JavaScript Integrity

The New Attack Surface

John Elliott, Security Advisor

Agenda

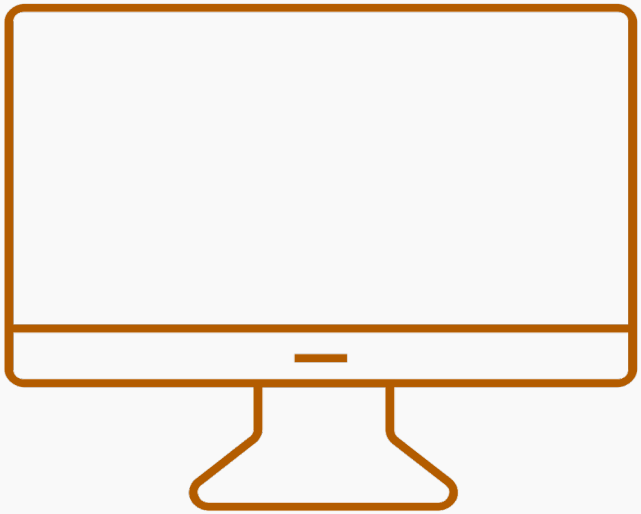
- The JavaScript attack surface
- New requirements in PCI DSS version 4
- Defeating JavaScript Integrity attacks

JavaScript

- **What is it?**
  - **JavaScript is a programming language**
  - **It runs in the customer's browser**
    - Chrome, IE, Edge, Firefox, Safari etc...
  - **It is how the web works now**
  - **It's what made gmail and then rest of the web changed**

