

Trace

By Kenny, Jacob, Ethan, Deepti, and Quadir

Table of Contents

01

Introductions

Meet Team Trace.

02

Problem Space

Premium digital art
and reuploading.

03

Solution

Where Trace comes
in.

04

Monetization of Trace

Stakeholders,
Competitors and
Value Proposition

Table of Contents

05

Steganography
Intro to
Steganography

06

Structure
What our project
does

07

Demo
Seeing Trace in
action

08

Future Additions
Features we could
implement

09

Questions
Any Questions?



01

Introductions/Mic Check

Introduction / Mic Check



Jacob Dyer
Sophomore
ITWS



Kenny Lee
Sophomore
Dual ITWS and BSAN



Quadir Russell
Junior
Dual ITWS and BSAN

Introduction / Mic Check



Deepti Sachin
Junior
ITWS & CS



Ethan Whitton
Senior
ITWS

02

Problem Space

Problem Space

- Artists, photographers, and other visual creators online can make a living producing and selling access to their works online.
- It's a fragile arrangement: creators risk their buyers reuploading or redistributing their images elsewhere on the internet and cutting into what could have been another sale.
- This is unavoidable. Once an image is sent to the buyer's computer, there is no way to prevent unauthorized distribution.

What recourse does the creator have?

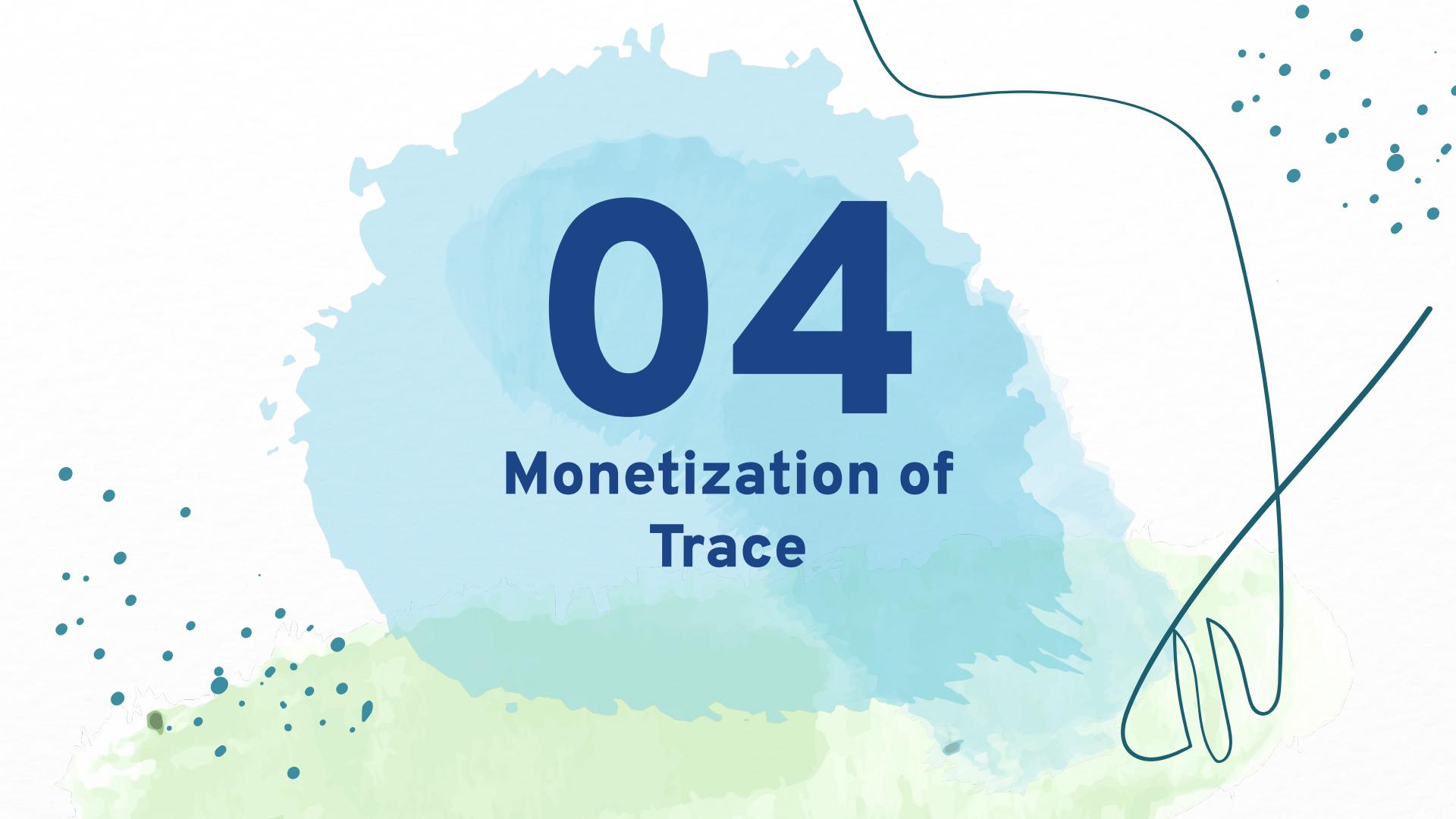


03

Solution

Trace

- Trace is an image gallery website.
- However, we also provides a special steganographic service for image creators who wish to distribute and enforce exclusive content among a select number of invitees.
- Trace is an environment where creators can enforce their own distribution arrangements to minimize the risk of their content being leaked.



04

Monetization of Trace

Artists/Creators

Creators get the benefits of hosting their images along with the benefit of securing their images

Users/Stakeholders

Followers of Creators

Higher resolution content alongside secondary beneficiaries of creators' security

Moderators

Straightforward concrete proof when unauthorized redistribution is reported

Different User Permissions

