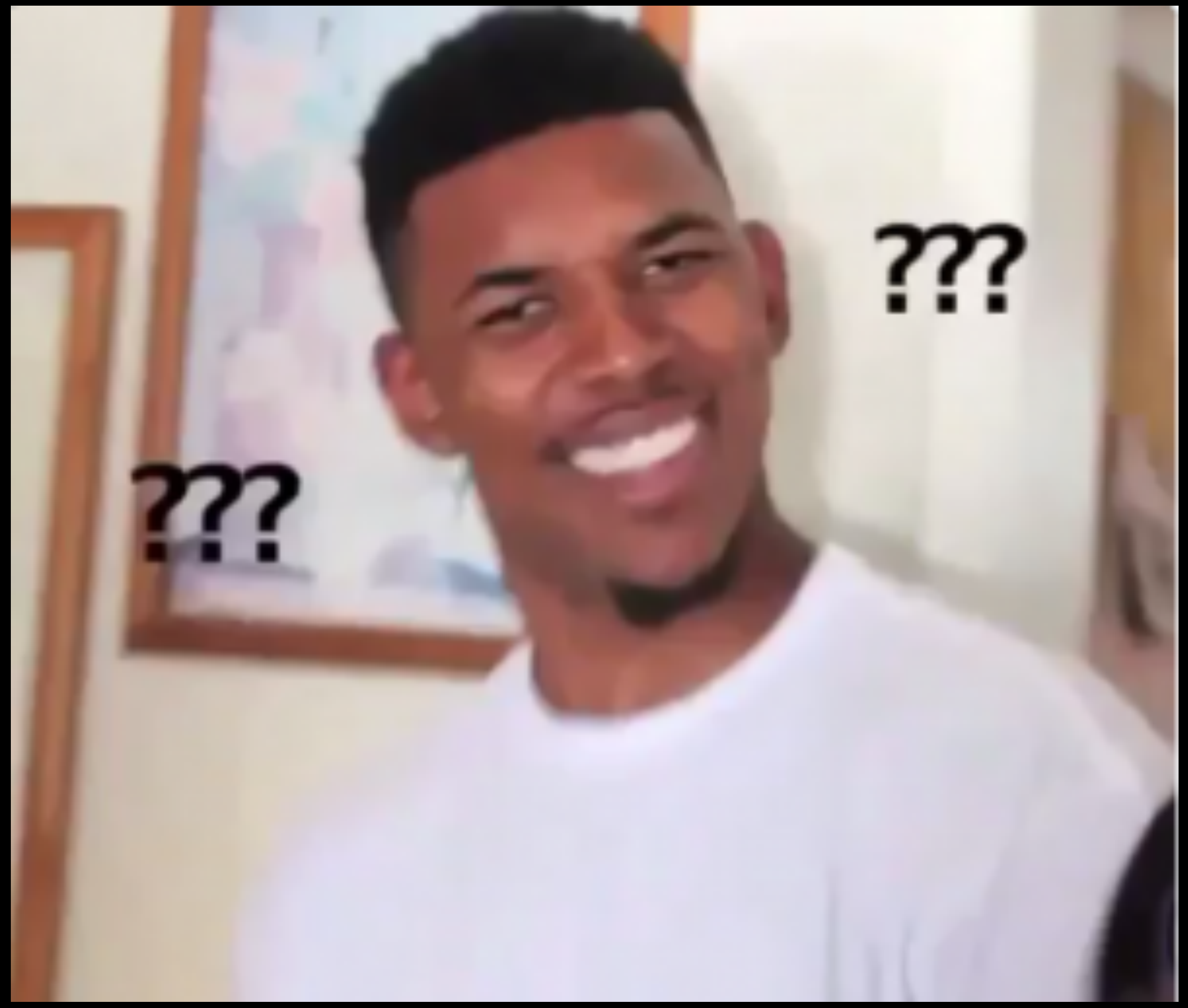
Mental Model Problem

easy

