NortonLifeLock™

DECEMBER 12TH - 2019

BakingTimer: Privacy Analysis of Server-Side Request Processing Time

Iskander Sanchez-Rola, Davide Balzarotti, and Igor Santos

EURECOM
Sophia Antipolis

DeustoTech

Motivation

Cookies were originally introduced as a way to provide state awareness to websites, but nowadays they are not limited to store the login information or the current state of user. In several cases, third-party **cookies are deliberately used for web tracking**.

Motivation

Cookies were originally introduced as a way to provide state awareness to websites, but nowadays they are not limited to store the login information or the current state of user. In several cases, third-party **cookies are deliberately used for web tracking**.

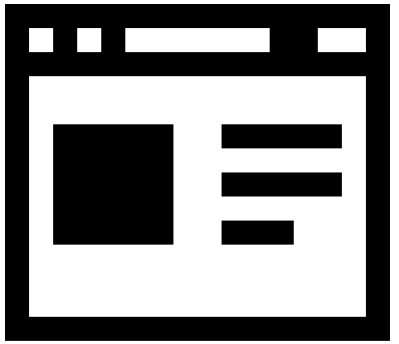
But even if the most famous, cookies are not the only technique capable of retrieving the users' browsing history. In fact, history sniffing techniques can do it without relying on **any specific code in a third-party website**, but only on code executed in one site.