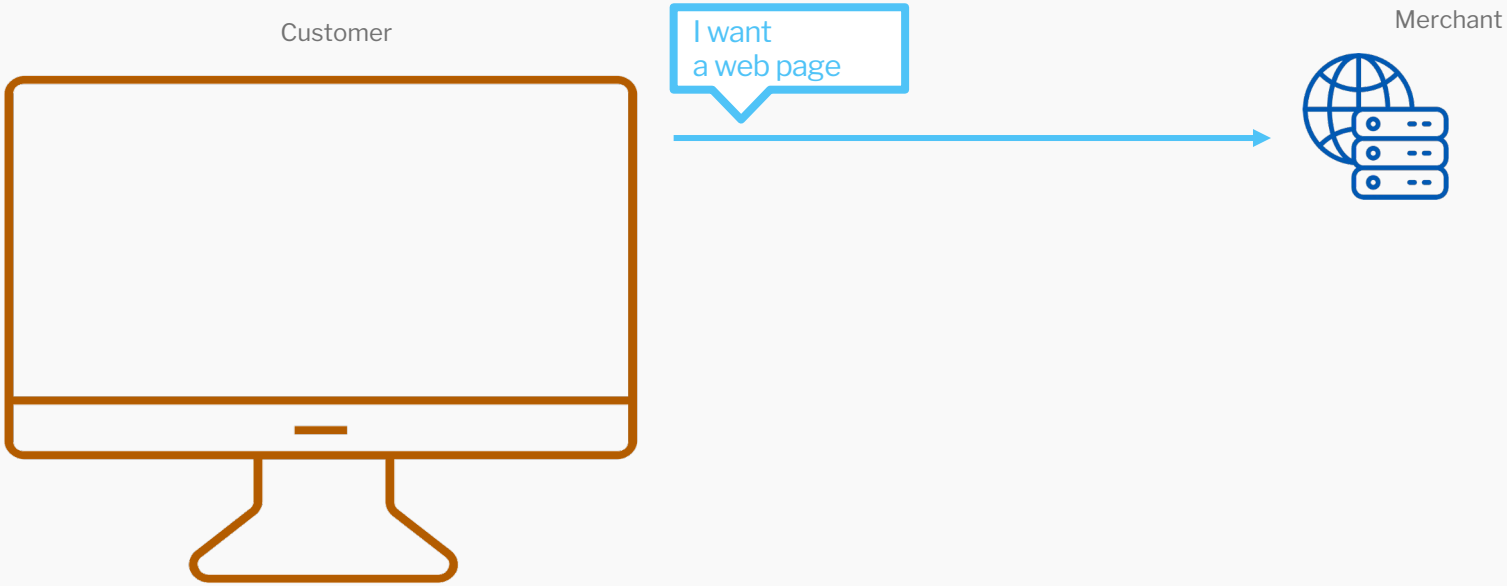
How the web works

Customer

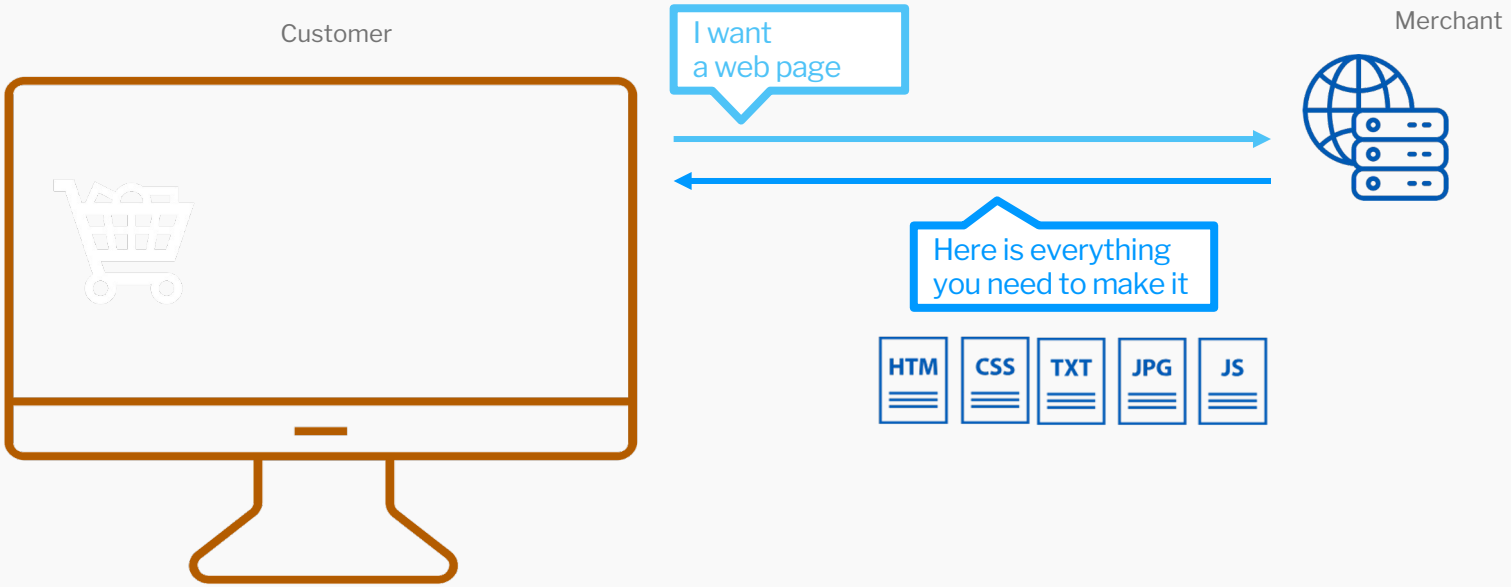


Merchant

How the web works

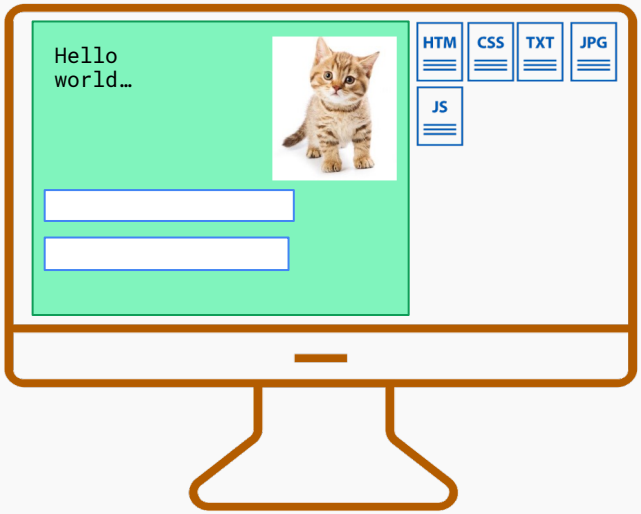


How the web works



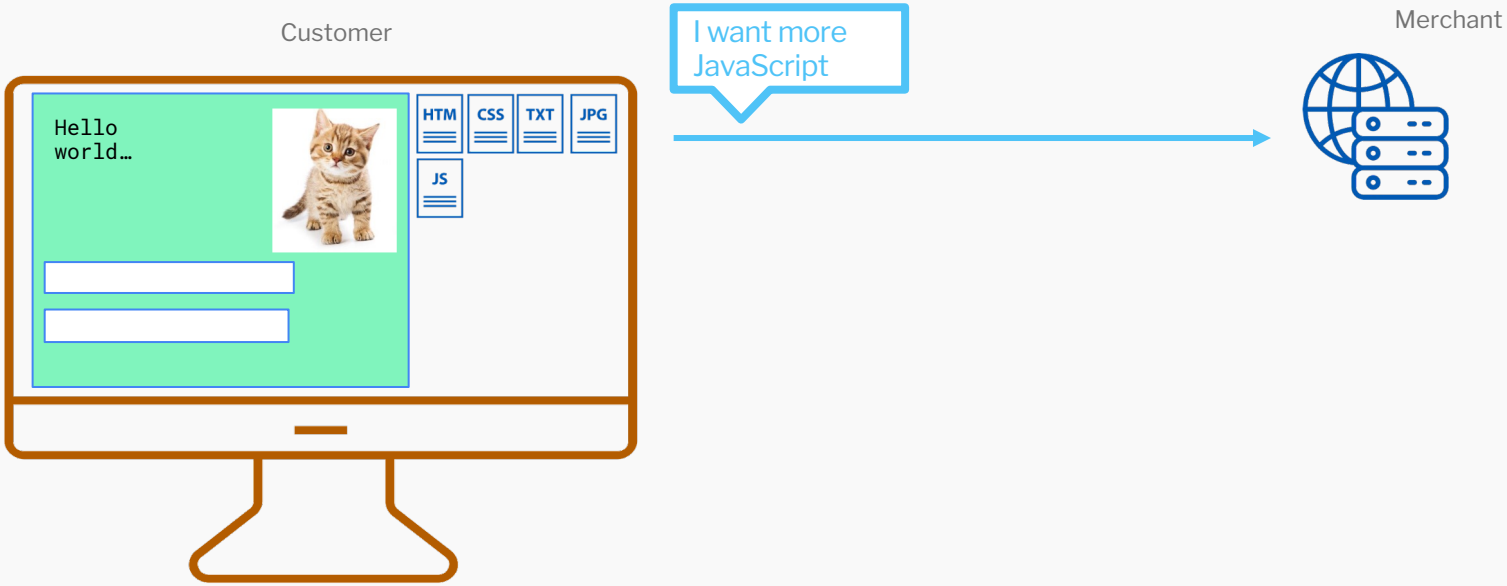
How the web works

Customer

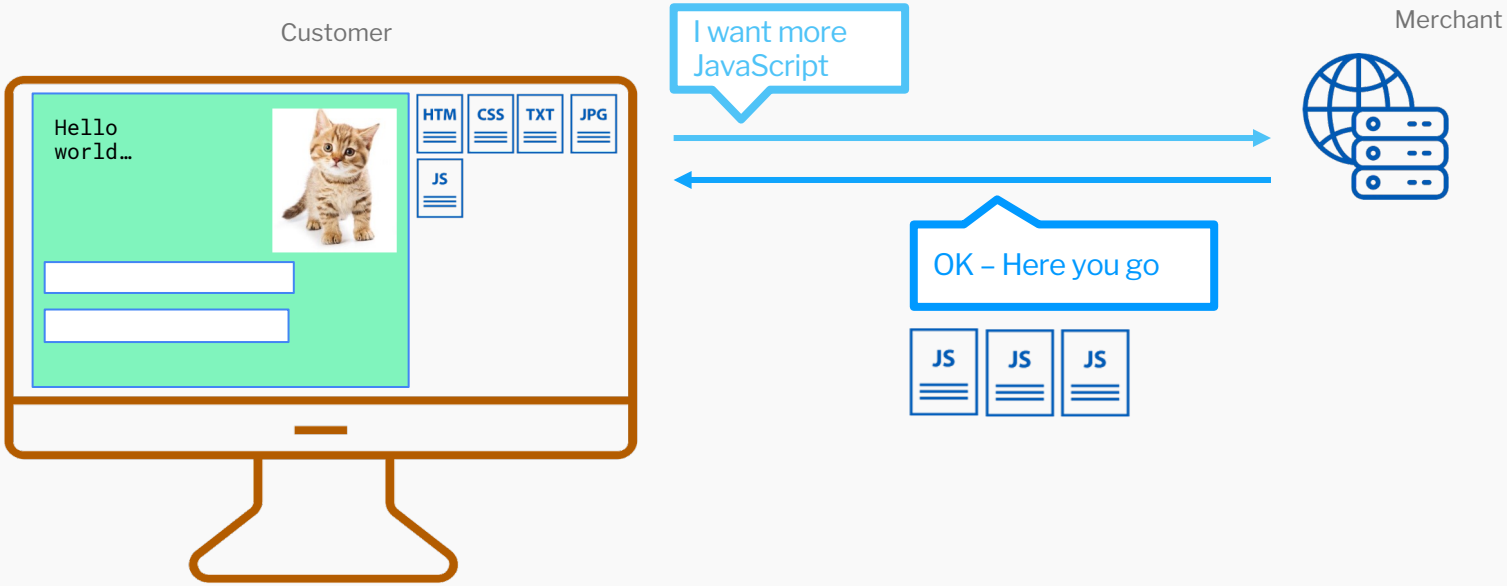


Merchant

How the web works

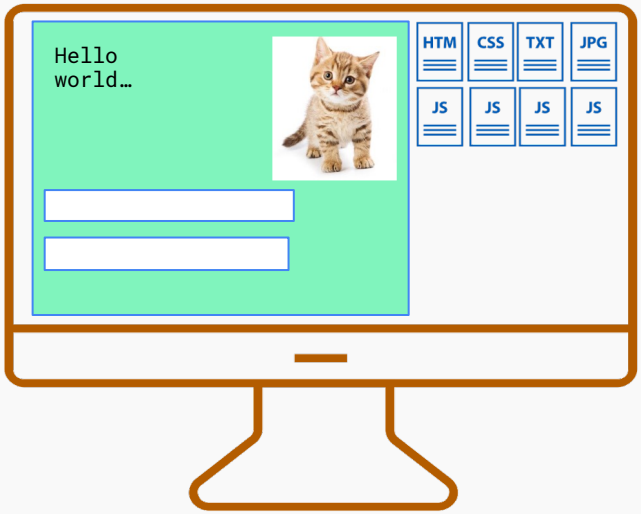


How the web works



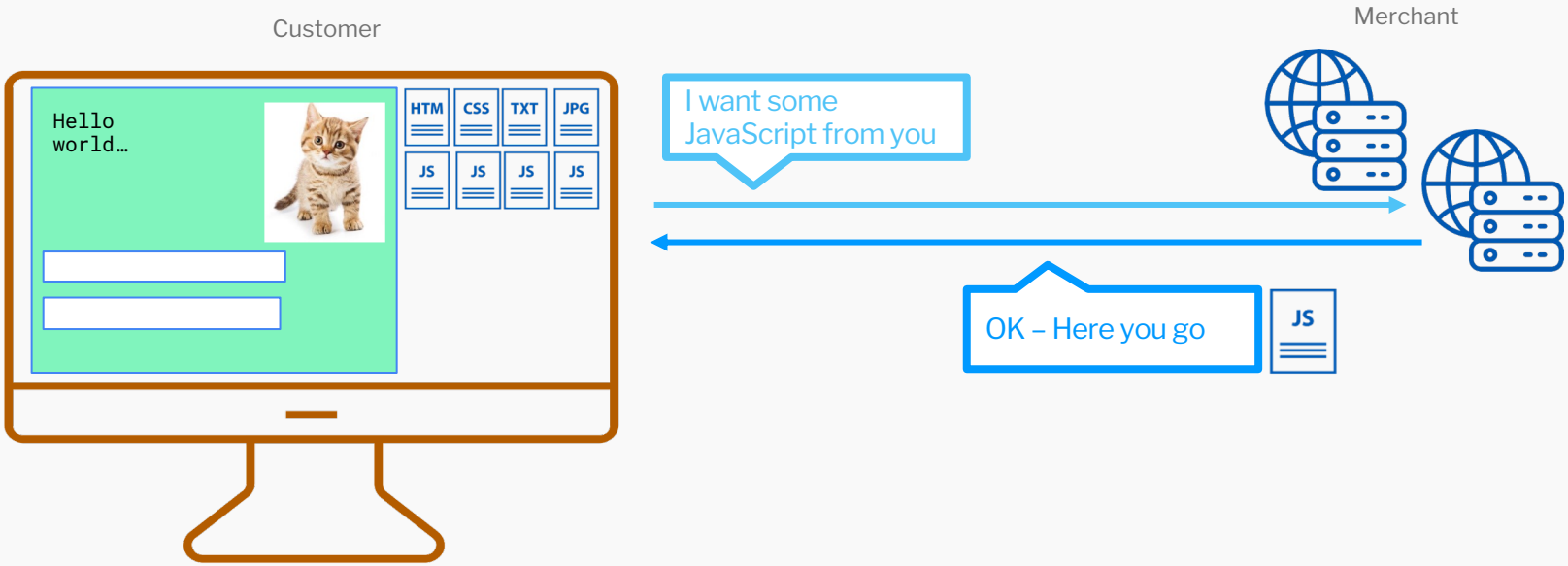
How the web works

Customer



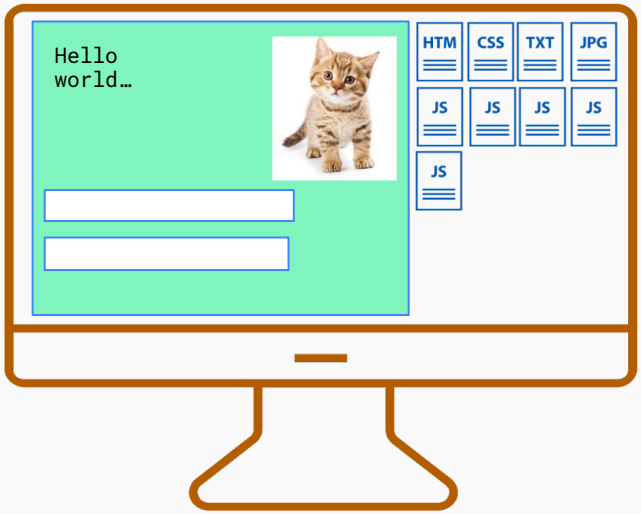
Merchant

How the web works



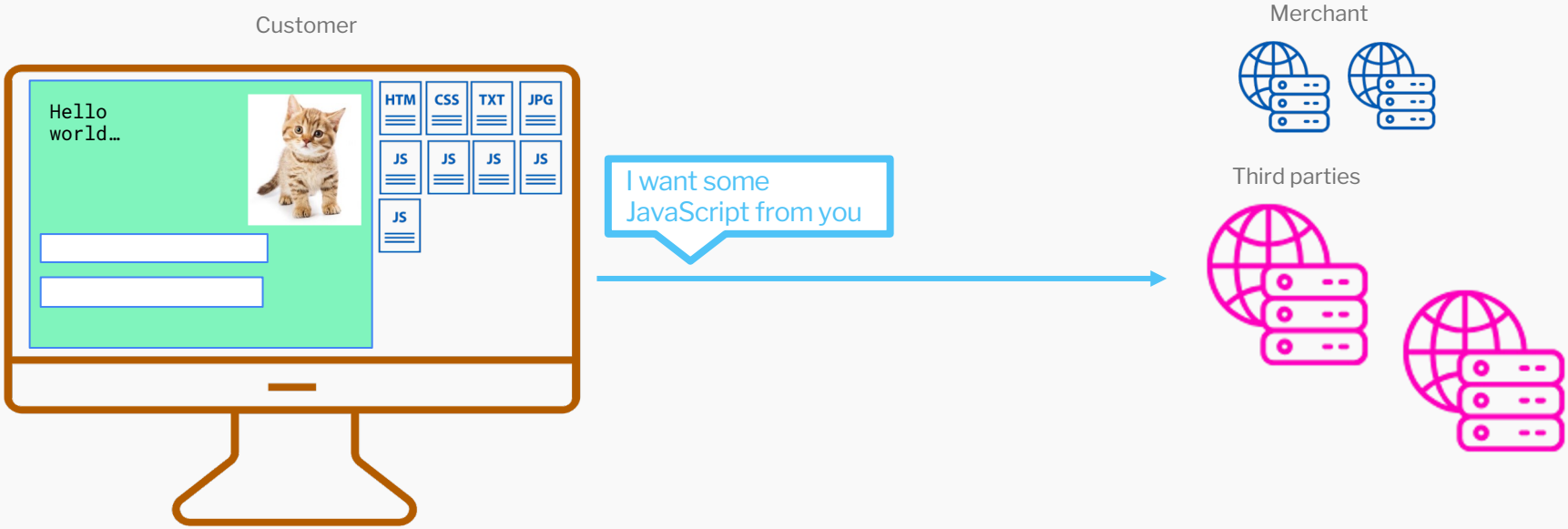
How the web works

Customer

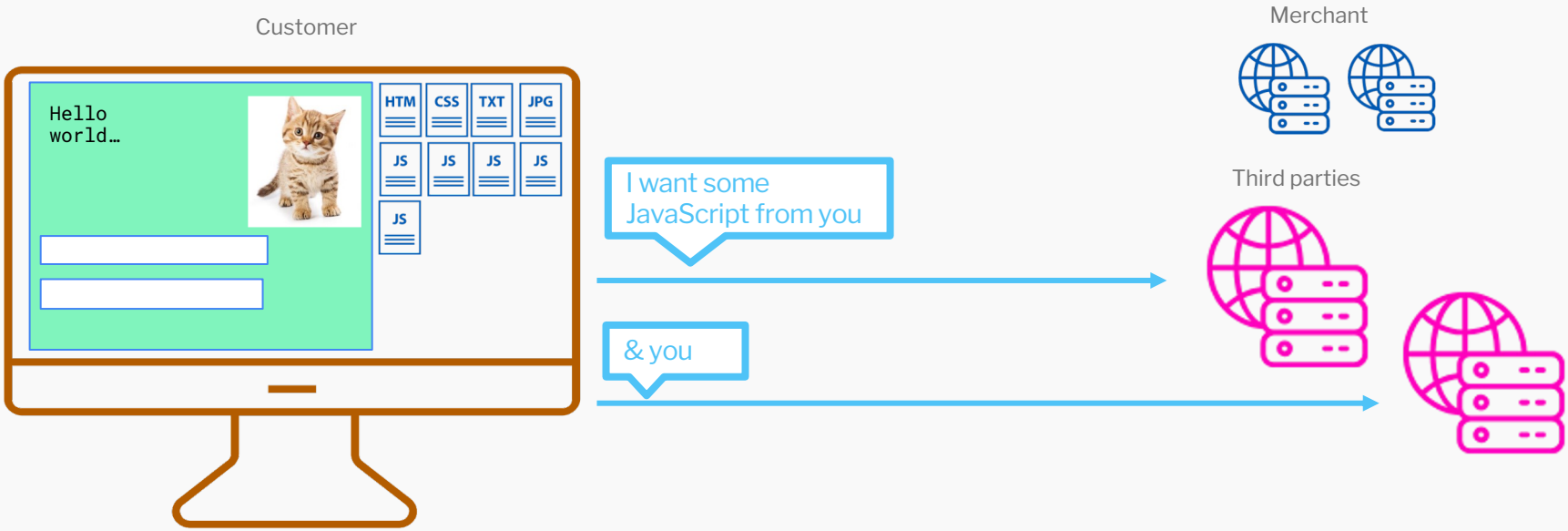


Merchant

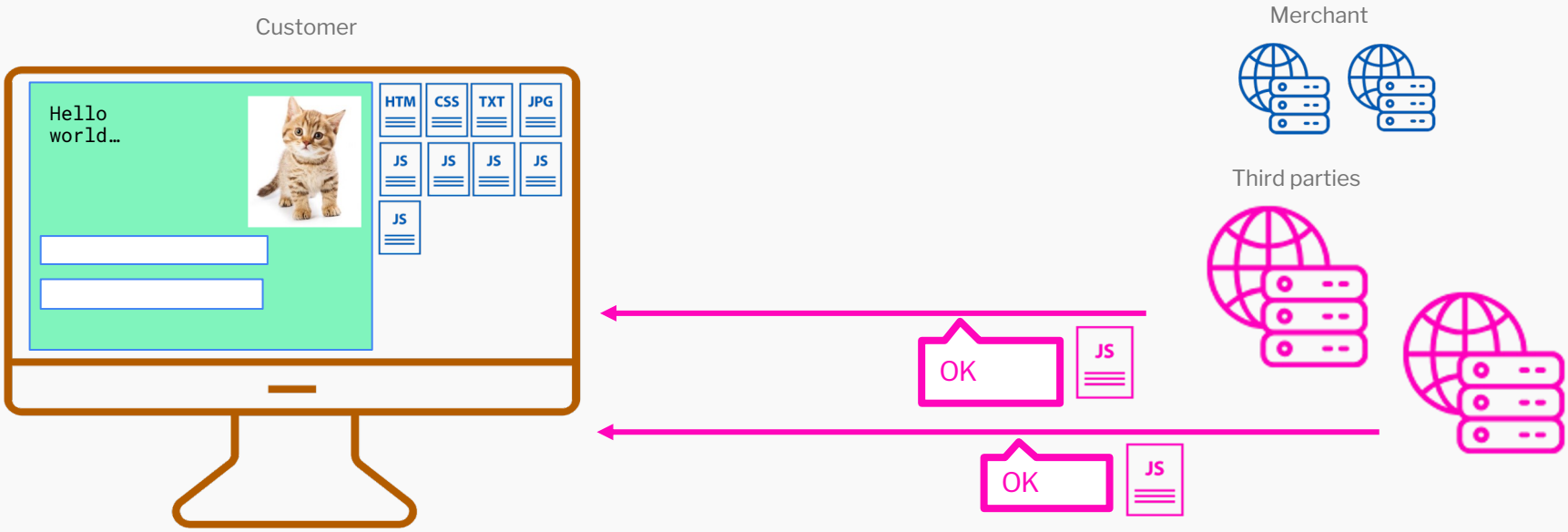
How the web works



How the web works

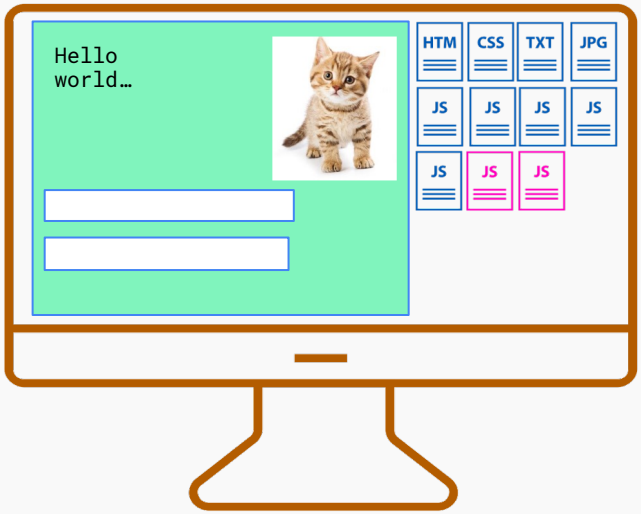


