

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

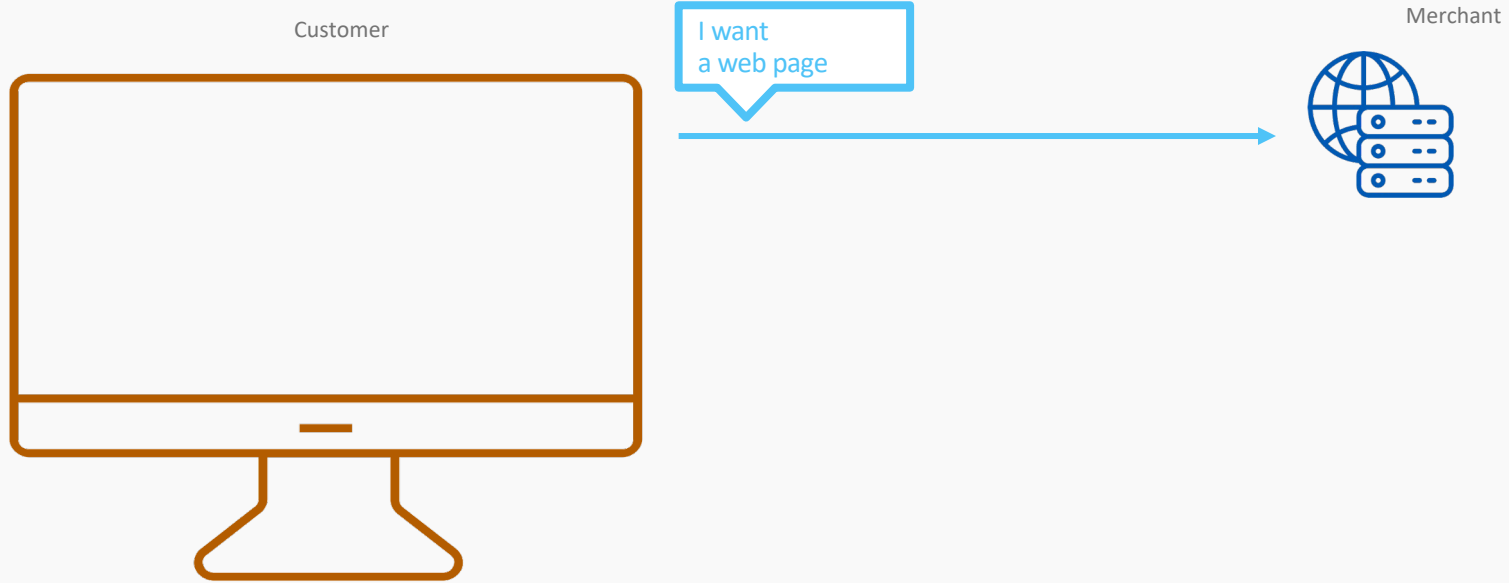
Customer



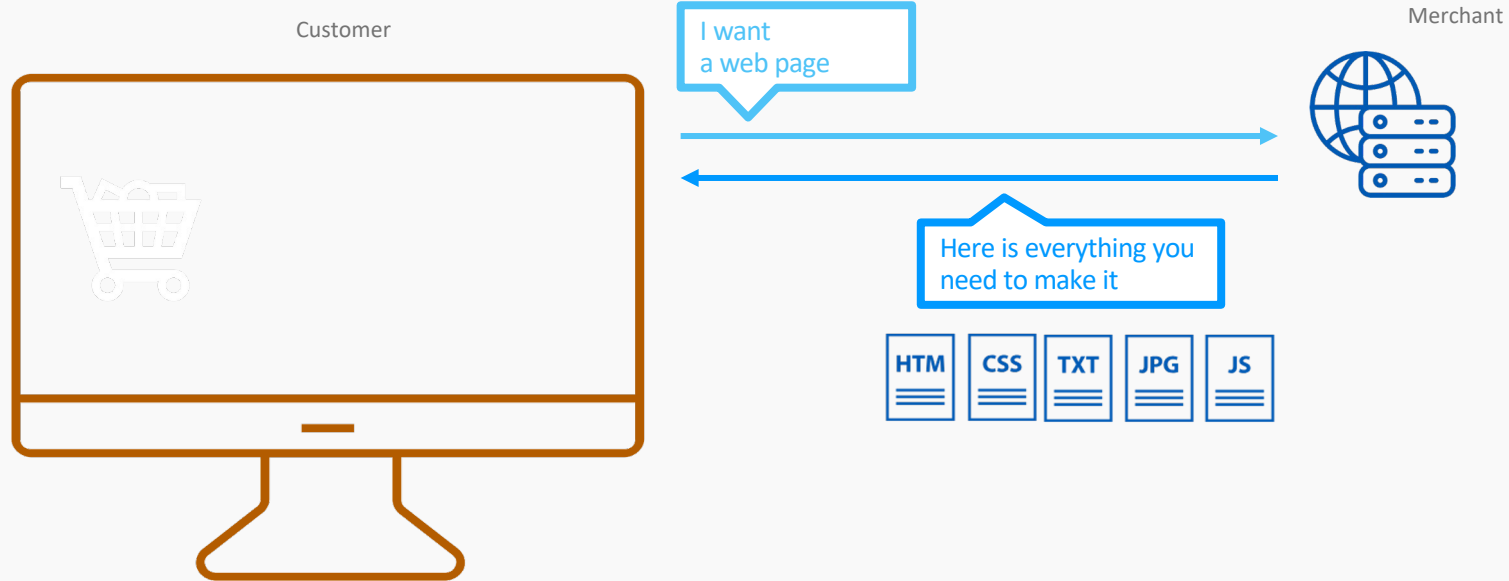
Merchant



How the web works

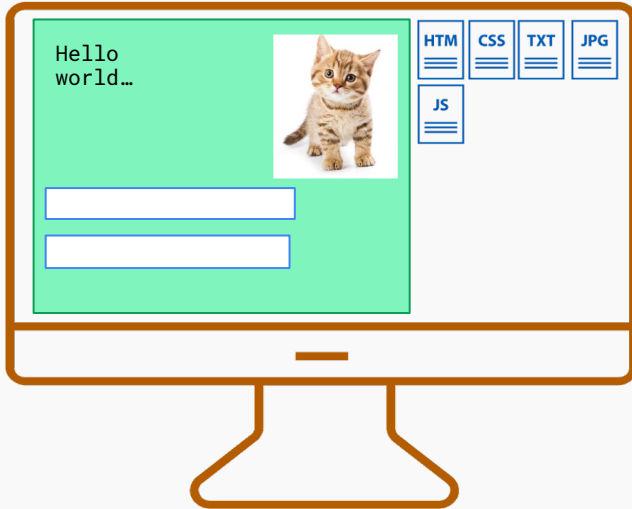


How the web works



How the web works

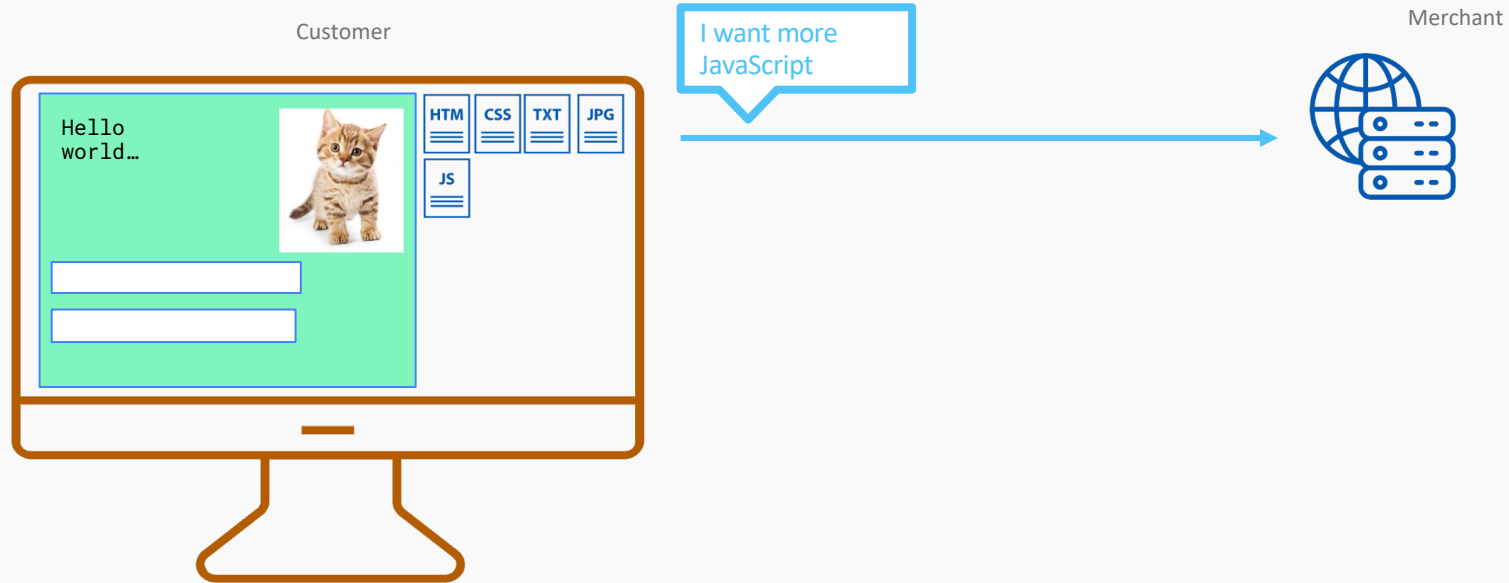
Customer



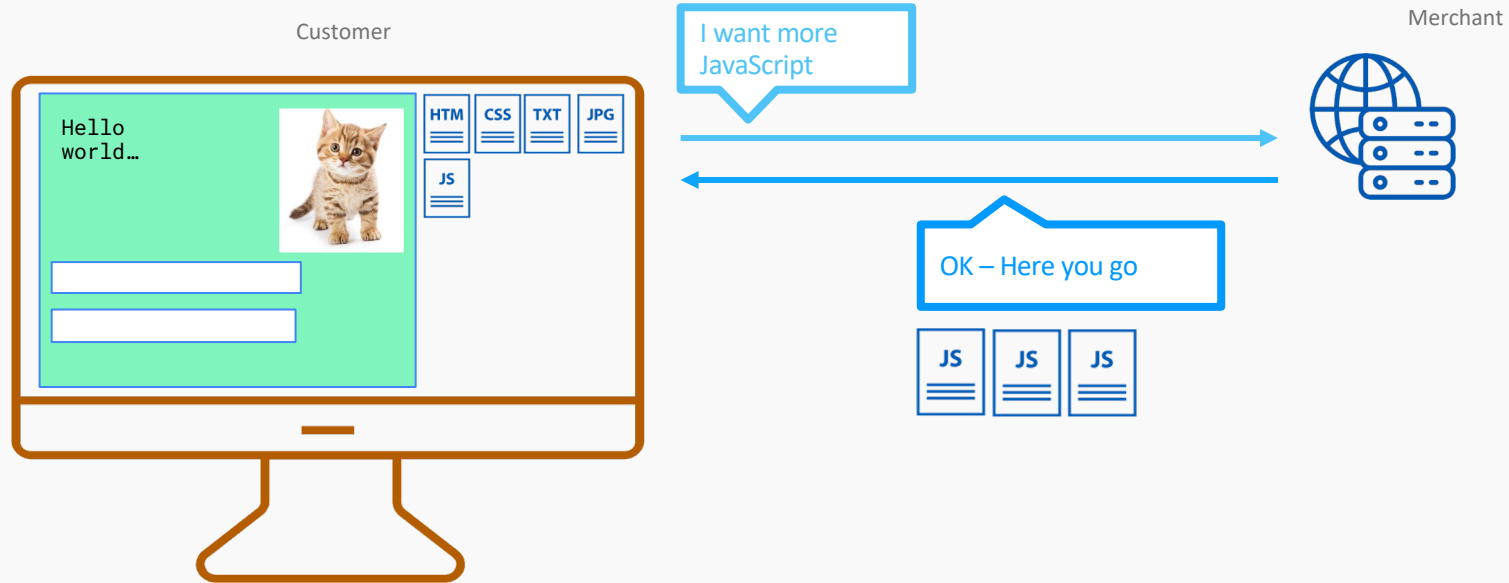
Merchant



How the web works

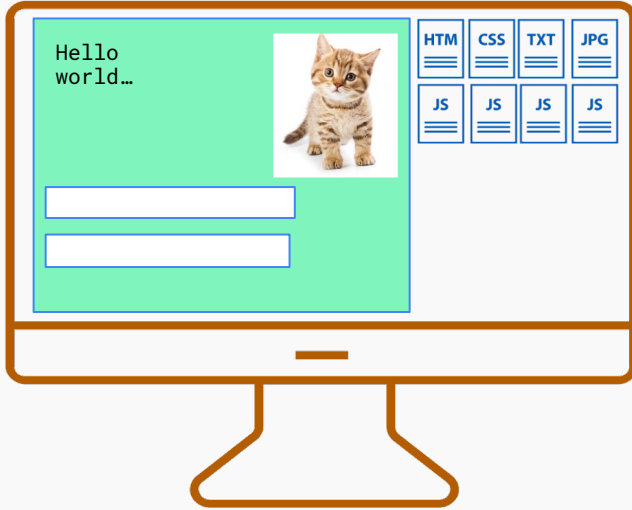


How the web works



How the web works

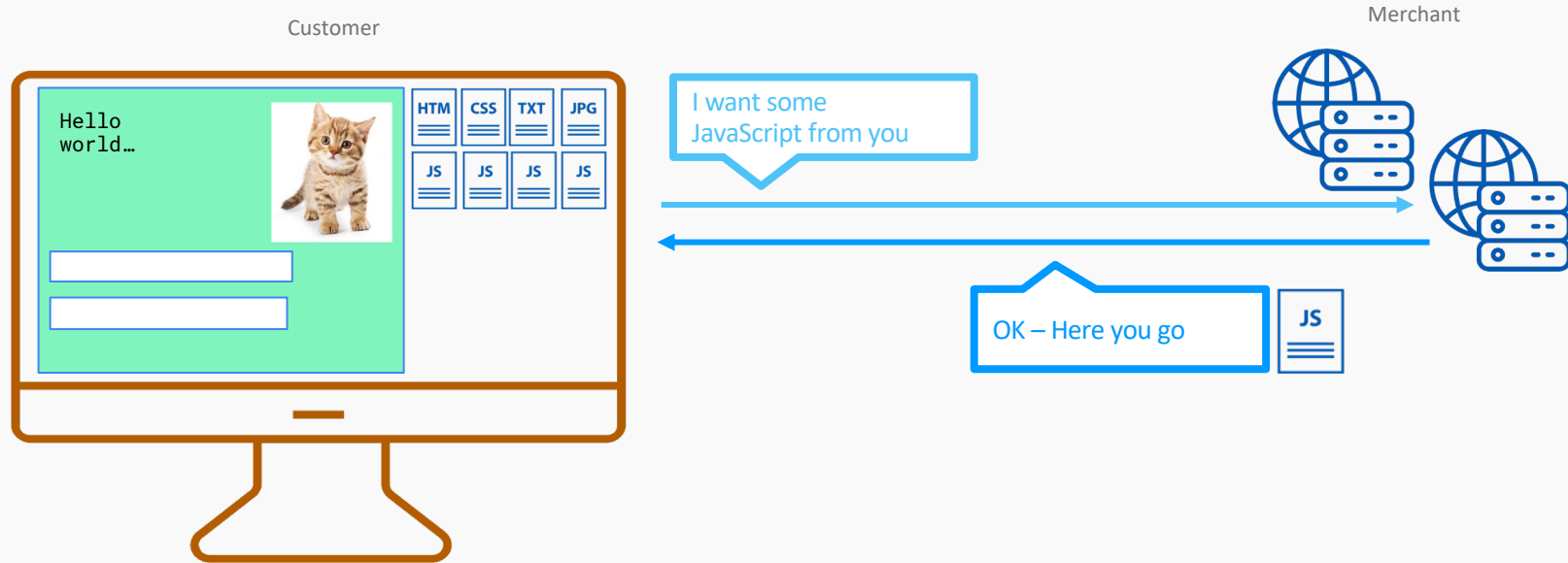
Customer