Threat Model

Alice

Threat Model

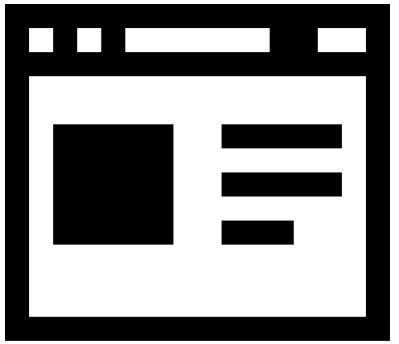
Alice



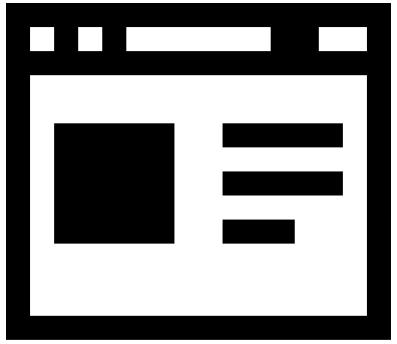
Shopping

Threat Model

Alice

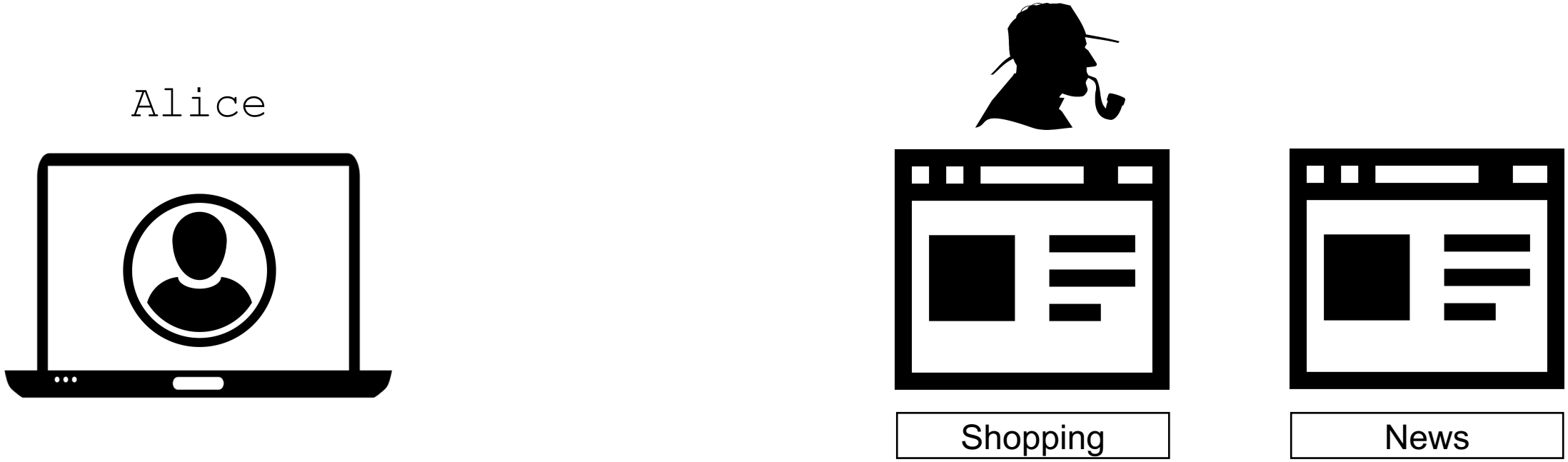


Shopping

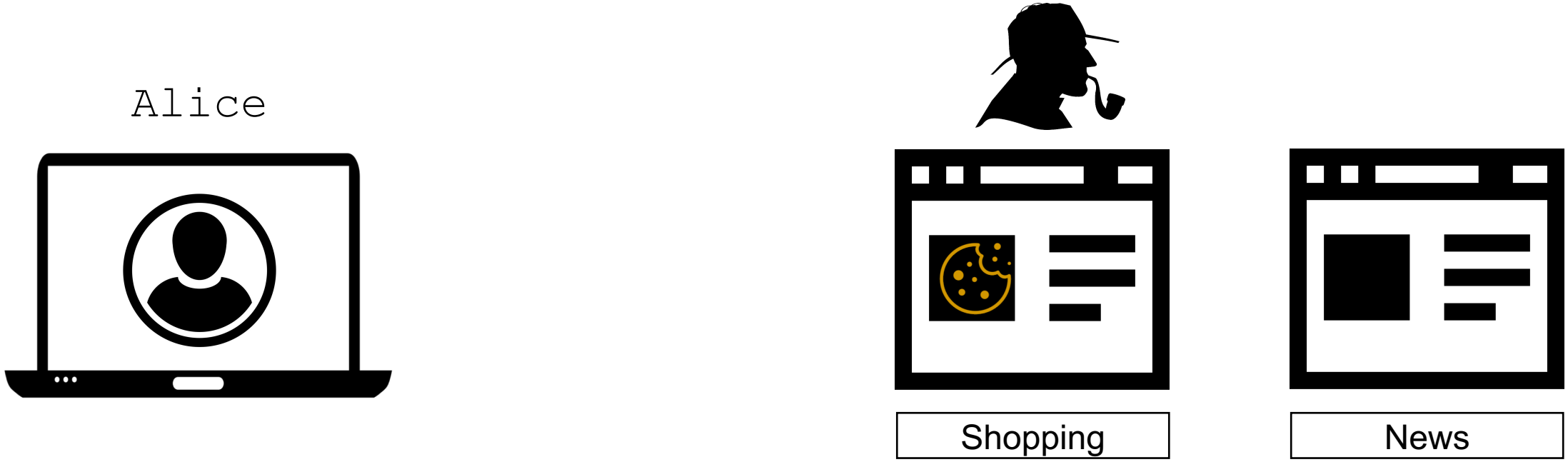


News

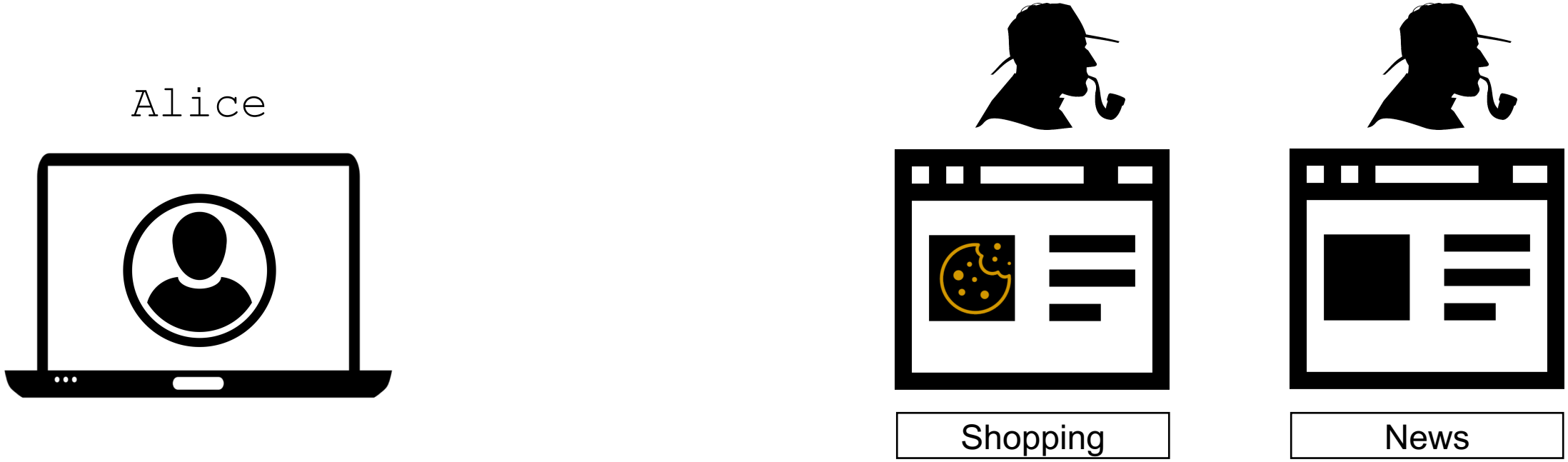
Threat Model



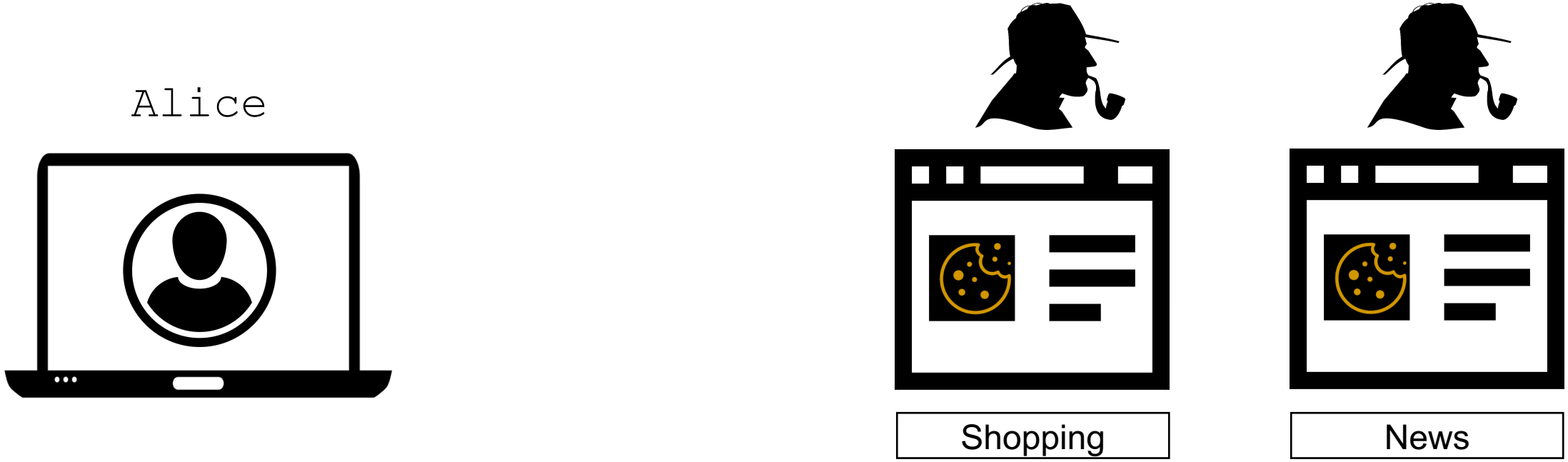
Threat Model



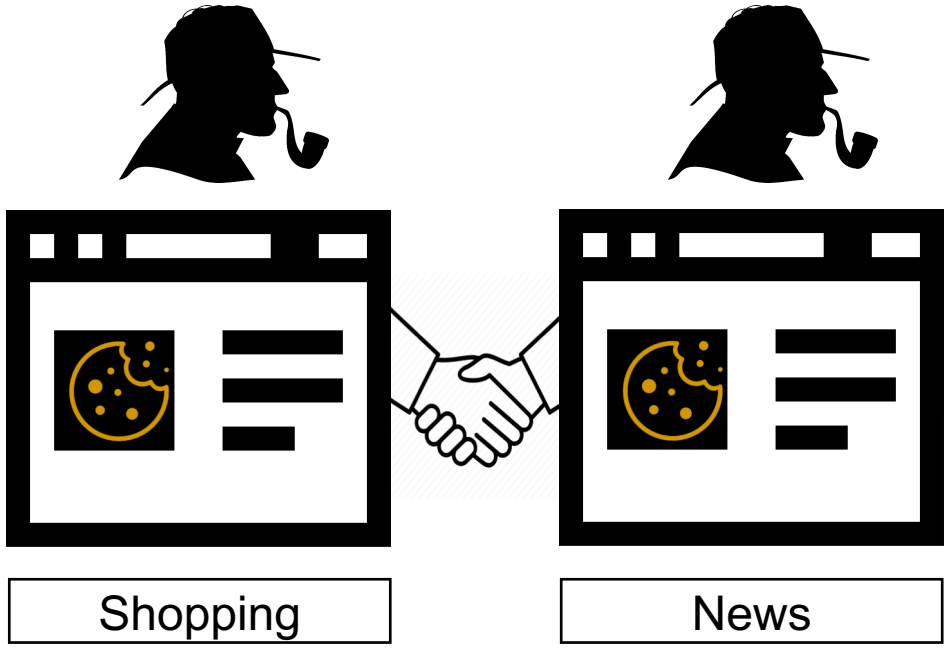
Threat Model



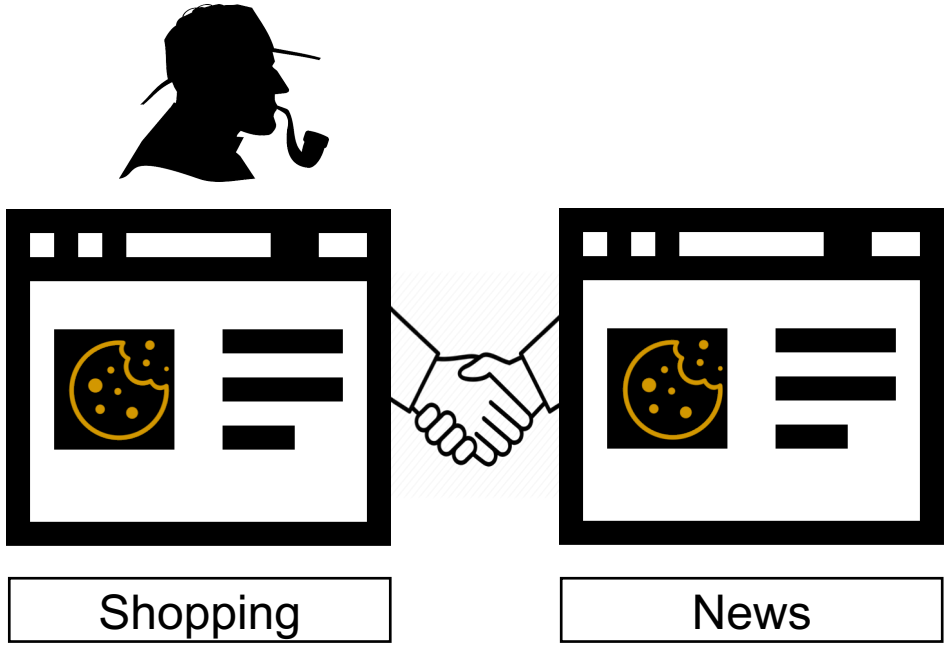
Threat Model



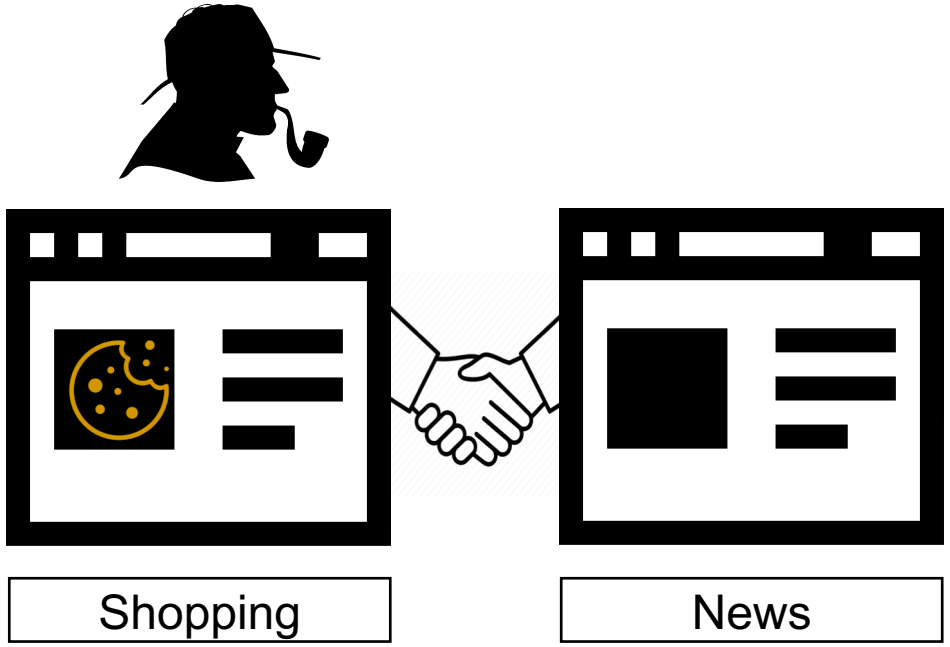
Threat Model



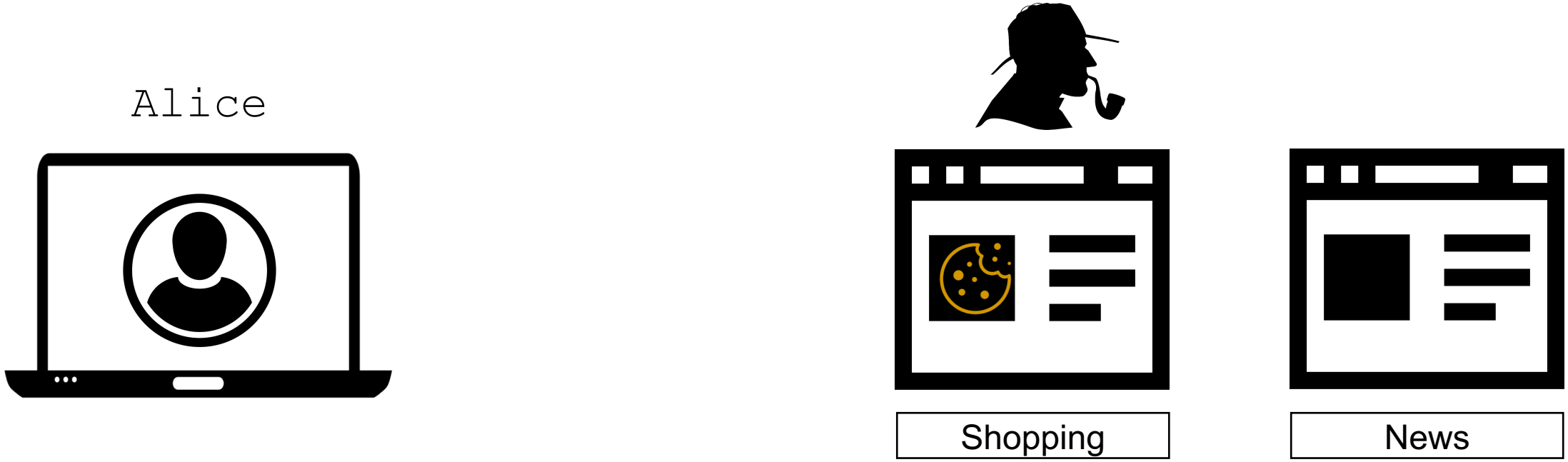
Threat Model



Threat Model



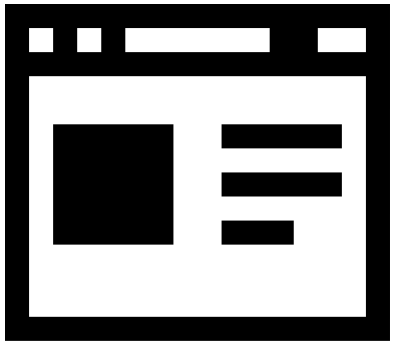
Threat Model



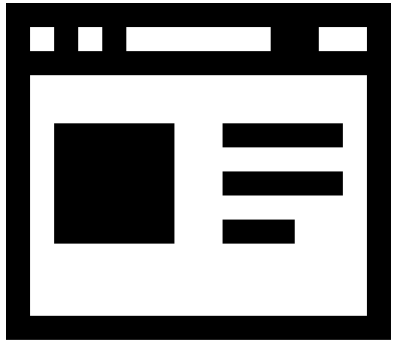
Threat Model

Threat Model

Alice

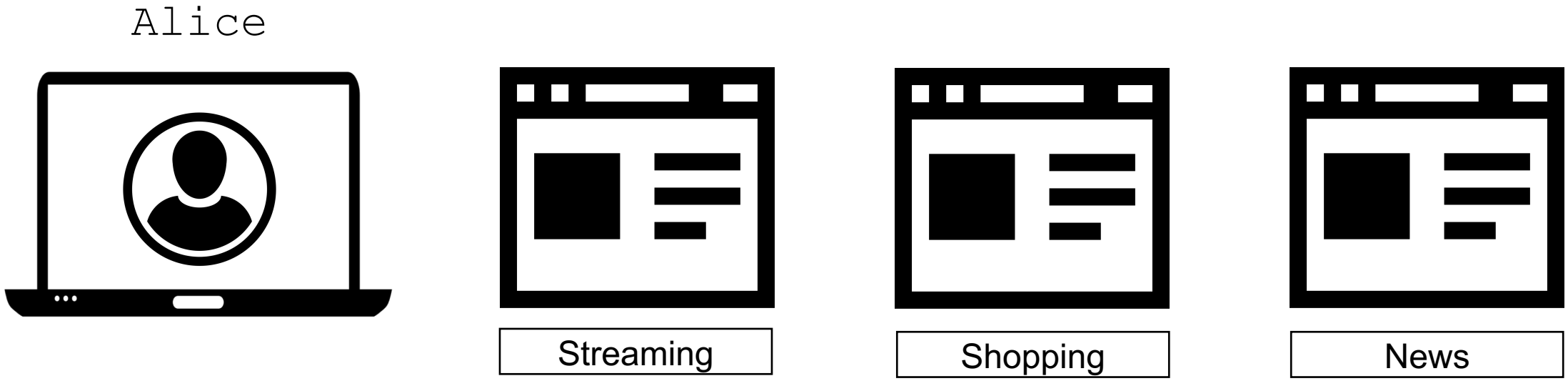


Shopping

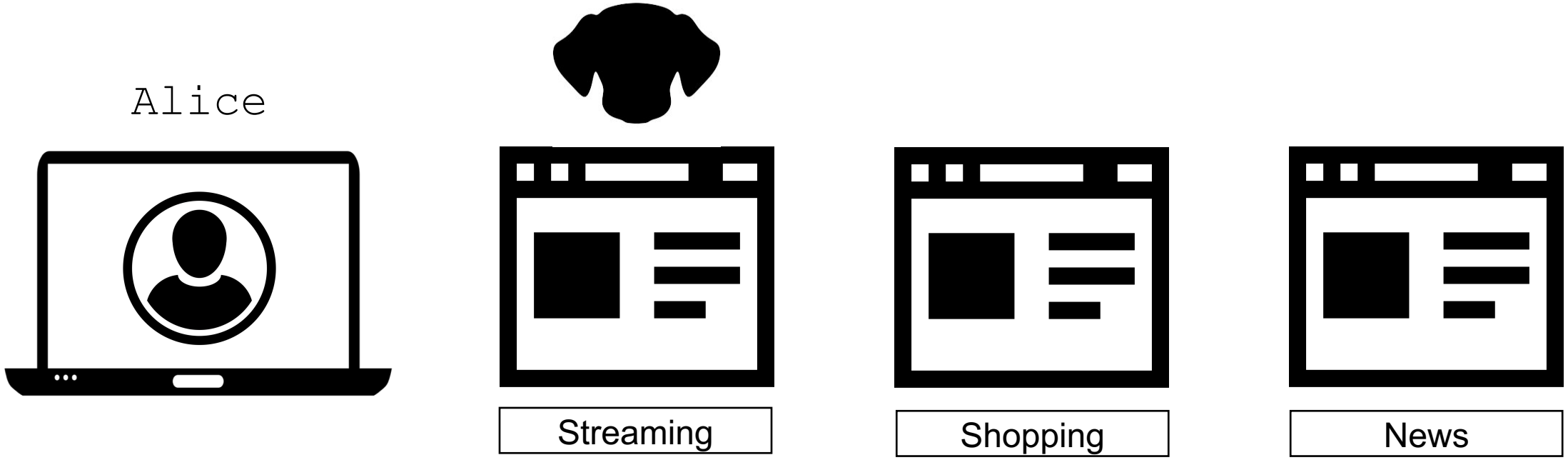


News

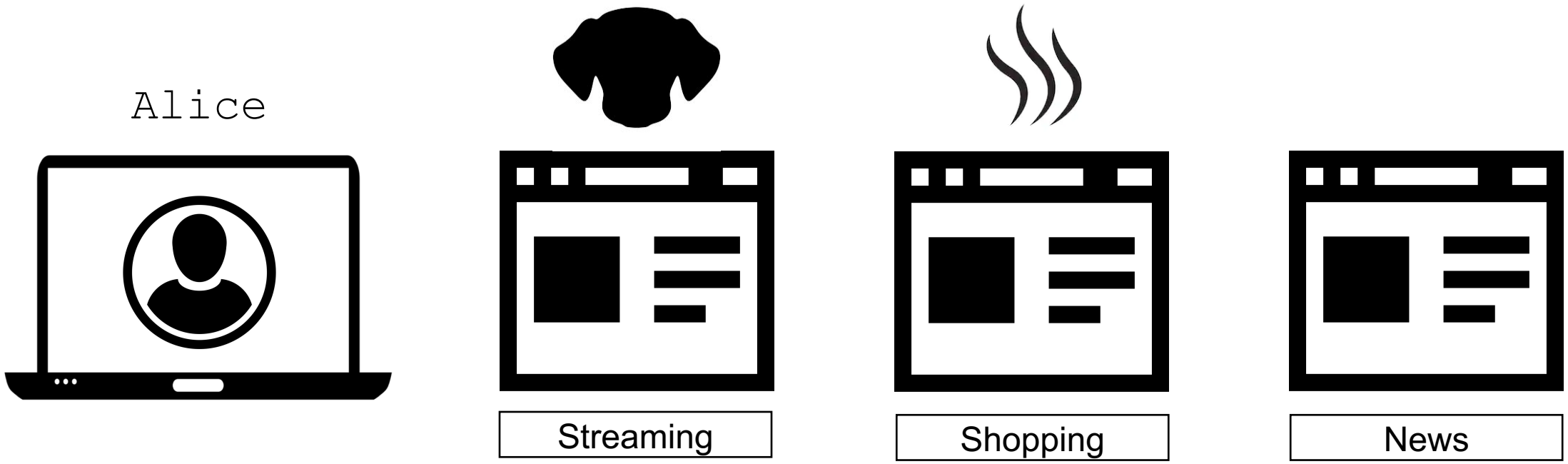
Threat Model



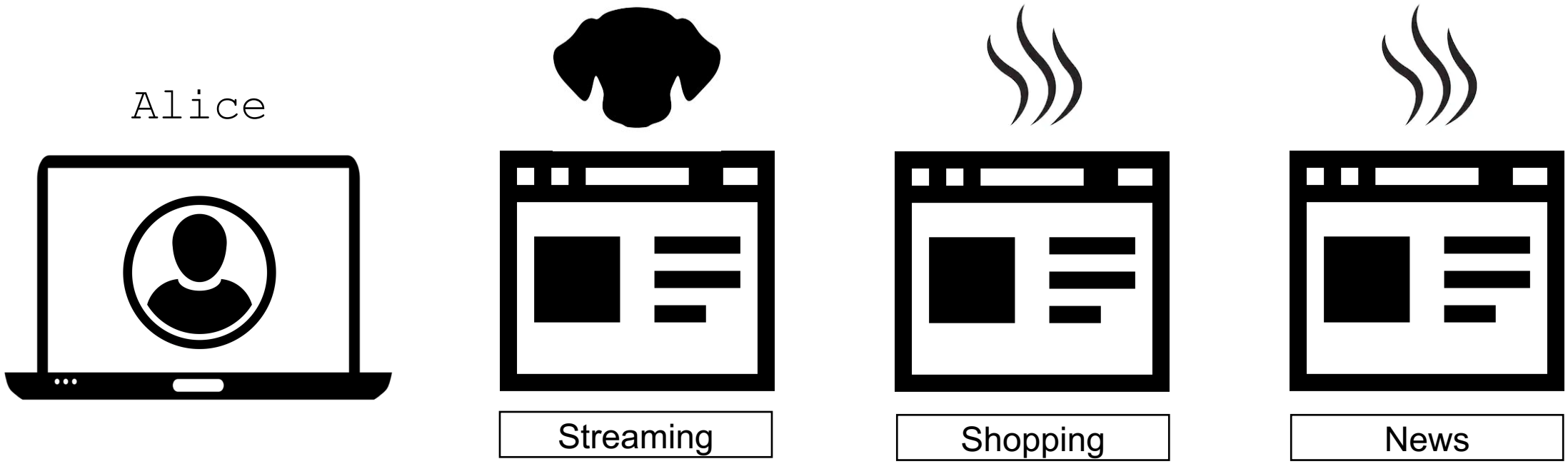
Threat Model



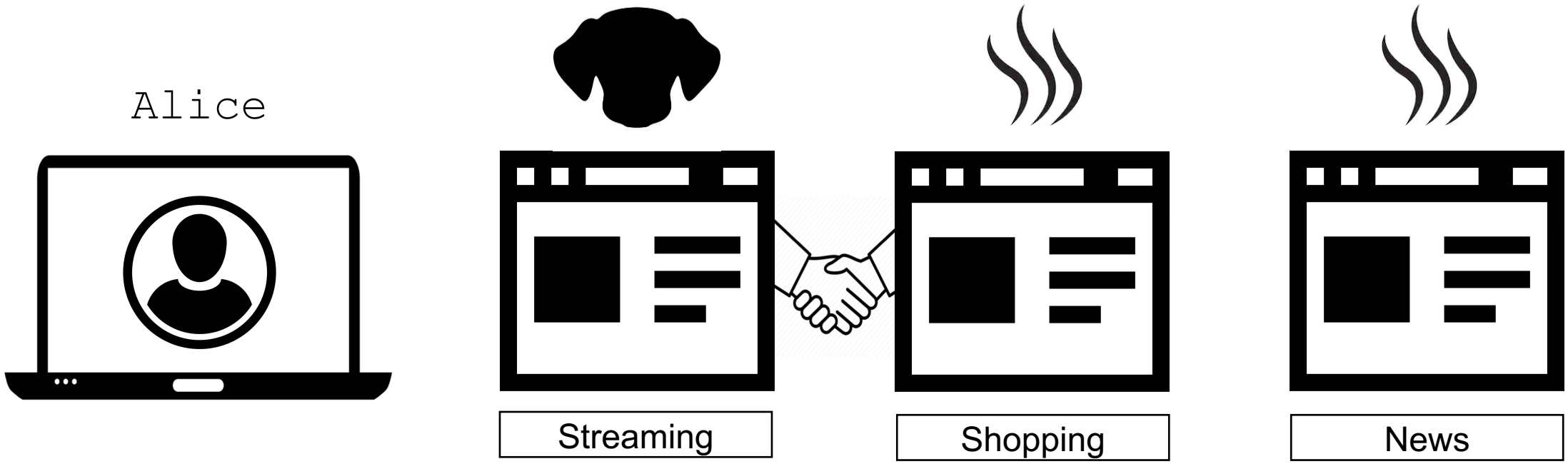
Threat Model



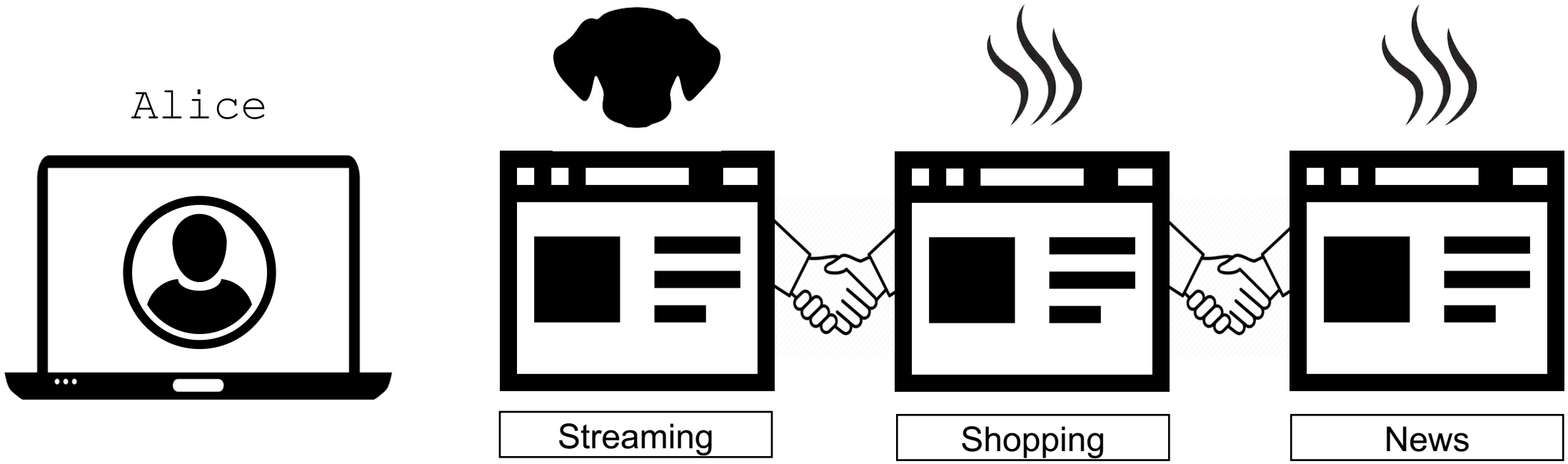
Threat Model



Threat Model



Threat Model



BakingTimer

We present a new timing side-channel attack, that relies on the **presence of first party cookies** set by the target websites to perform history sniffing.

BakingTimer

We present a new timing side-channel attack, that relies on the **presence of first party cookies** set by the target websites to perform history sniffing.

Our system is based on the analysis of how servers **process HTTP requests**, and by using this information, is able to detect both if the user previously visited the website and whether she is currently logged in.

BakingTimer

The main observation behind our approach is that, when the browser sends a cookie along with a request, it is reasonable to assume that the **server will check its value**.

BakingTimer

The main observation behind our approach is that, when the browser sends a cookie along with a request, it is reasonable to assume that the **server will check its value**.

Then, it may use the value to retrieve the associated user session and load additional data from the database, or that it will simply **execute a different path** with respect to a request that does not contain any cookie.

BakingTimer

```
1   <?php
2   $userID = "0bc63ecec05112d03544fde0b5a18c70";
19  ?>
```

BakingTimer

```
1  <?php
2  $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4  if (isset($_COOKIE[["consent"]) {
19 ?>
```

BakingTimer

```
1  <?php
2  $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4  if (isset($_COOKIE[["consent"]) {
19 ?>
```

BakingTimer

```
1  <?php
2  $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4  if (isset($_COOKIE[["consent"]) {
```

```
16 } else {
17     askConsent();
18 }
19 ?>
```

BakingTimer

```
1   <?php
2   $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4   if (isset($_COOKIE[["consent"]) {
```

```
16  } else {
17      askConsent();
18  }