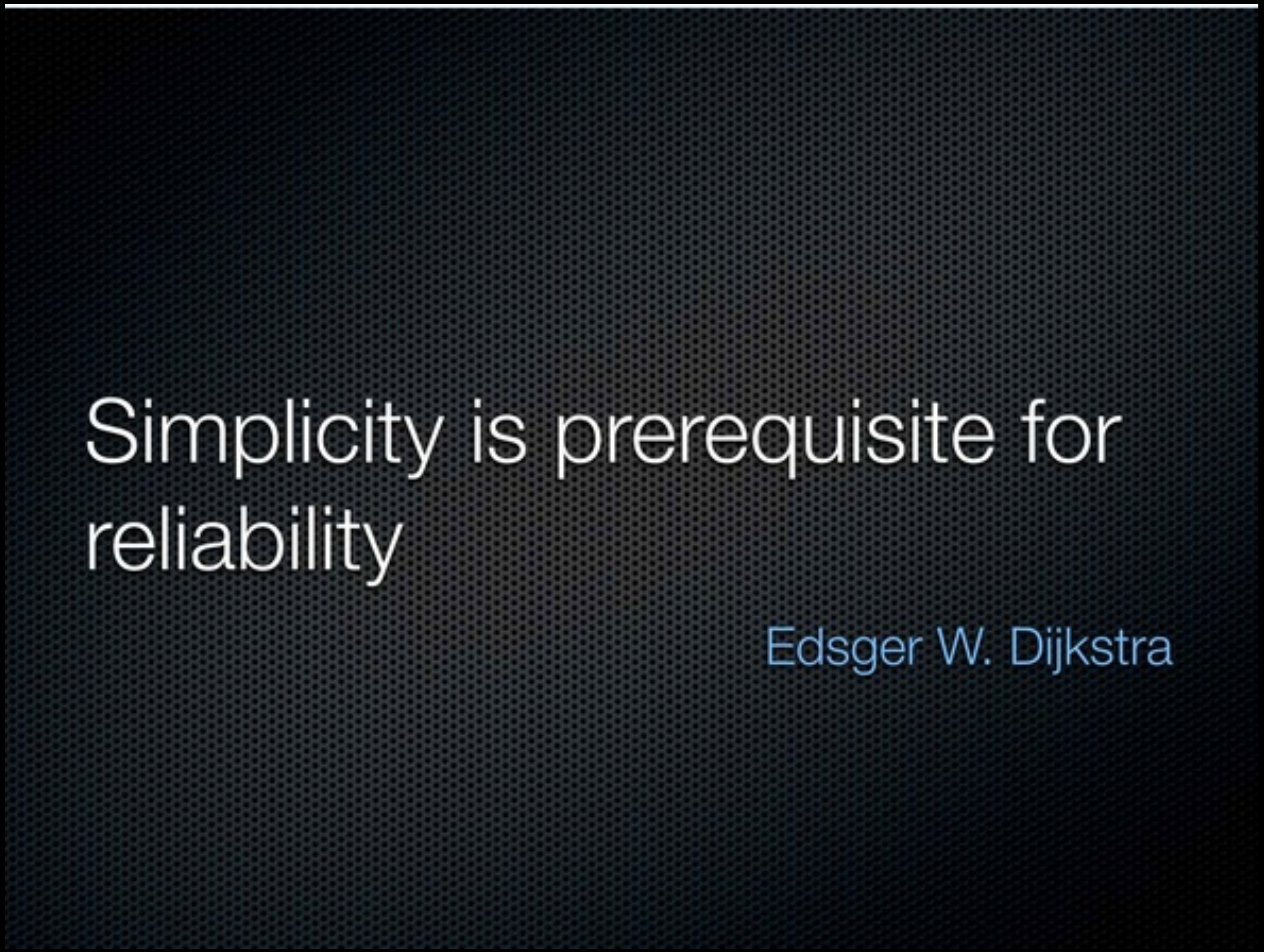
Well it’s ~~simple~~ in XYZ!