User

Users will be separated
into two categories

Free User

- Capped number of Uploads per month
- Limited features

Paid User

- Unlimited uploads and image hosting
- Access to all features Trace has to offer

Competitors

Image Hosting

In the image hosting space we would be competing with companies like imgur, Imgbox, or ImgPile.

Social Media

In the social media realm we compete with companies like Twitter, and Instagram

Creator Patron Services

In the creator-patreon services space we compete with companies like Patreon or OnlyFans.

Security

Using Trace's fingerprint feature you will know who leaked your image

Image Hosting

You will feel comfortable giving away your content given how easily it can be taken down if it is sent or posted by unauthorized users

Transparency

With this almost forced transparency users will be less likely to share your content which in turn forces more people to pay for premium art

Trace's Value Proposition

Cost Effective

With Trace's different membership levels users can choose whether or not they want to pay to use the service or not

Clients



Commercial

Our commercial clients will likely be businesses looking to use our steg service to protect their work



Freelance

Our freelance clients will likely be photographers or artists looking to sell their work online



05

Intro to Steganography

Steganography (n.): “The art or practice of concealing a message, image, or file within another message, image, or file.”

“A Picture’s Worth A Thousand Words”



Can you spot the difference?

“A Picture’s Worth A Thousand Words”



The one on the right has a message embedded!

```
In [8]: decodeLSB(stegImage)
Out[8]: 'But most importantly, be kind.'
```

Least-Significant Bit Steganography

- All images are made of pixels.
- Pixels are colored according to RGB in most picture formats.

Color = (255, 255, 255)

- Each number corresponds to a color channel, Red, Green, and Blue respectively.
- Each number is represented in binary:

R = 1111 1111

Underlined: the least-significant bit.