Merchant

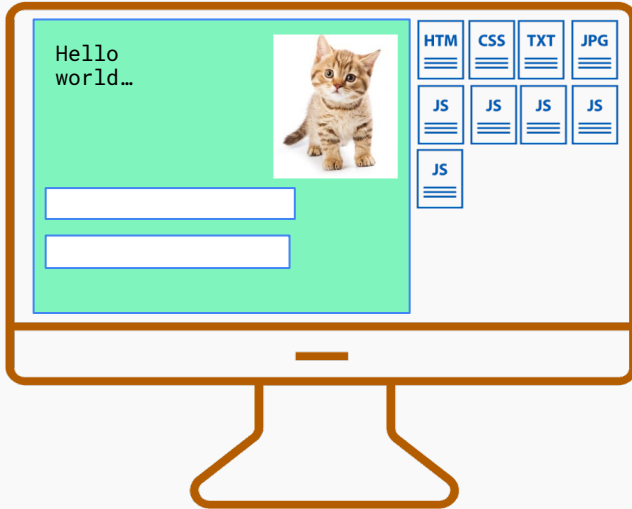


How the web works



How the web works

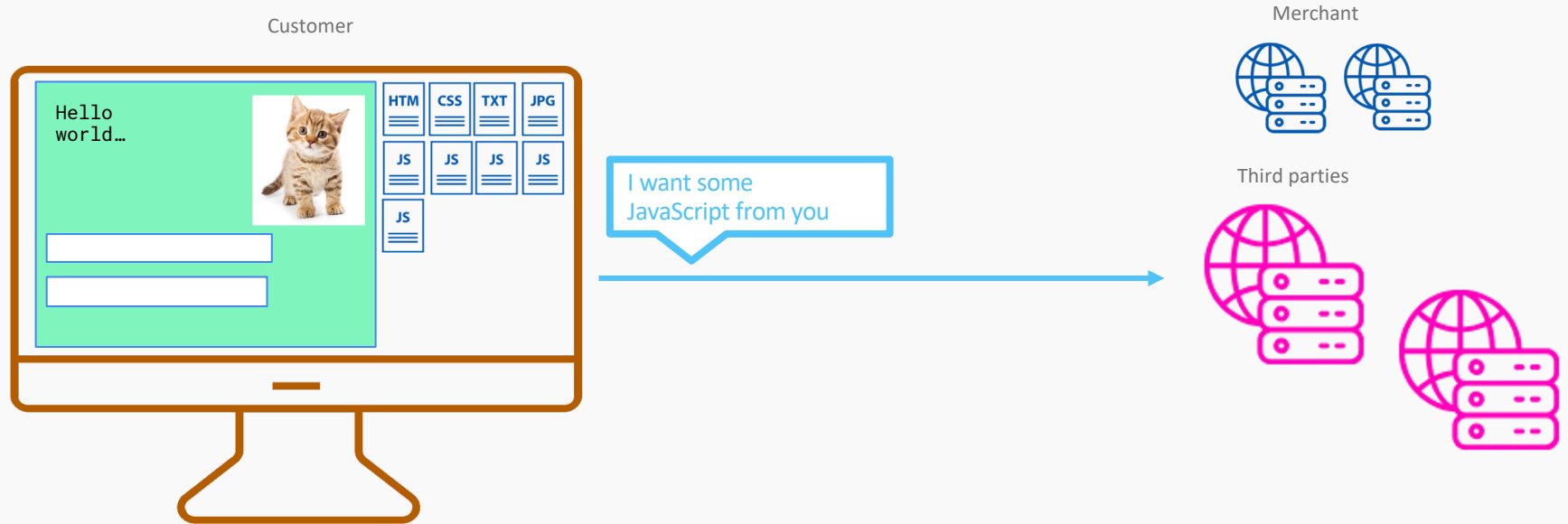
Customer



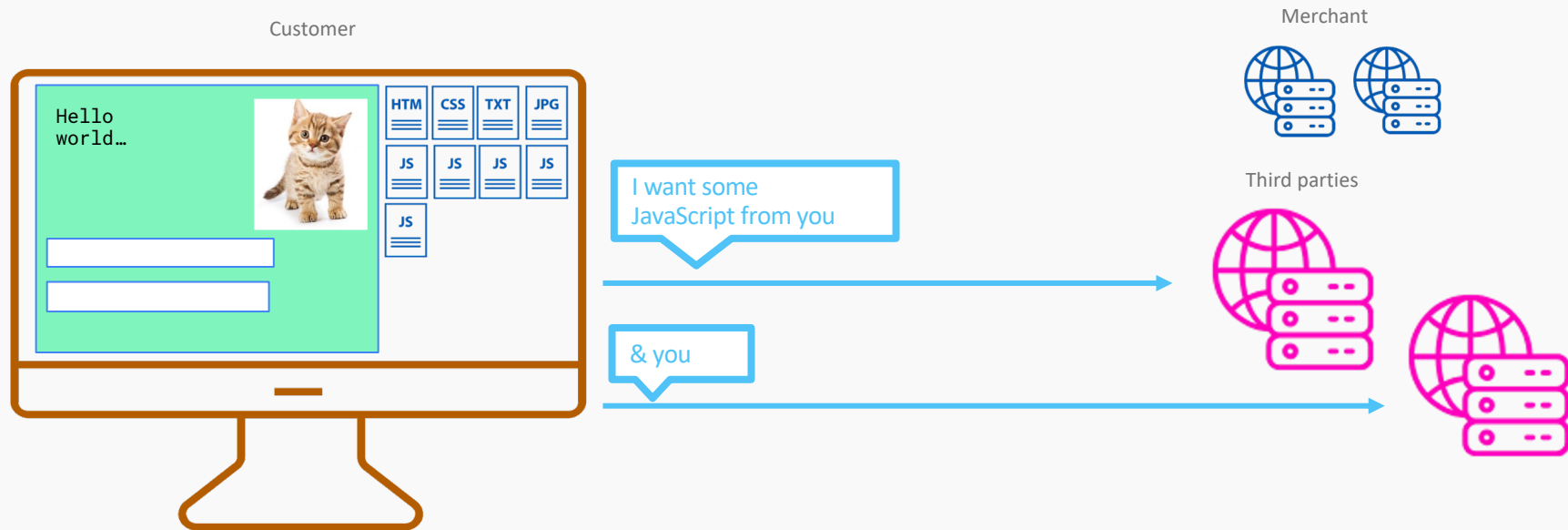
Merchant



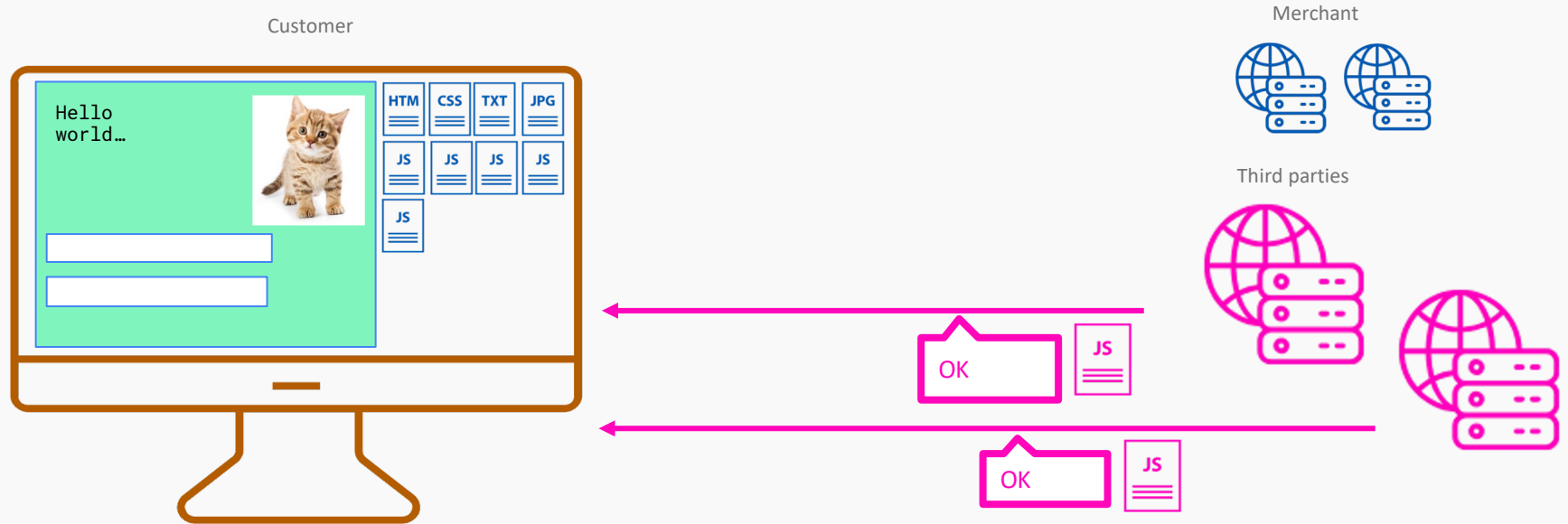
How the web works



How the web works

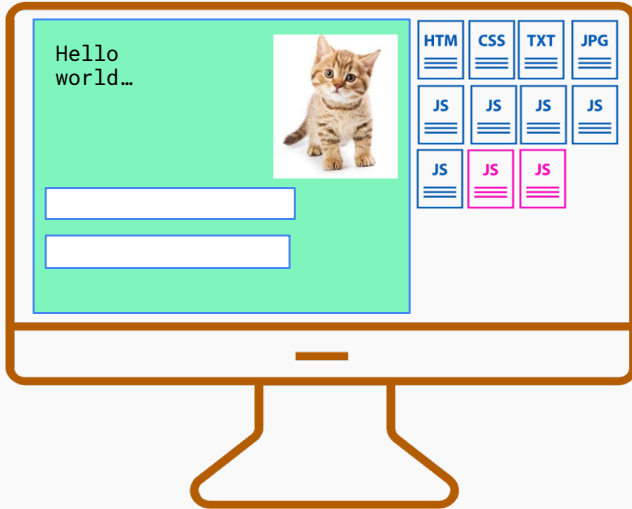


How the web works



How the web works

Customer



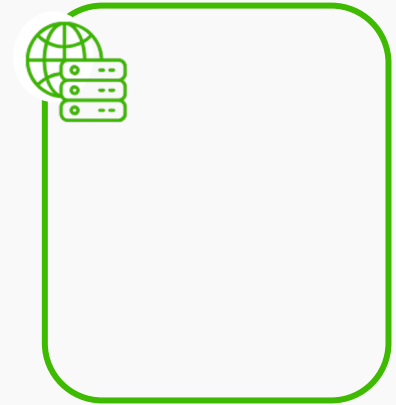
Merchant



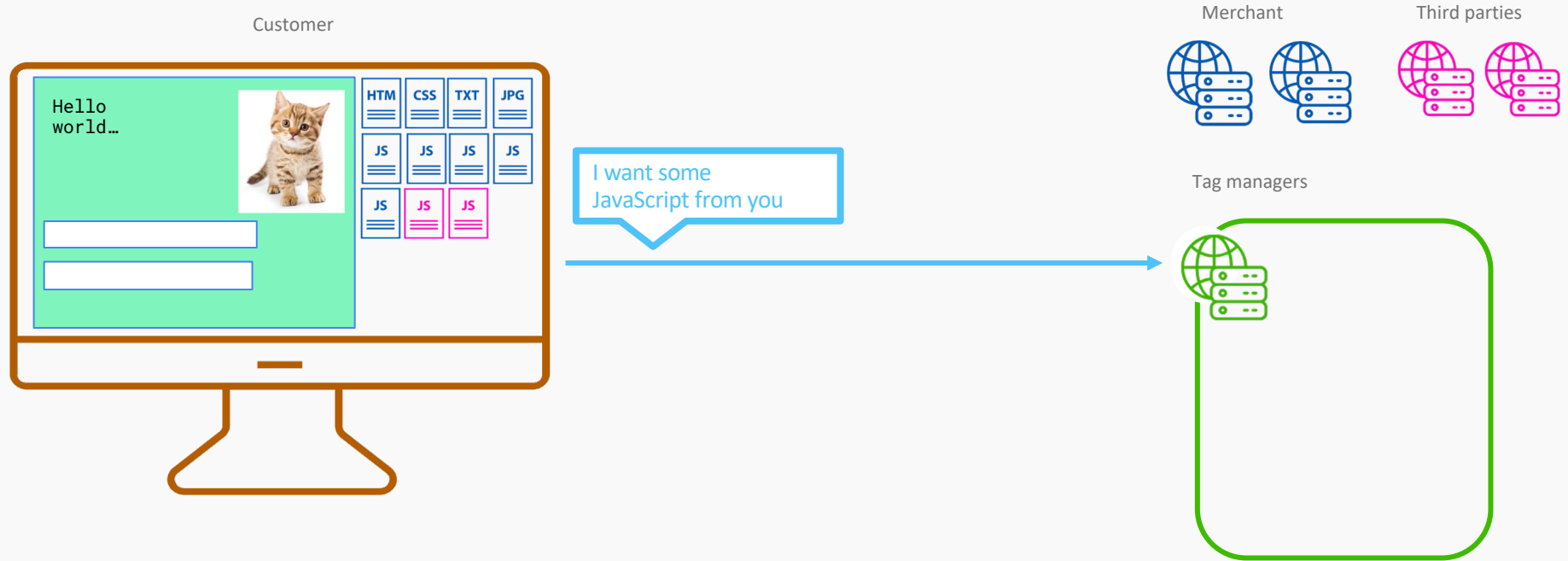
Third parties



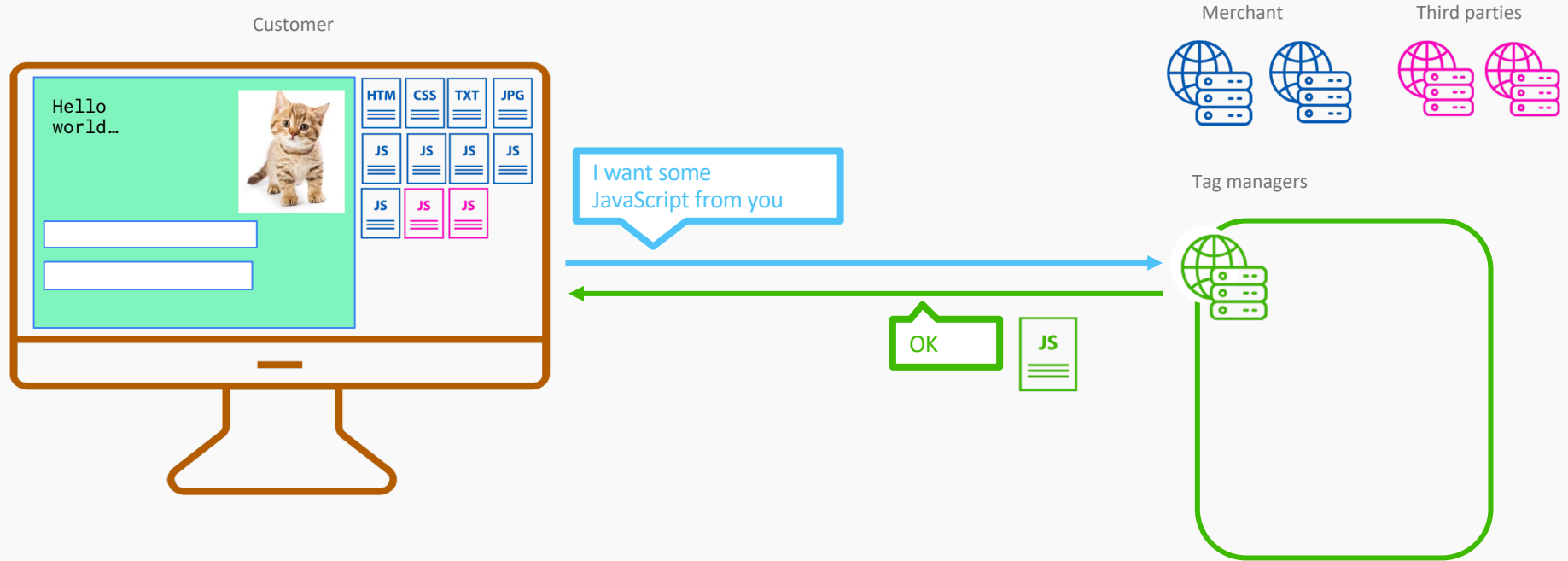
