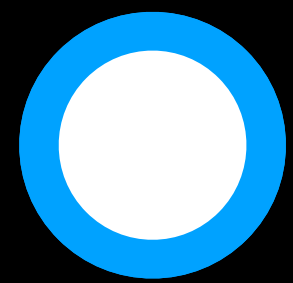


# CS50 Beyond

**Security**

Git

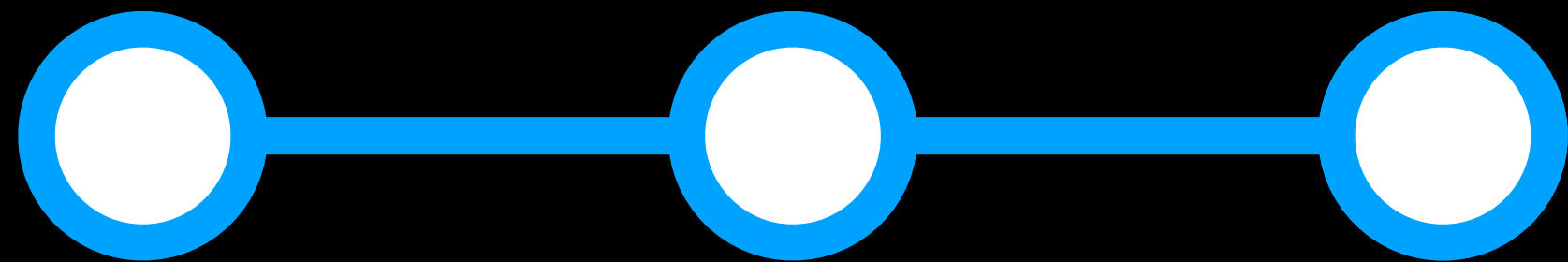


first  
commit



first  
commit

credentials  
exposed



first  
commit

credentials  
exposed

credentials  
removed

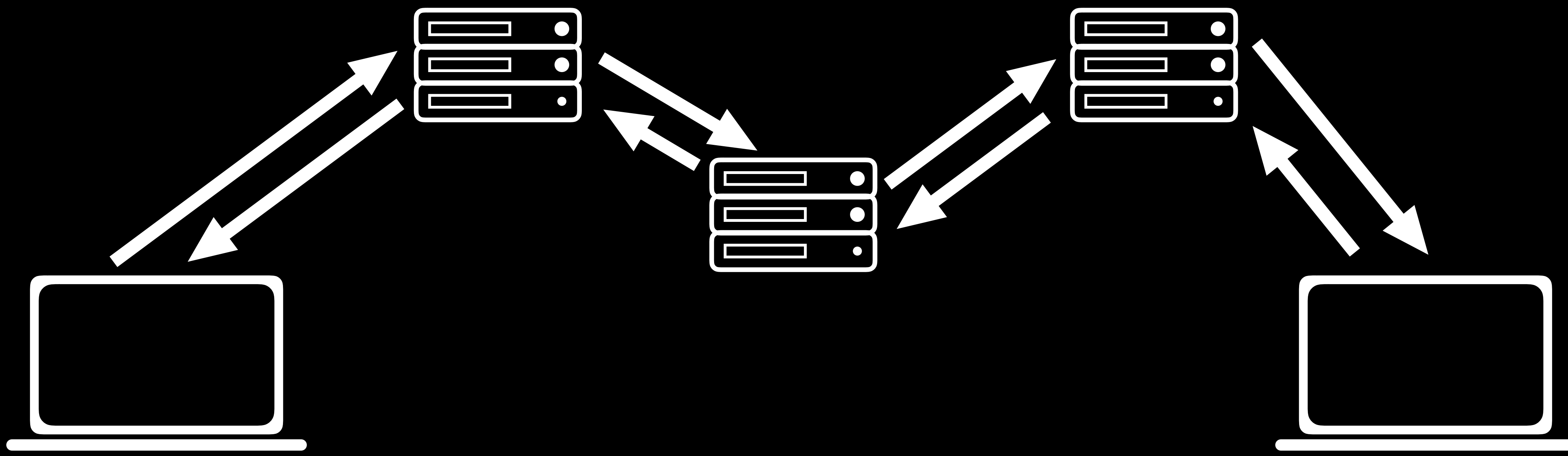
HTML

```
<a href="ur11">  
    ur12  
</a>
```