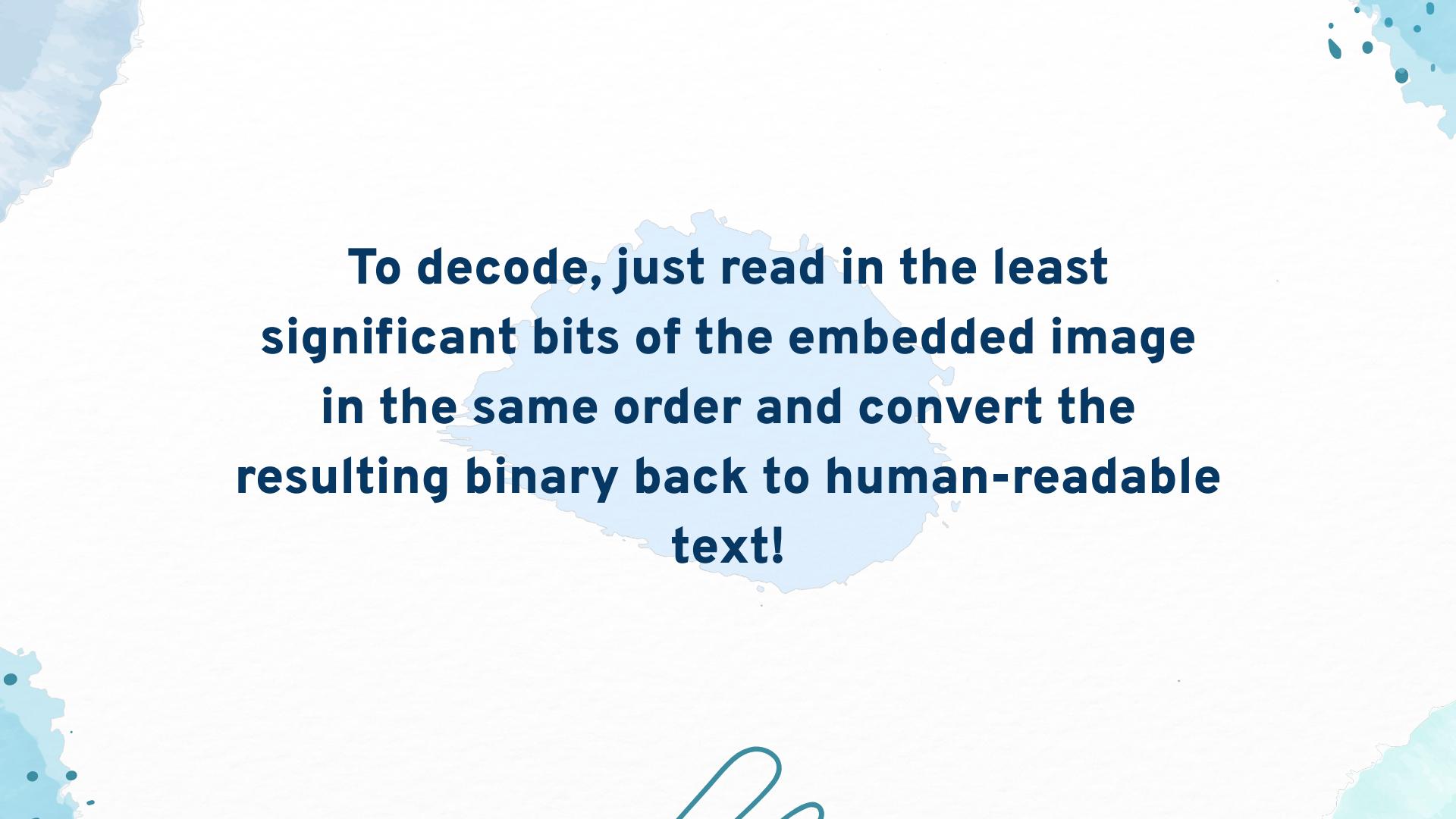
Least-Significant Bit Steganography

We wrote a function to embed a message within an image:

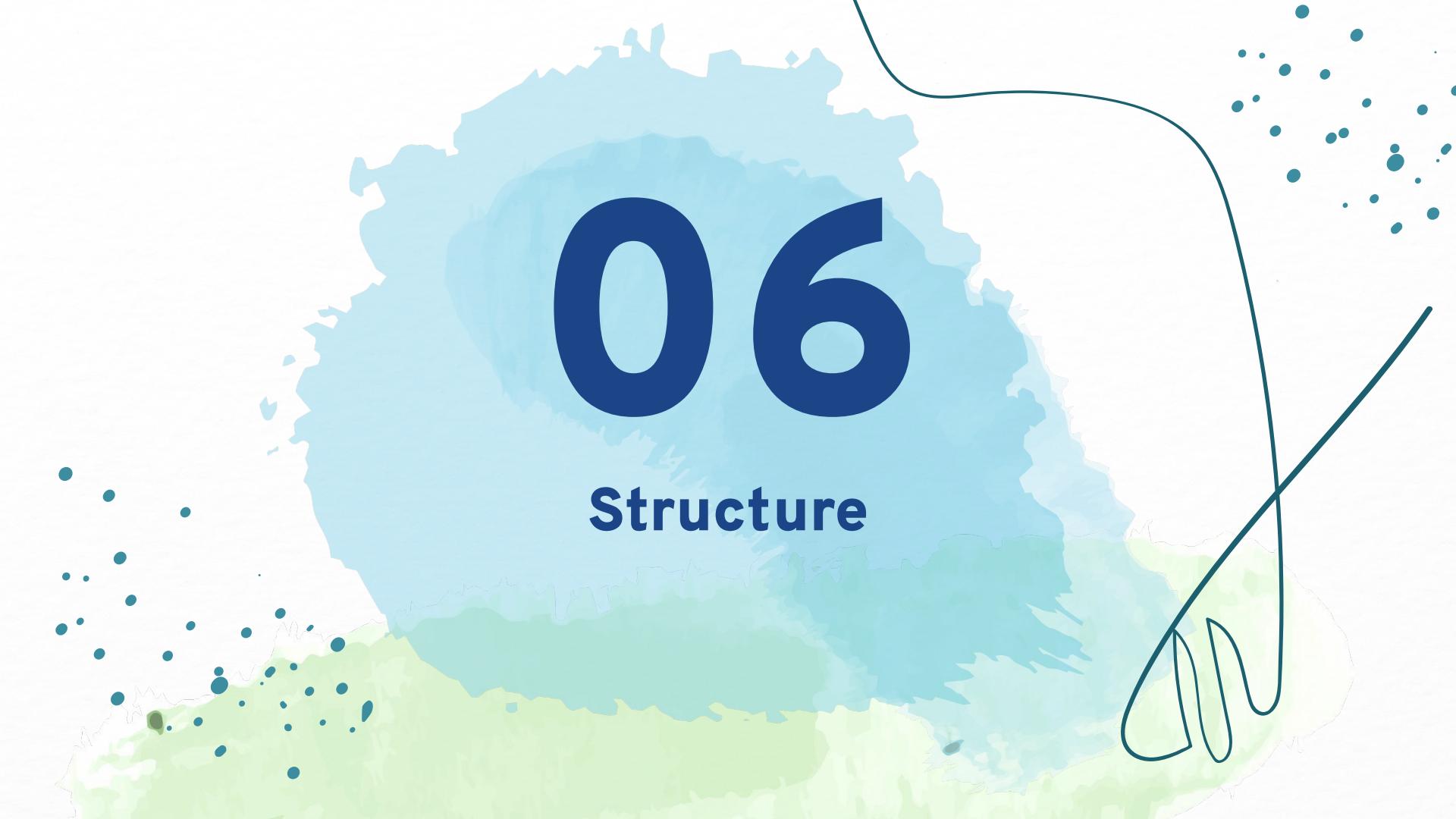
encodeLSB(Cover Image, Message)

1. Convert the message to binary.
2. For each bit in the binary message:
 - a. Open up the next color channel. (3 channels per 1 pixel)
 - b. If the message bit is 0, replace the color channel LSB with 0.
 - c. If the message bit is 1, replace the color channel LSB with 1.
3. Return the image.

Now you have an embedded image!