19  ?>
```


CASE **A**

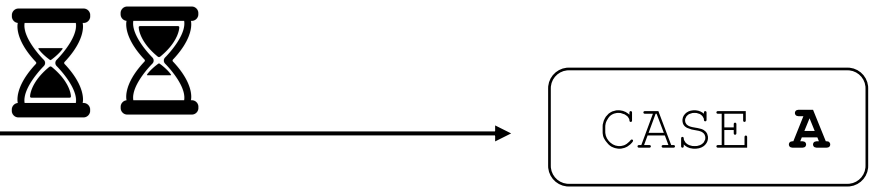
BakingTimer

```
1  <?php
2  $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4  if (isset($_COOKIE[["consent"]) {
16 } else {
17     askConsent();
18 }
19 ?>
```

CASE **A**

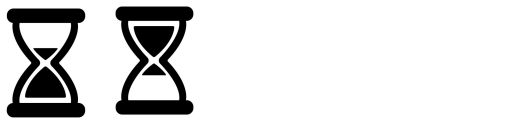
BakingTimer

```
1   <?php
2   $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4   if (isset($_COOKIE[["consent"]) {
16  } else {
17      askConsent();
18  }
19  ?>
```



BakingTimer

```
1   <?php
2   $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4   if (isset($_COOKIE[["consent"]) {
5
6       if (isset($_COOKIE[["userID"]) {
```



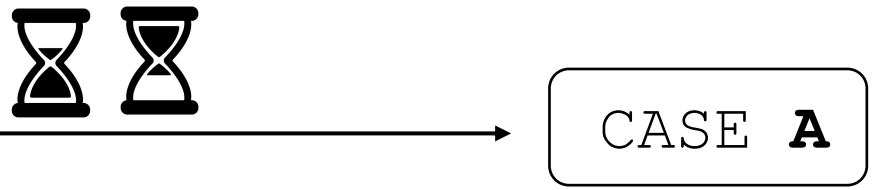
```
16  } else {
17      askConsent();
18  }
19  ?>
```

CASE **A**

BakingTimer

```
1   <?php
2   $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4   if (isset($_COOKIE[["consent"]) {
5
6       if (isset($_COOKIE[["userID"]) {
```

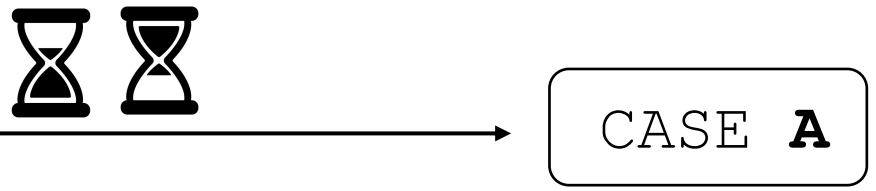
```
16  } else {
17      askConsent();
18  }
19  ?>
```



BakingTimer

```
1   <?php
2   $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4   if (isset($_COOKIE[["consent"]) {
5
6       if (isset($_COOKIE[["userID"]) {
```

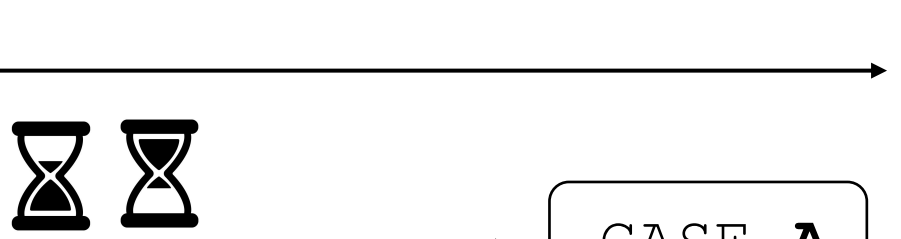
```
12      } else {
13          saveNavigation();
14      }
15
16  } else {
17      askConsent();
18  }
19  ?>
```



BakingTimer

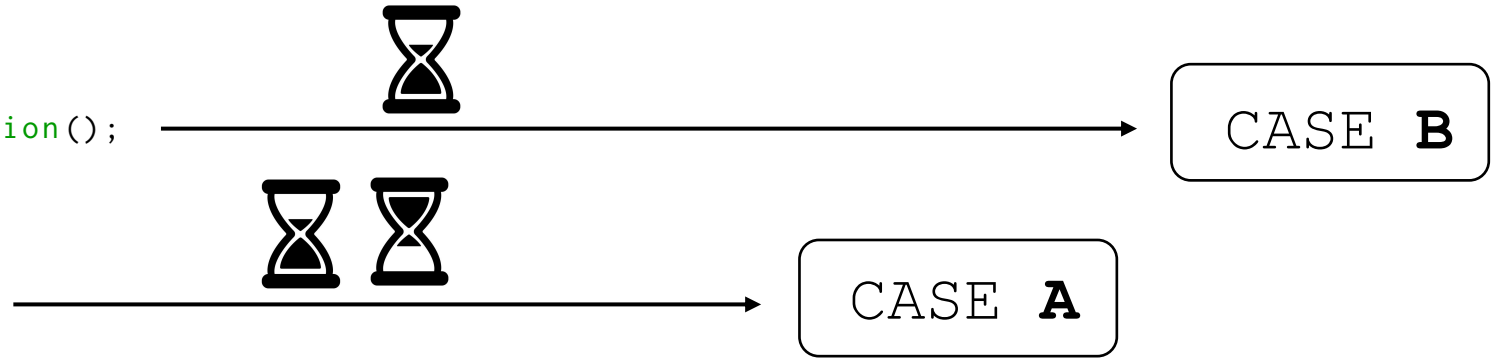
```
1   <?php
2   $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4   if (isset($_COOKIE[["consent"]) {
5
6       if (isset($_COOKIE[["userID"]) {
```

```
12      } else {
13          saveNavigation();
14      }
15
16  } else {
17      askConsent();
18  }
19  ?>
```



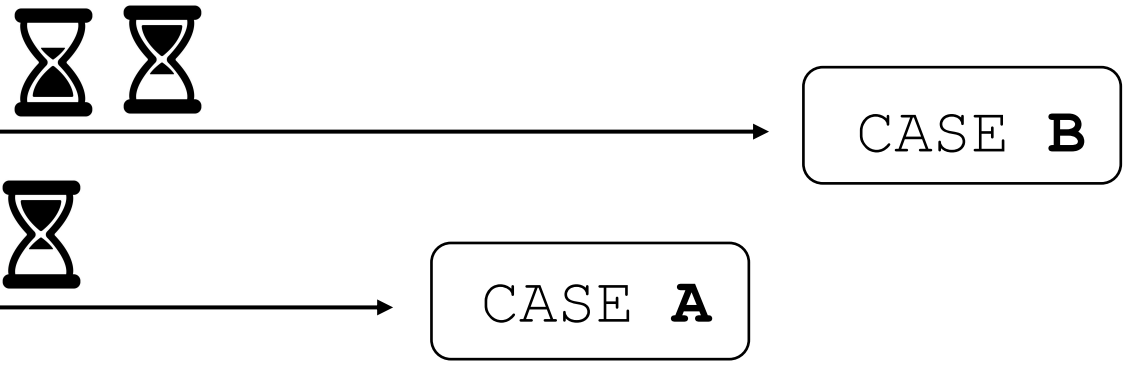
BakingTimer

```
1   <?php
2   $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4   if (isset($_COOKIE[["consent"]) {
5
6       if (isset($_COOKIE[["userID"]) {
12      } else {
13          saveNavigation();
14      }
15
16  } else {
17      askConsent();
18  }
19  ?>
```



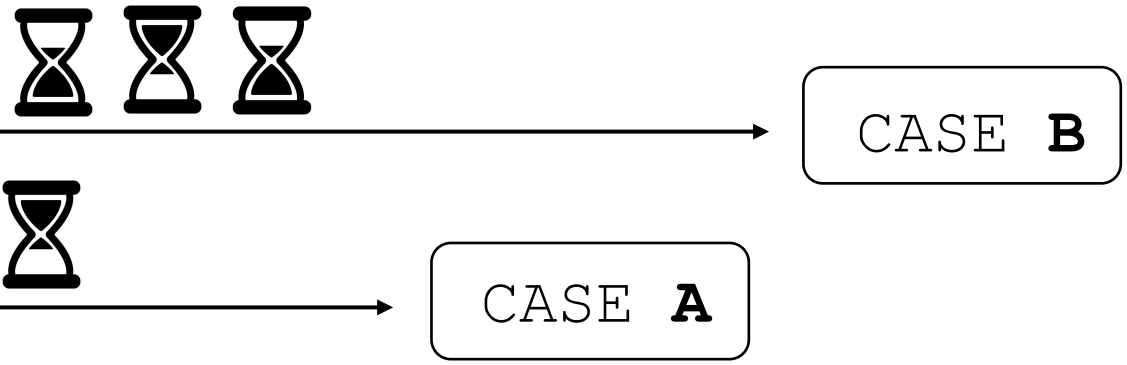
BakingTimer

```
1   <?php
2   $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4   if (isset($_COOKIE[["consent"]) {
5
6       if (isset($_COOKIE[["userID"]) {
12      } else {
13          saveNavigation();
14      }
15
16  } else {
17      askConsent();
18  }
19  ?>
```



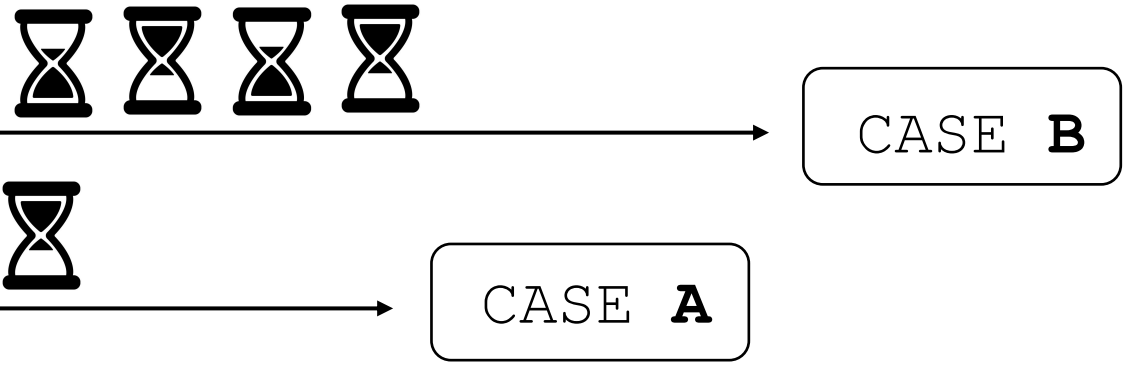
BakingTimer

```
1   <?php
2   $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4   if (isset($_COOKIE[["consent"]) {
5
6       if (isset($_COOKIE[["userID"]) {
12      } else {
13          saveNavigation();
14      }
15
16  } else {
17      askConsent();
18  }
19  ?>
```



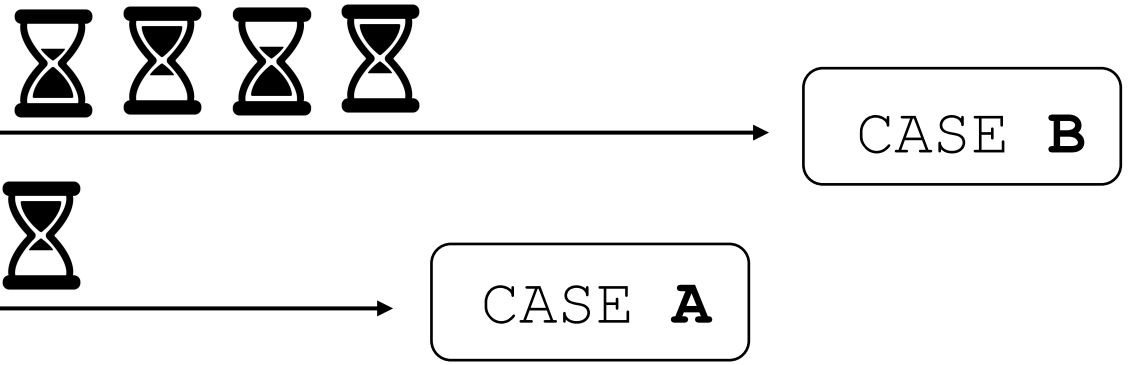
BakingTimer

```
1   <?php
2   $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4   if (isset($_COOKIE[["consent"]) {
5
6       if (isset($_COOKIE[["userID"]) {
12      } else {
13          saveNavigation();
14      }
15
16  } else {
17      askConsent();
18  }
19  ?>
```