Tag managers



How the web works

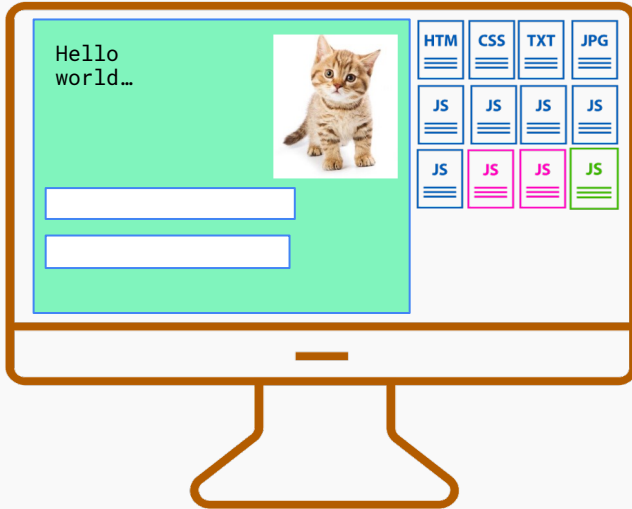


How the web works



How the web works

Customer



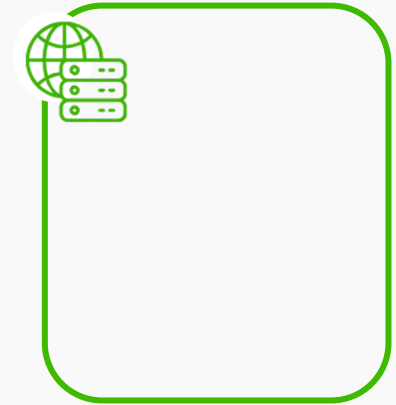
Merchant



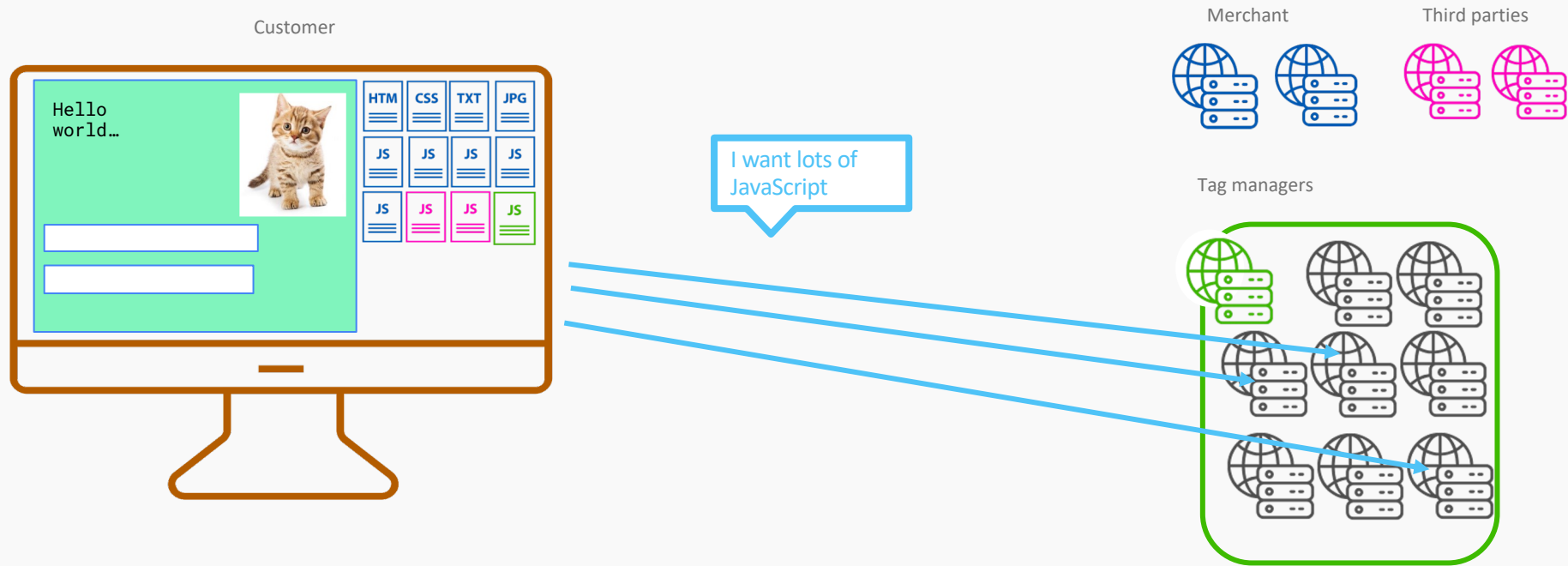
Third parties



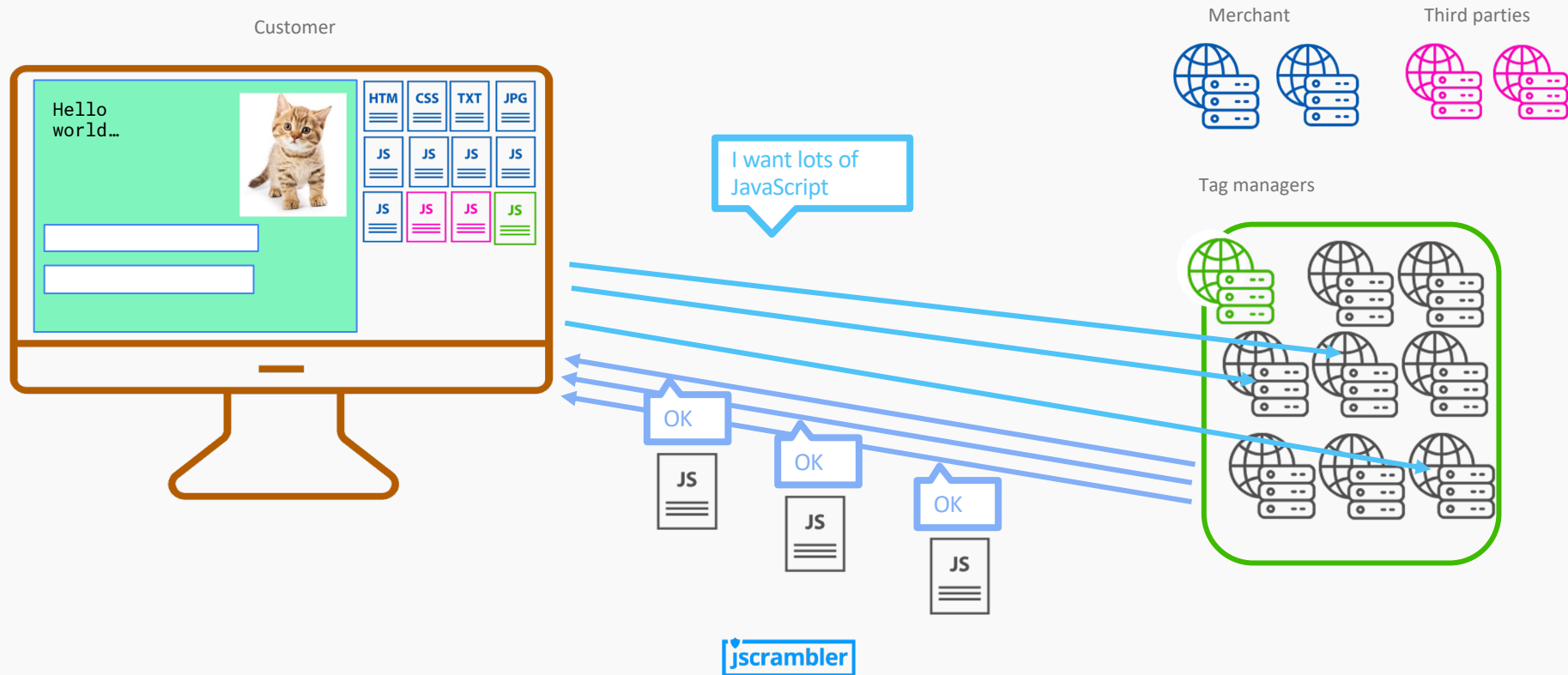
Tag managers



How the web works

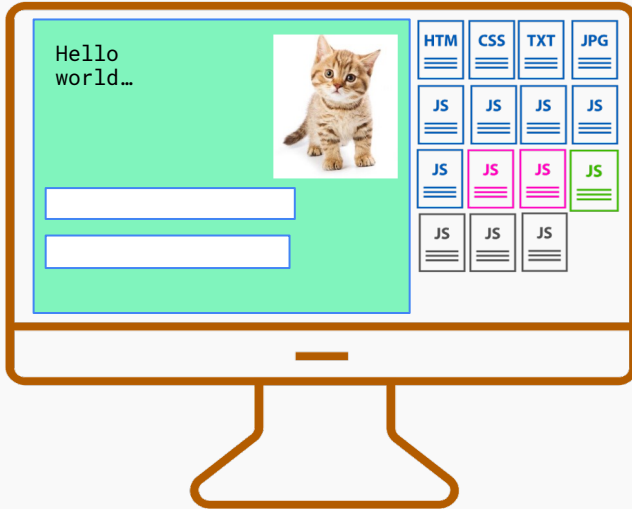


How the web works



How the web works

Customer



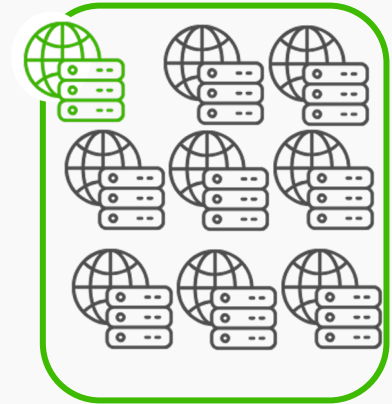
Merchant



Third parties

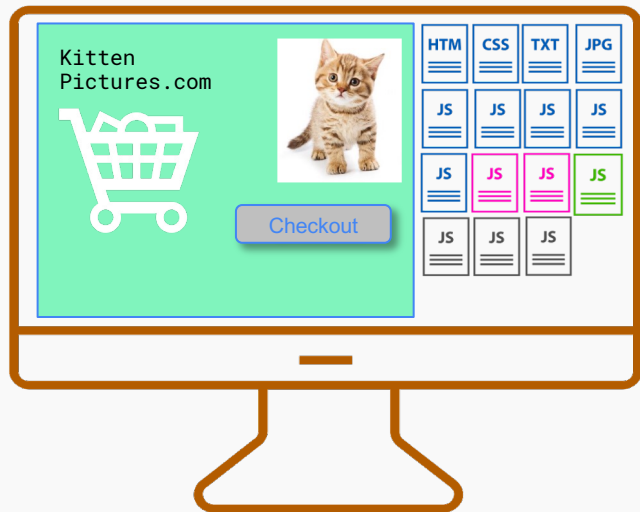


Tag managers