How the web works



How the web works

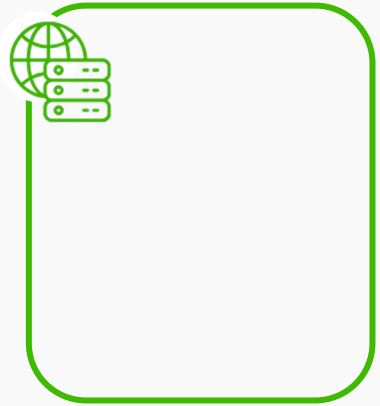
Customer



Merchant

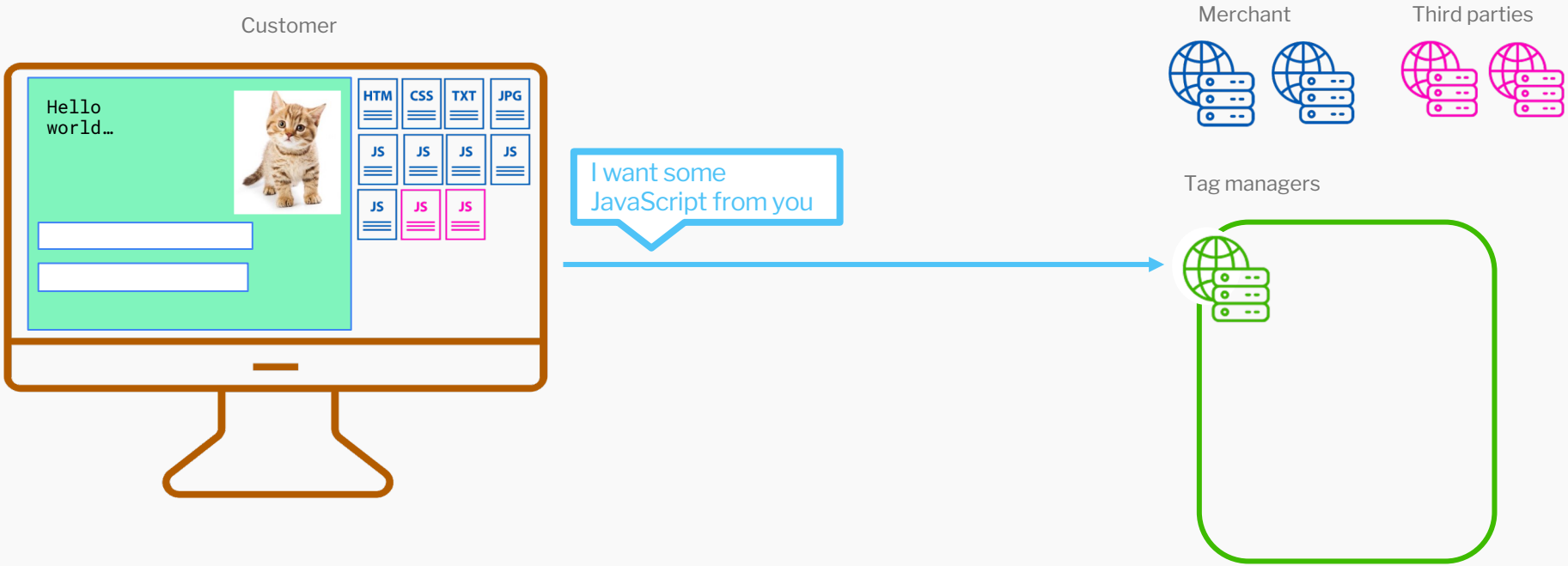
Third parties

Tag managers

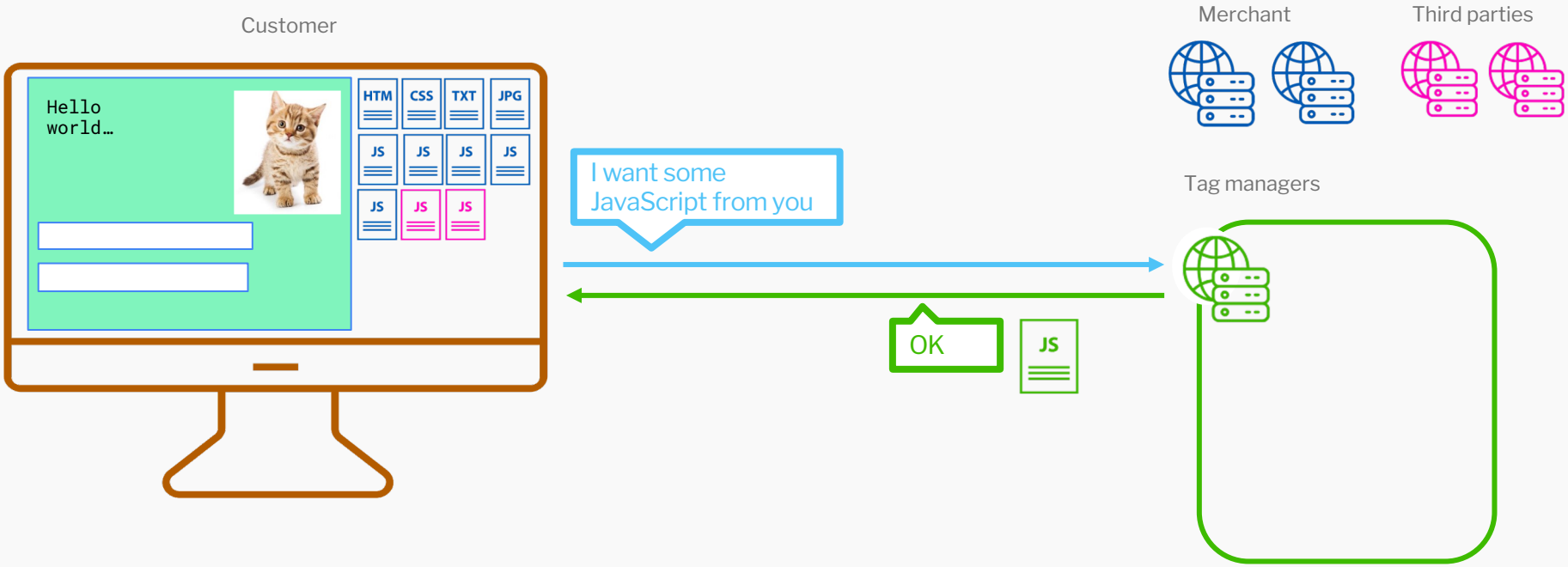


jscrambler

How the web works

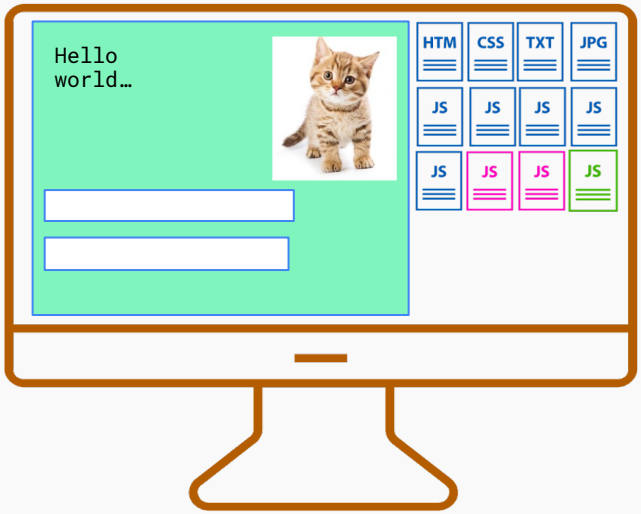


How the web works



How the web works

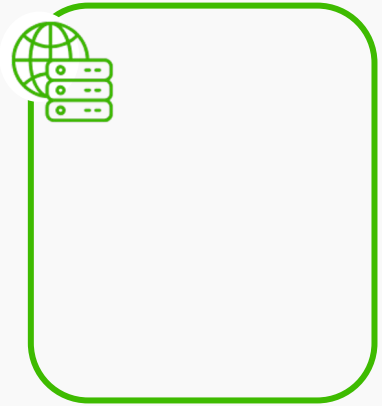
Customer



Merchant

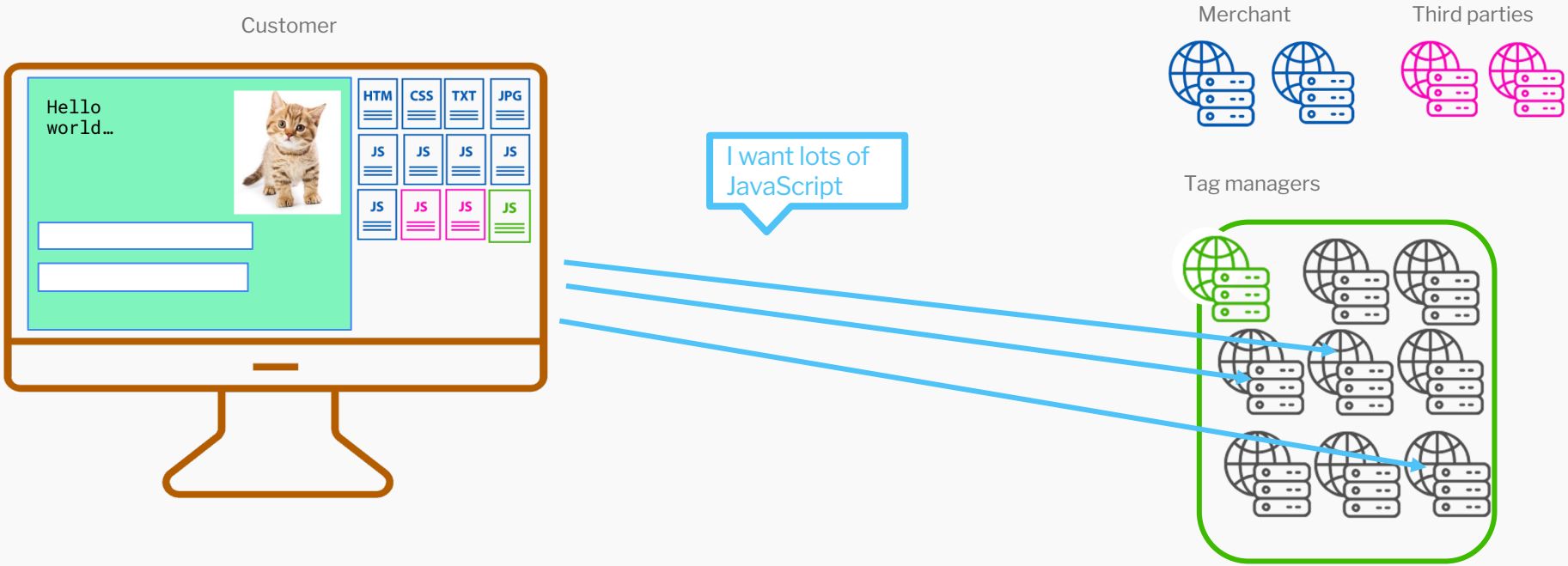
Third parties

Tag managers

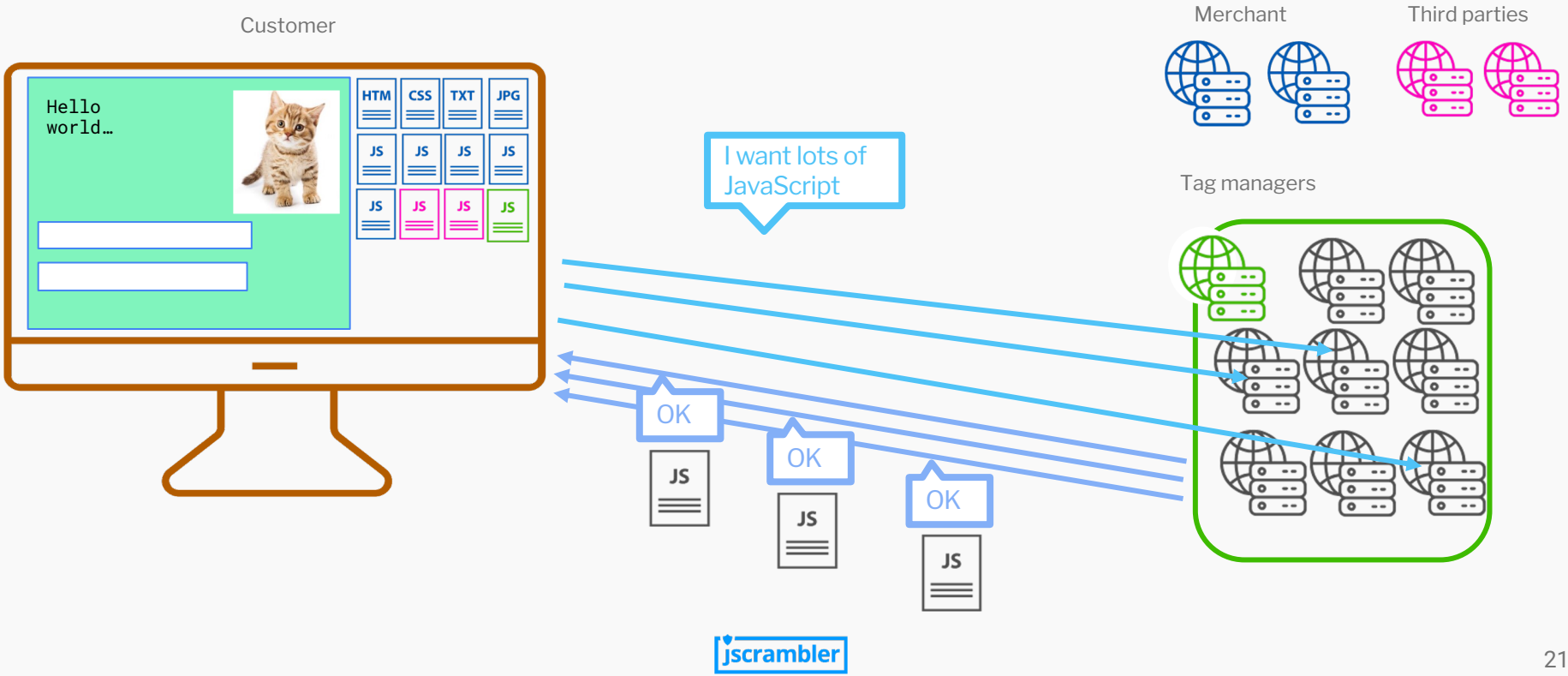


jscrambler

How the web works

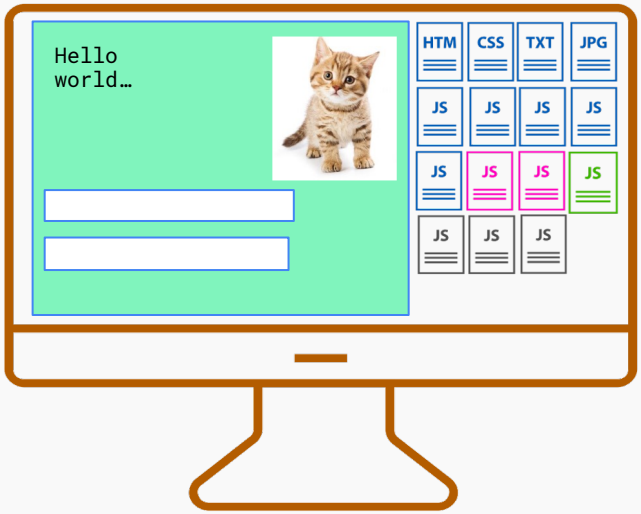


How the web works



How the web works

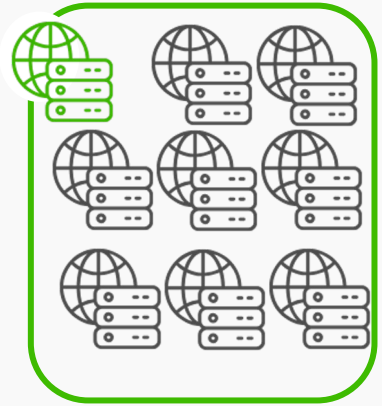
Customer



Merchant

Third parties

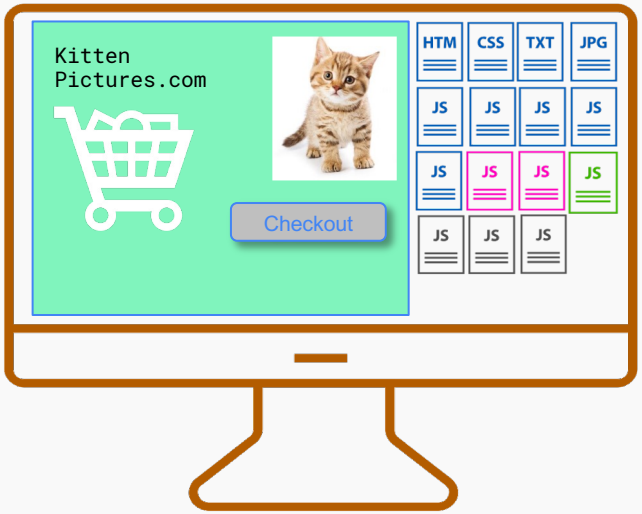
Tag managers



jscrambler

But this is an ecommerce site