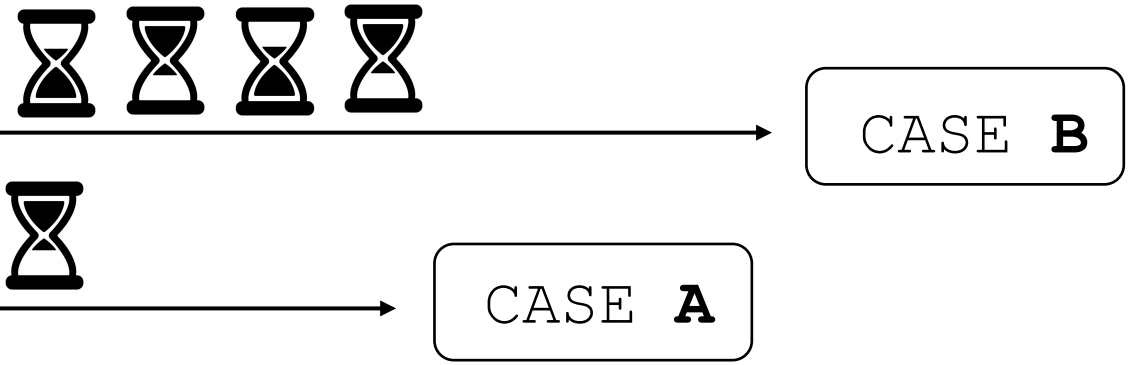
BakingTimer

```php
<?php
$userID = "0bc63ecec05112d03544fde0b5a18c70";

if (isset($_COOKIE[["consent"]) {

    if (isset($_COOKIE[["userID"]) {

        if ($_POST["userID"] == $userID) {
            getUserData();
        }

    } else {
        saveNavigation();
    }

} else {
    askConsent();
}
?>
```



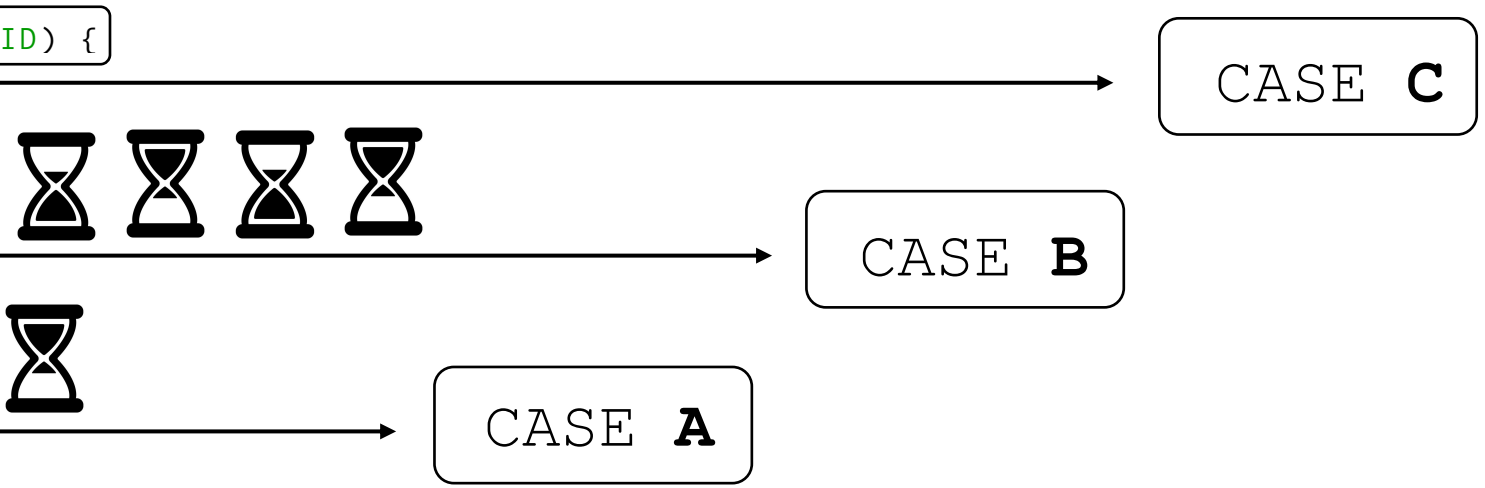
BakingTimer

```
1   <?php
2   $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4   if (isset($_COOKIE[["consent"]) {
5
6       if (isset($_COOKIE[["userID"]) {
7
8           if ($_POST["userID"] == $userID) {
9               getUserData();
10          }
11
12      } else {
13          saveNavigation();
14      }
15
16  } else {
17      askConsent();
18  }
19  ?>
```



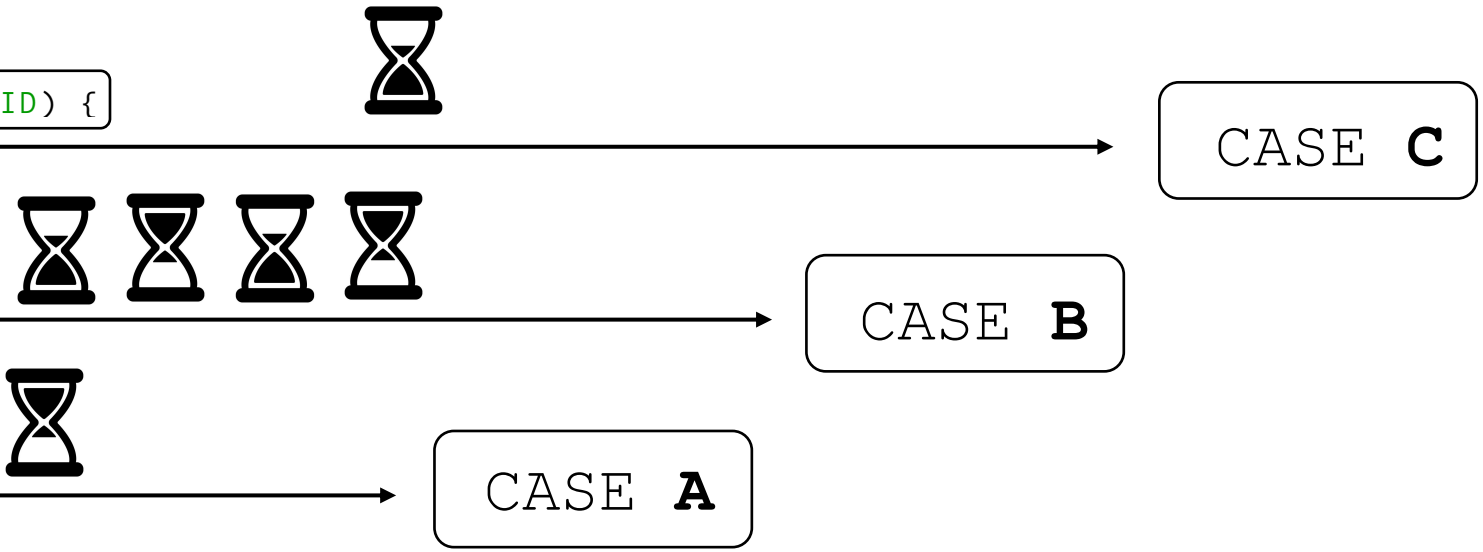
BakingTimer

```
1  <?php
2  $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4  if (isset($_COOKIE[["consent"]) {
5
6      if (isset($_COOKIE[["userID"]) {
7
8          if ($_POST["userID"] == $userID) {
9              getUserData();
10         }
11
12     } else {
13         saveNavigation();
14     }
15
16 } else {
17     askConsent();
18 }
19 ?>
```



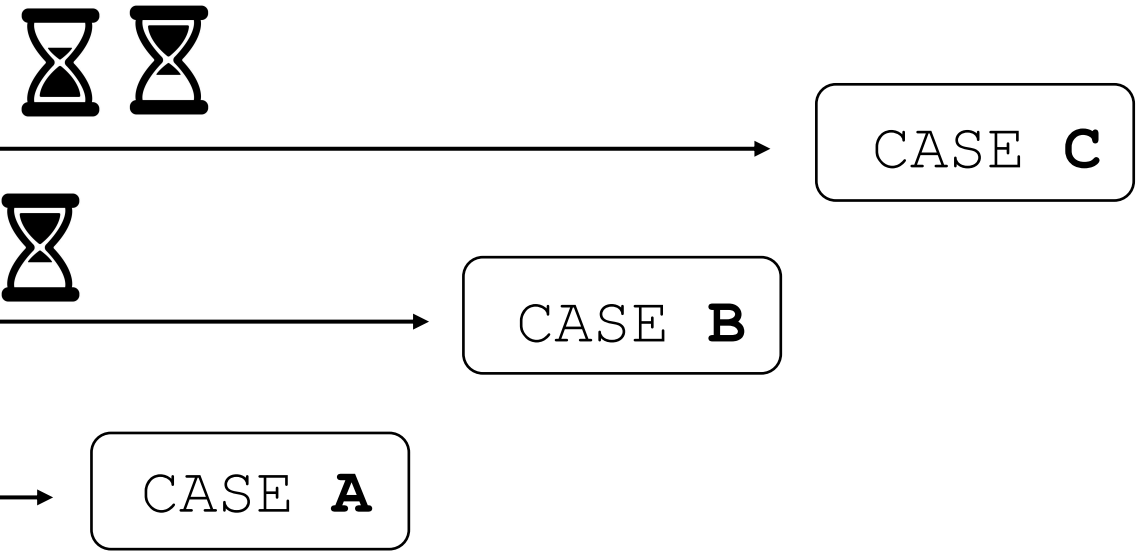
BakingTimer

```php
<?php
$userID = "0bc63ecec05112d03544fde0b5a18c70";

if (isset($_COOKIE[["consent"]) {

    if (isset($_COOKIE[["userID"]) {

        if ($_POST["userID"] == $userID) {
            getUserData();
        }

    } else {
        saveNavigation();
    }

} else {
    askConsent();
}
?>
```



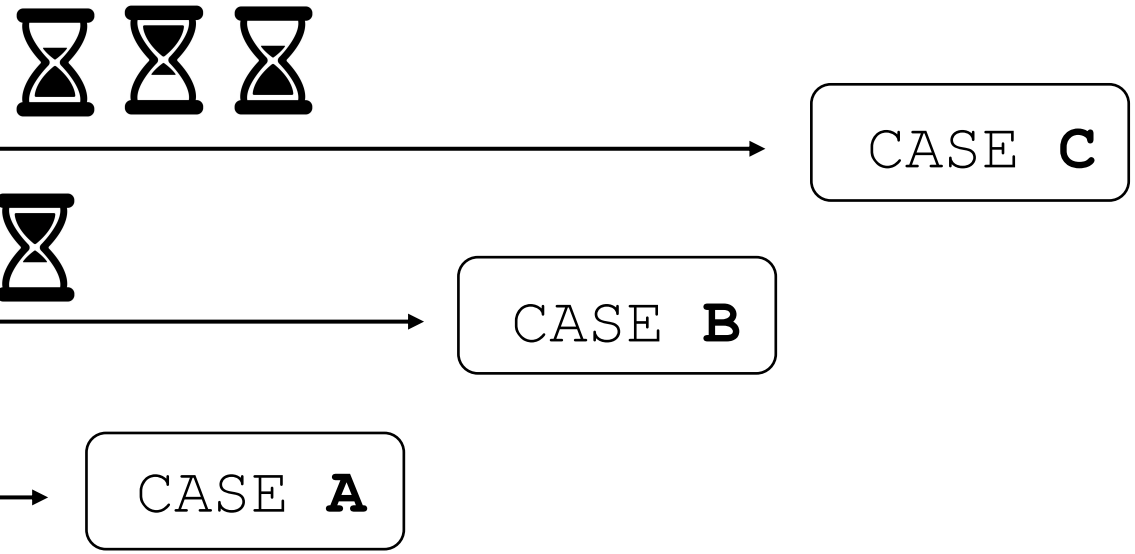
BakingTimer

```
1   <?php
2   $userID = "0bc63ecec05112d03544fde0b5a18c70";
3
4   if (isset($_COOKIE[["consent"]) {
5
6       if (isset($_COOKIE[["userID"]) {
7
8           if ($_POST["userID"] == $userID) {
9               getUserData();
10          }
11
12      } else {
13          saveNavigation();
14      }
15
16  } else {
17      askConsent();
18  }
19  ?>
```



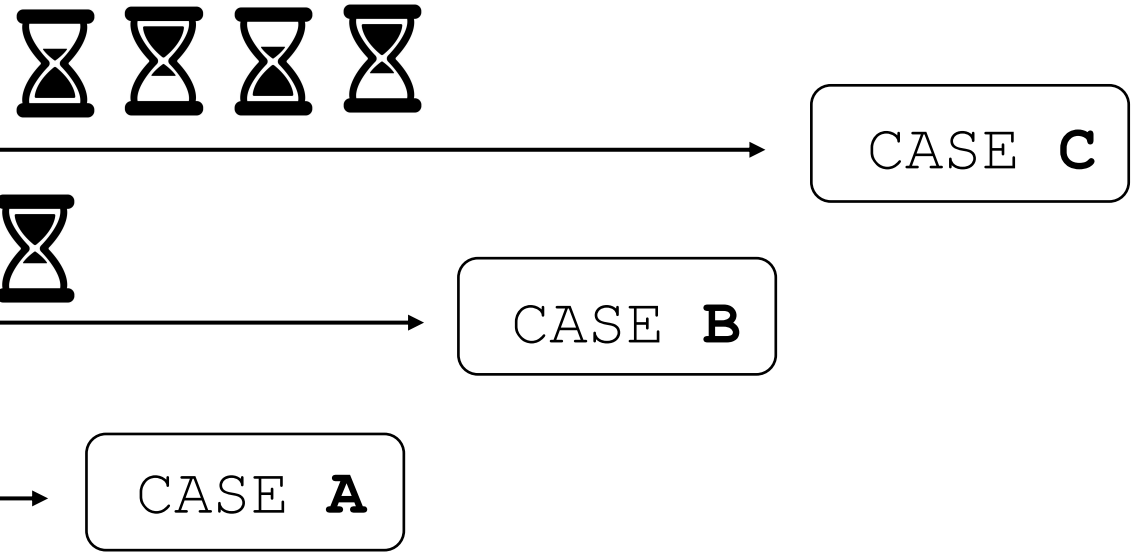
BakingTimer

```php
<?php
$userID = "0bc63ecec05112d03544fde0b5a18c70";

if (isset($_COOKIE[["consent"]) {

    if (isset($_COOKIE[["userID"]) {

        if ($_POST["userID"] == $userID) {
            getUserData();
        }

    } else {
        saveNavigation();
    }

} else {
    askConsent();
}
?>
```