But this is an ecommerce site

Customer



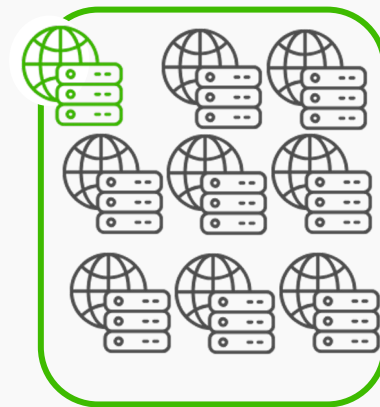
Merchant



Third parties



Tag managers



But this is now a payment form!

Customer

Name	<input type="text"/>	HTM	CSS	TXT	JPG
Number	<input type="text"/>	JS	JS	JS	JS
Expire	<input type="text"/>	JS	JS	JS	JS
CVV2	<input type="text"/>	JS	JS	JS	JS
<input type="button" value="Pay now"/>		JS	JS	JS	JS

Acquirer / Processor



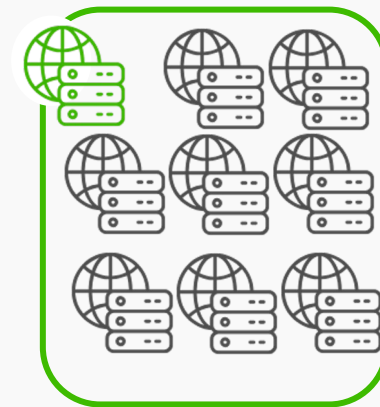
Merchant



Third parties



Tag managers



But this is now a payment form!