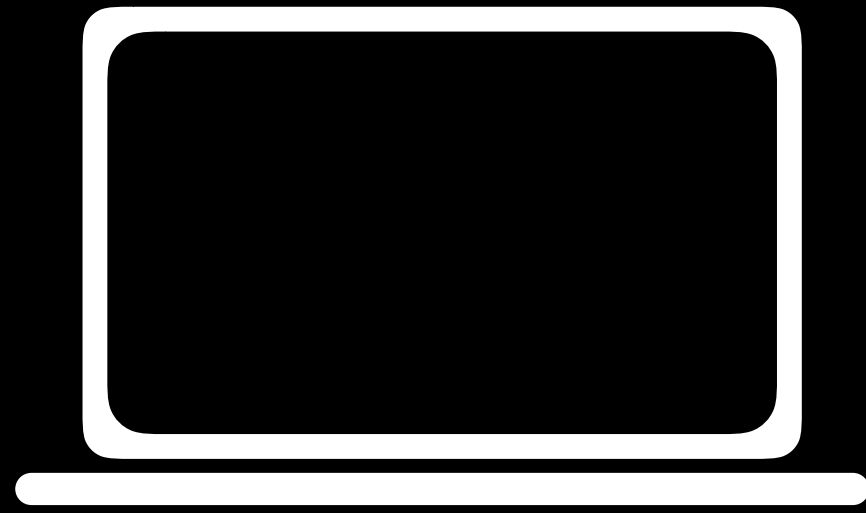
**Flask**

# HTTP and HTTPS



# Cryptography

# Secret-Key Cryptography



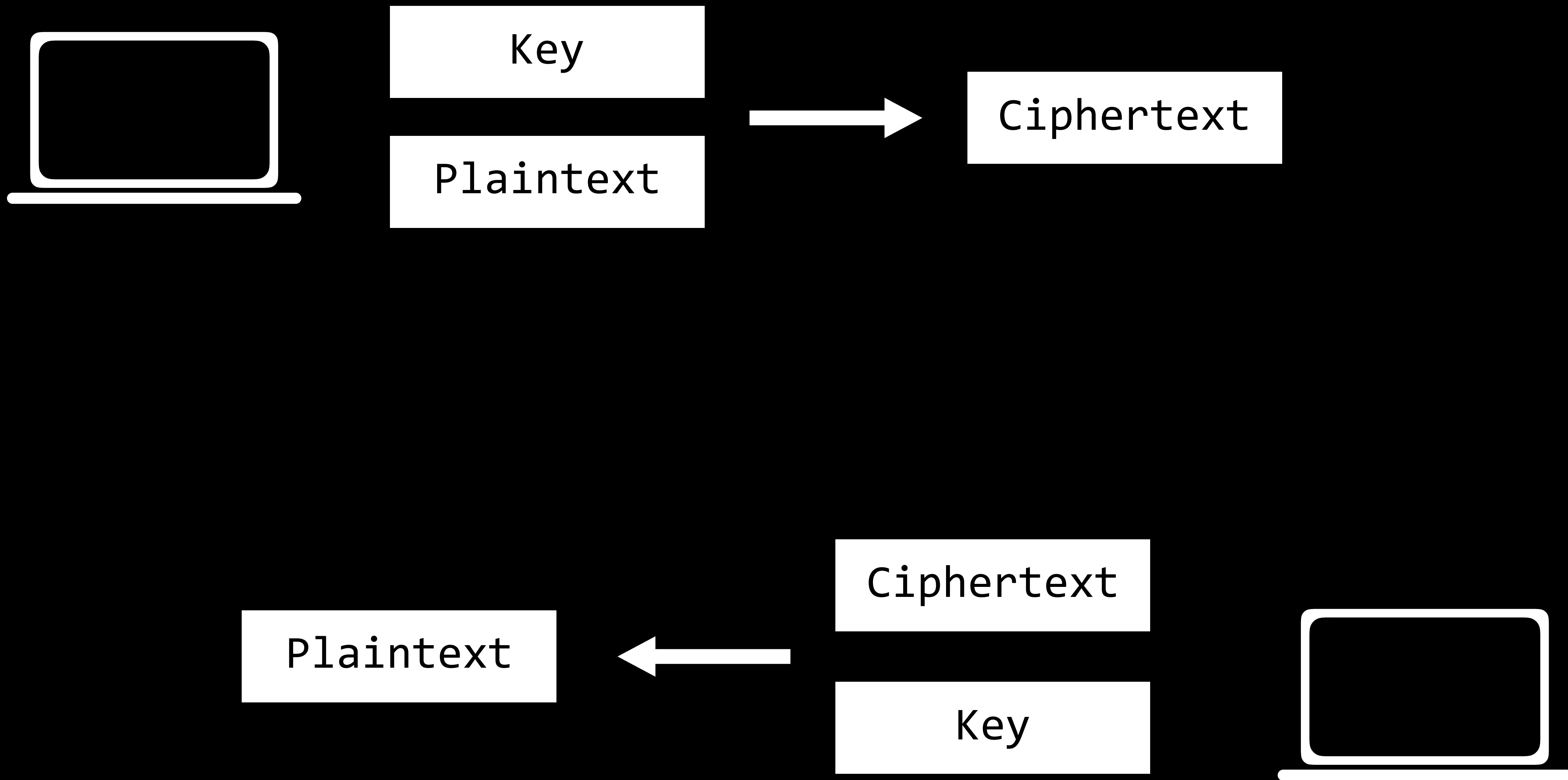
Key

Plaintext



Ciphertext



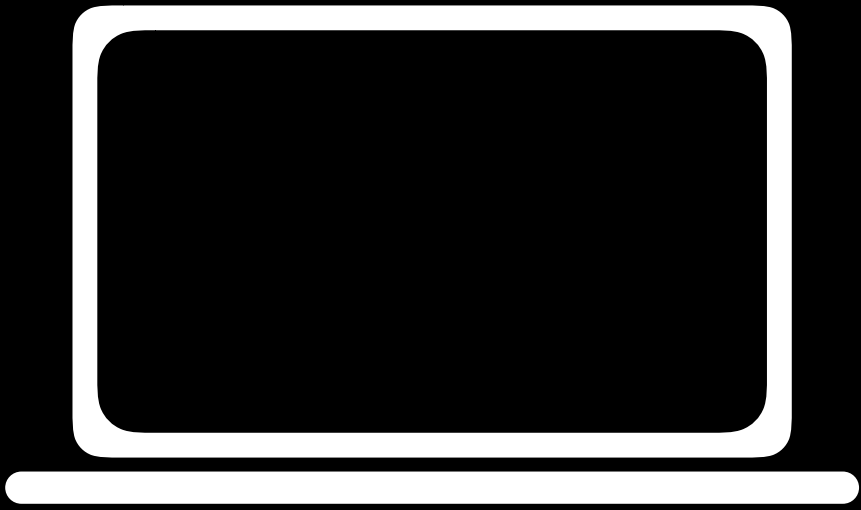


# Public-Key Cryptography



Public Key

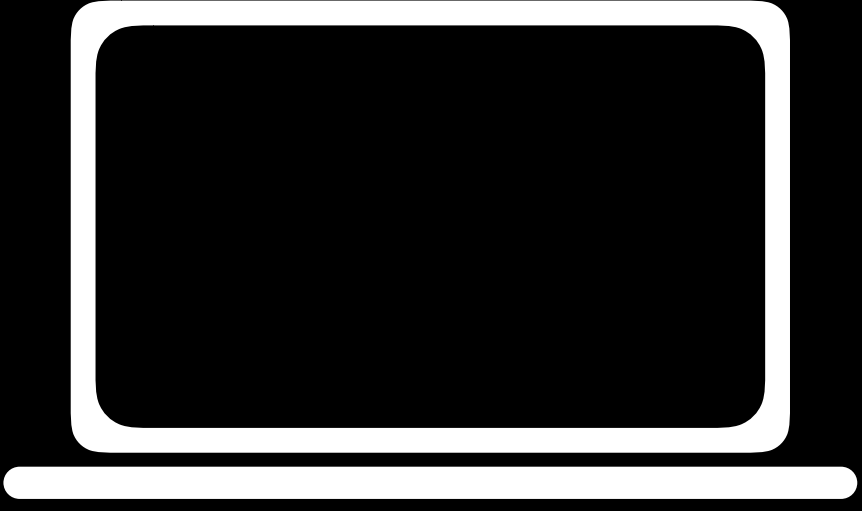