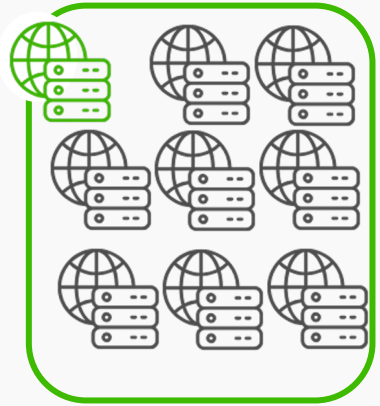
Customer



Merchant

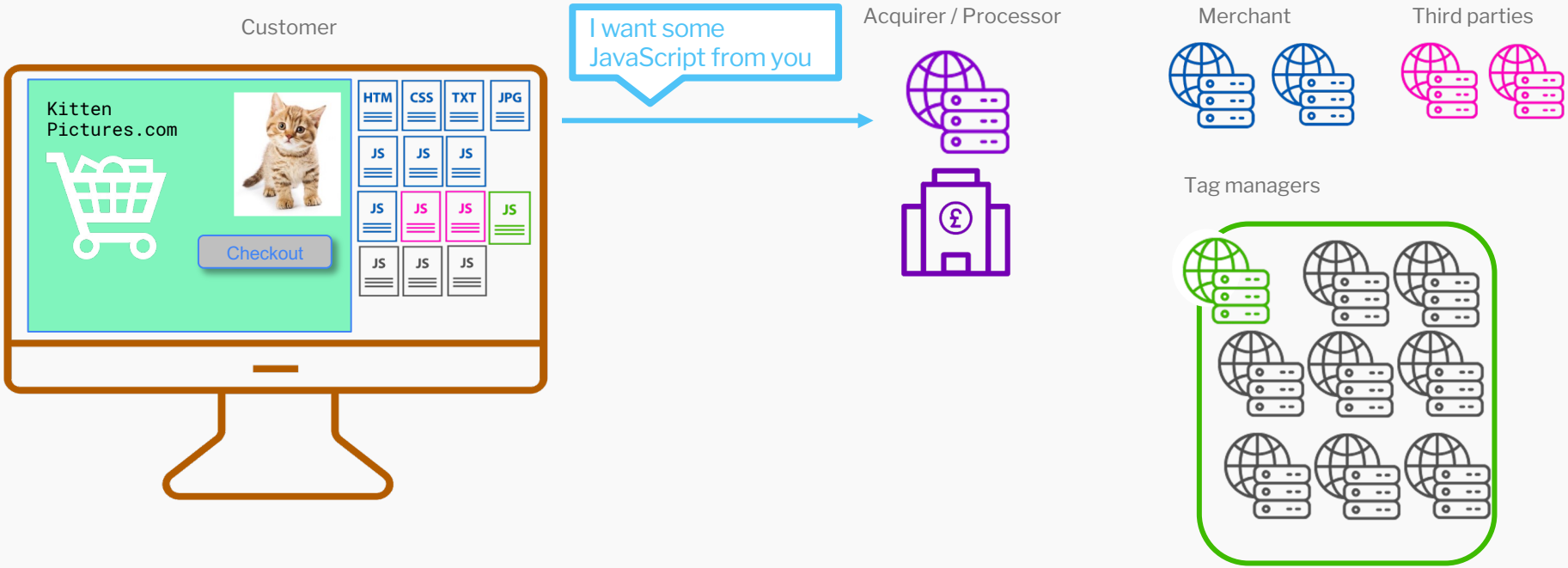
Third parties

Tag managers

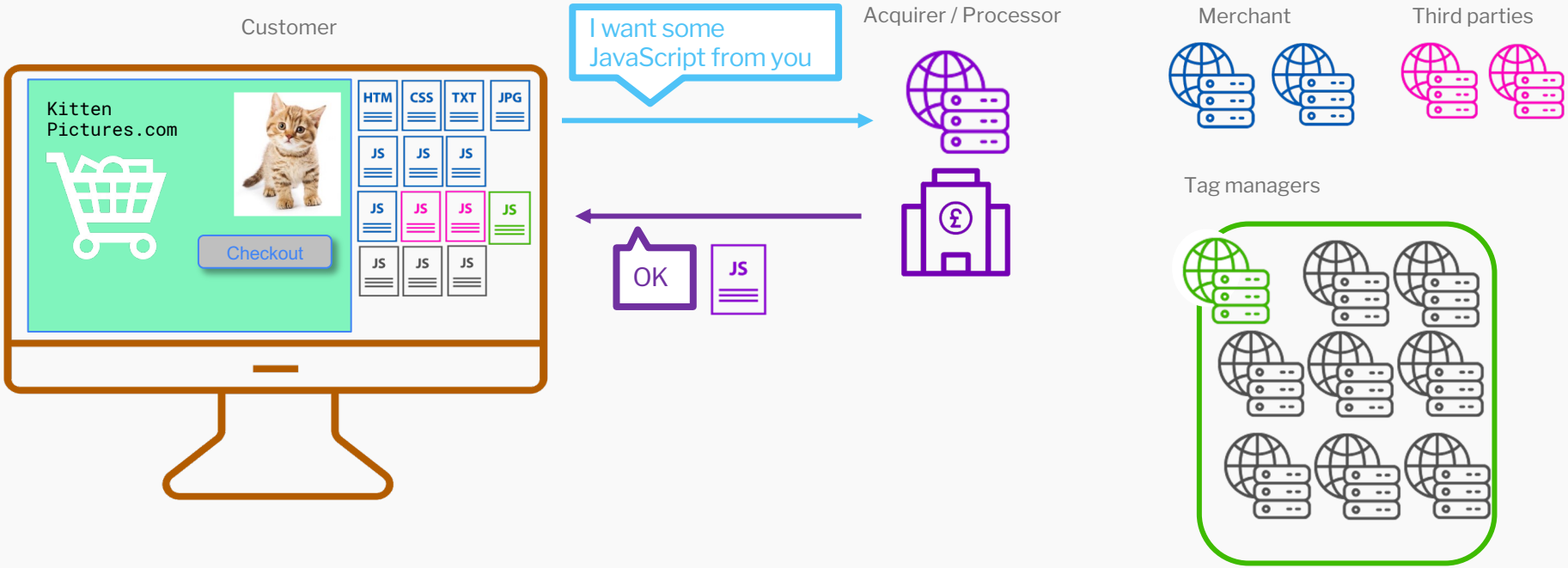


jscrambler

But this is an ecommerce site

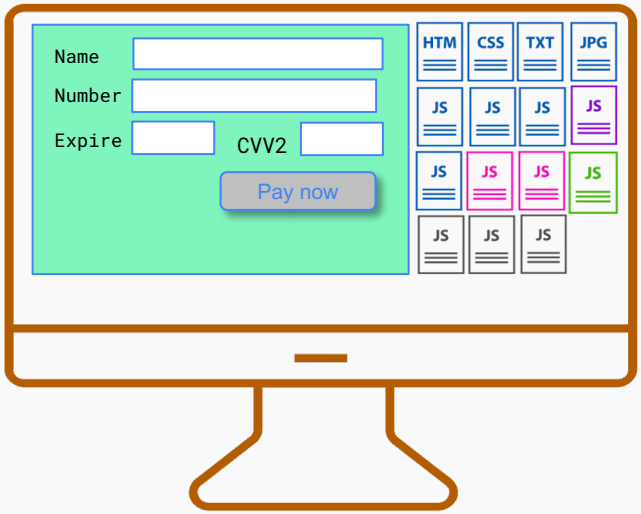


But this is an ecommerce site



But this is now a payment form!

Customer

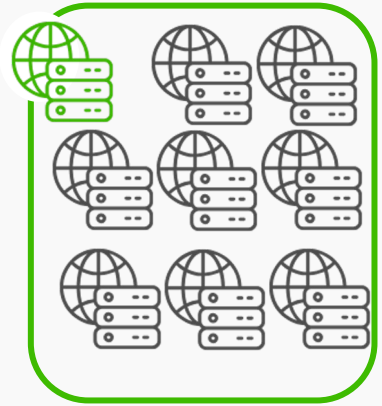


Acquirer / Processor

Merchant

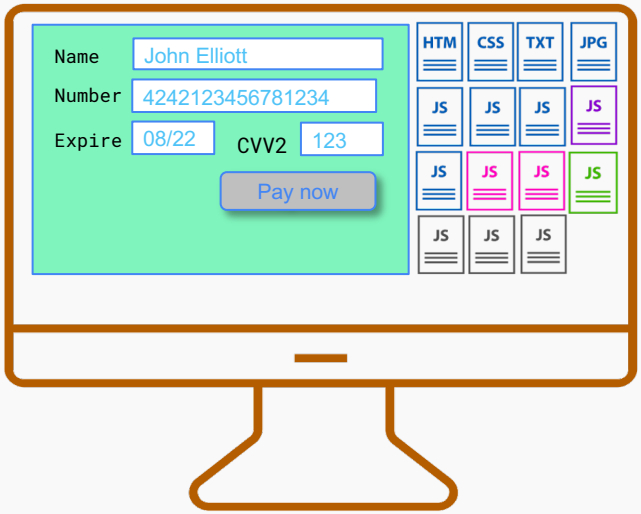
Third parties

Tag managers



But this is now a payment form!

Customer

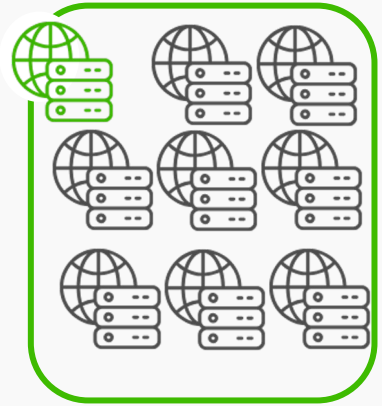


Acquirer / Processor

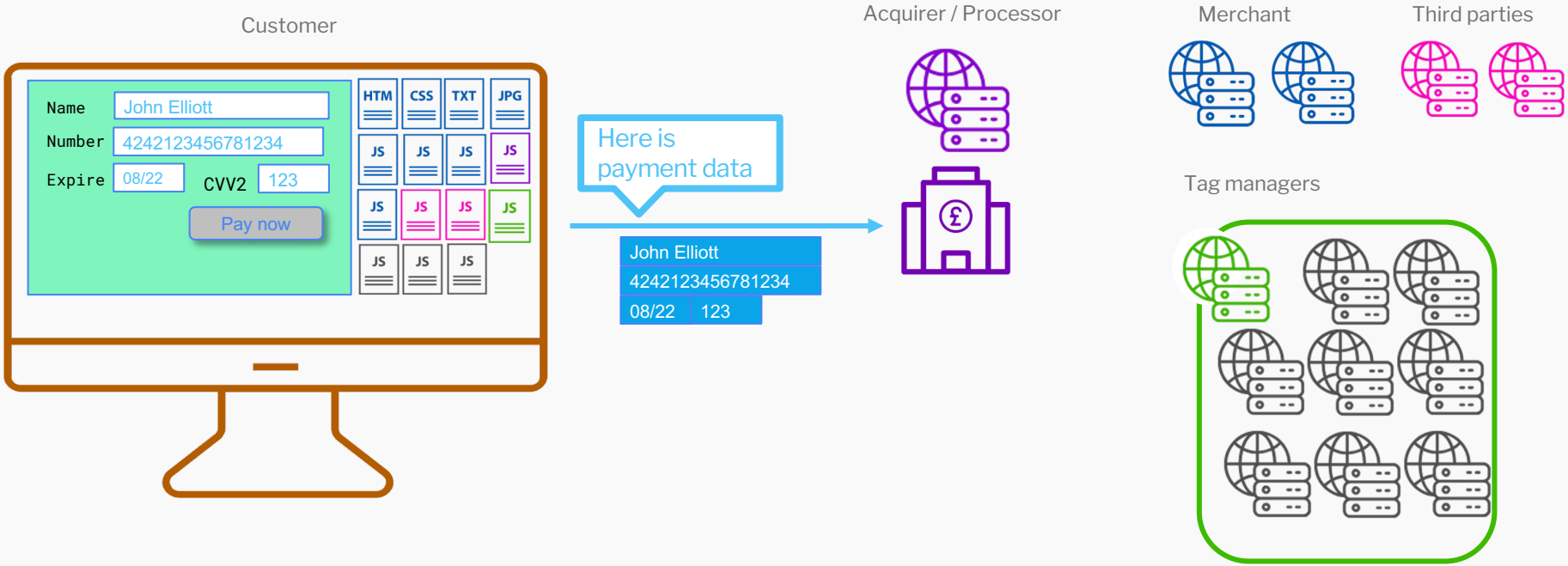
Merchant

Third parties

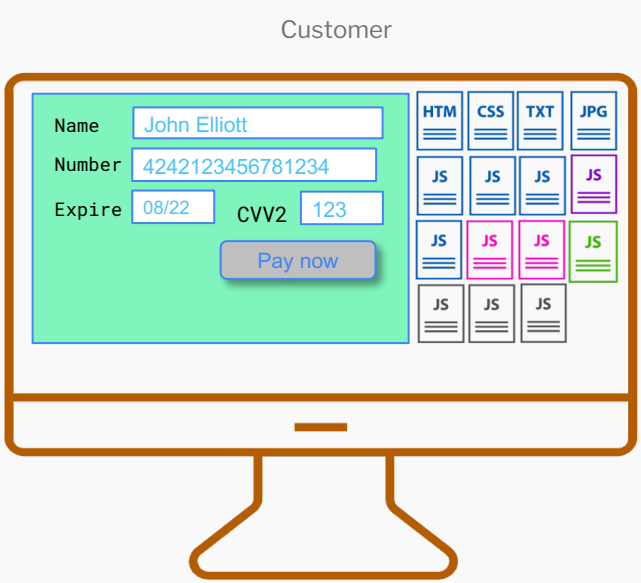
Tag managers

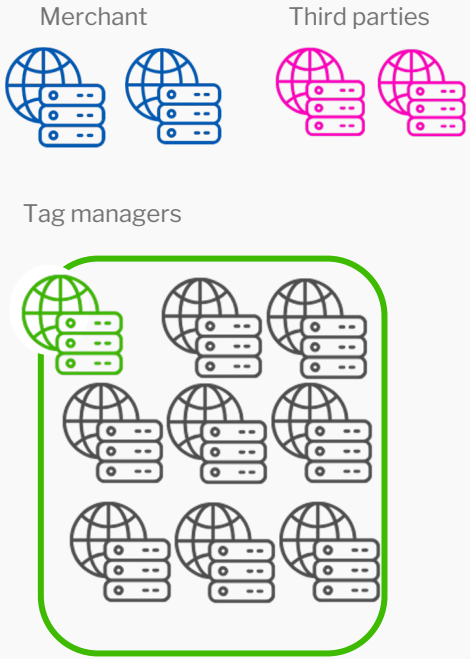


But this is now a payment form!



But this is now a payment form!