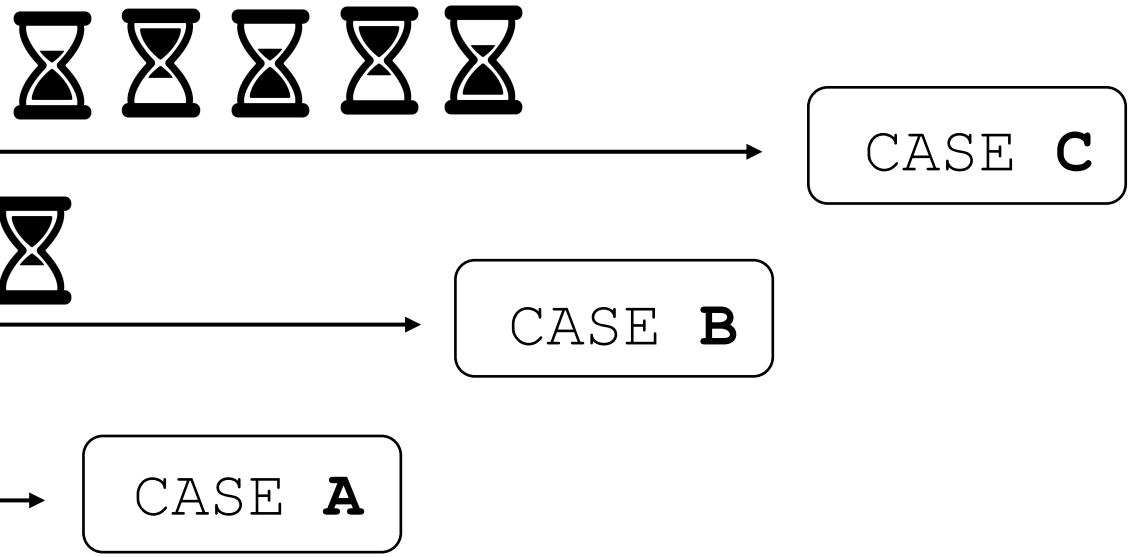
BakingTimer

```php
<?php
$userID = "0bc63ecec05112d03544fde0b5a18c70";

if (isset($_COOKIE[["consent"]) {

    if (isset($_COOKIE[["userID"]) {

        if ($_POST["userID"] == $userID) {
            getUserData();
        }

    } else {
        saveNavigation();
    }

} else {
    askConsent();
}
?>
```



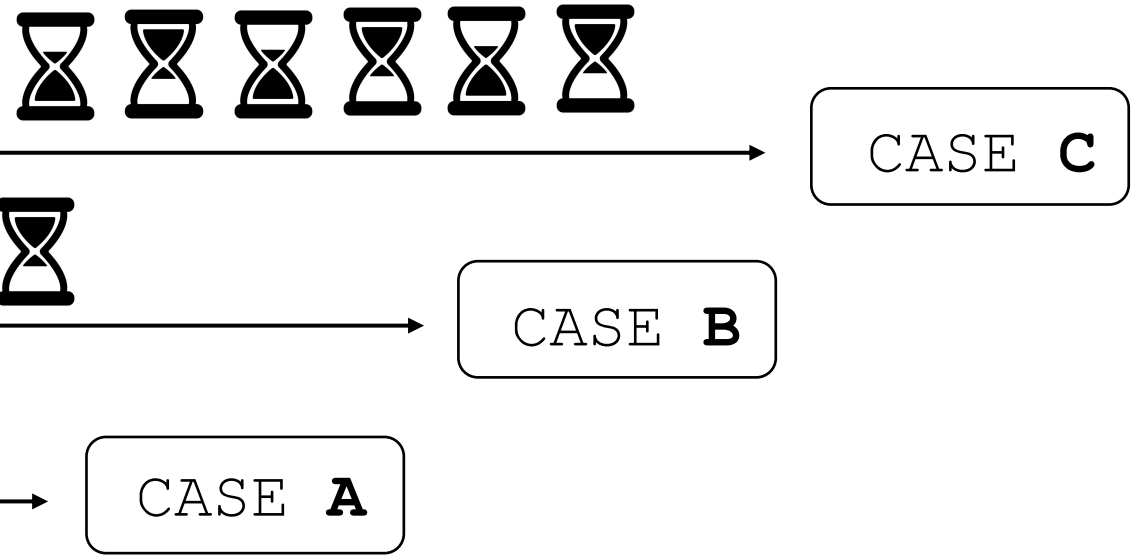
BakingTimer

```php
<?php
$userID = "0bc63ecec05112d03544fde0b5a18c70";

if (isset($_COOKIE[["consent"]) {

    if (isset($_COOKIE[["userID"]) {

        if ($_POST["userID"] == $userID) {
            getUserData();
        }

    } else {
        saveNavigation();
    }

} else {
    askConsent();
}
?>
```



BakingTimer

```php
<?php
$userID = "0bc63ecec05112d03544fde0b5a18c70";

if (isset($_COOKIE[["consent"]) {

    if (isset($_COOKIE[["userID"]) {

        if ($_POST["userID"] == $userID) {
            getUserData();
        }

    } else {
        saveNavigation();
    }

} else {
    askConsent();
}
?>
```



BakingTimer

REQUEST 1 + REQUEST 2

BakingTimer

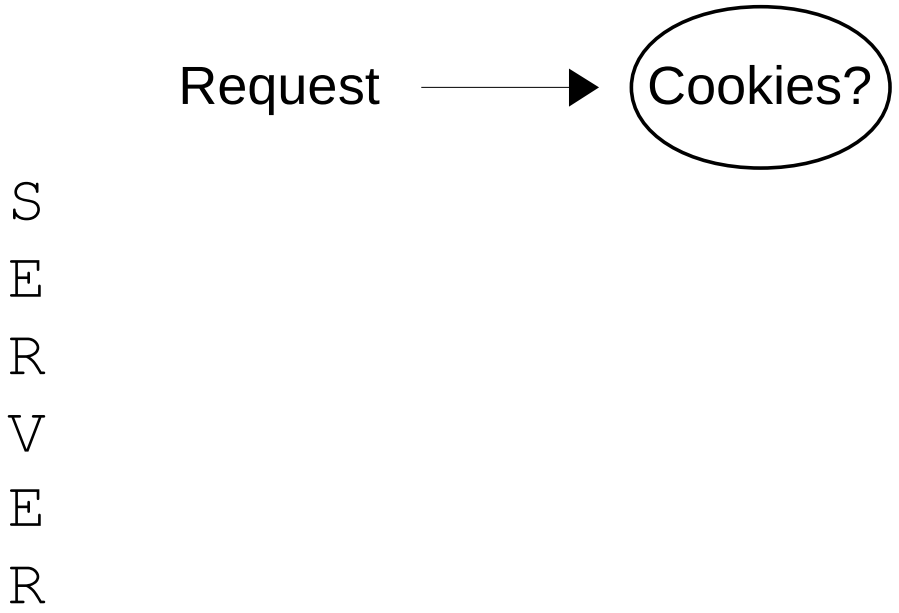
REQUEST 1

BakingTimer

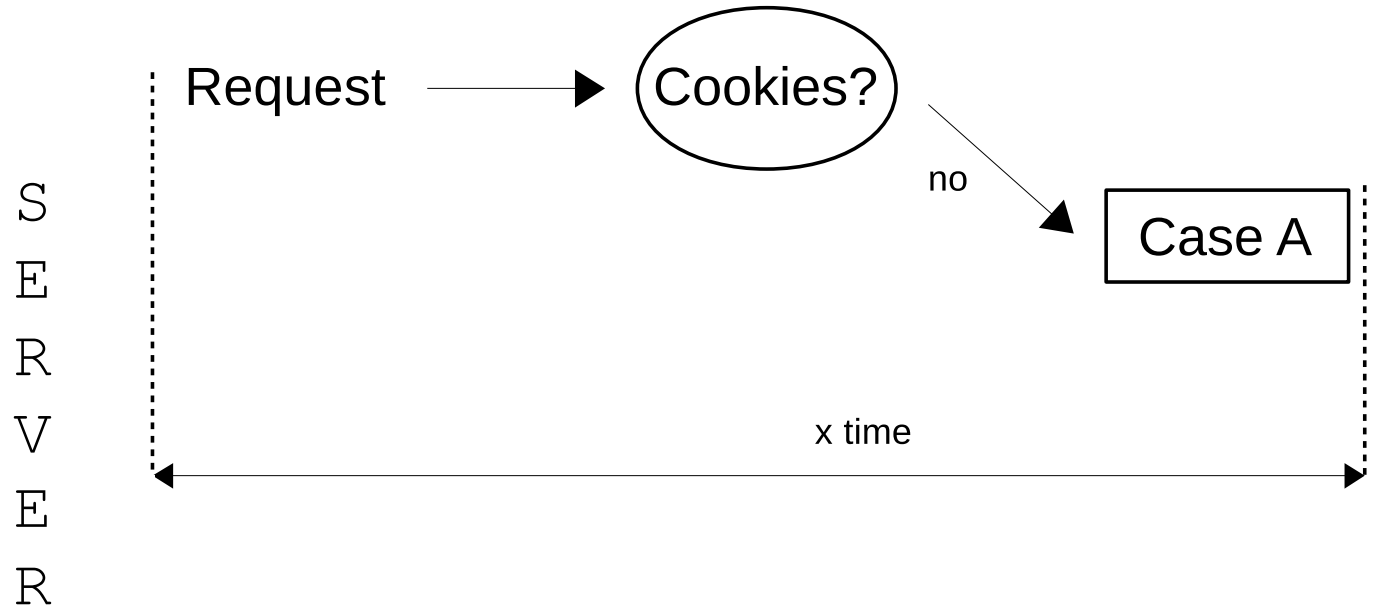
REQUEST 1

```
xmlHttpRequest.withCredentials = FALSE;
```

BakingTimer



BakingTimer



BakingTimer

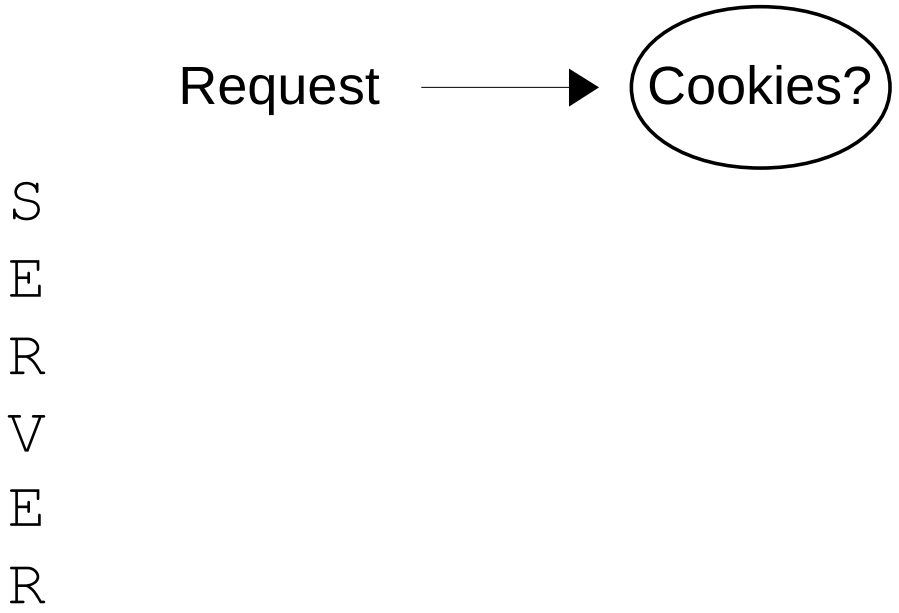
REQUEST 2

BakingTimer

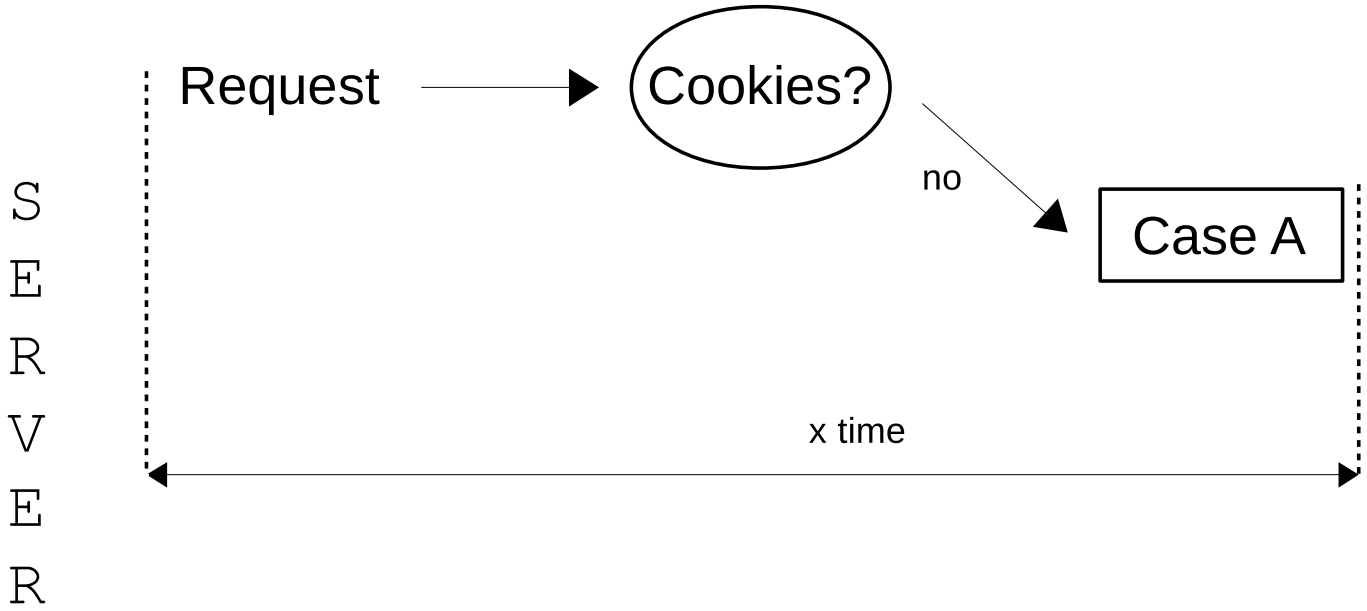
REQUEST 2

```
xmlHttpRequest.withCredentials = TRUE;
```