Private Key



Plaintext

Public Key

Private Key





Plaintext

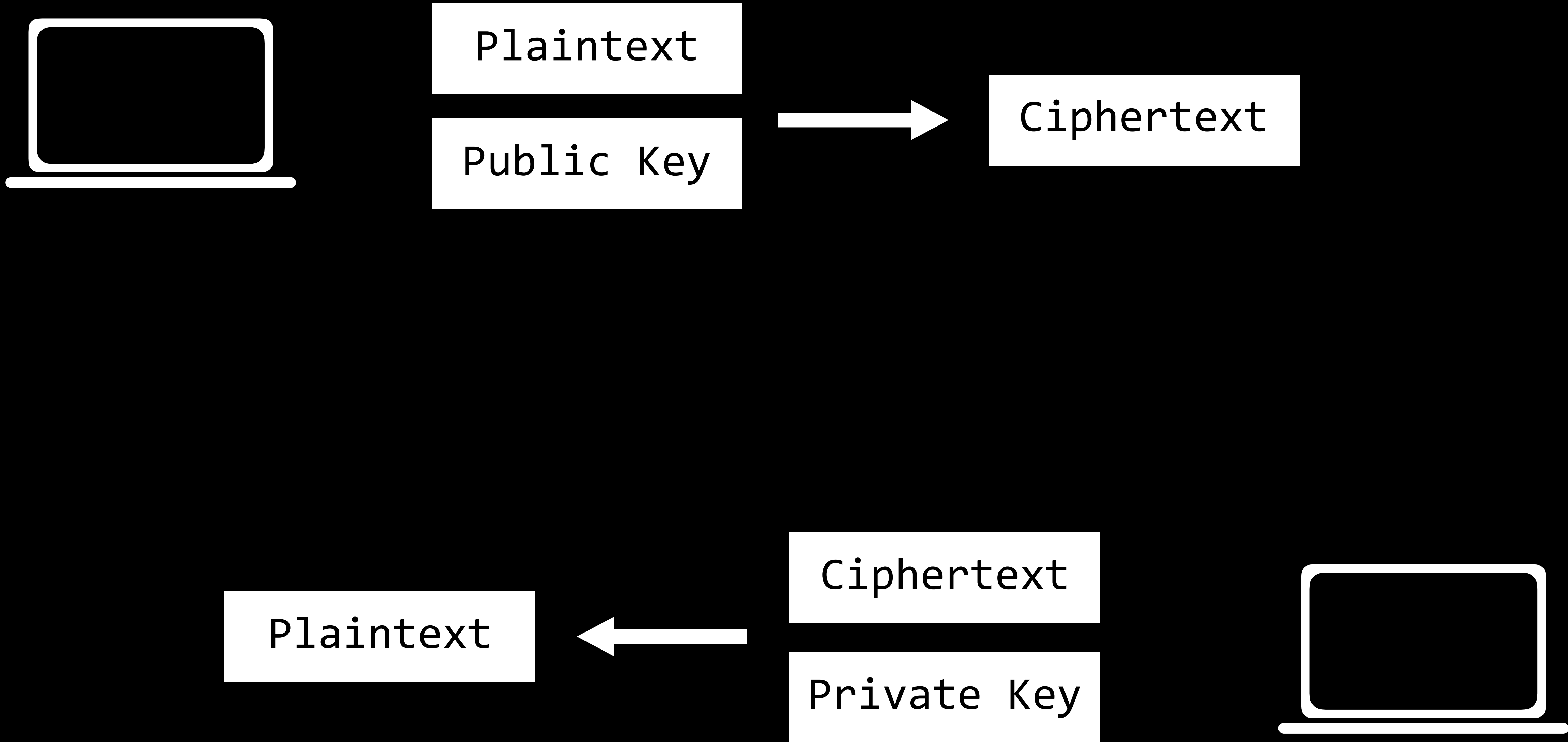
Public Key



Ciphertext

Private Key





# Environment Variables

```
app.config["SECRET_KEY"] = "dHd1bnR5ZW1naHQ"
```

```
app.config["SECRET_KEY"] = os.environ.get("SECRET_KEY")
```