easy

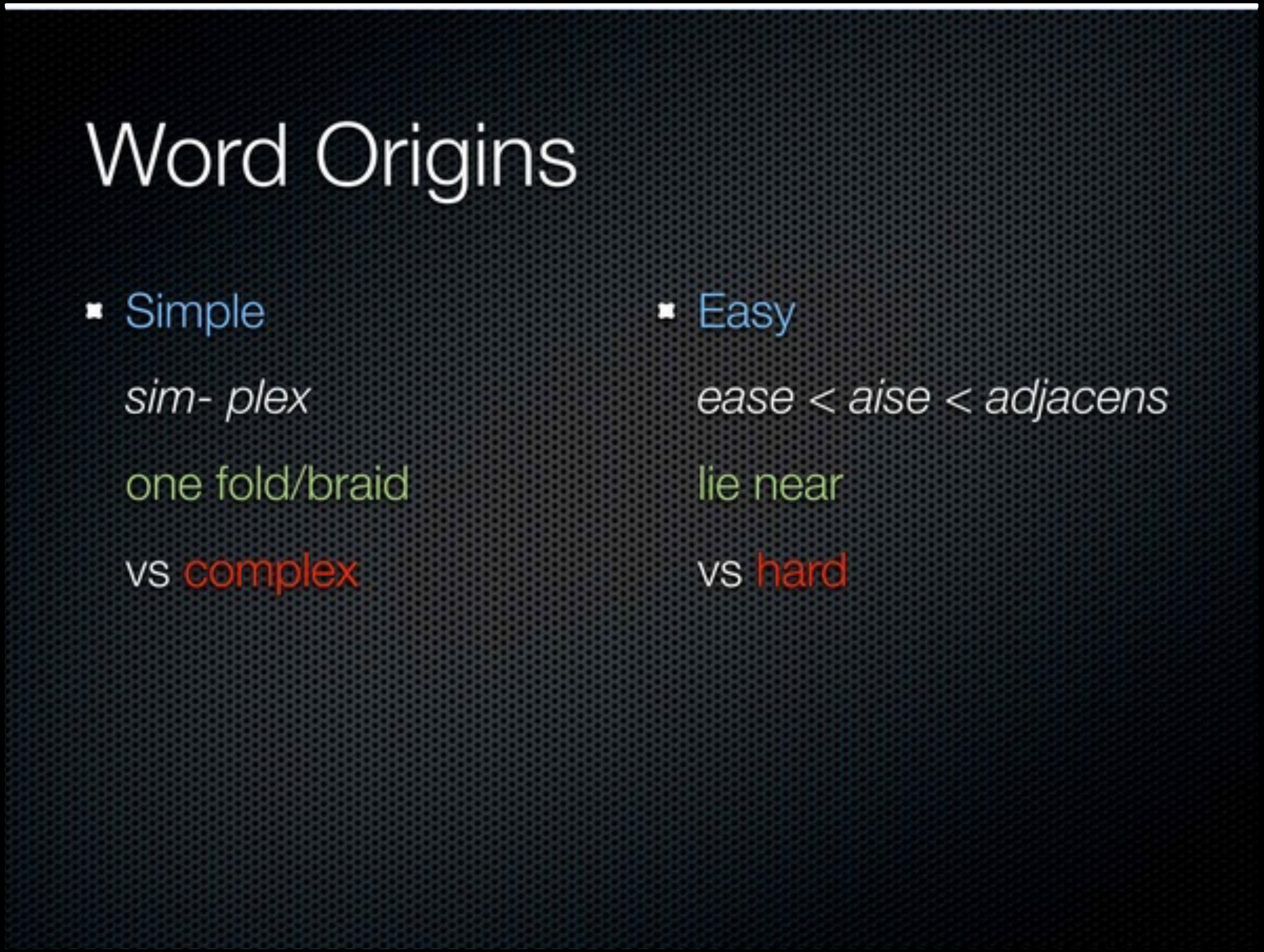
Well it's ~~simple~~ in XYZ!



Well it’s ~~simple~~ in XYZ!