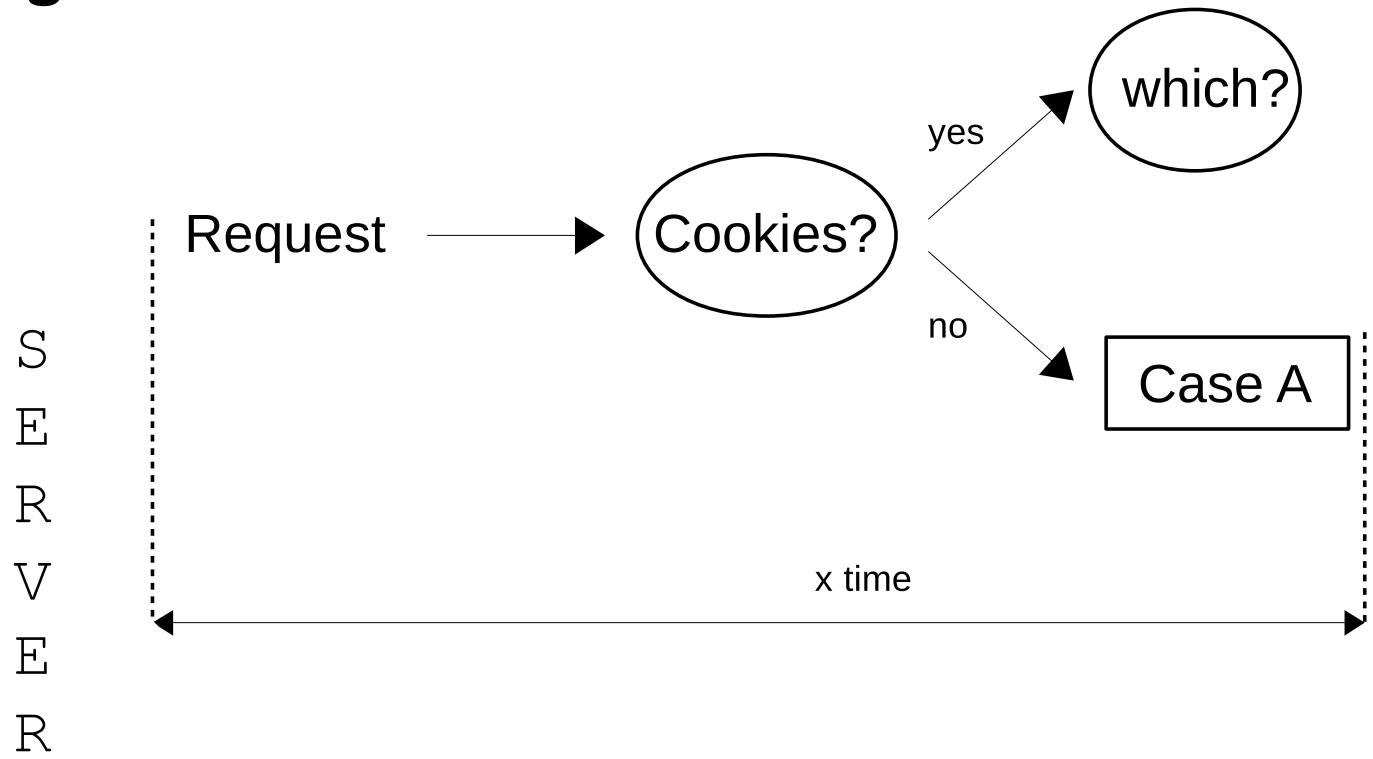
BakingTimer



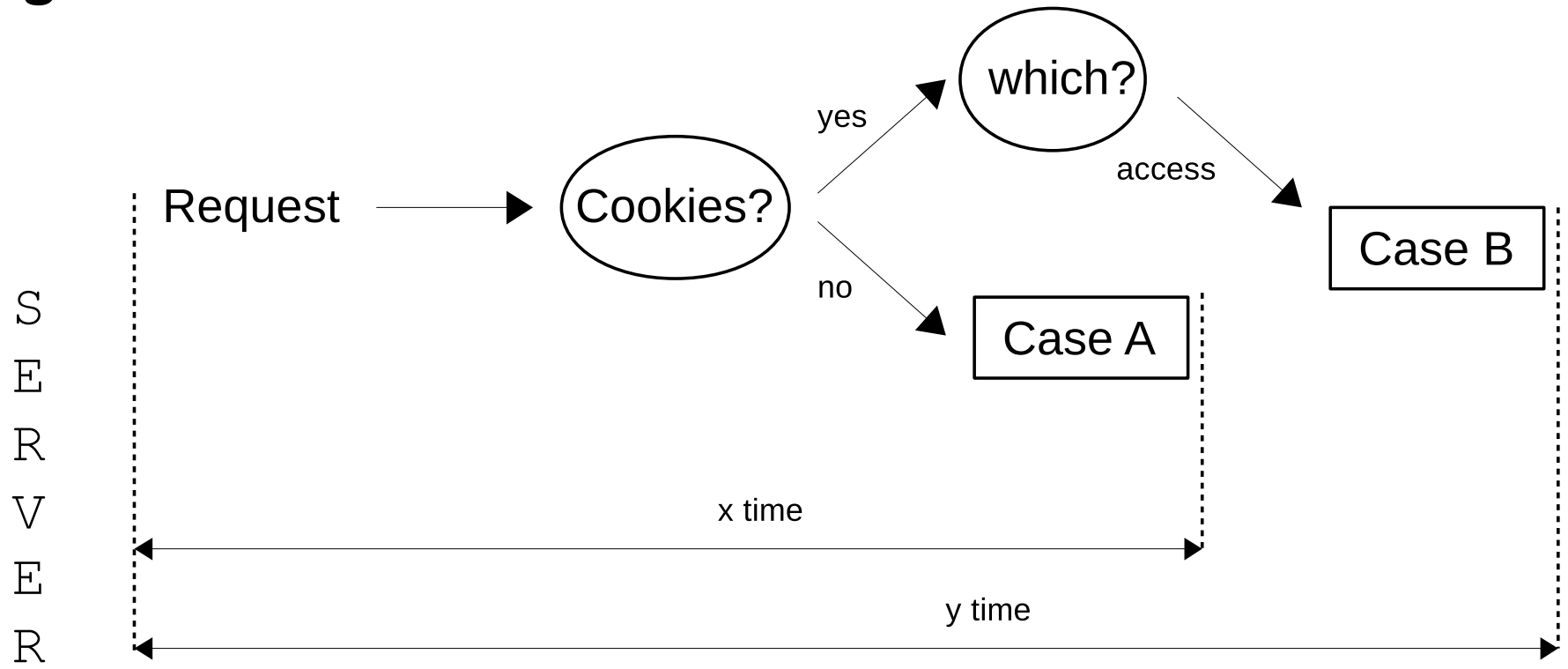
BakingTimer



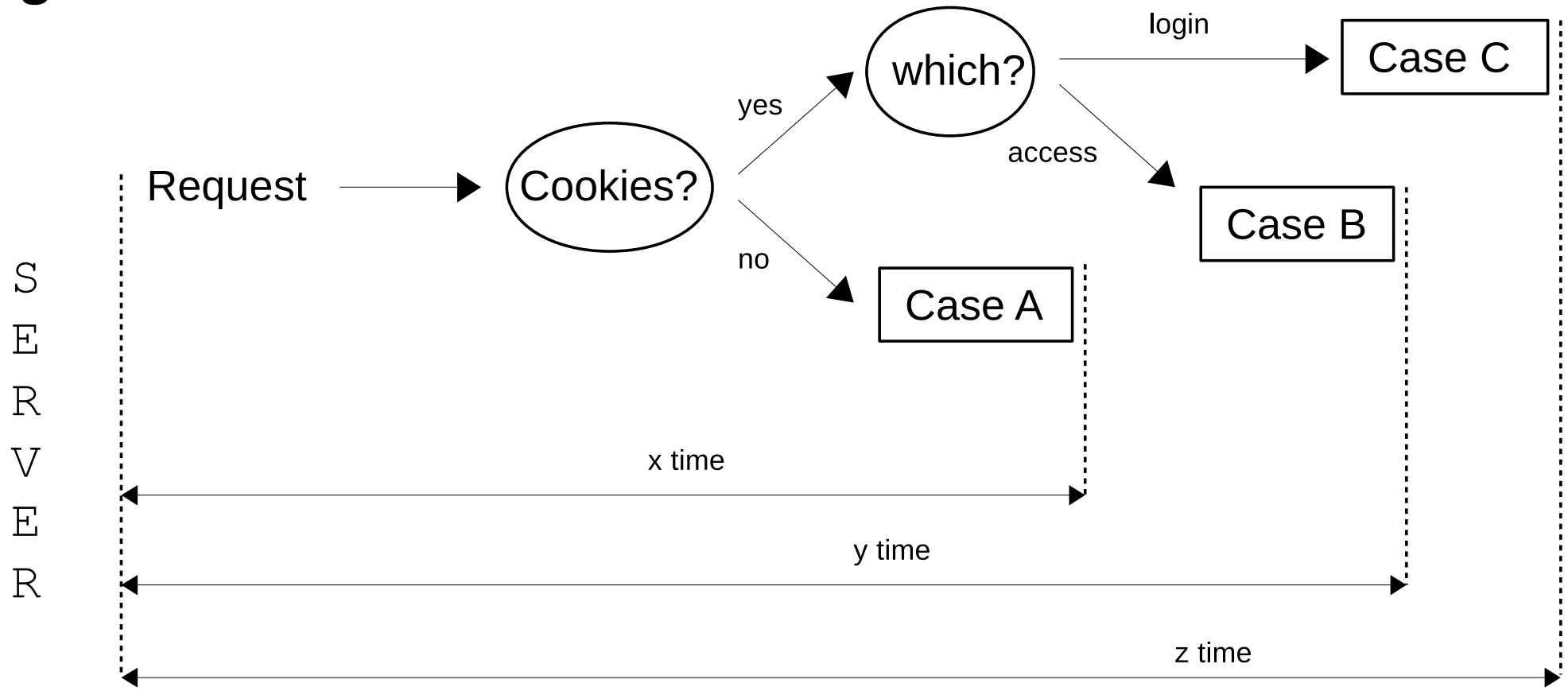
BakingTimer



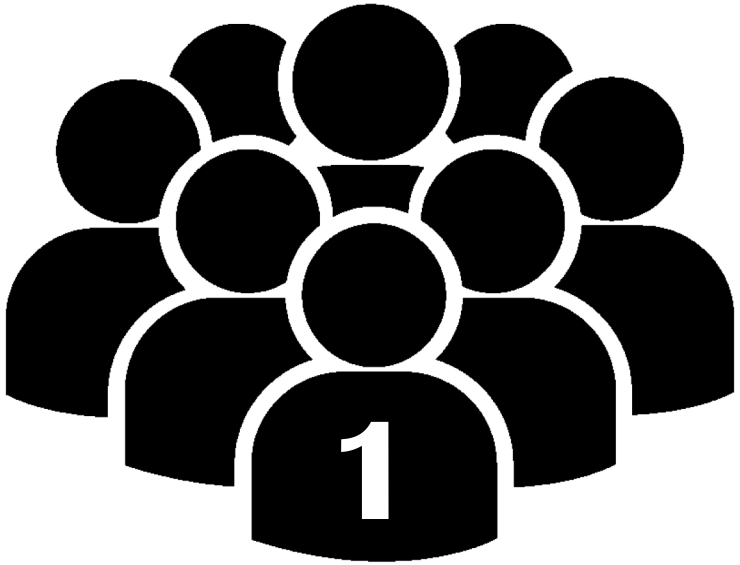
BakingTimer

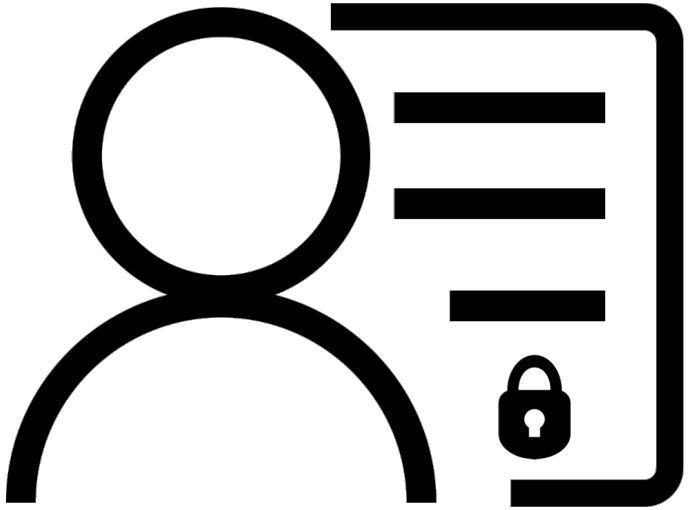


BakingTimer



Experiment Dataset





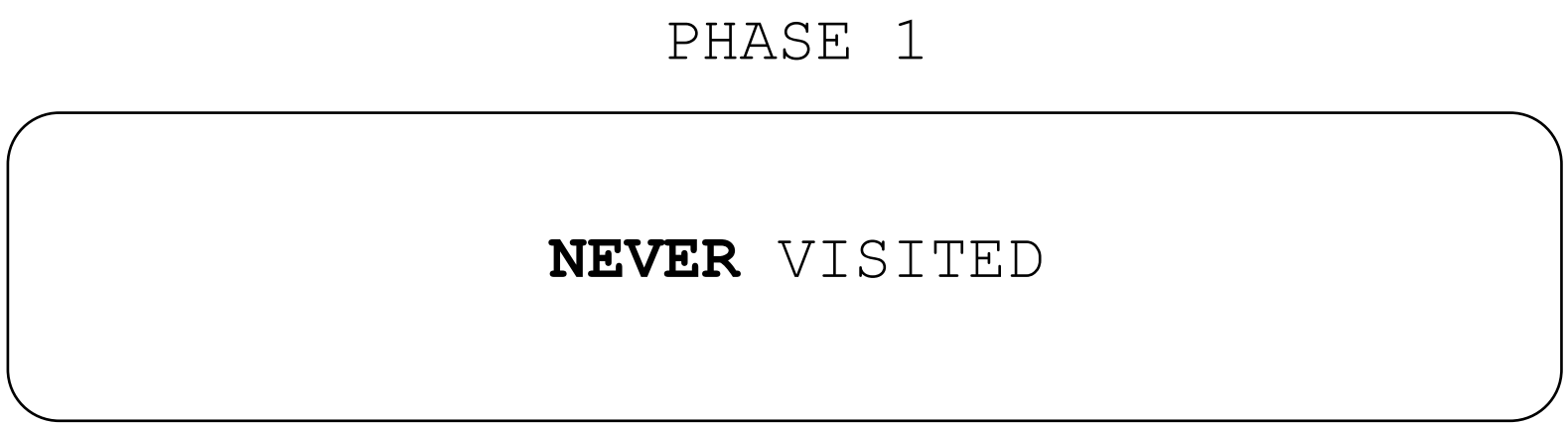
Access Detection

PHASE 1 + PHASE 2

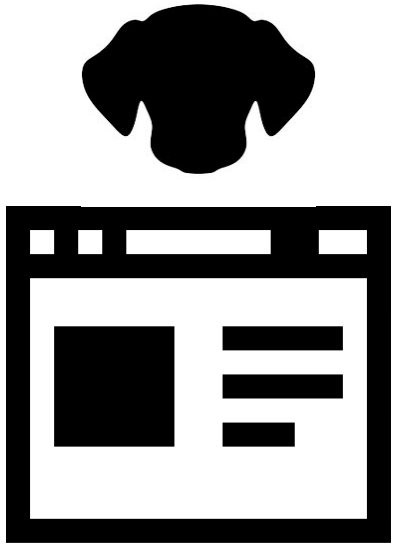
Access Detection

PHASE 1

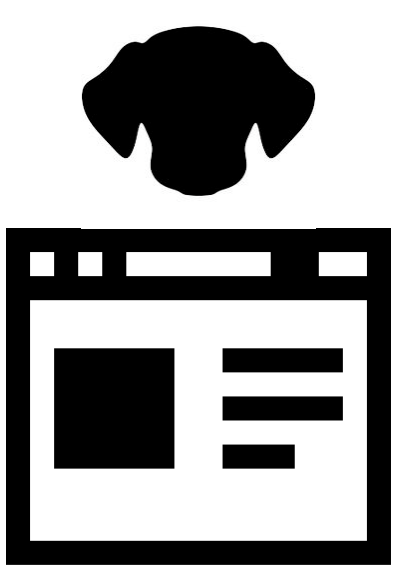
Access Detection

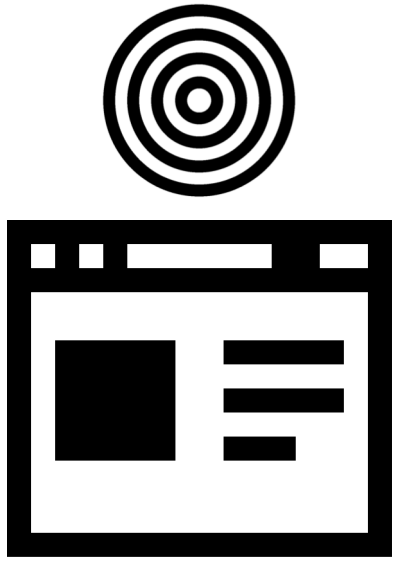


Access Detection

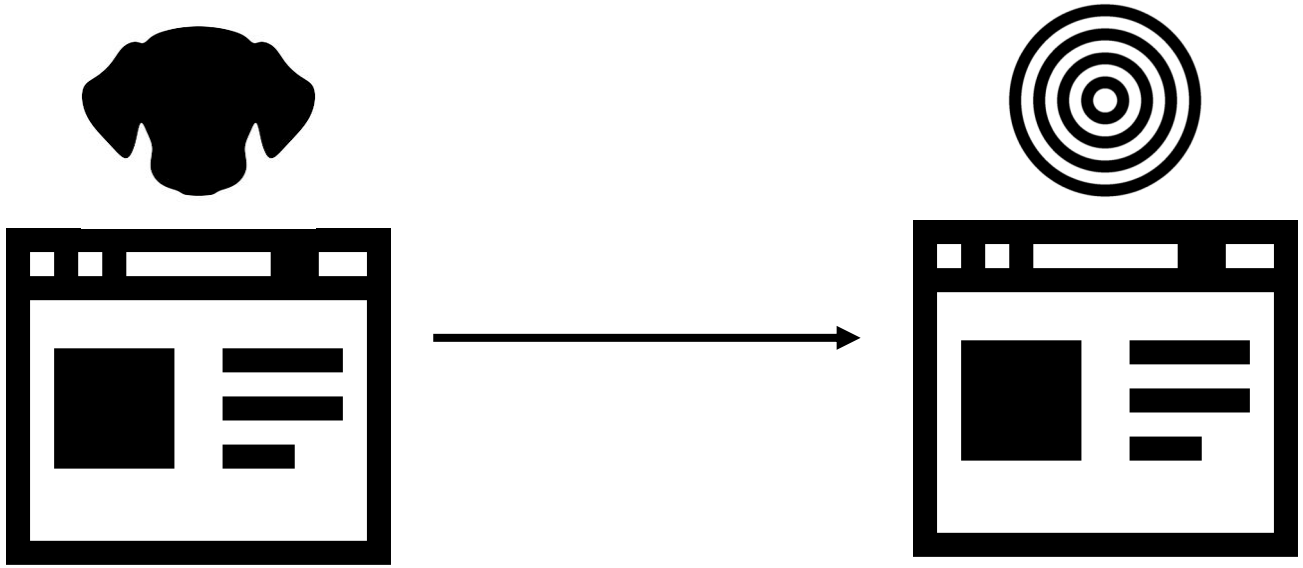


Access Detection

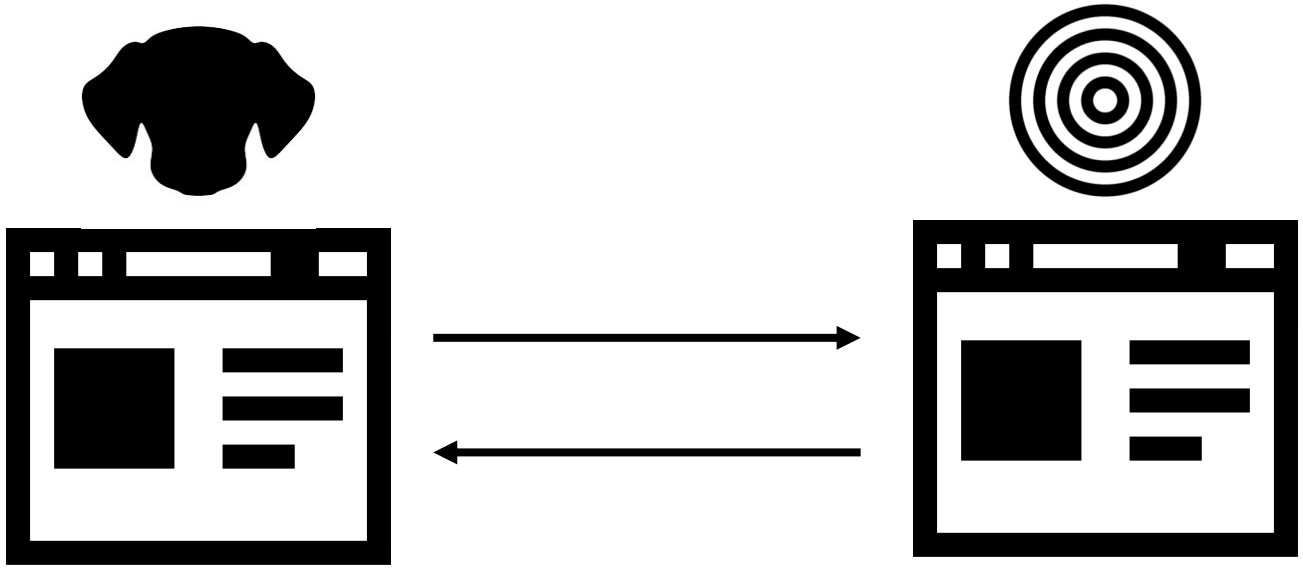




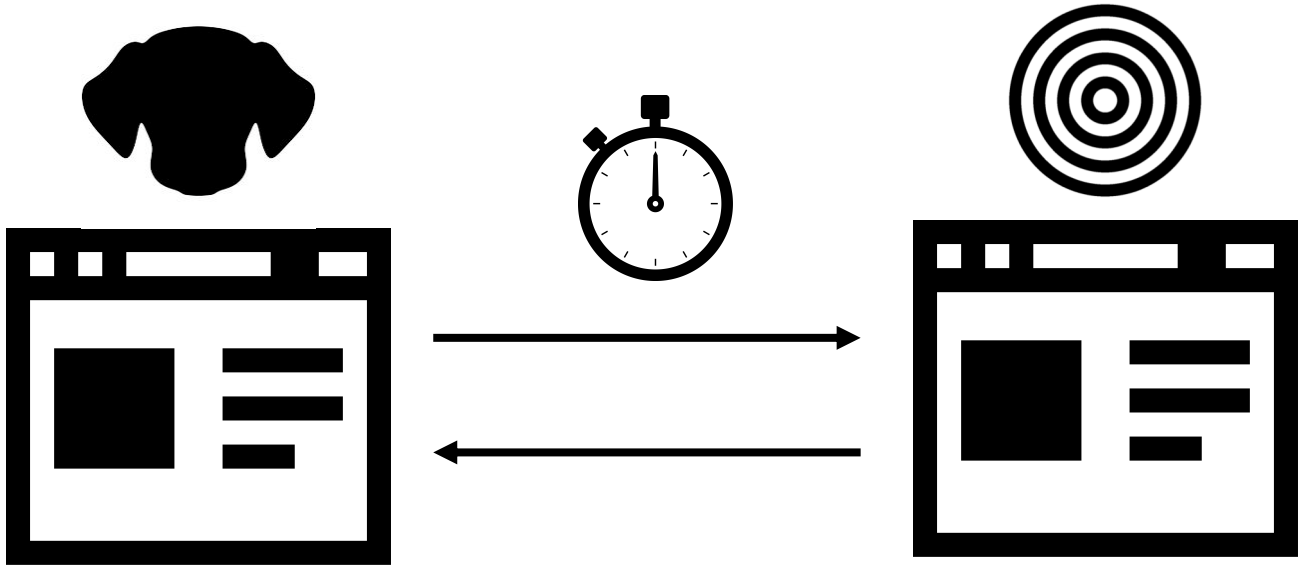
Access Detection



Access Detection



Access Detection



Access Detection

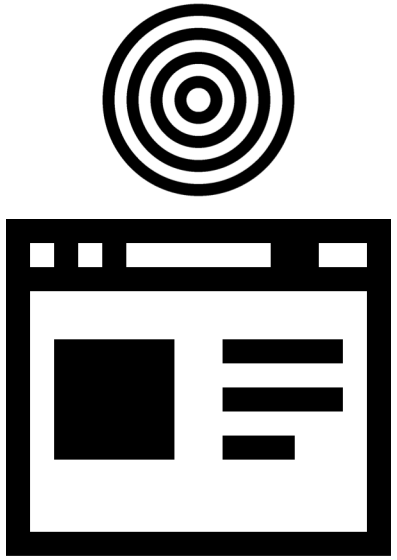
PHASE 2

Access Detection

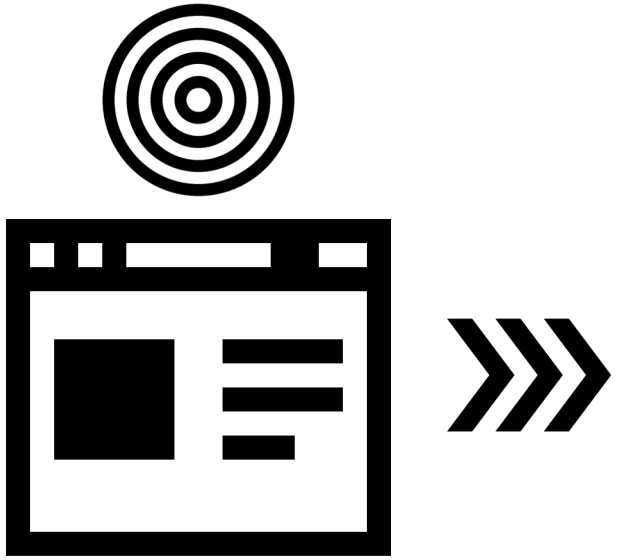
PHASE 2

PREVIOUSLY VISITED

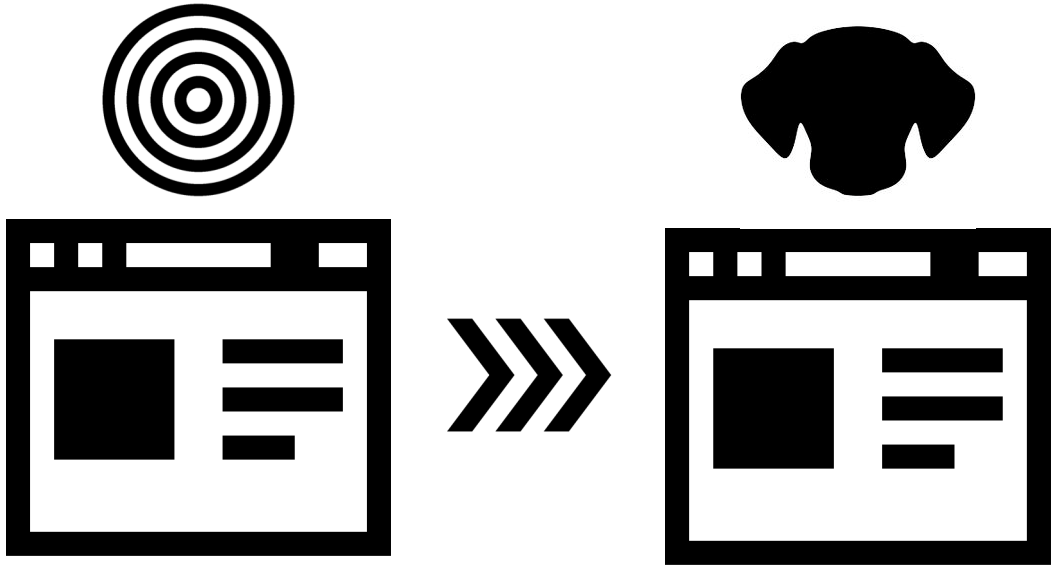
Access Detection



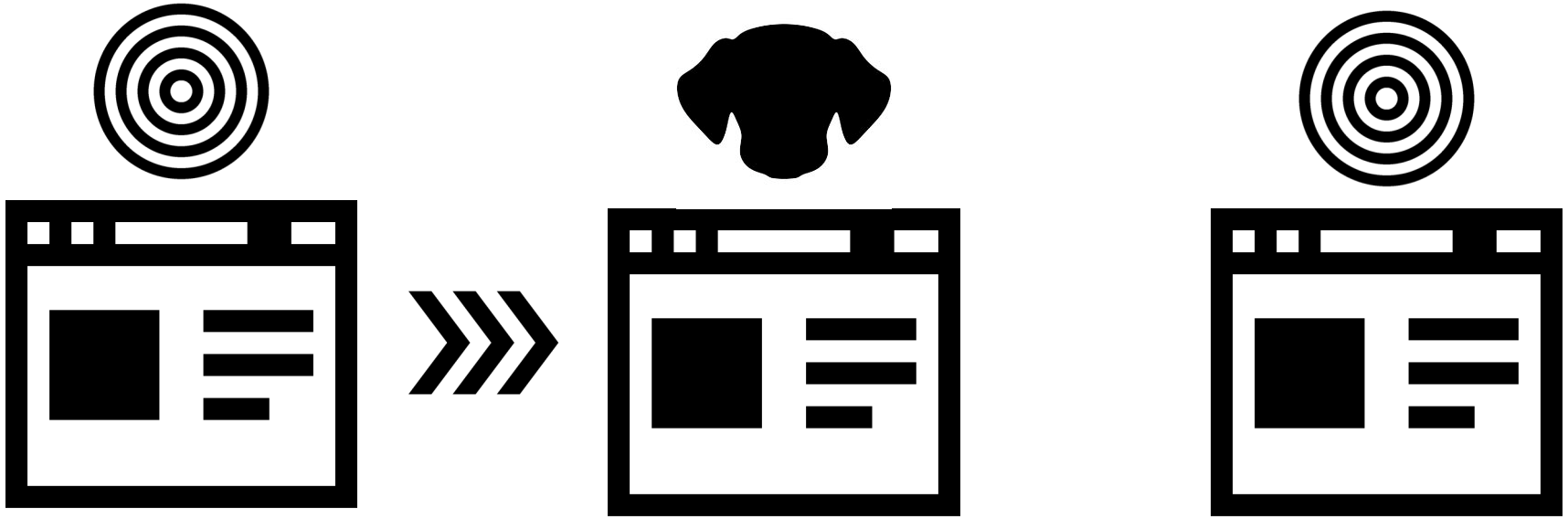
Access Detection



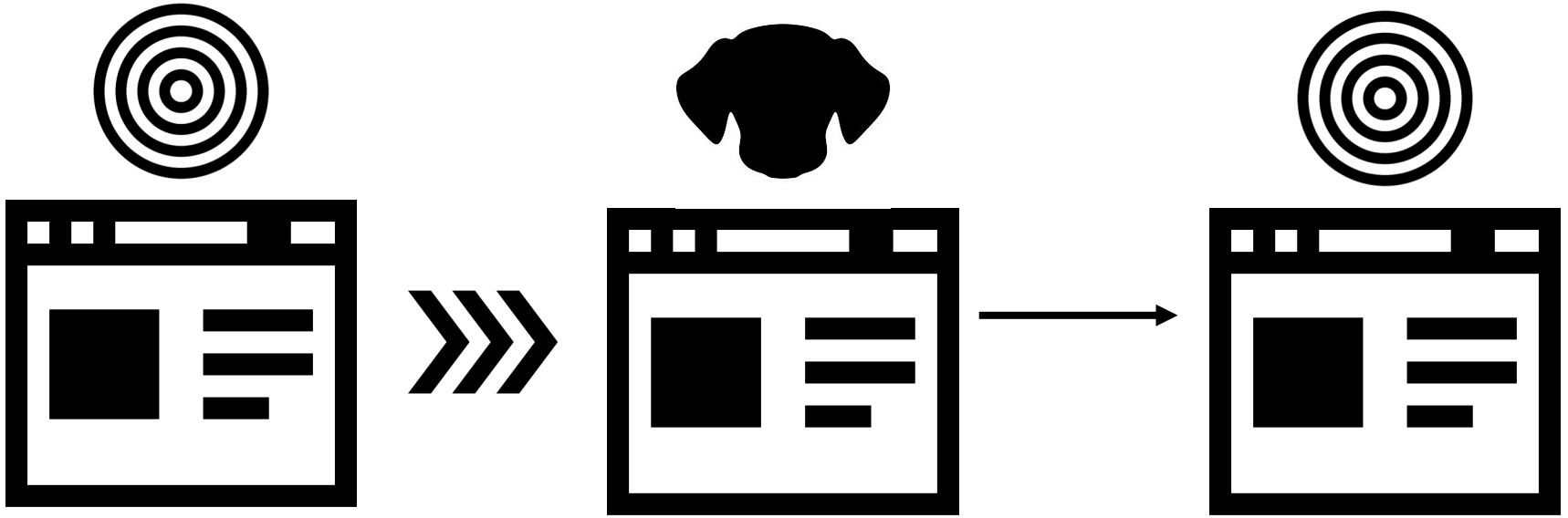
Access Detection



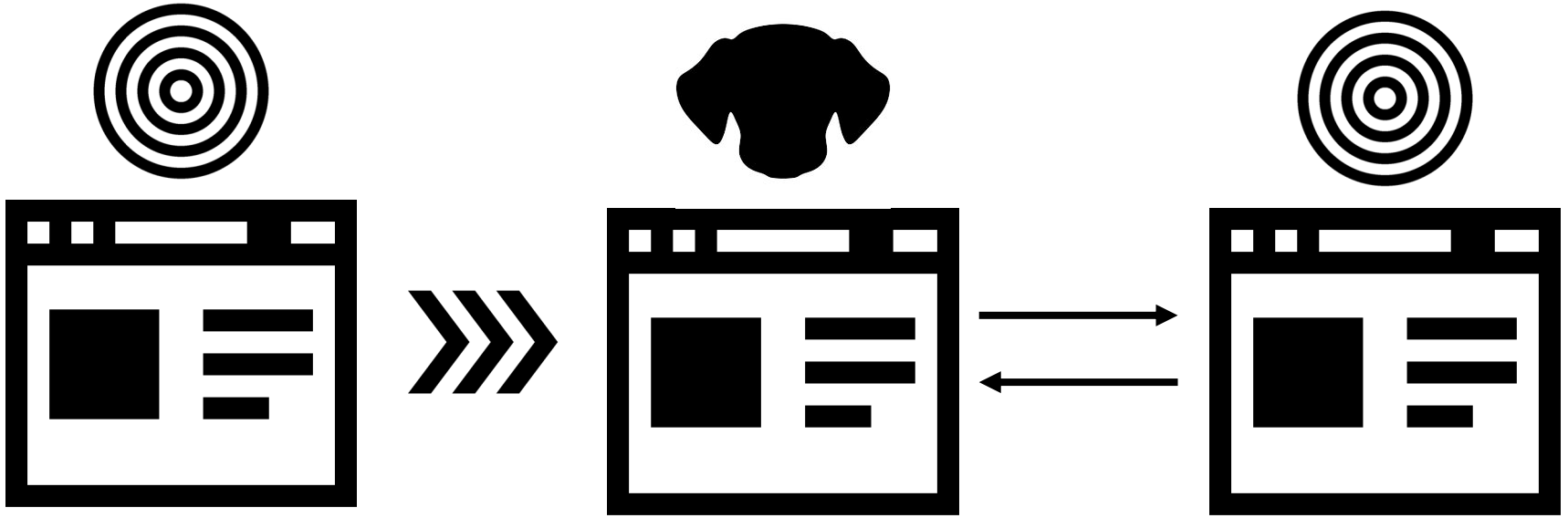
Access Detection



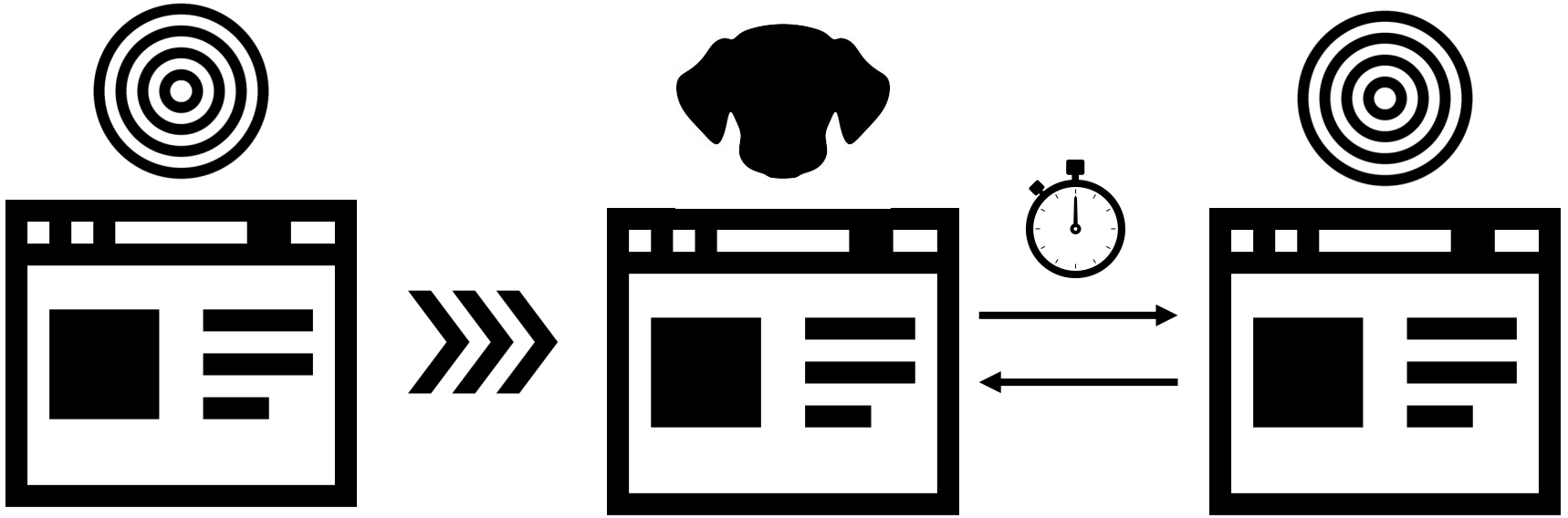
Access Detection