SQL

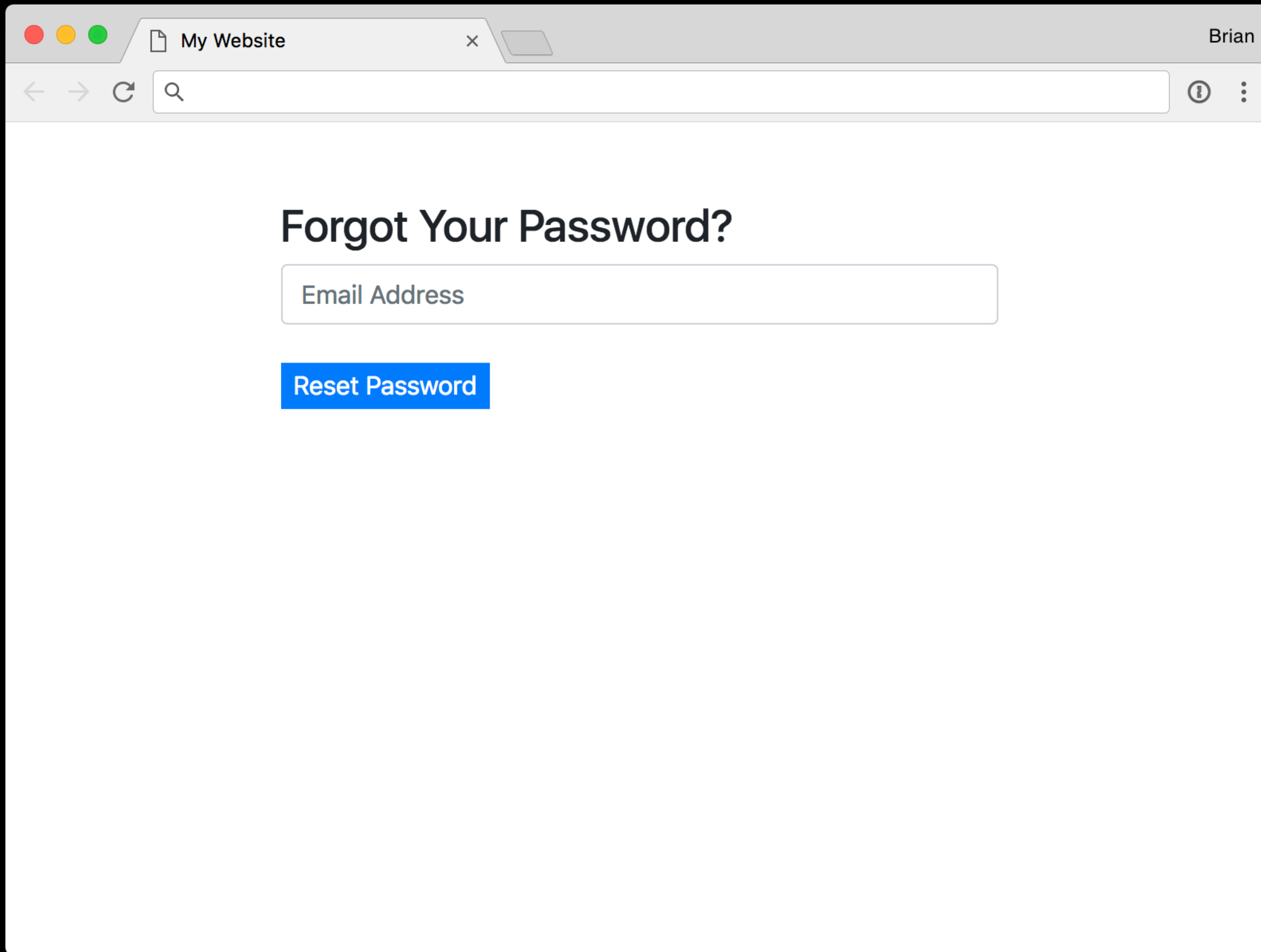


# users

<b>id</b>	<b>username</b>	<b>password</b>
<b>1</b>	<b>andrew</b>	<b>hello</b>
<b>2</b>	<b>athena</b>	<b>password</b>
<b>3</b>	<b>josh</b>	<b>12345</b>
<b>4</b>	<b>julia</b>	<b>abcdef</b>
<b>5</b>	<b>krishna</b>	<b>qwerty</b>

# users

<b>id</b>	<b>username</b>	<b>password</b>
<b>1</b>	<b>andrew</b>	<b>48c8e8c3f9e80b68ac67304c7c510e9fcb</b>
<b>2</b>	<b>athena</b>	<b>6024aba15e3f9be95e3c9e6d3bf261d78e</b>
<b>3</b>	<b>josh</b>	<b>90112701066c0a536f2f6b2761e5edb09e</b>
<b>4</b>	<b>julia</b>	<b>b053b7574c8a25751e2a896377e5d477c5</b>
<b>5</b>	<b>krishna</b>	<b>a4048eaaee50680532845b2025996b44a9</b>