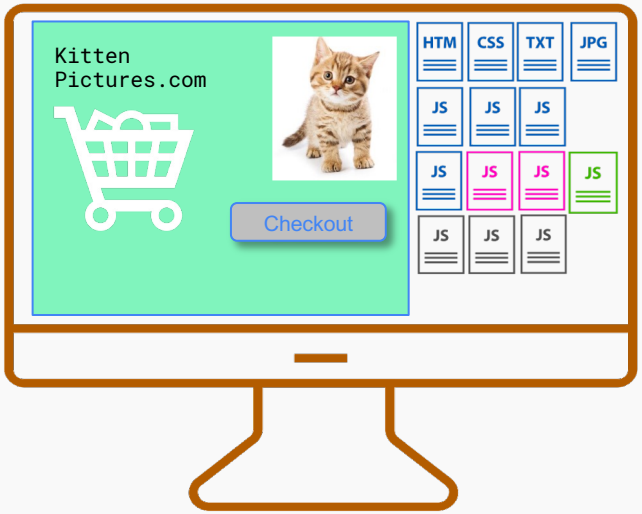


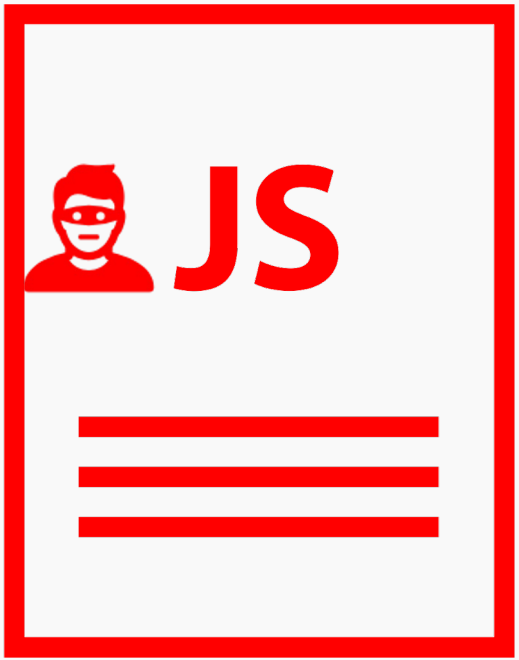


The attack explained

The attack

Customer

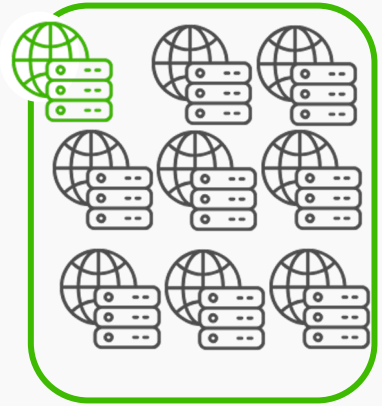




Merchant

Third parties

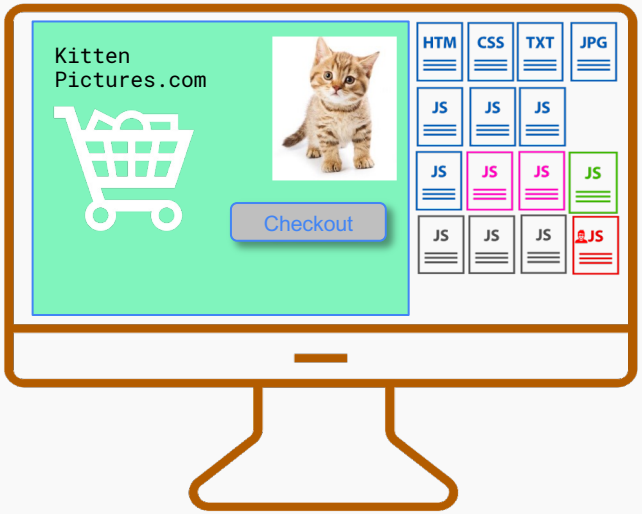
Tag managers



jscrambler

The attack

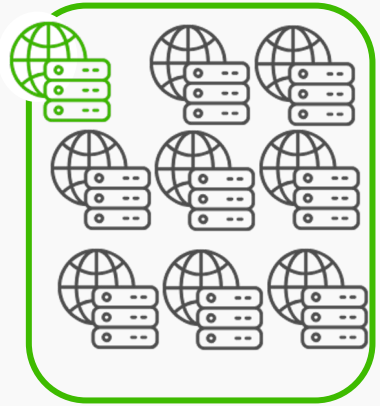
Customer



Merchant

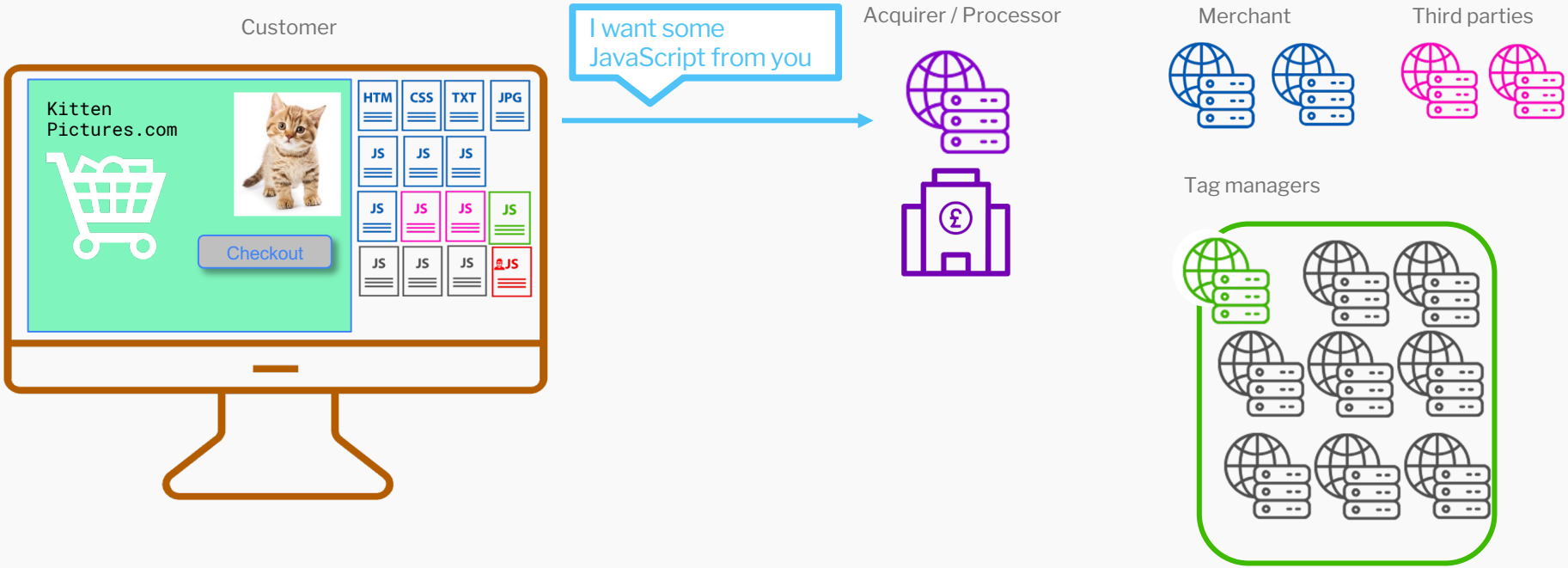
Third parties

Tag managers

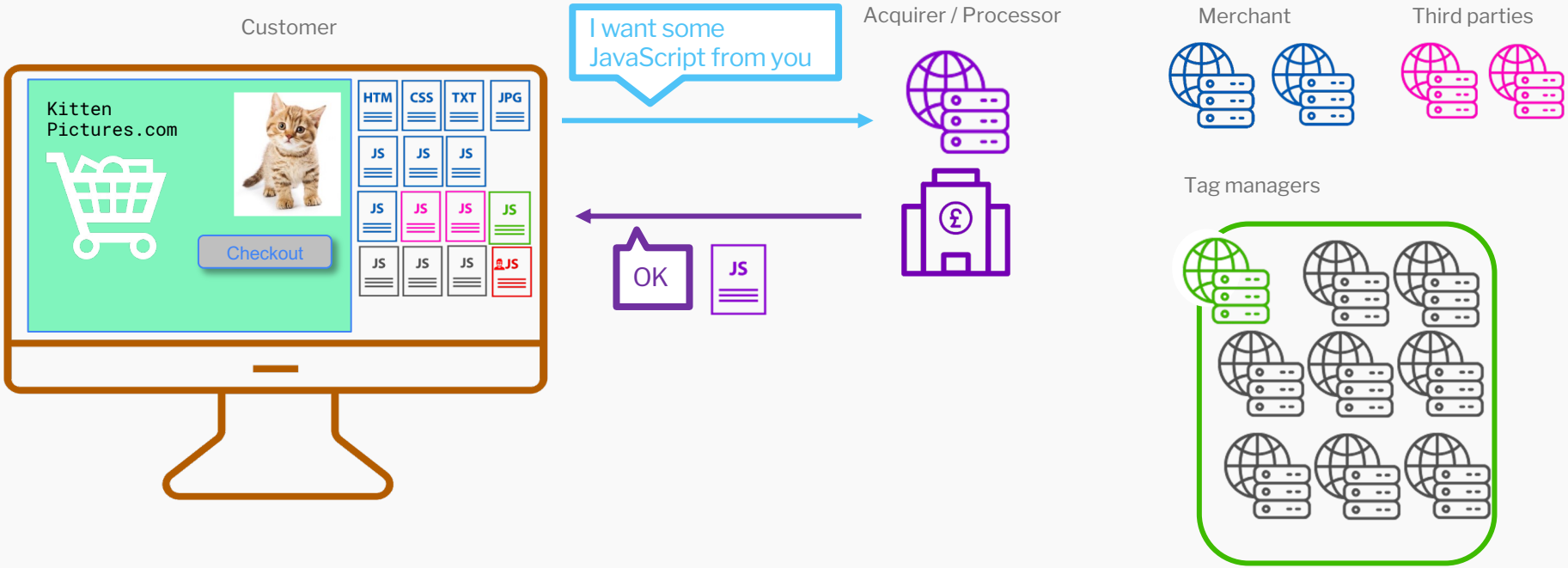


jscrambler

The attack


Customer
Kitten Pictures.com
Checkout
HTM
CSS
TXT
JPG
JS
I want some JavaScript from you
Acquirer / Processor
Merchant
Third parties
Tag managers