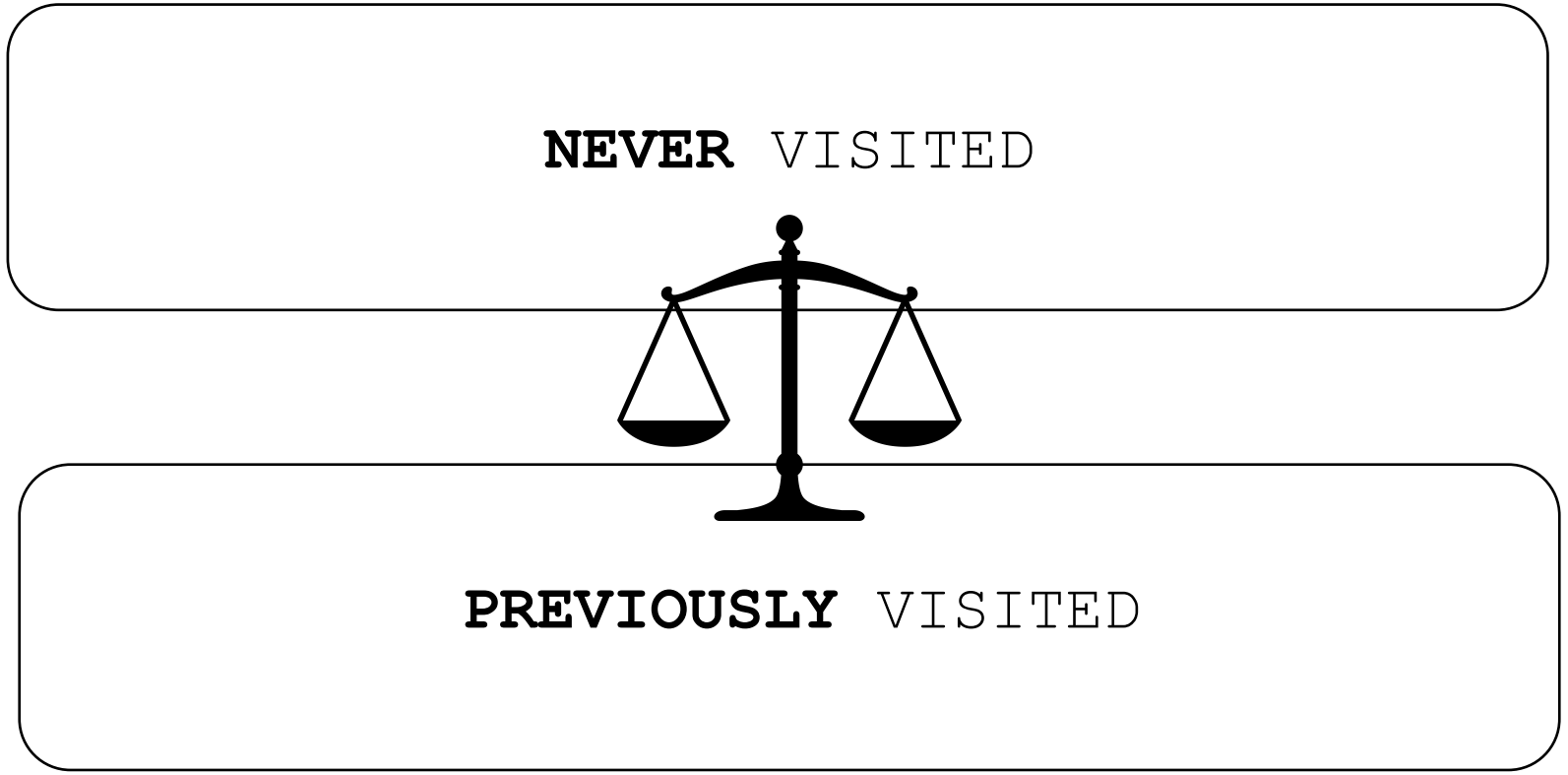
Access Detection



Access Detection



Access Detection



Access Detection

More than **half of the websites analyzed** were vulnerable to our attack. More concretely, around 70% with private personal information, and around 40% of highly accesible.

Access Detection

More than **half of the websites analyzed** were vulnerable to our attack. More concretely, around 70% with private personal information, and around 40% of highly accesible.

We compared the mean and standard deviation of the number of cookies, and results show that highly accessed websites have a higher number of cookies. This hints that **slower servers or less optimized code** seem the resposible of the difference.

Login Detection

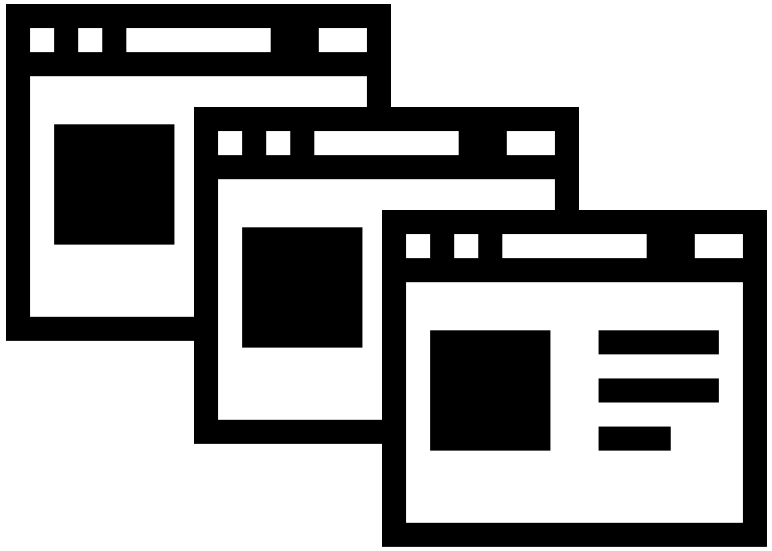
We can also check if the **user is logged in**. In our dataset, we found highly accesible website such as World of Warcraft (WoW) or Gucci, and websites related to private personal information such as LGBTchat or Dynamic Catholic.

Login Detection

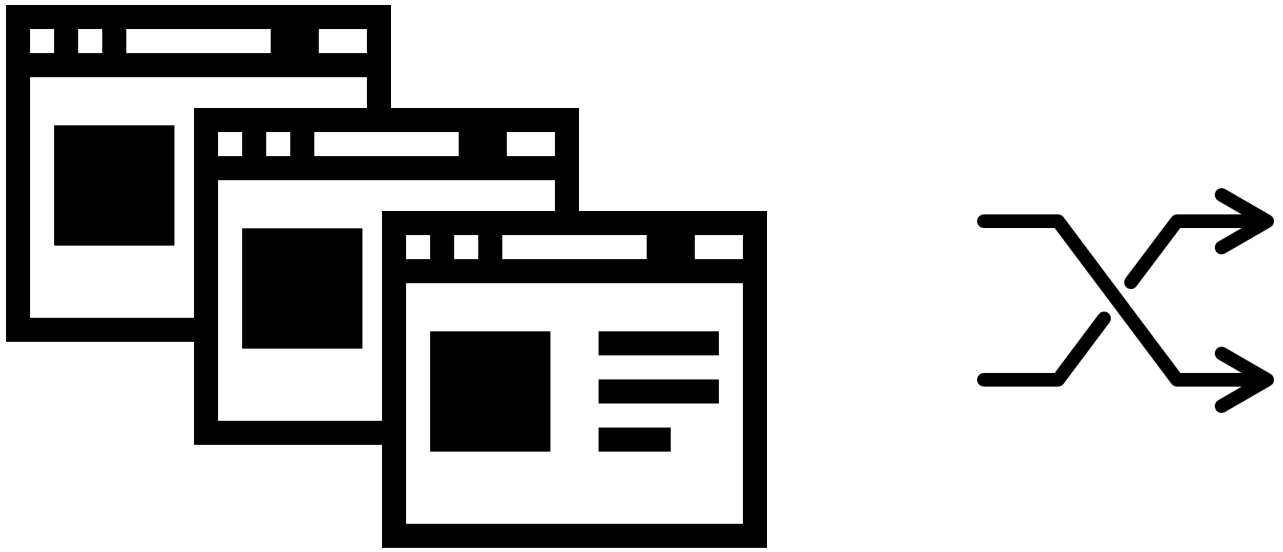
We can also check if the **user is logged in**. In our dataset, we found highly accesible website such as World of Warcraft (WoW) or Gucci, and websites related to private personal information such as LGBTchat or Dynamic Catholic.