My Website

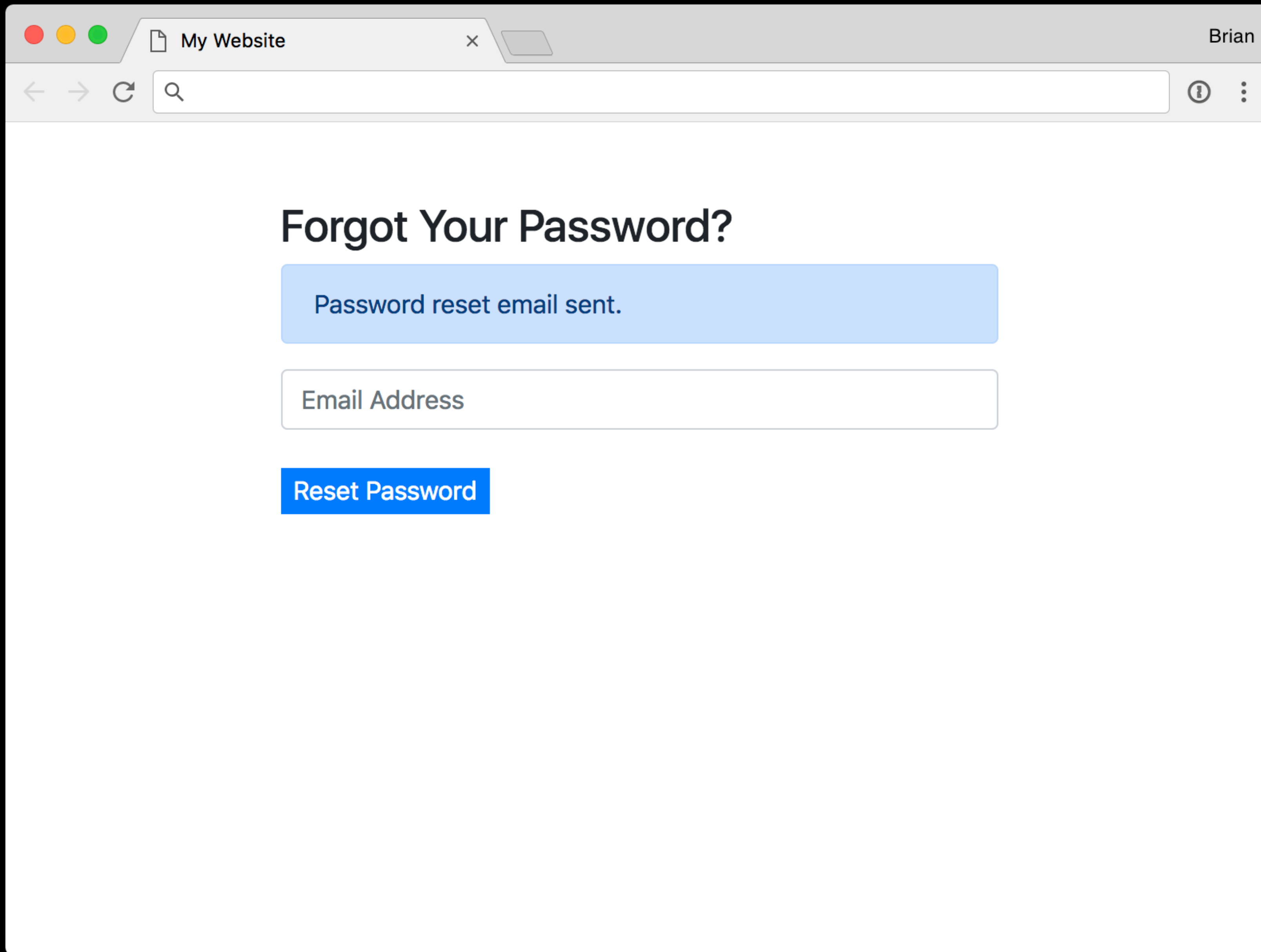


Brian



## Forgot Your Password?

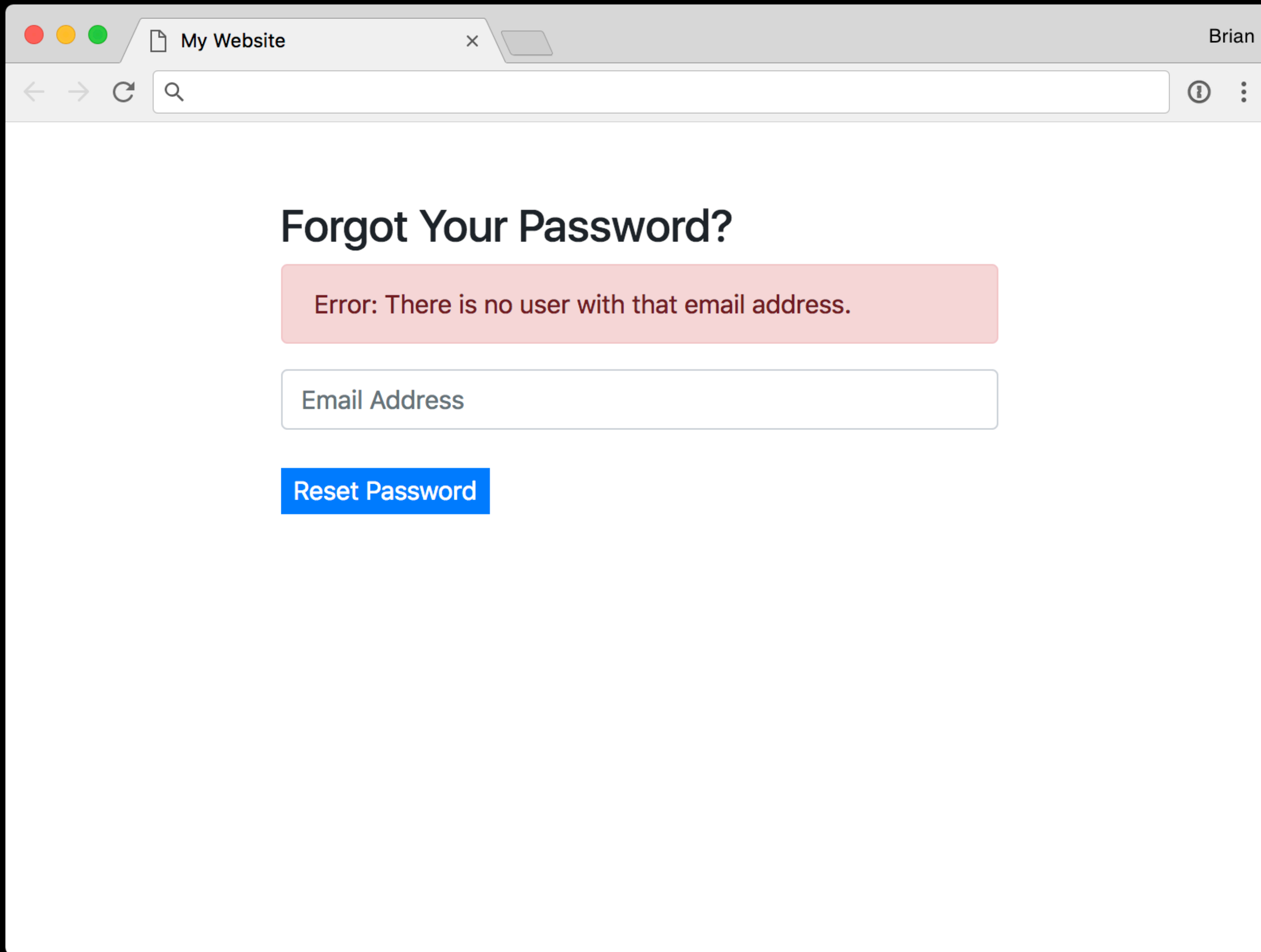
[Reset Password](#)



# Forgot Your Password?

Password reset email sent.

[Reset Password](#)



## Forgot Your Password?

Error: There is no user with that email address.

[Reset Password](#)

# SQL Injection

JavaScript

# Cross-Site Scripting



```
from flask import Flask, request

app = Flask(__name__)

@app.route("/")
def index():
    return "Hello, world!"

@app.errorhandler(404)
def page_not_found(e):
    return "Not Found: " + request.path
```

```
@app.errorhandler(404)
def page_not_found(e):
    return "Not Found: " + request.path
```

/foo

```
@app.errorhandler(404)
def page_not_found(e):
    return "Not Found: " + request.path
```

```
/<script>alert('hi')</script>
```

```
@app.errorhandler(404)  
def page_not_found(e):  
    return "Not Found: " + request.path
```

```
/<script>document.write(  
  '')</script>
```

```
@app.errorhandler(404)  
def page_not_found(e):  
    return "Not Found: " + request.path
```

# Cross-Site Request Forgery

```
<body>  
  <a href="http://yourbank.com/transfer?to=brian&amt=2800">  
    Click Here!  
  </a>  
</body>
```

```
<body>
```

```
  
```

```
</body>
```



```
<body>
  <form action="https://yourbank.com/transfer"
        method="post">
    <input type="hidden" name="to" value="brian">
    <input type="hidden" name="amt" value="2800">
    <input type="submit" value="Click Here!">
  </form>
</body>
```

```
<body onload="document.forms[0].submit()">
  <form action="https://yourbank.com/transfer"
    method="post">
    <input type="hidden" name="to" value="brian">
    <input type="hidden" name="amt" value="2800">
    <input type="submit" value="Click Here!">
  </form>
</body>
```

**Scalability**



Server



Server

# Benchmarking

# Vertical Scaling



Server





Server

# Horizontal Scaling

Server

Server

Server

```
graph TD; LB[Load Balancer] --- S1[Server]; LB --- S2[Server];
```

Load Balancer

Server

Server

# Load Balancing

# Load Balancing Methods

- Random Choice
- Round Robin
- Fewest Connections
- ...