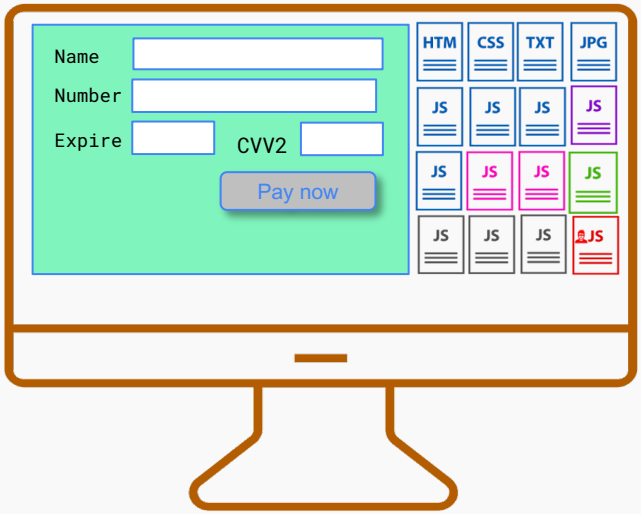
The attack



jscrambler

The attack

Customer

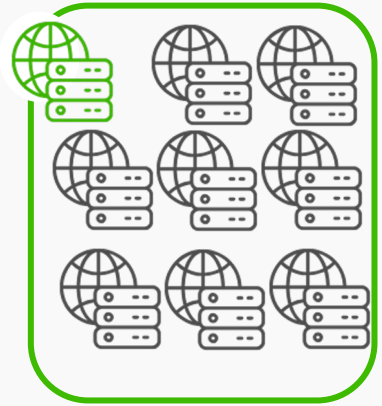


Acquirer / Processor

Merchant

Third parties

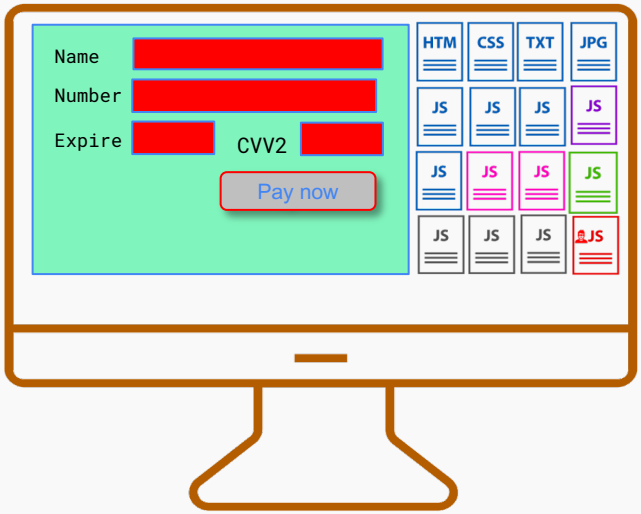
Tag managers



jscrambler

The attack

Customer

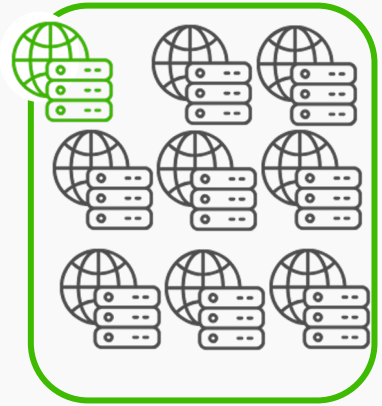


Acquirer / Processor

Merchant

Third parties

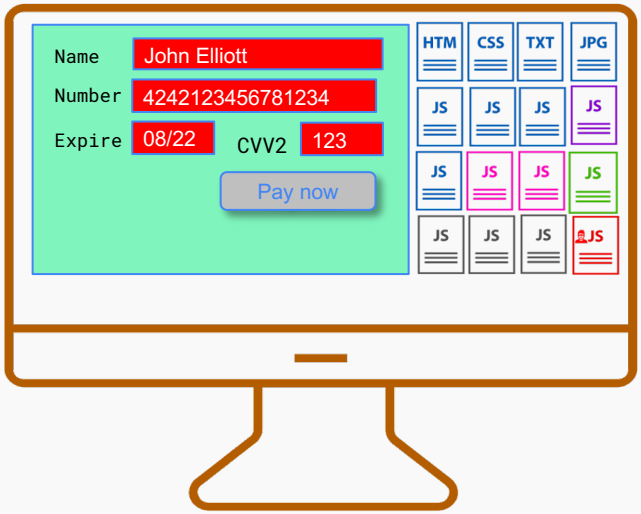
Tag managers



jscrambler

The attack

Customer

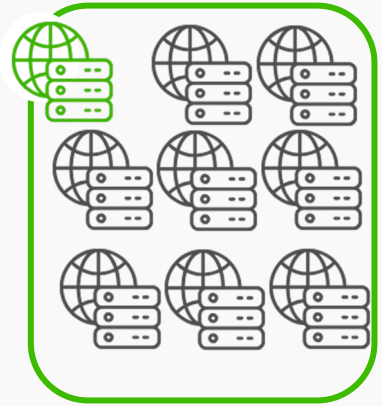


Acquirer / Processor

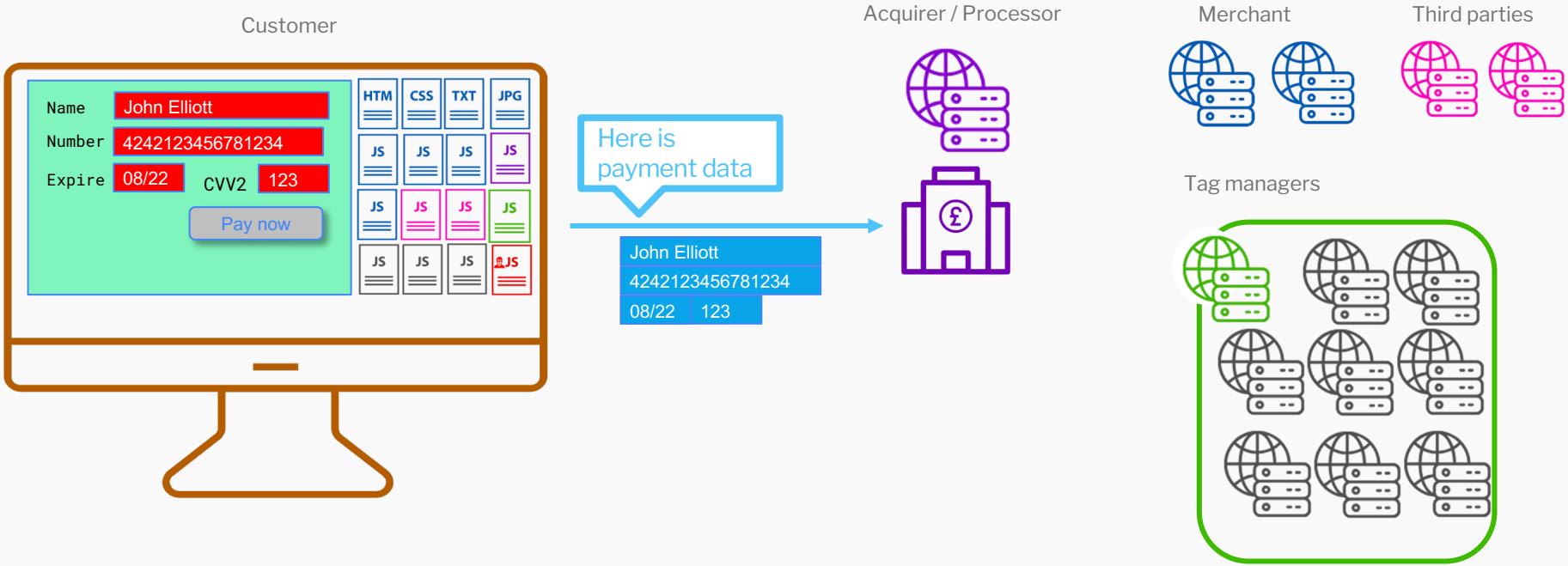
Merchant

Third parties

Tag managers

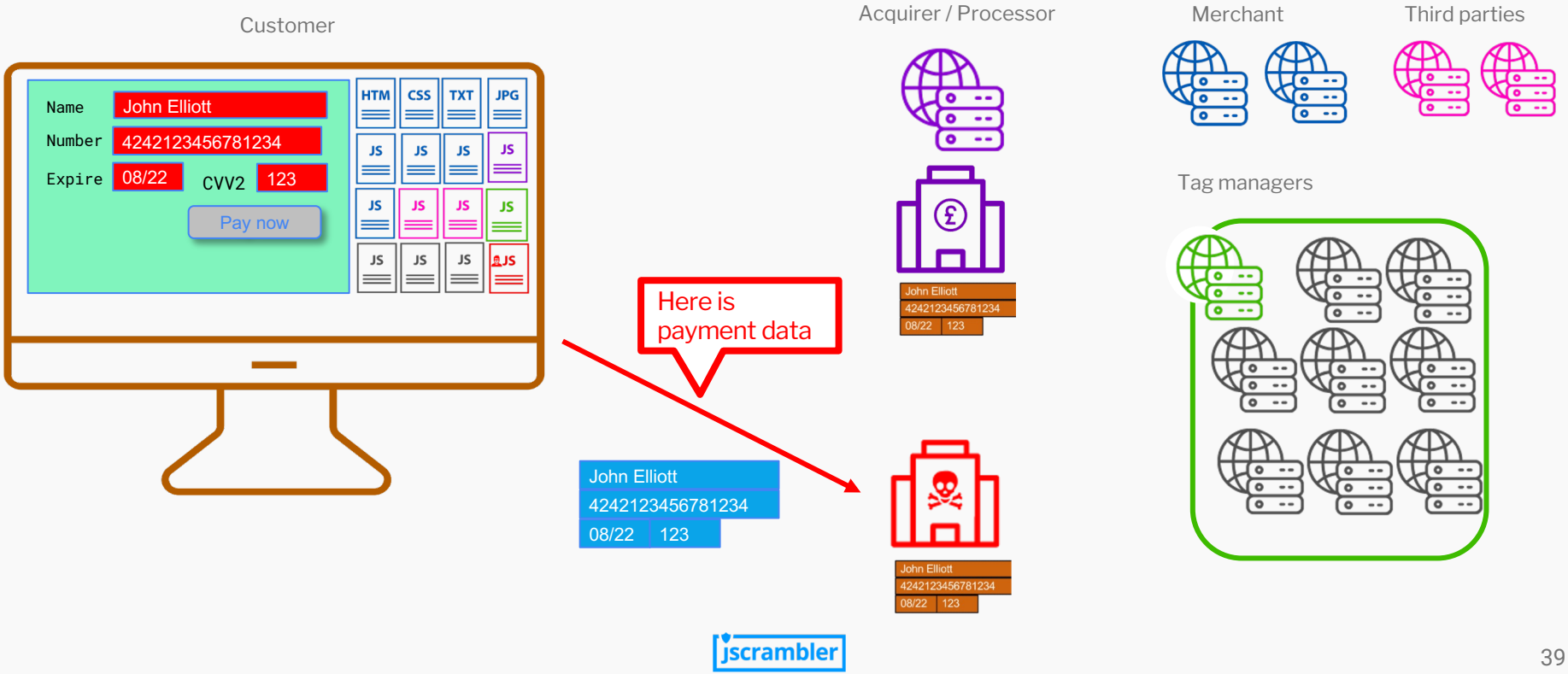


The attack

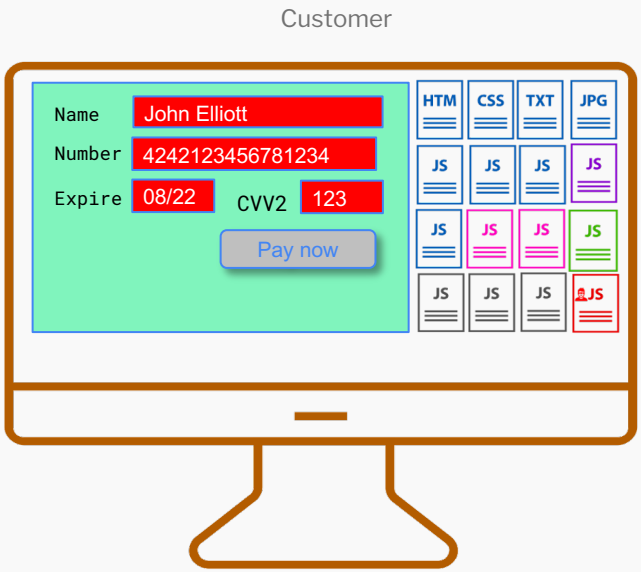


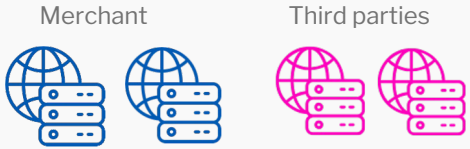
jscrambler

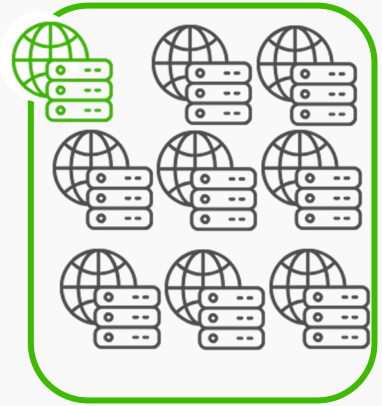
The attack


Customer
Name
John Elliott
Number
4242123456781234
Expire
08/22
CVV2
123
Pay now
HTM
CSS
TXT
JPG
JS
Here is payment data
John Elliott
4242123456781234
08/22
123
Acquirer / Processor
Merchant
Third parties
Tag managers
jscrambler