Customer

Name	<input type="text" value="John Elliott"/>	HTM	CSS	TXT	JPG
Number	<input type="text" value="4242123456781234"/>	JS	JS	JS	JS
Expire	<input type="text" value="08/22"/>	JS	JS	JS	JS
CVV2	<input type="text" value="123"/>	JS	JS	JS	JS
<input type="button" value="Pay now"/>		JS	JS	JS	

Acquirer / Processor



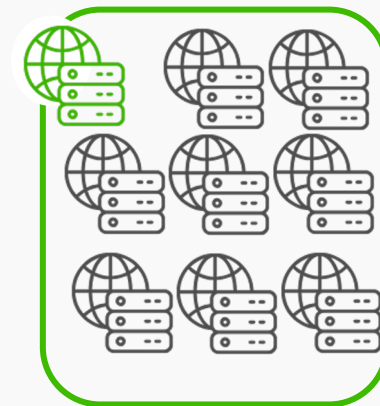
Merchant



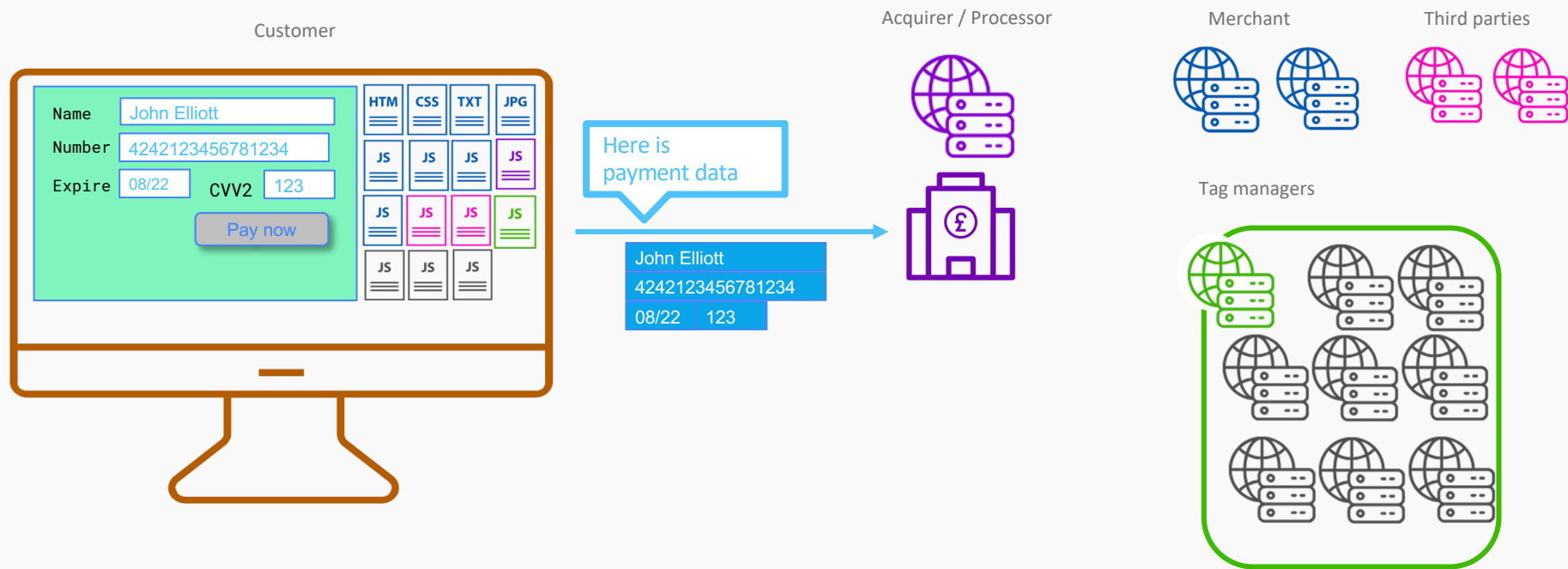
Third parties



Tag managers



But this is now a payment form!



But this is now a payment form!

Customer

Name	<input type="text" value="John Elliott"/>	HTM	CSS	TXT	JPG
Number	<input type="text" value="4242123456781234"/>	JS	JS	JS	JS
Expire	<input type="text" value="08/22"/>	JS	JS	JS	JS
	CVV2 <input type="text" value="123"/>	JS	JS	JS	JS
	<input type="button" value="Pay now"/>	JS	JS	JS	

Acquirer / Processor



Merchant



Third parties

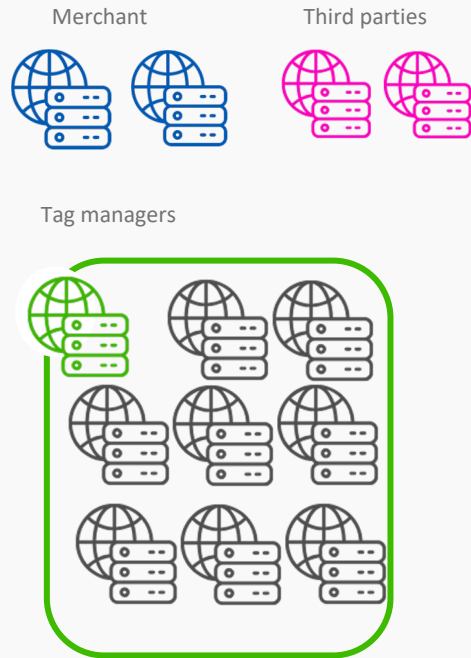
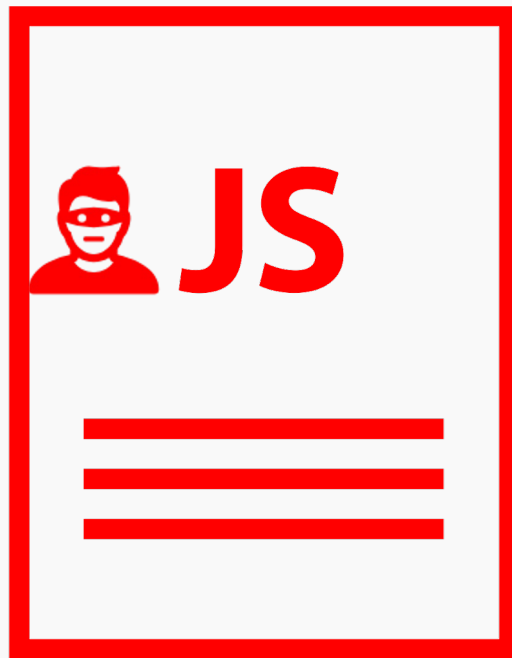
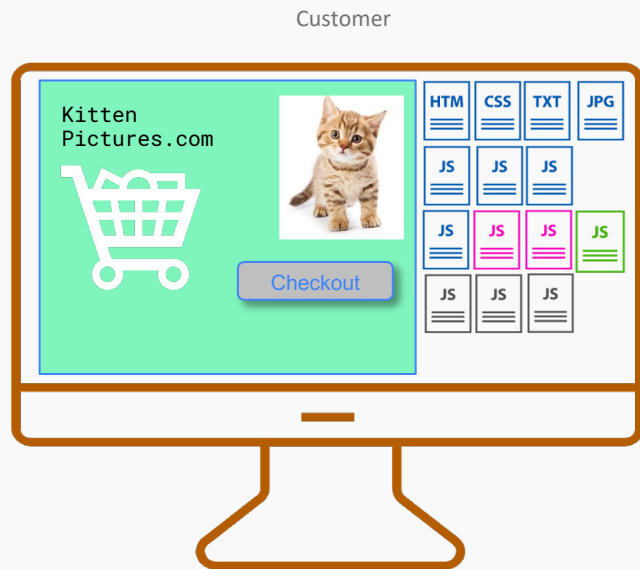