# Session-Aware Load Balancing

- Sticky Sessions
- Sessions in Database
- Client-Side Sessions
- ...



# Autoscaling

```
graph TD; LB[Load Balancer] --- S1[Server]; LB --- S2[Server];
```

Load Balancer

Server

Server

```
graph TD; LB[Load Balancer] --- S1[Server]; LB --- S2[Server]; LB --- S3[Server];
```

Load Balancer

Server

Server

Server

```
graph TD; LB[Load Balancer] --- S1[Server]; LB --- S2[Server]; LB --- S3[Server]; LB --- S4[Server];
```

Load Balancer

Server

Server

Server

Server

```
graph TD; LB[Load Balancer] --- S1[Server]; LB --- S2[Server]; LB --- S3[Server];
```

Load Balancer

Server

Server

Server

```
graph TD; LB[Load Balancer] --- S1[Server]; LB --- S2[Server]; LB --- S3[Server]; LB --- S4[Server];
```

Load Balancer

Server

Server

Server

Server

```
graph TD; LB[Load Balancer] --- S1[Server]; LB --- S2[Server]; LB --- S3[Server]; LB --- S4[Server];
```

Load Balancer

Server

Server

Server

Server

# Scaling Databases

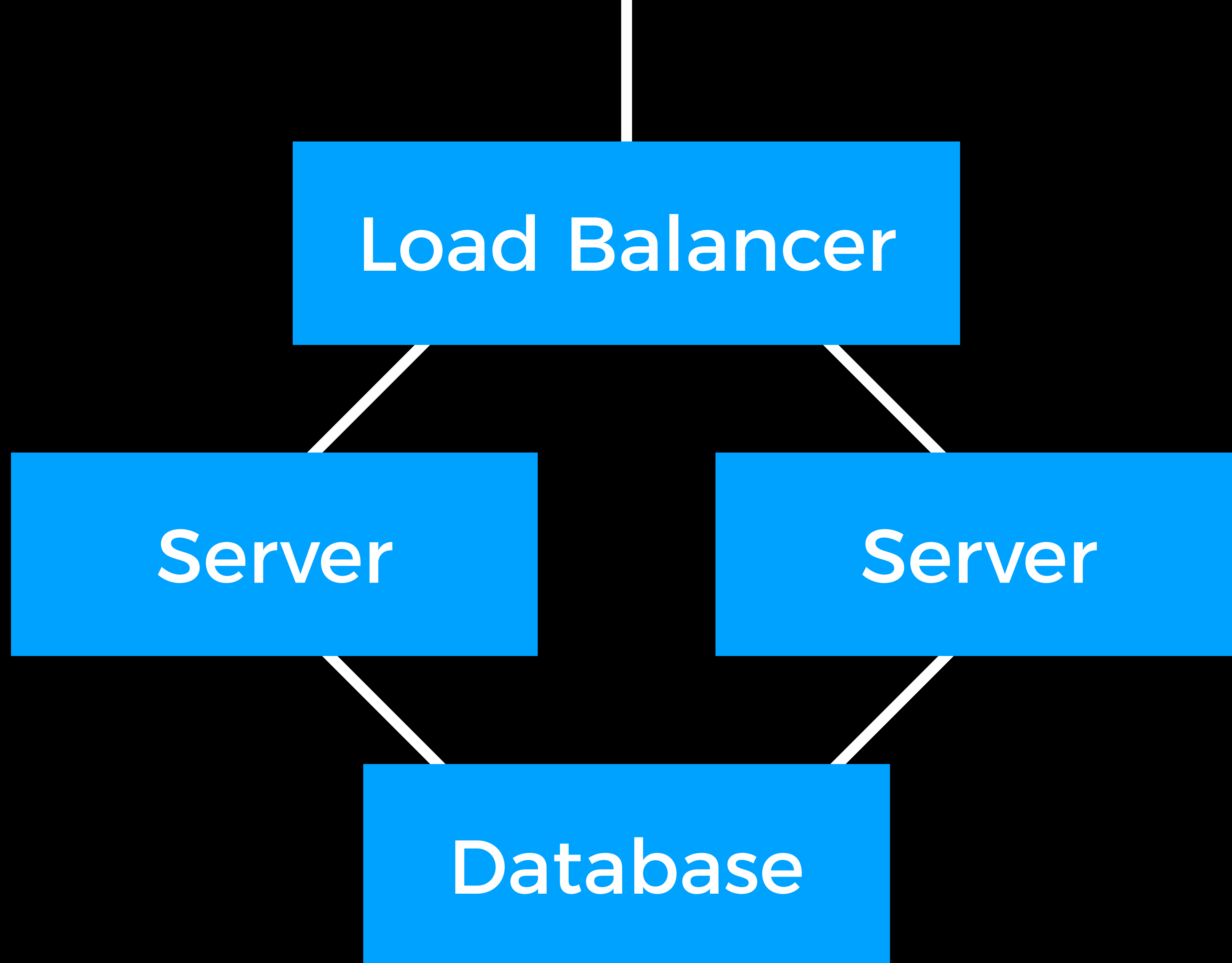


```
graph TD; LB[Load Balancer] --- S1[Server]; LB --- S2[Server];
```

Load Balancer

Server

Server



# Database Partitioning

# flights

<b>id</b>	<b>origin</b>	<b>origin_code</b>	<b>destination</b>	<b>destination_code</b>	<b>duration</b>
1	New York	JFK	London	LHR	415
2	Shanghai	PVG	Paris	CDG	760
3	Istanbul	IST	Tokyo	NRT	700
4	New York	JFK	Paris	CDG	435
5	Moscow	SVO	Paris	CDG	245
6	Lima	LIM	New York	JFK	455

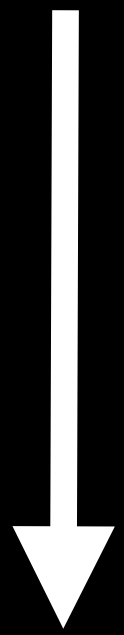
# locations

<b>id</b>	<b>code</b>	<b>name</b>
<b>1</b>	<b>JFK</b>	<b>New York</b>
<b>2</b>	<b>PVG</b>	<b>Shanghai</b>
<b>3</b>	<b>IST</b>	<b>Istanbul</b>
<b>4</b>	<b>LHR</b>	<b>London</b>
<b>5</b>	<b>SVO</b>	<b>Moscow</b>
<b>6</b>	<b>LIM</b>	<b>Lima</b>
<b>7</b>	<b>CDG</b>	<b>Paris</b>
<b>8</b>	<b>NRT</b>	<b>Tokyo</b>