The attack





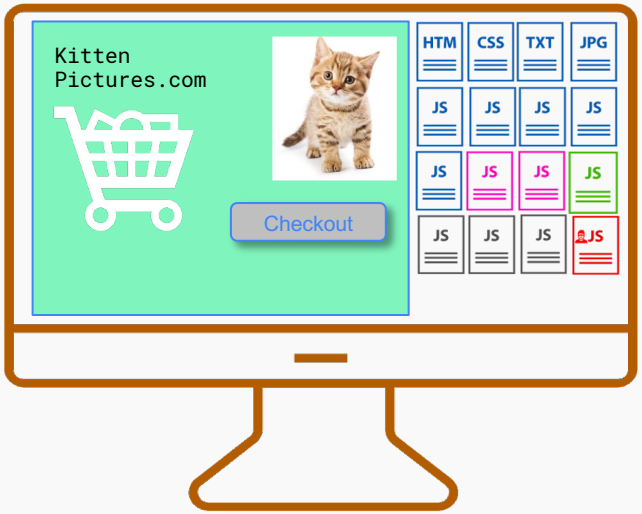


jscrambler

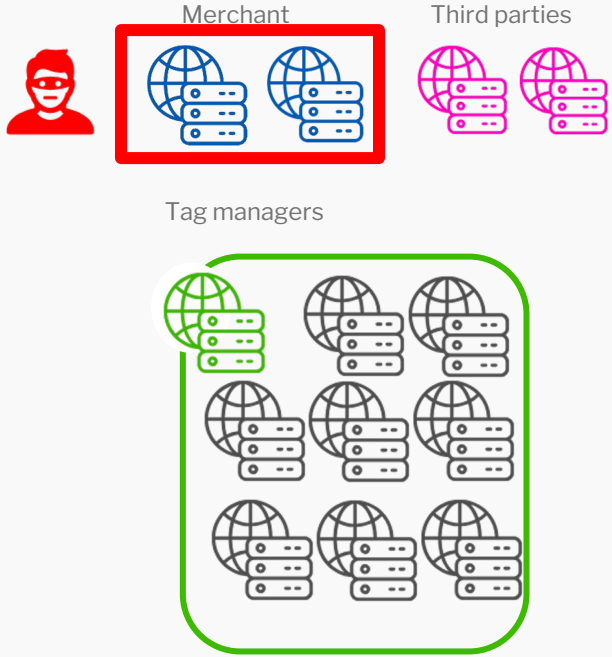
The attack surface

The attack surface

Customer

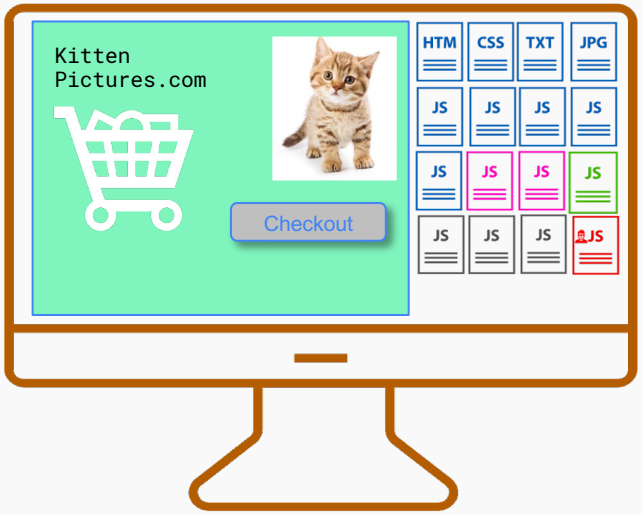


Any of the merchant's web servers

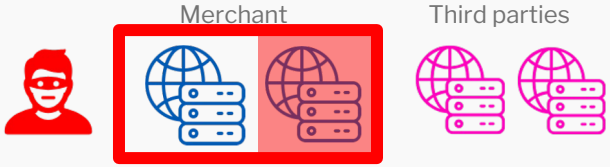


The attack surface

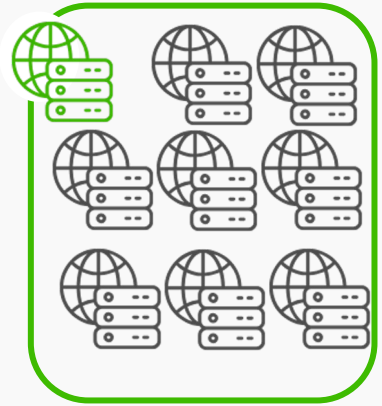
Customer



JS in the BA breach was loaded from a secondary web server

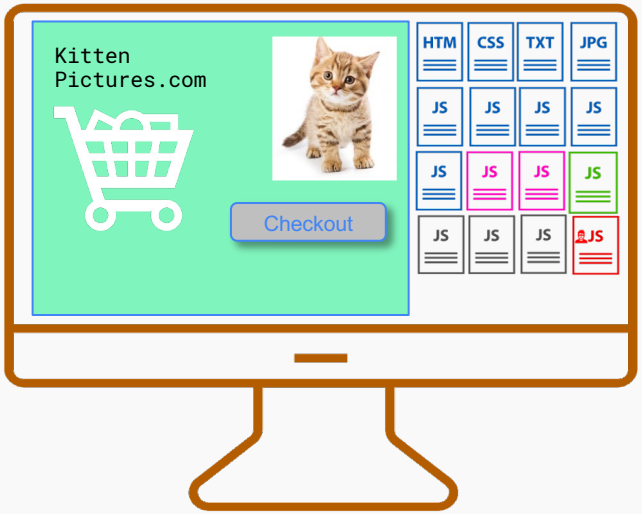


Tag managers



The attack surface

Customer

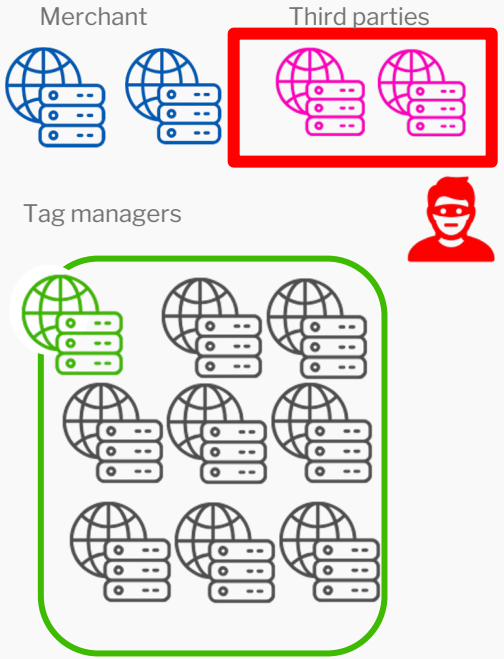


Any third party web servers

Merchant

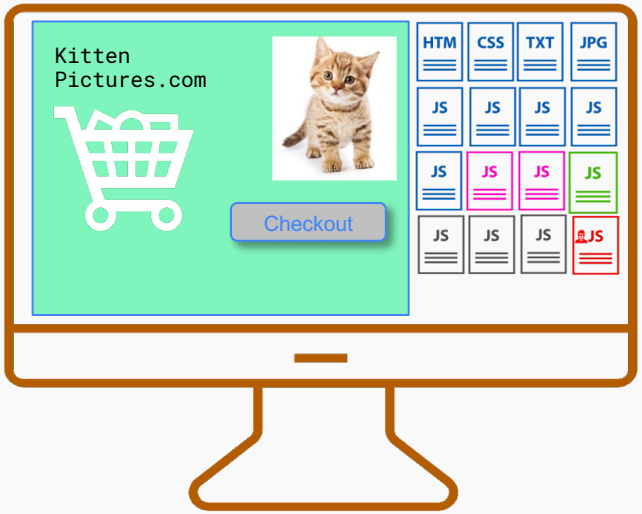
Third parties

Tag managers

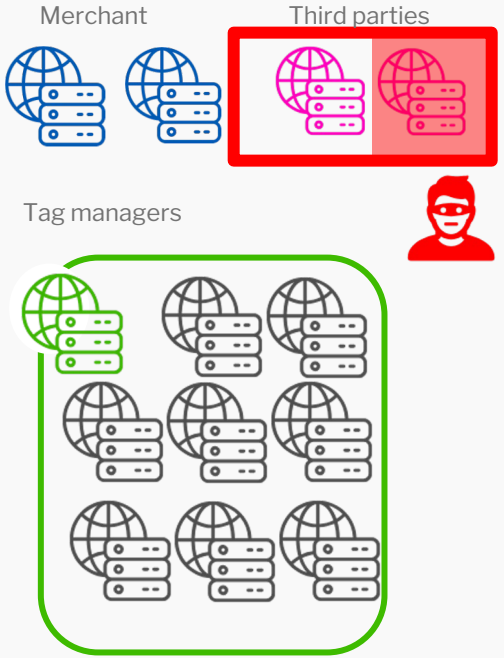


The attack surface

Customer



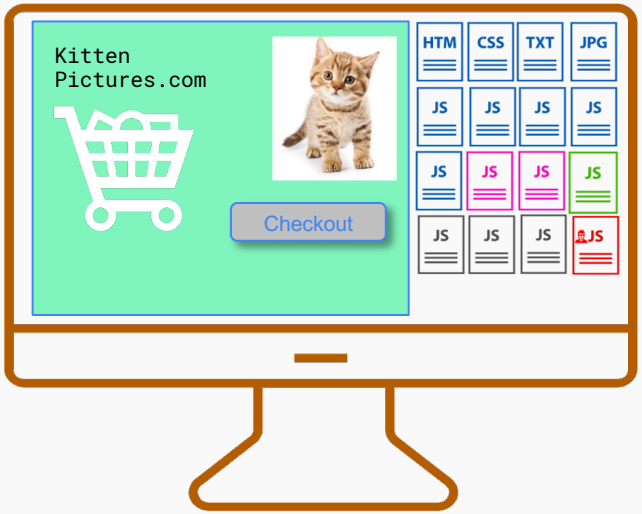
JS in the Ticketmaster breach was loaded from a third party web server



jscrambler

The attack surface

Customer

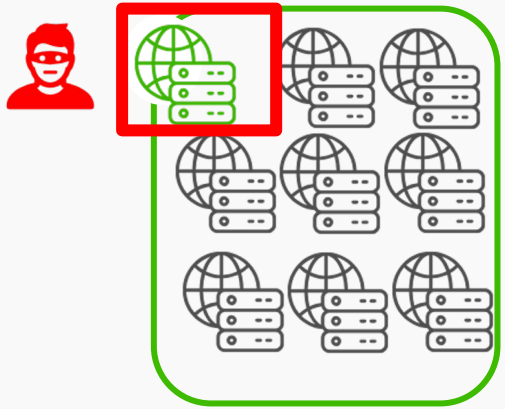


The tag manager

Merchant

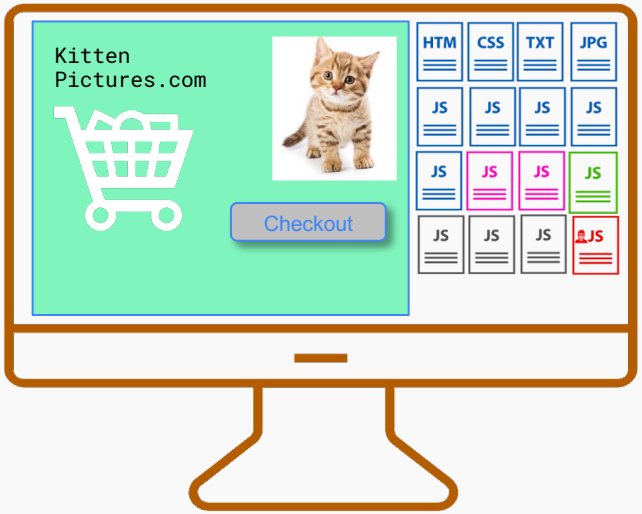
Third parties

Tag managers



The attack surface

Customer

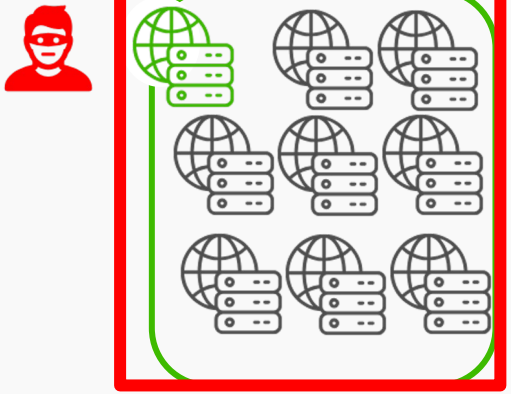


Any servers specified in the tag manager

Merchant

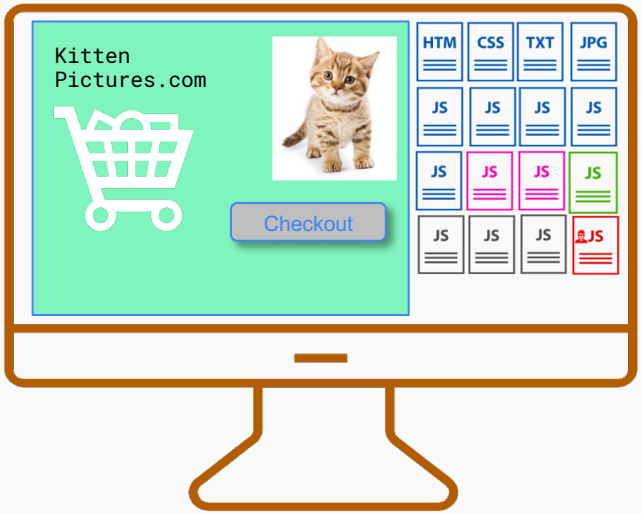
Third parties

Tag managers



The attack surface

Customer

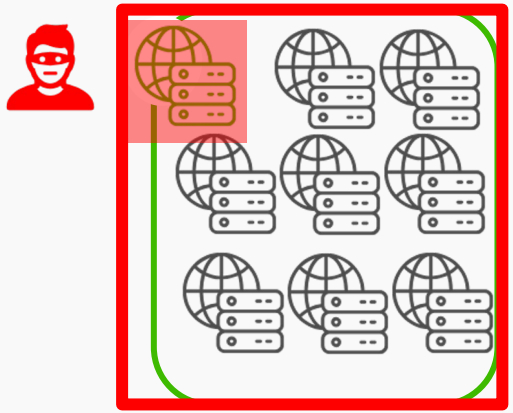


Adverline
breach
277 sites

Merchant

Third parties

Tag managers



jscrambler

How to stop attacks

Script Integrity

When the script is added to the page

If (when) it changes

jscrambler

When a script changes

- **Do you know what a script does?**
- **What functionality is in the script**
  - Read field contents?
- **Does it need to be on the payment page?**
- **Is where it comes from compliant with PCI DSS?**
  - Because if it is on the payment page, it can affect the security of cardholder data

You need to manage new scripts

jscrambler

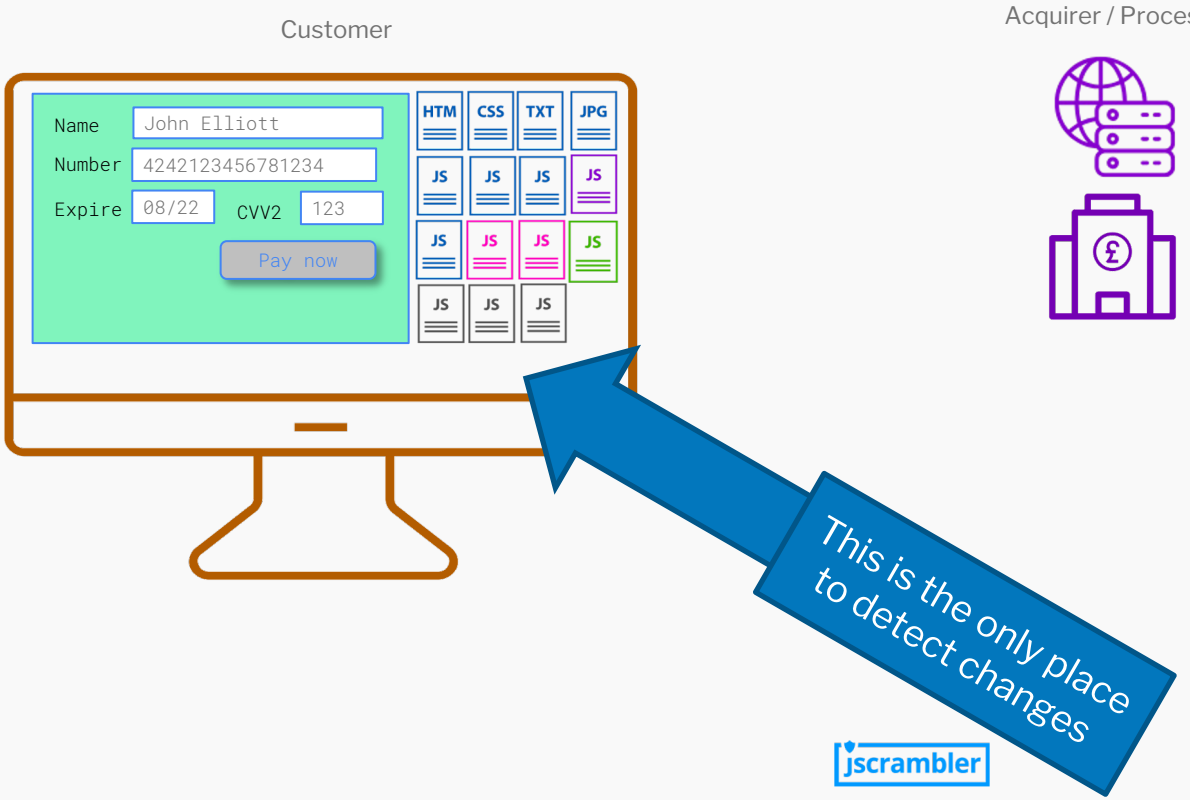
When a script changes

- **Authorized?**
- **What's the new functionality**
  - Read form fields
  - Hook into submit events
- **Malicious or benign?**
  - Trust indicators
  - Reputation

You need to monitor changes

jscrambler

Where to monitor changes?



The new requirements in PCI DSS 4.0

jscrambler

6.4.3

Defined Approach

All payment page scripts that are loaded and executed in the consumer's browser are managed as follows:

- A method is implemented to confirm that each script is authorized.
- A method is implemented to assure the integrity of each script.
- An inventory of all scripts is maintained with written justification as to why each is necessary

Customized Approach

Unauthorized code cannot be present in the payment page as it is rendered in the consumer's browser.

11.6.1

Defined Approach

A change- and tamper-detection mechanism is deployed as follows:

- To alert personnel to unauthorized modification (including indicators of compromise, changes, additions, and deletions) to the HTTP headers and the contents of payment pages as received by the consumer browser.
- The mechanism is configured to evaluate the received HTTP header and payment page.
- The mechanism functions are performed as follows:
  - At least once every seven days, OR
  - Periodically

Customized Approach

E-commerce skimming code or techniques cannot be added to payment pages as received by the consumer browser without a timely alert being generated.
Anti-skimming measures cannot be removed from payment pages without a prompt alert being generated.