Tag managers

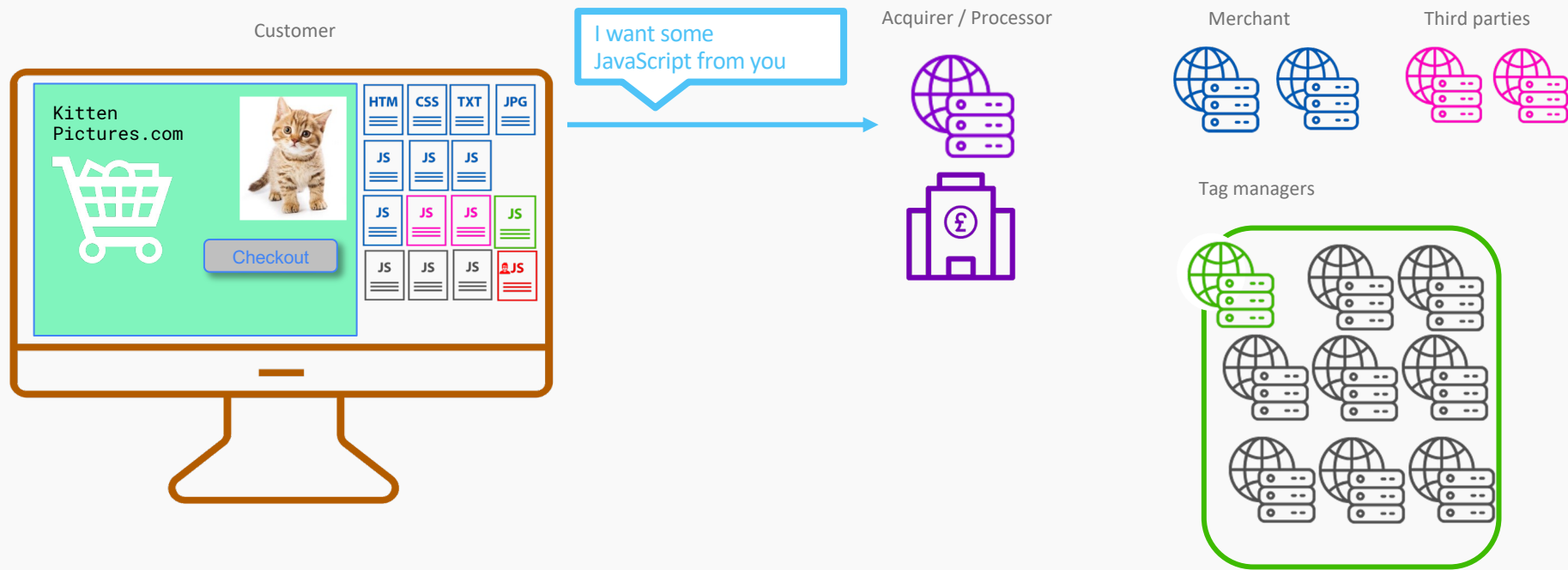


The attack explained

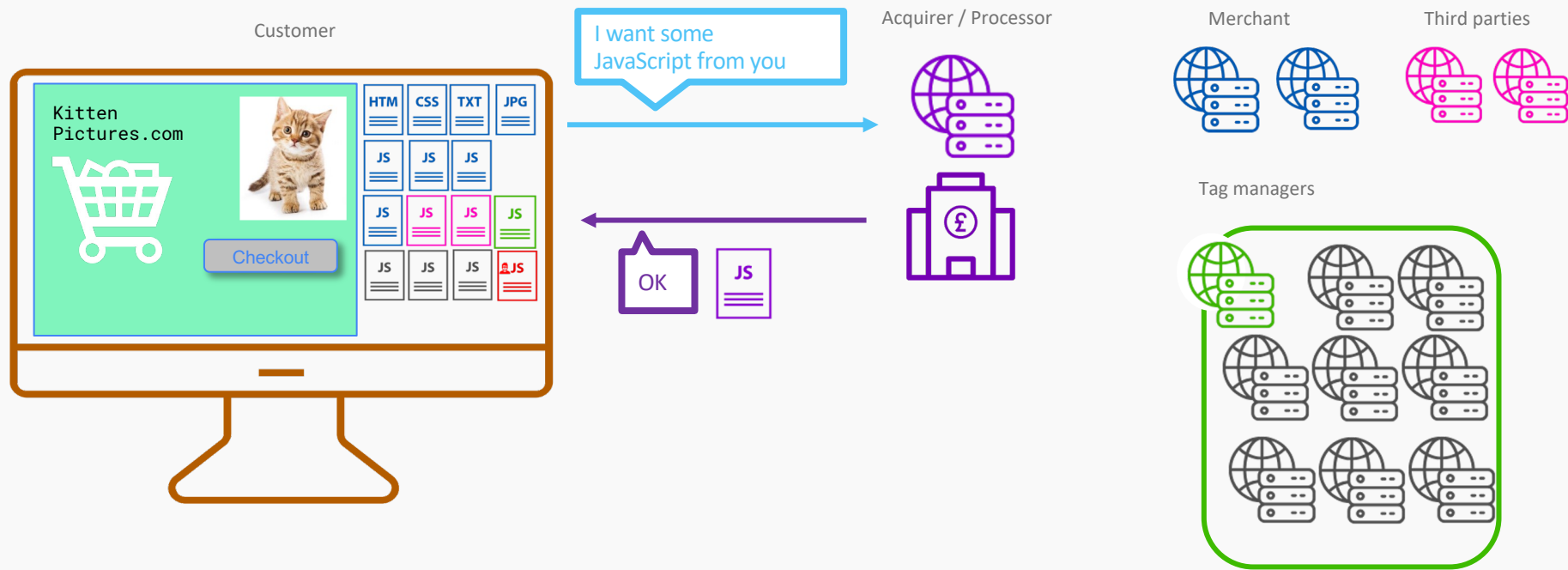
The attack



The attack

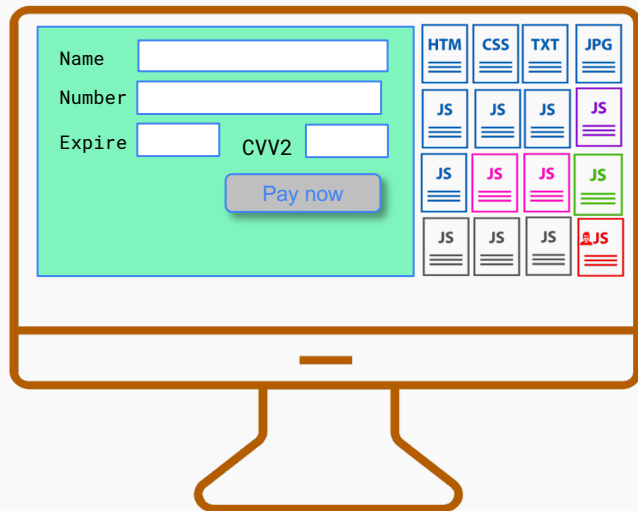


The attack



The attack

Customer



Acquirer / Processor



Merchant



Third parties

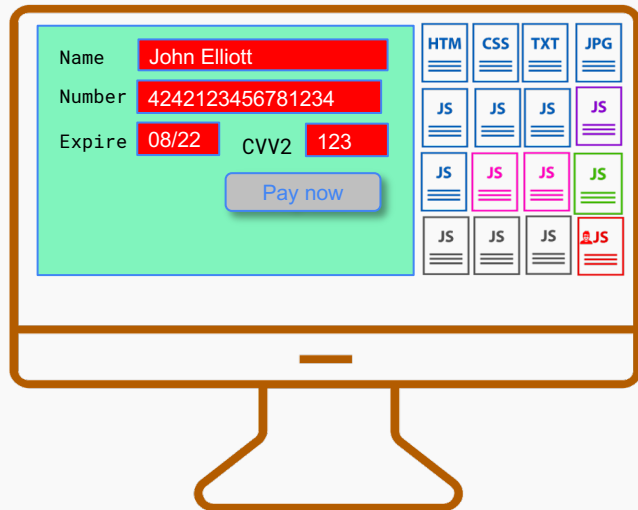


Tag managers



The attack

Customer



Acquirer / Processor



Merchant



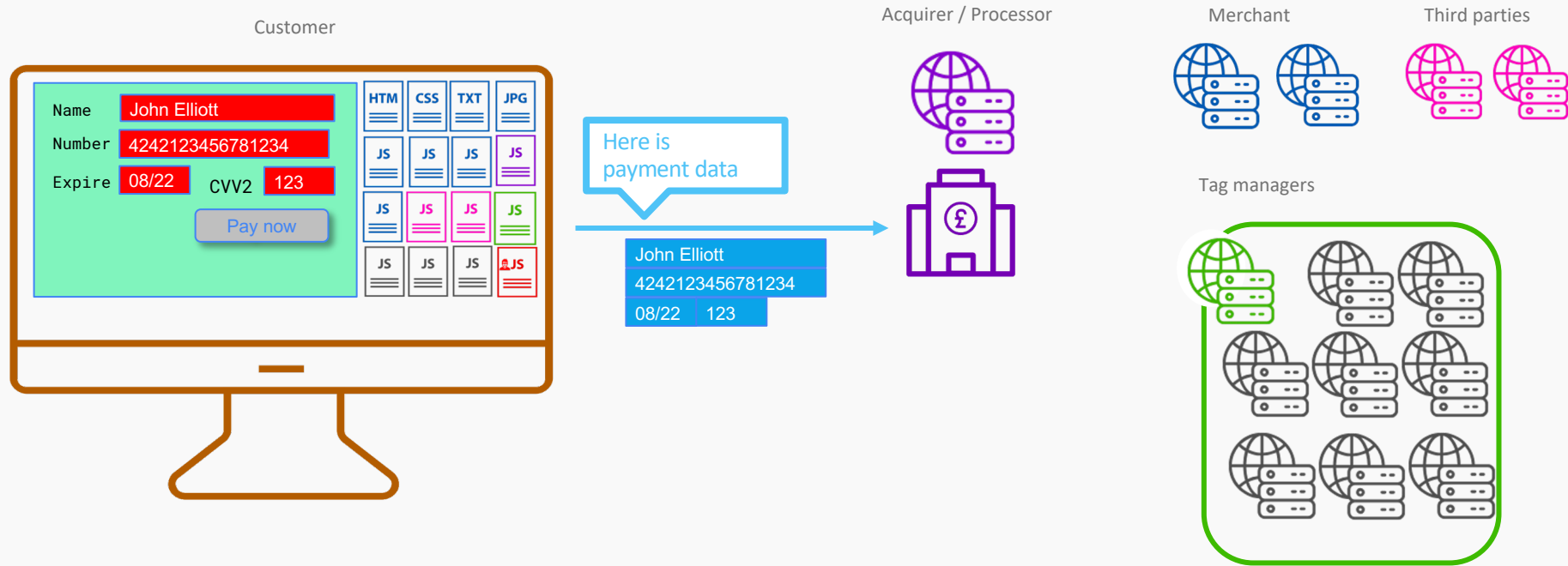
Third parties



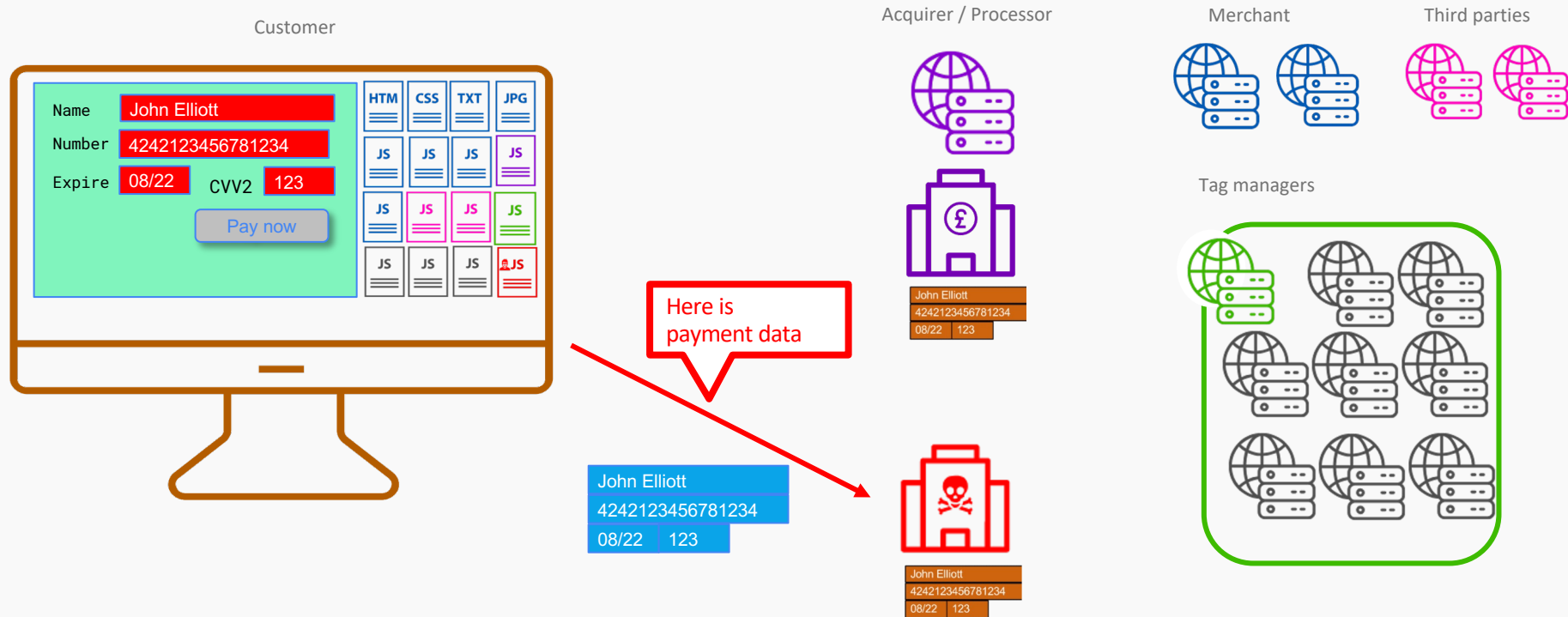
Tag managers



The attack

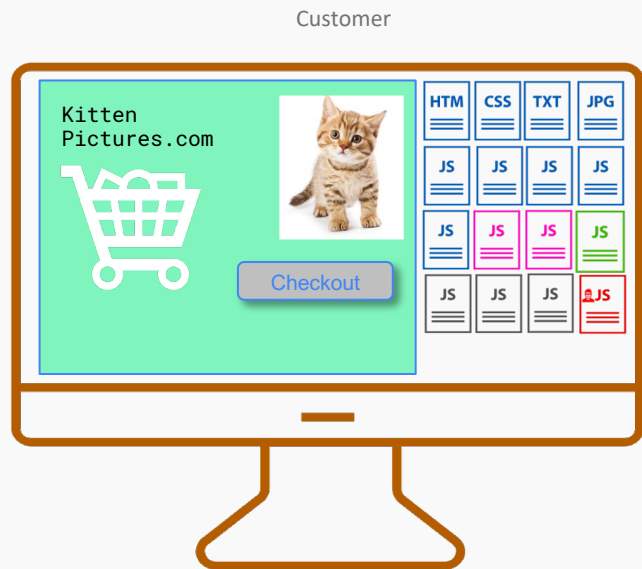


The attack

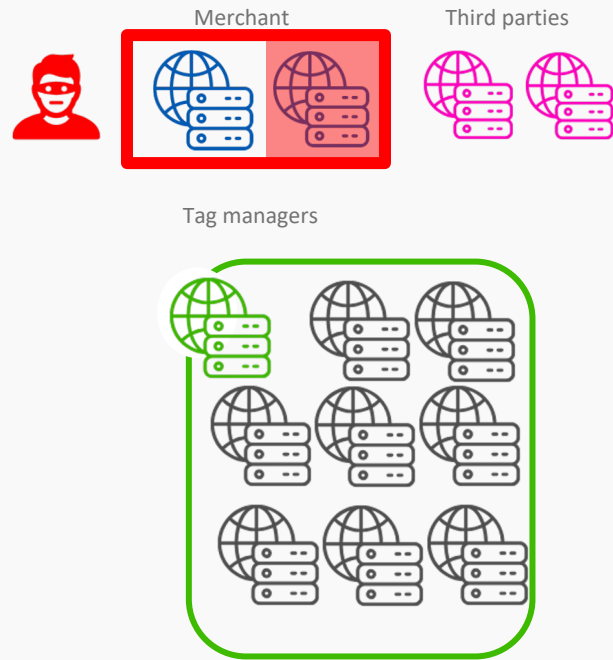


The attack surface

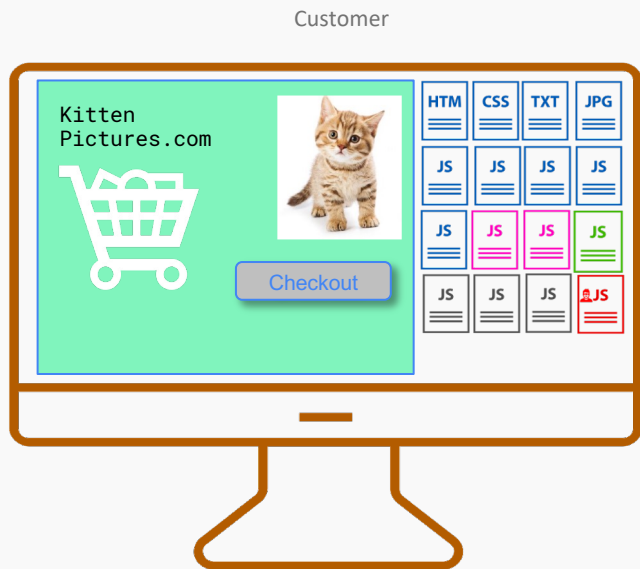
The attack surface



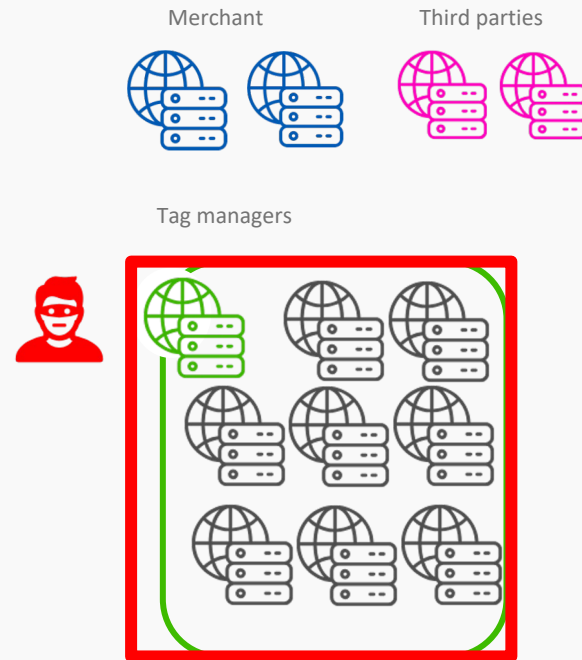
JS in the BA breach
was loaded from a
secondary web
server



The attack surface

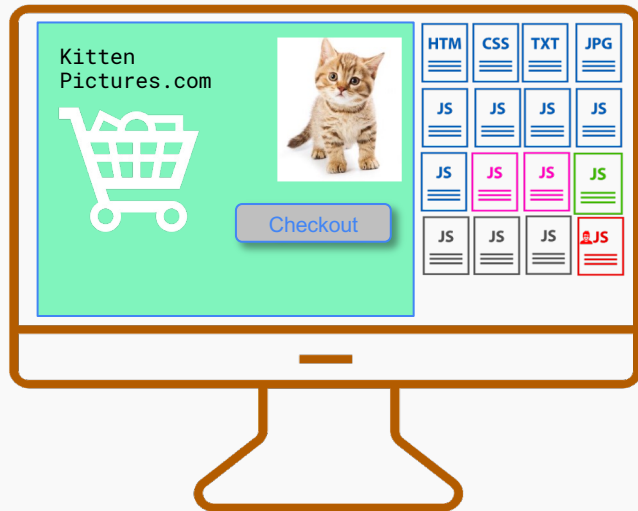


Any servers specified in the tag manager



The attack surface