# flights

<b>id</b>	<b>origin_id</b>	<b>destination_id</b>	<b>duration</b>
<b>1</b>	<b>1</b>	<b>4</b>	<b>415</b>
<b>2</b>	<b>2</b>	<b>7</b>	<b>760</b>
<b>3</b>	<b>3</b>	<b>8</b>	<b>700</b>
<b>4</b>	<b>1</b>	<b>7</b>	<b>435</b>
<b>5</b>	<b>5</b>	<b>7</b>	<b>245</b>
<b>6</b>	<b>6</b>	<b>1</b>	<b>455</b>

flights



flights\_domestic

flights\_international

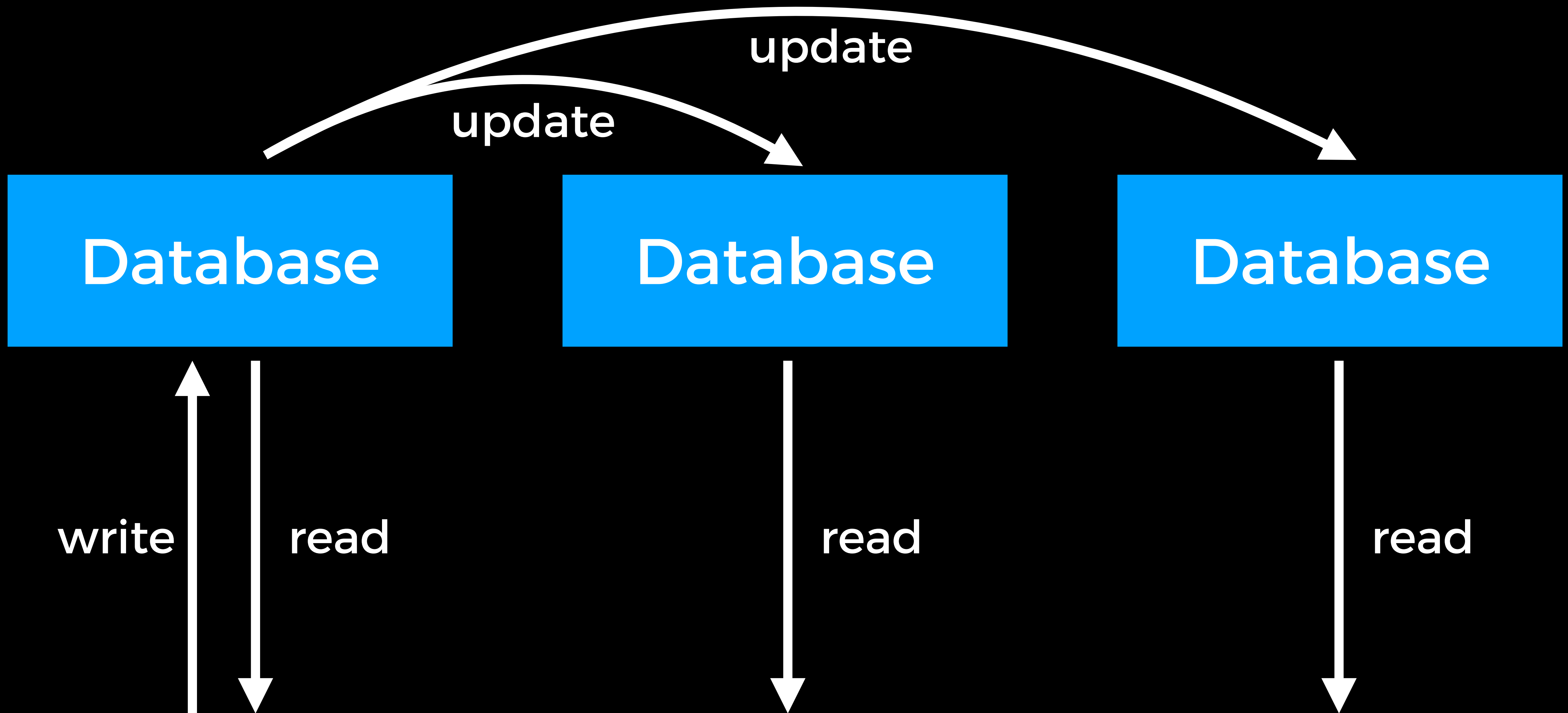