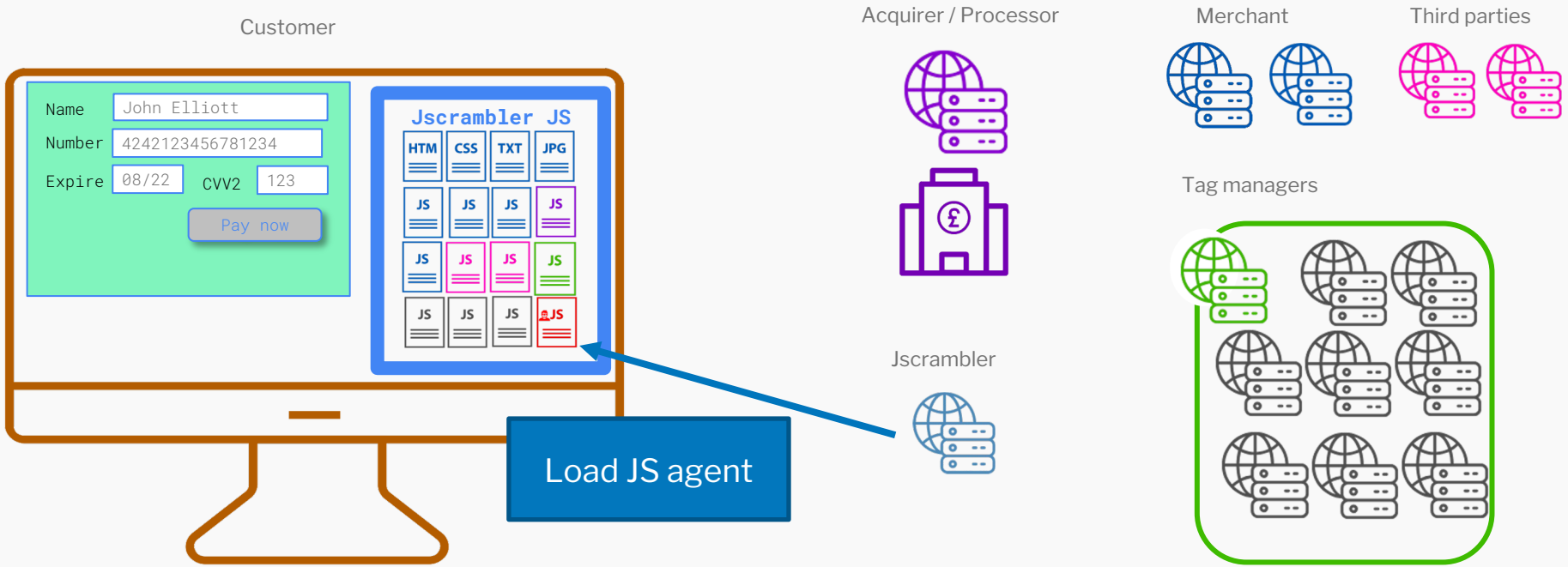
Simply

Continuous assurance of the integrity of JavaScript on the payment page

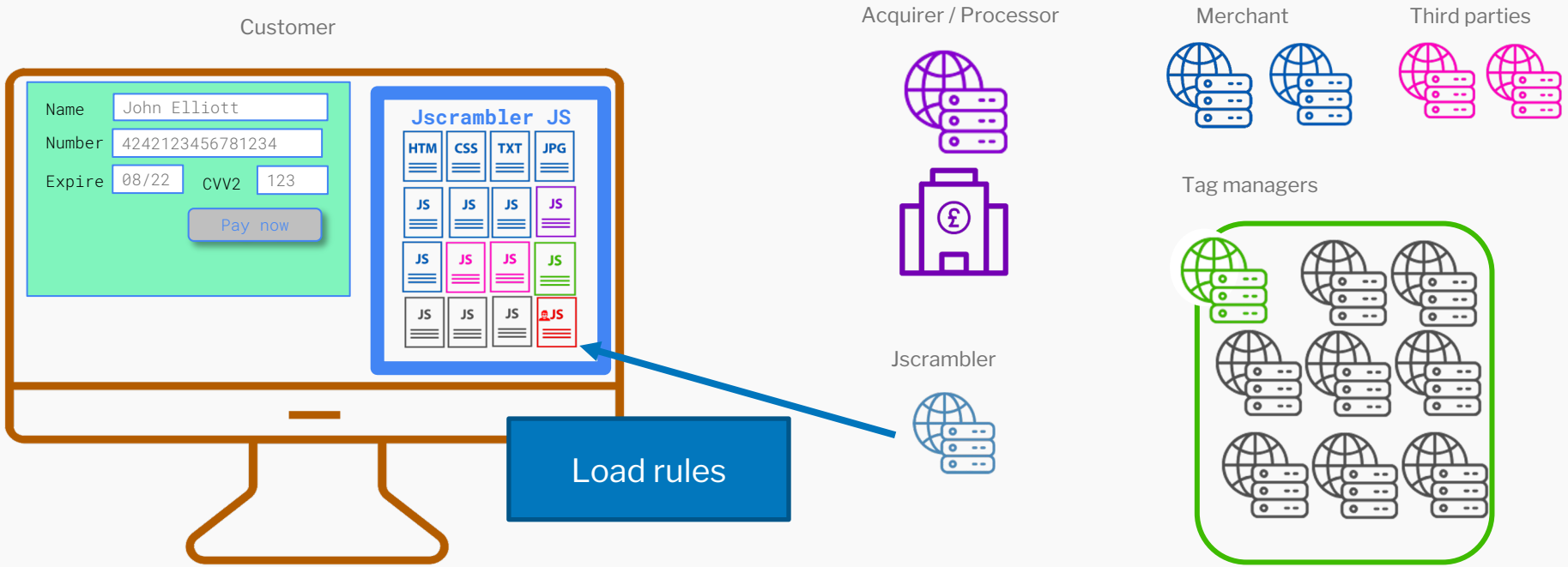
jscrambler

So what does Jscrambler do?

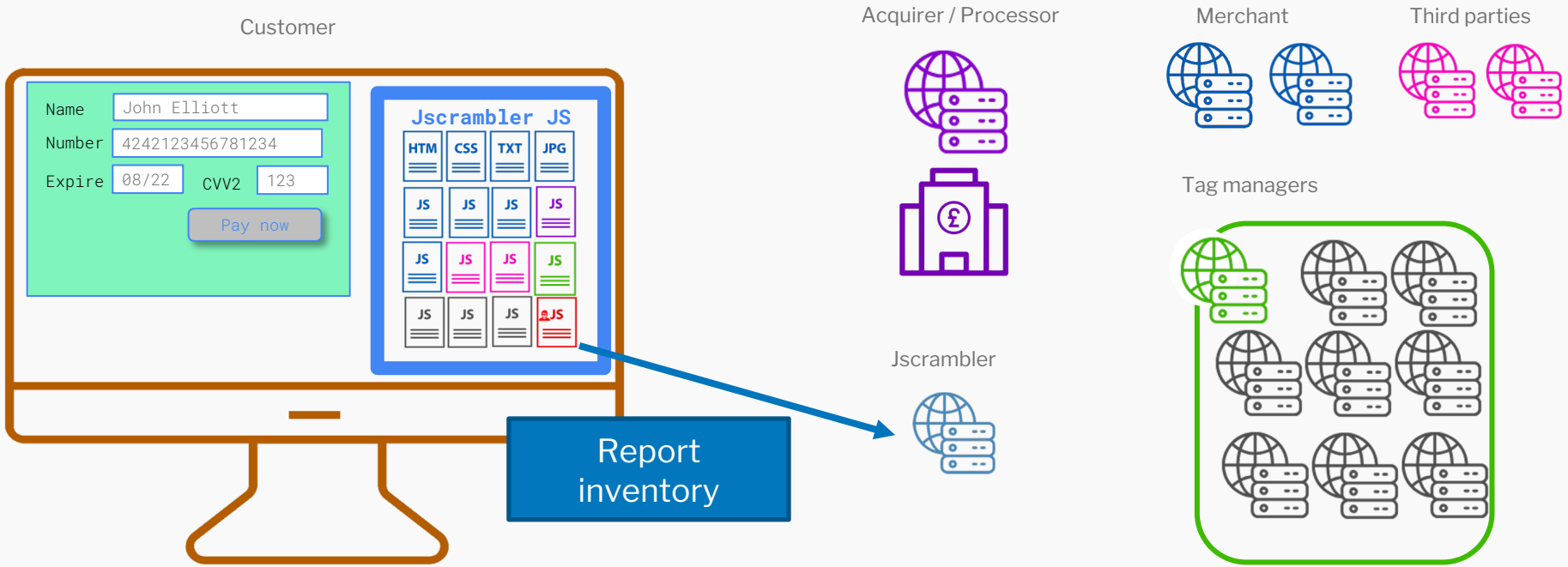
JavaScript isolation



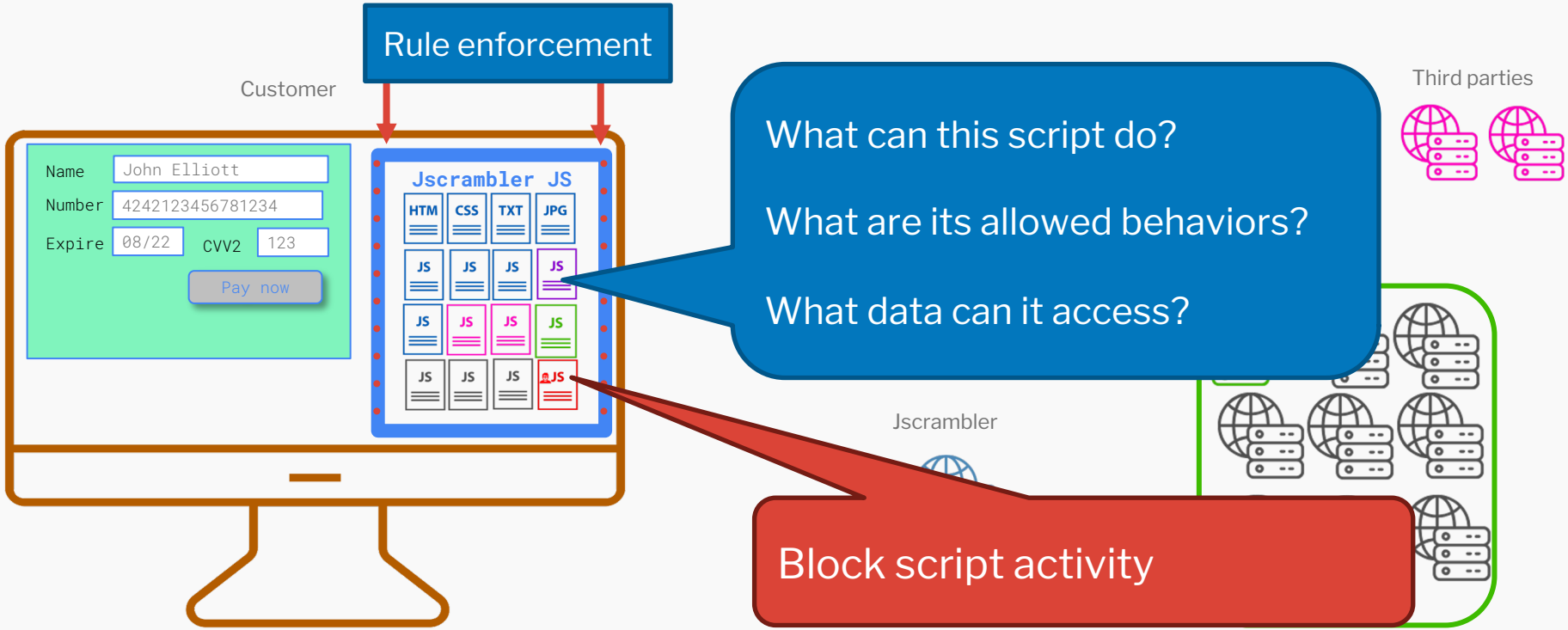
JavaScript isolation



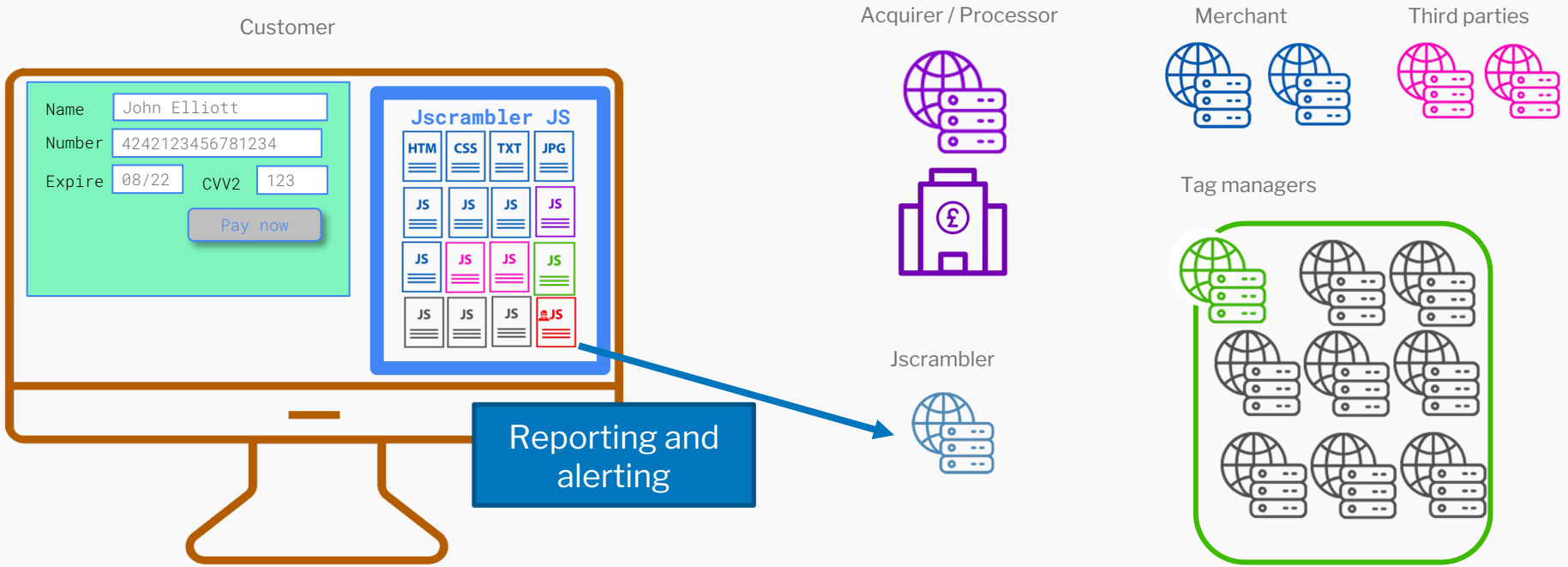
JavaScript isolation



JavaScript isolation



JavaScript isolation

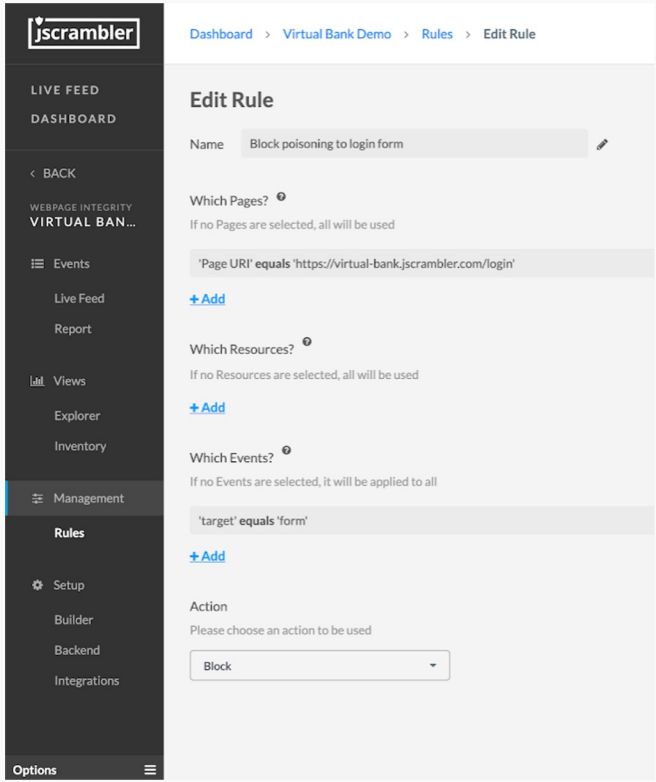


Let’s go Shopping

jscrambler

Webpage Integrity

- Agent (JavaScript) added to your page
- Real-time inventory of scripts
- Script behavior analyzed
- New scripts can be blocked until approved (configurable)
- If a script changes it can be blocked and an alert generated



jscrambler

Code Integrity

- **Polymorphic Obfuscation**
  - Makes reverse-engineering (almost) impossible
  - Guarantees integrity
- **Code Locks**
  - Version and environment control
- **Runtime Protection**
  - Tamper detection and tamper prevention
- **JavaScript Threat Monitoring**
  - Provides alerts on tamper

jscrambler

Q&A

contact@jscrambler.com

john.elliott@jscrambler.com

jscrambler