Customer



Adverline breach
277 sites

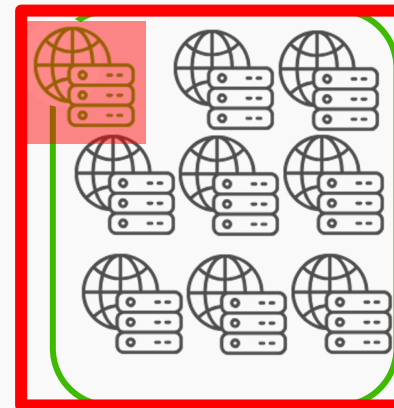
Merchant



Third parties



Tag managers



How to stop attacks

Script Integrity

When the
script is added
to the page

If (when) it
changes

When a script changes

appears!

- **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

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

Where to monitor changes?

Customer

HTM	CSS	TXT	JPG
JS	JS	JS	JS
JS	JS	JS	JS
JS	JS	JS	JS

Acquirer / Processor



Merchant



Third parties



Tag managers



This is the only place to detect changes

The new requirements in PCI DSS 4.0

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.

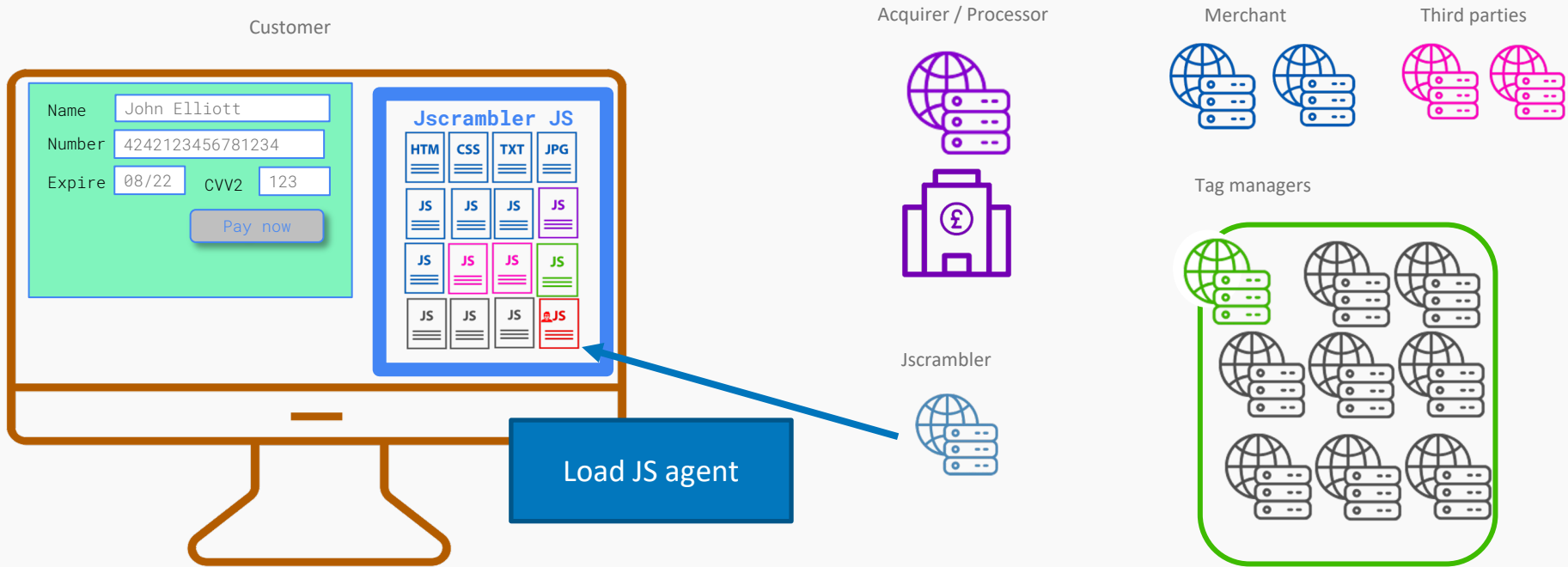
Anti-skimming measures cannot be removed from payment pages without a prompt alert being generated.

Simply

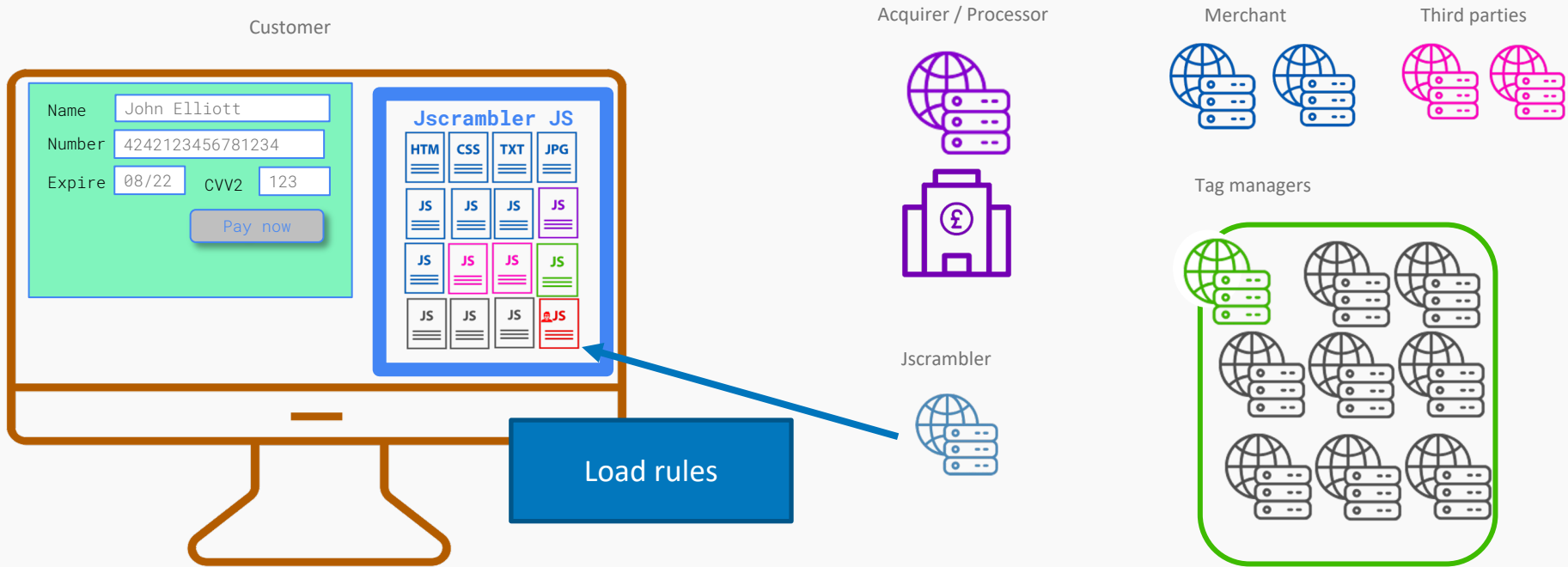
Continuous assurance of the
integrity of JavaScript on the
payment page