# Database Replication



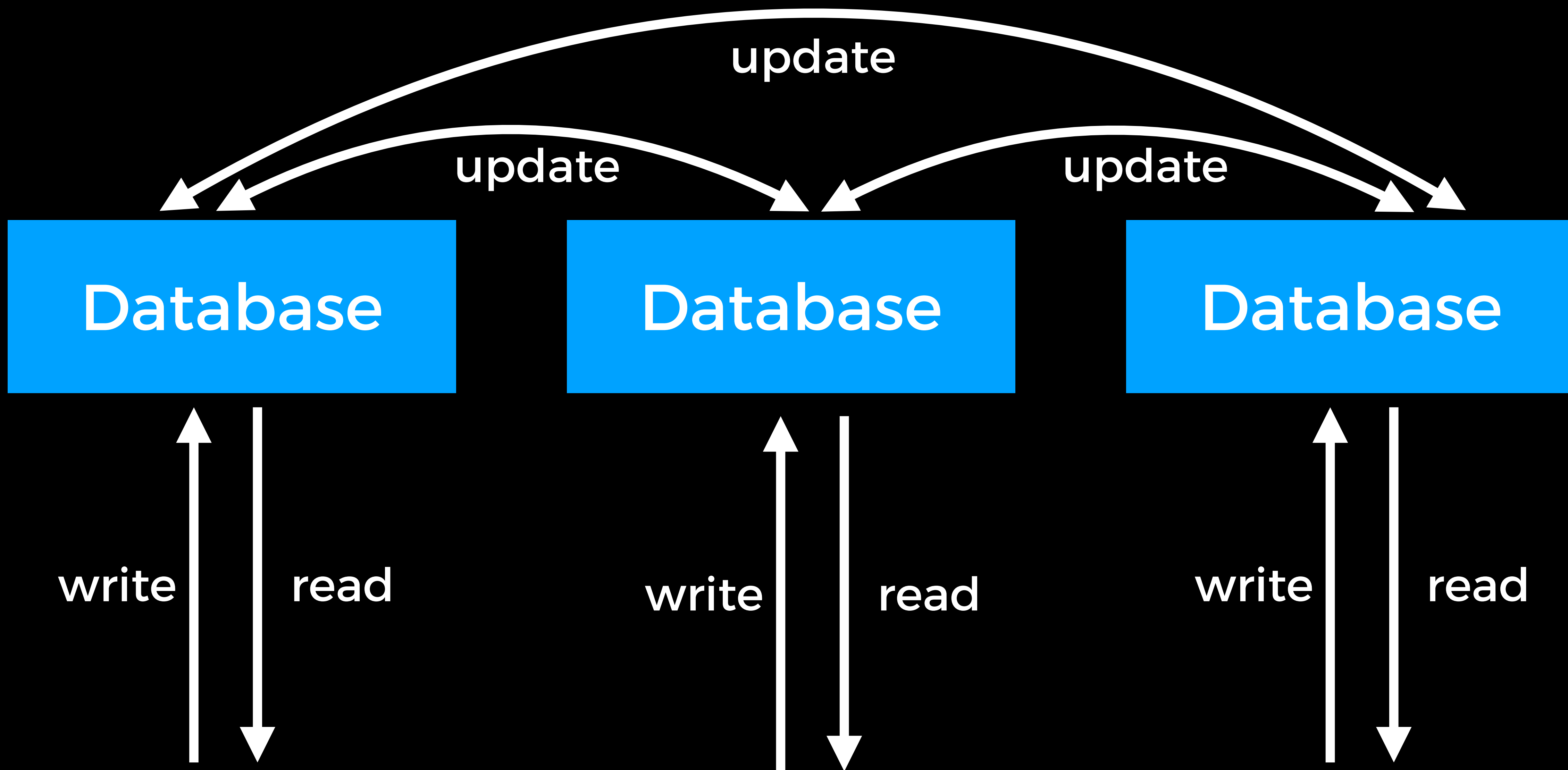
# Database Replication

- **Single-Primary Replication**
- **Multi-Primary Replication**

# Single-Primary Replication



# Multi-Primary Replication



Caching

# Client-Side Caching

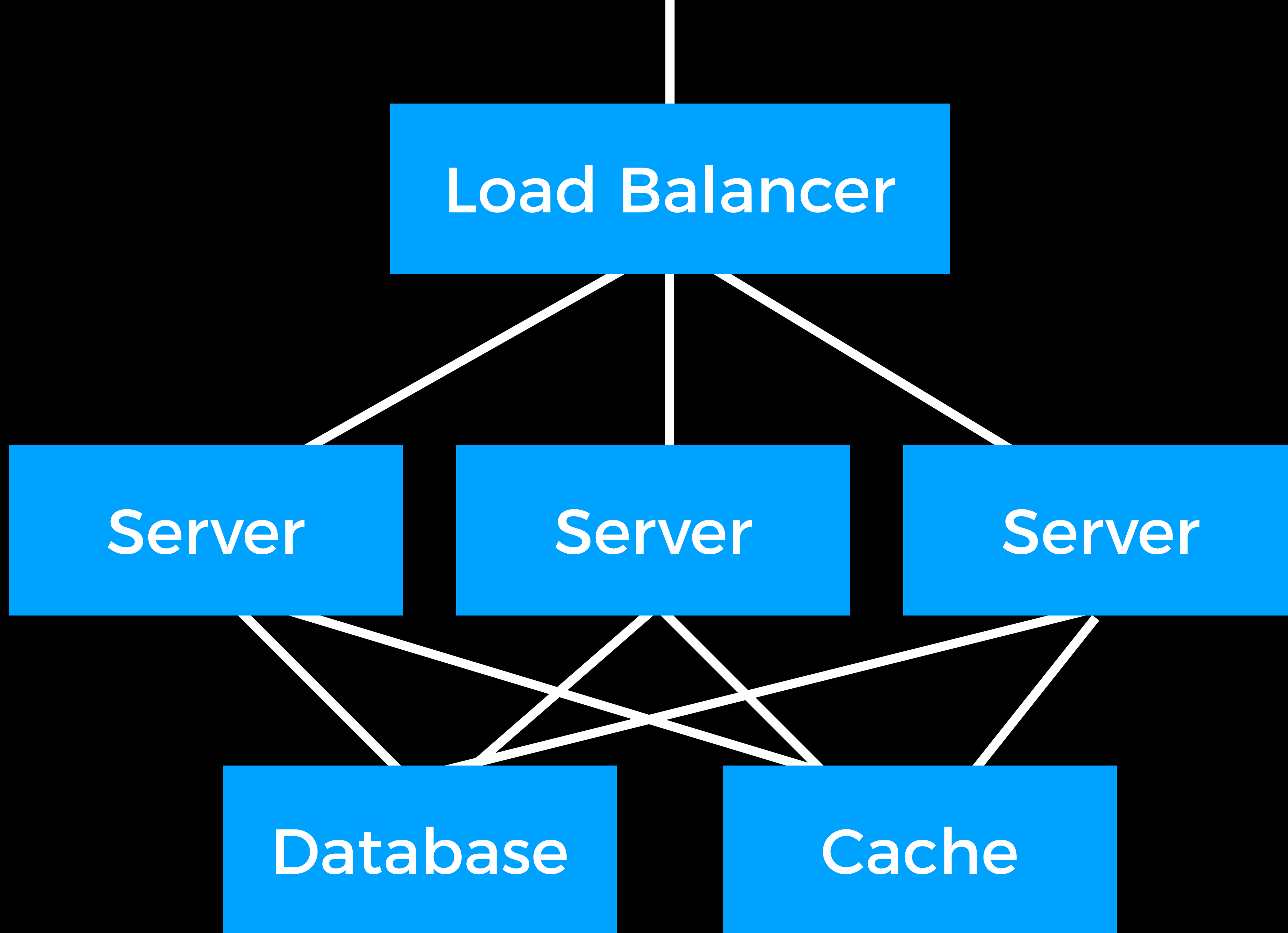
Cache-Control: max-age=86400

Cache-Control: max-age=86400

ETag: "7477656E74796569676874"

# Server-Side Caching





# CS50 Beyond