> Simplicity is prerequisite for reliability
>
> Edsger W. Dijkstra



Word Origins

- Simple

  sim- plex

  one fold/braid

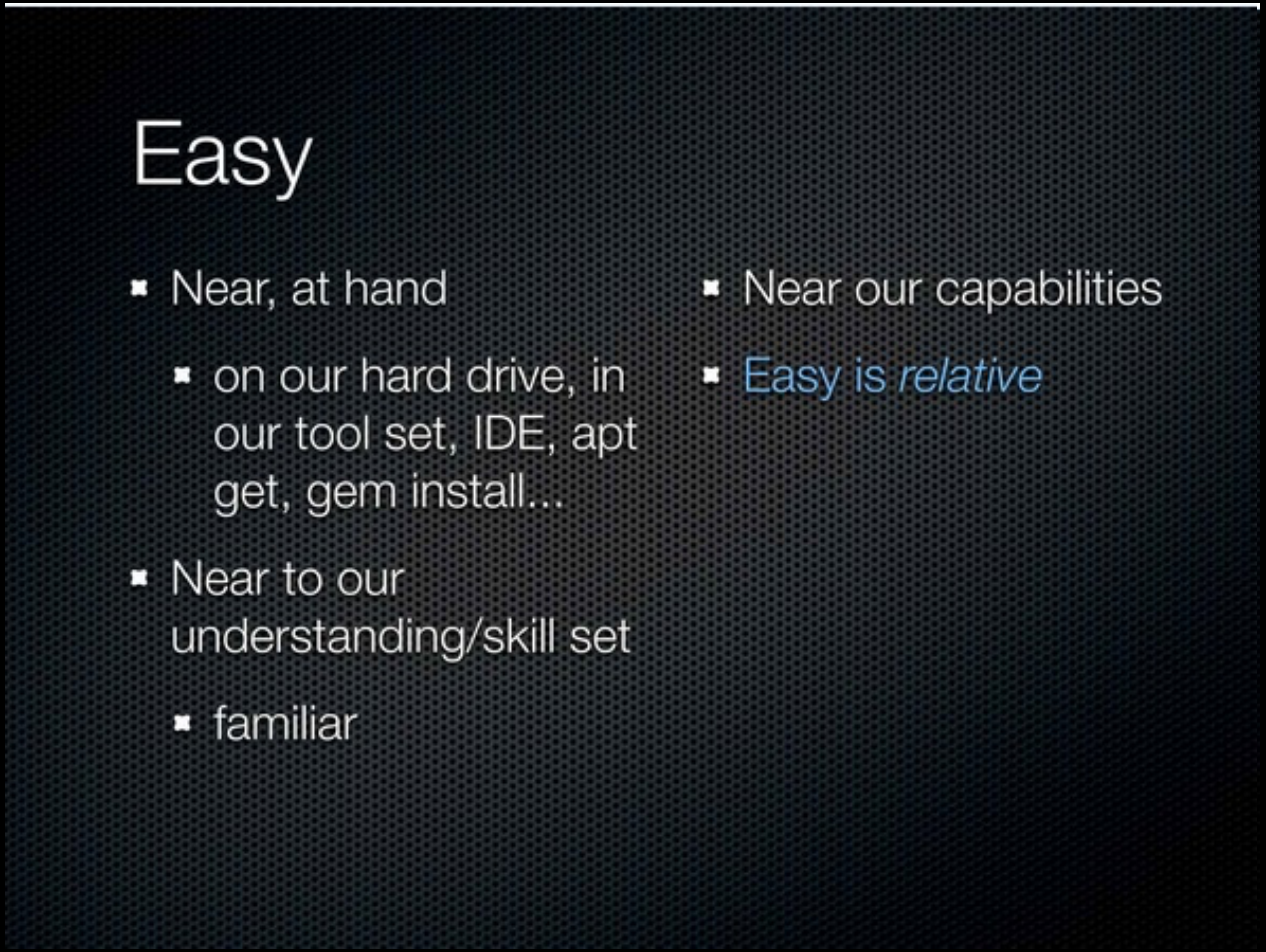
  vs complex

- Easy

  ease < aise < adjacens

  lie near

  vs hard



Easy

- Near, at hand
  - on our hard drive, in our tool set, IDE, apt get, gem install...
- Near to our understanding/skill set
  - familiar
- Near our capabilities
- Easy is *relative*

Simple != Easy

Why need simplicity when ease makes life “easy”?

Simplicity leads to ease.

Ease of Understanding

Ease of Change

Ease of Debugging

Simple things are focussed.