**So what does
Jscrambler do?**

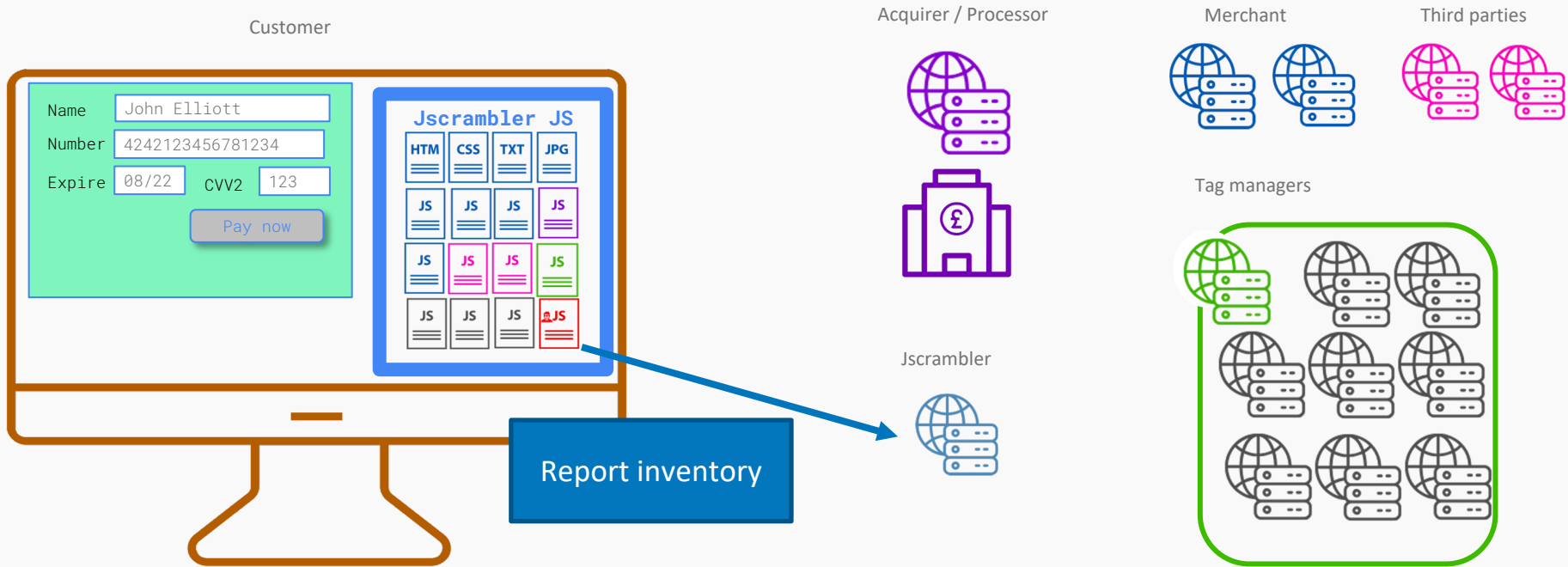
JavaScript isolation



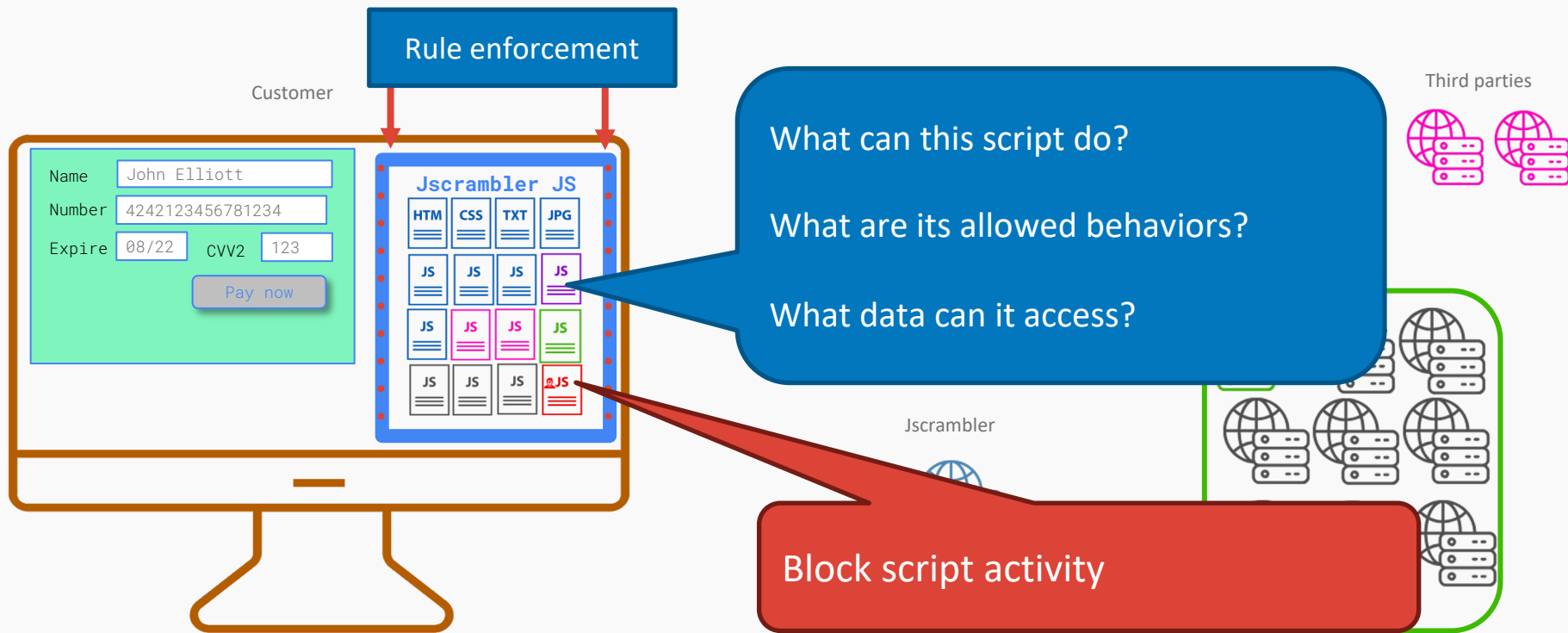
JavaScript isolation



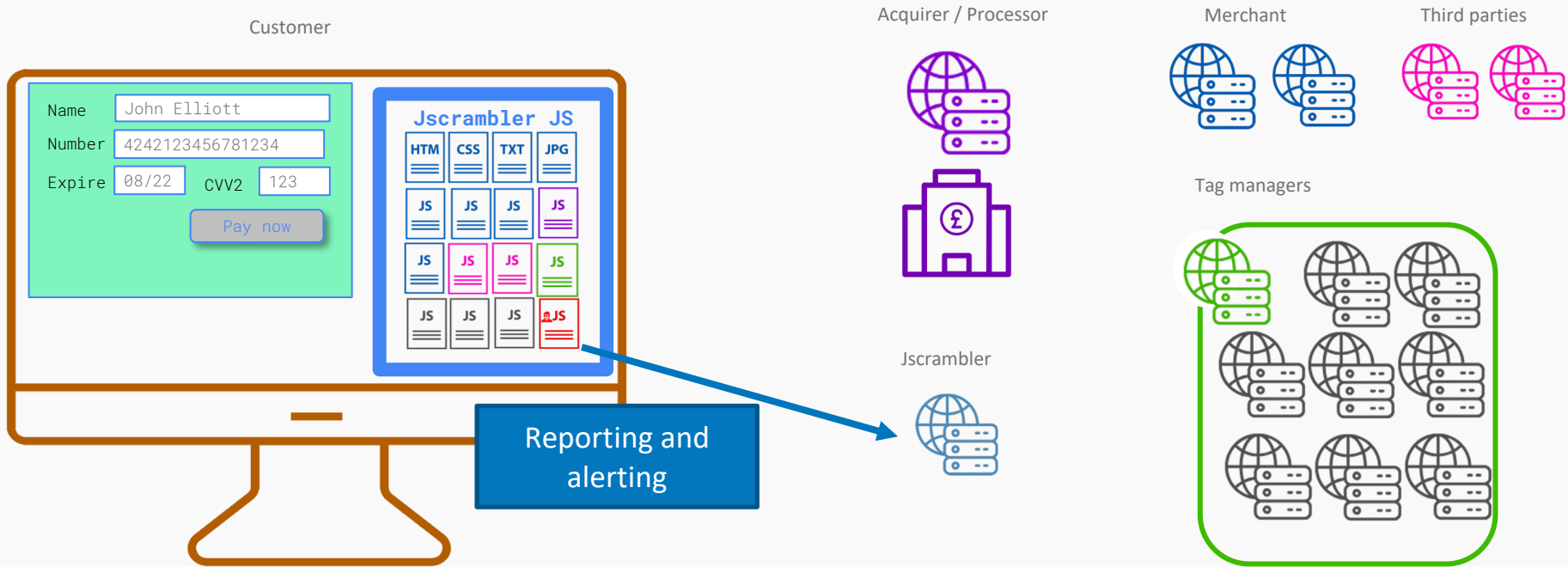
JavaScript isolation



JavaScript isolation



JavaScript isolation



Let's go Shopping

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

The screenshot displays the jsCrambler dashboard interface. The top navigation bar includes the jsCrambler logo and a breadcrumb trail: Dashboard > Virtual Bank Demo > Rules > Edit Rule. The left sidebar menu is dark-themed and lists various sections: LIVE FEED, DASHBOARD, WEBPAGE INTEGRITY (VIRTUAL BAN...), Events, Live Feed, Report, Views, Explorer, Inventory, Management (highlighted), Rules, Setup, Builder, Backend, and Integrations. The main content area is titled 'Edit Rule' and shows the configuration for a rule named 'Block poisoning to login form'. It includes sections for 'Which Pages?' (with a rule: 'Page URI' equals 'https://virtual-bank-jscrambler.com/login'), 'Which Resources?' (with a rule: 'target' equals 'form'), and 'Which Events?'. The 'Action' dropdown is set to 'Block'. The bottom of the sidebar shows 'Options' and a hamburger menu icon.

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

Q&A

contact@jscrambler.com

john.elliott@jscrambler.com