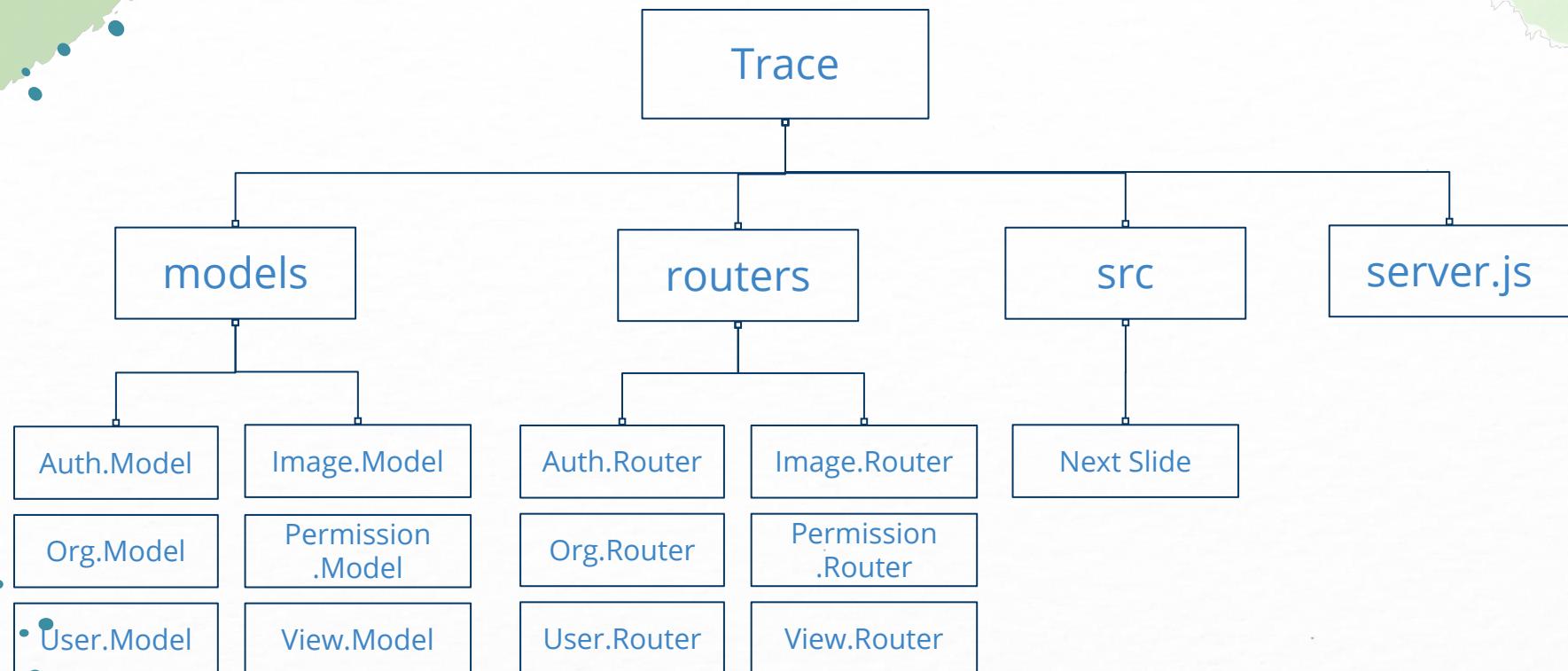
To decode, just read in the least significant bits of the embedded image in the same order and convert the resulting binary back to human-readable text!