Don't add complexity because you might need it.

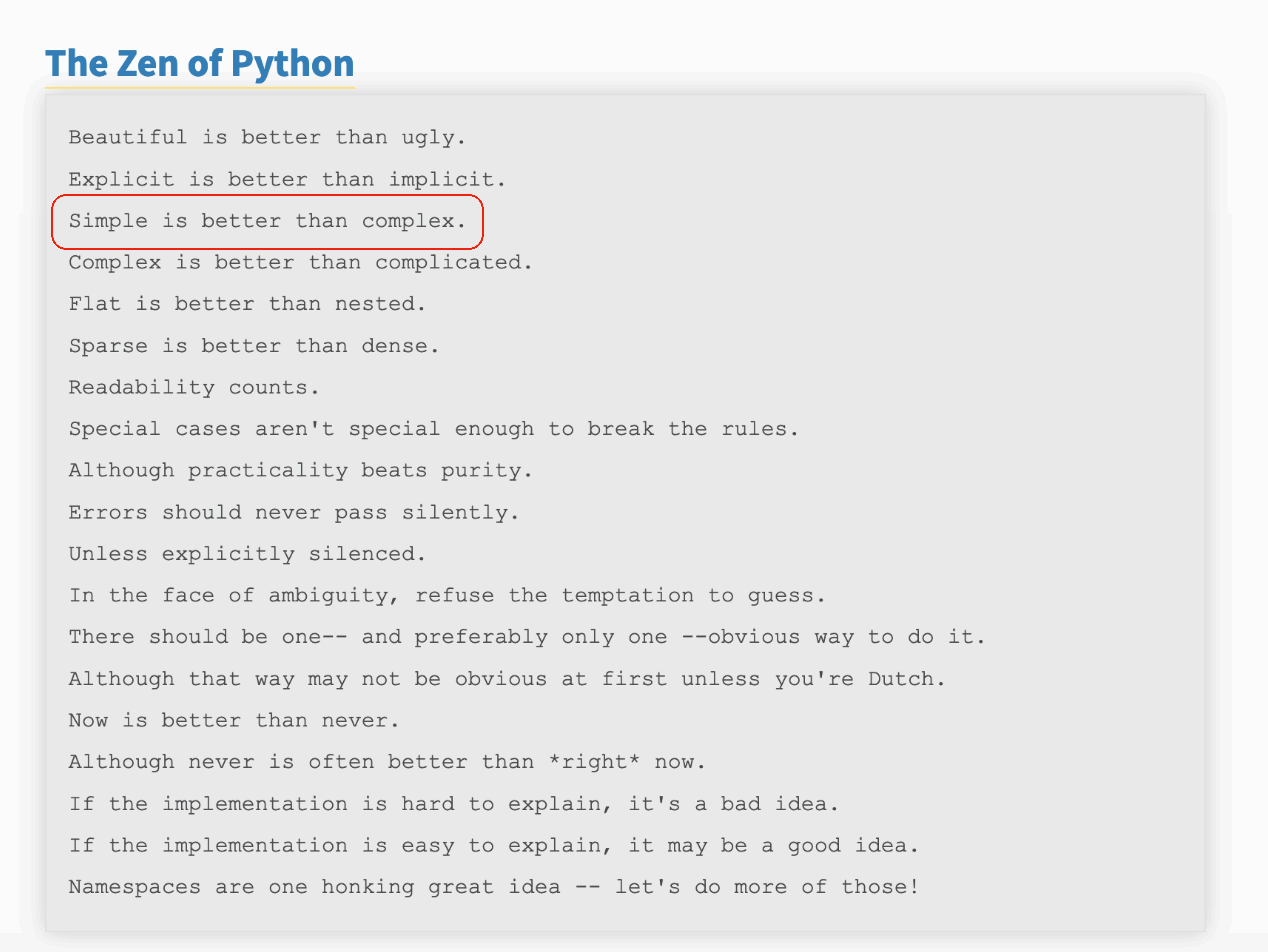
There is a need to things “quickly” in software, resist it.

We need to start asking the right questions

You might know Spring Boot but

We **read code** more than we write.

We re



The Zen of Python

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```