Curiously, some websites do not properly delete all cookies related to the login, what we call **persistant login**. In this cases, it is possible to detect a previous logged-in state even if not logged at that moment (e.g., Microsoft/MSN and Openload).

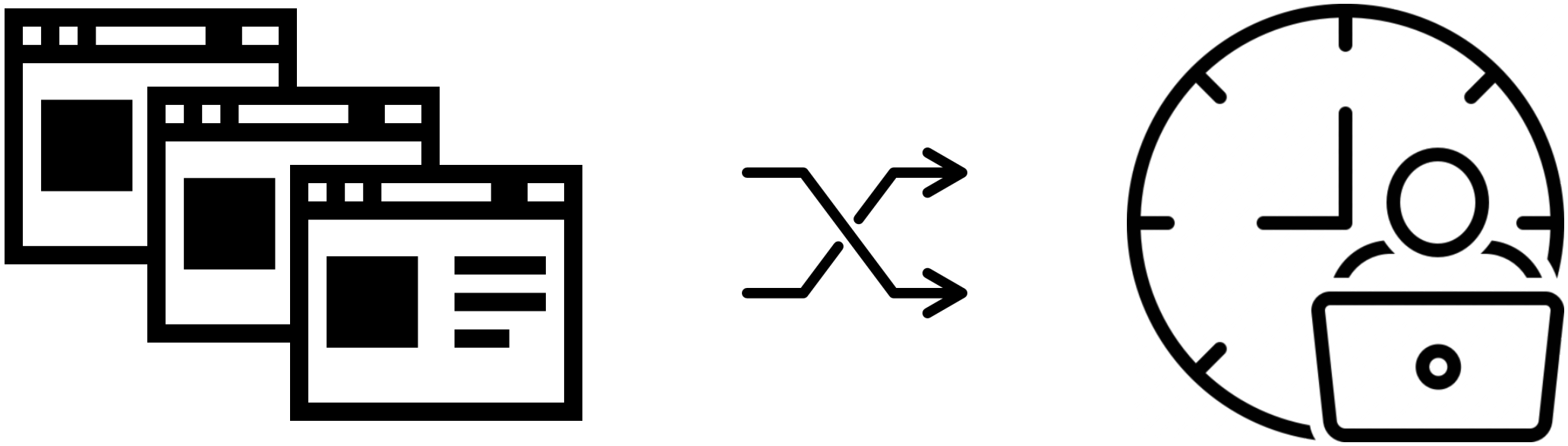
Stability Test



Stability Test



Stability Test



Countermeasures

Regular defenses for server-side timing attacks include, a random delays in the response time, or fixed response times for sensitive requests. But are difficult and impractical to implement in reality due to **performance issues**.

Countermeasures

Regular defenses for server-side timing attacks include, a random delays in the response time, or fixed response times for sensitive requests. But are difficult and impractical to implement in reality due to **performance issues**.

Another option would be cookies with the **SameSite attribute**, that can indicate that they do not want to be send in third-party requests. However, as long as one of the cookies involved does not indicate it, the attack would still work.

NortonLifeLock™

Thank You!

iskander_sanchez@symantec.com

iskander-sanchez-rola.com