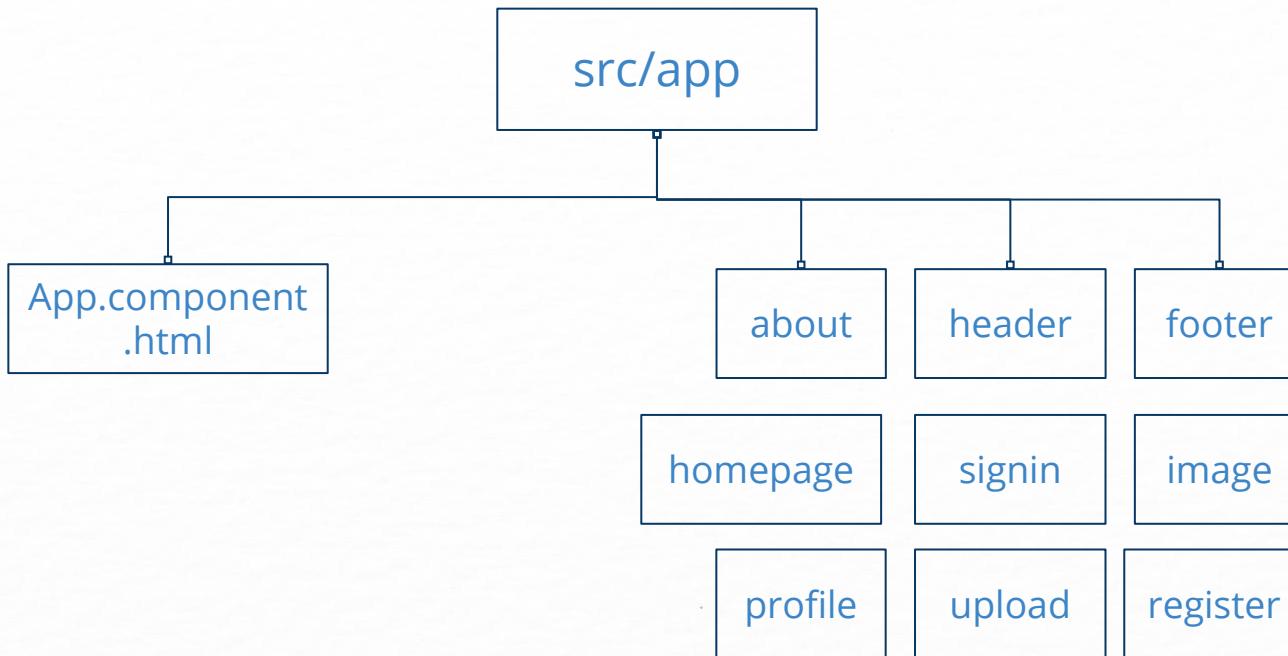
06

Structure

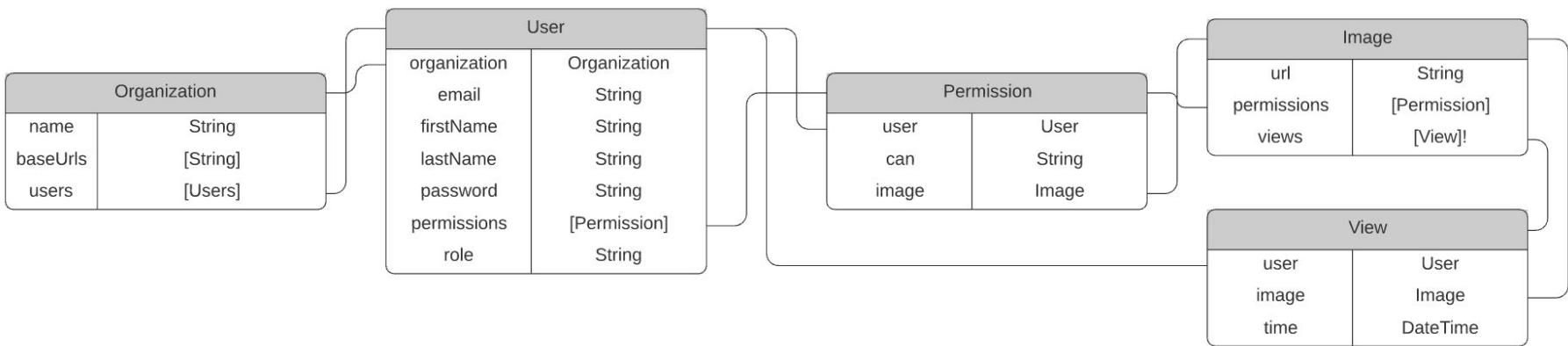
File Structure



File structure



Data and API Design



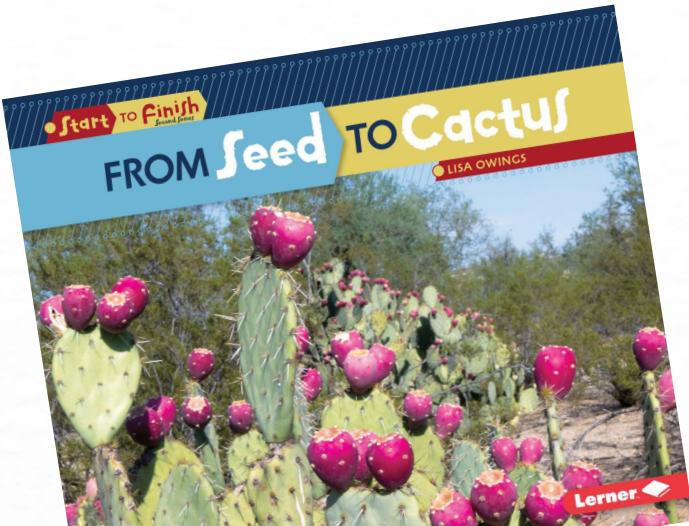
```
route("/organizations")
  post("/create")
  get("/:id")
  post("/update/:id")
  delete("/delete/:id")
```

```
route("/users")
  get("/get/images/:id")
  post("/update/:id")
  delete("/delete/:id")
```

```
route("/permissions")
  post("/create")
  get("/:id")
  post("/update/:id")
  delete("/delete/:id")
```

```
route("/image")
  post("/create")
  get("/:id") # modifies views
  post("/update/:id")
  delete("/delete/:id")
  post("/against/:id") # another image
  get("/permission/:perm") # retrieves image
```

Seeding



```
~/trace main ✓  
▶ |
```



07

Trace Demo



Demo

08

Future of Trace

Future Additions

- CI Testing Additions
 - API Unit
 - Accessibility
- Pagination, improve user experience and loading time
- Image albums

09

Thanks for listening!

Any Questions?