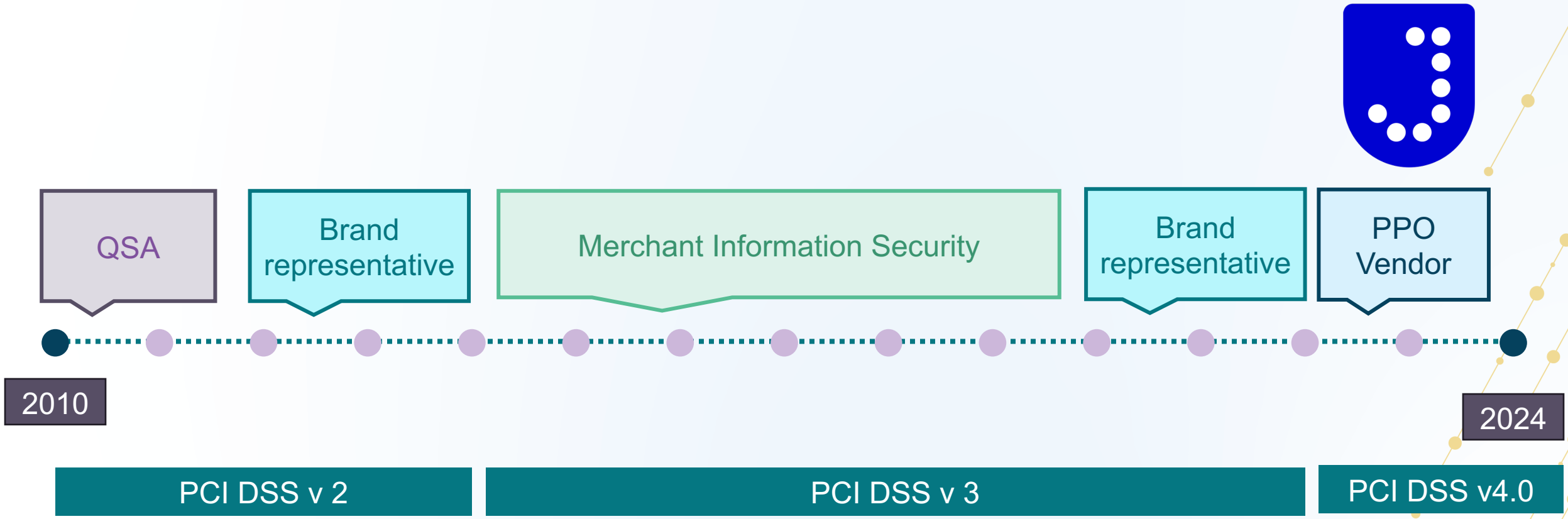
John Elliott | Security Advisor @ Jscrambler

The Challenge of Managing e-Commerce JavaScript

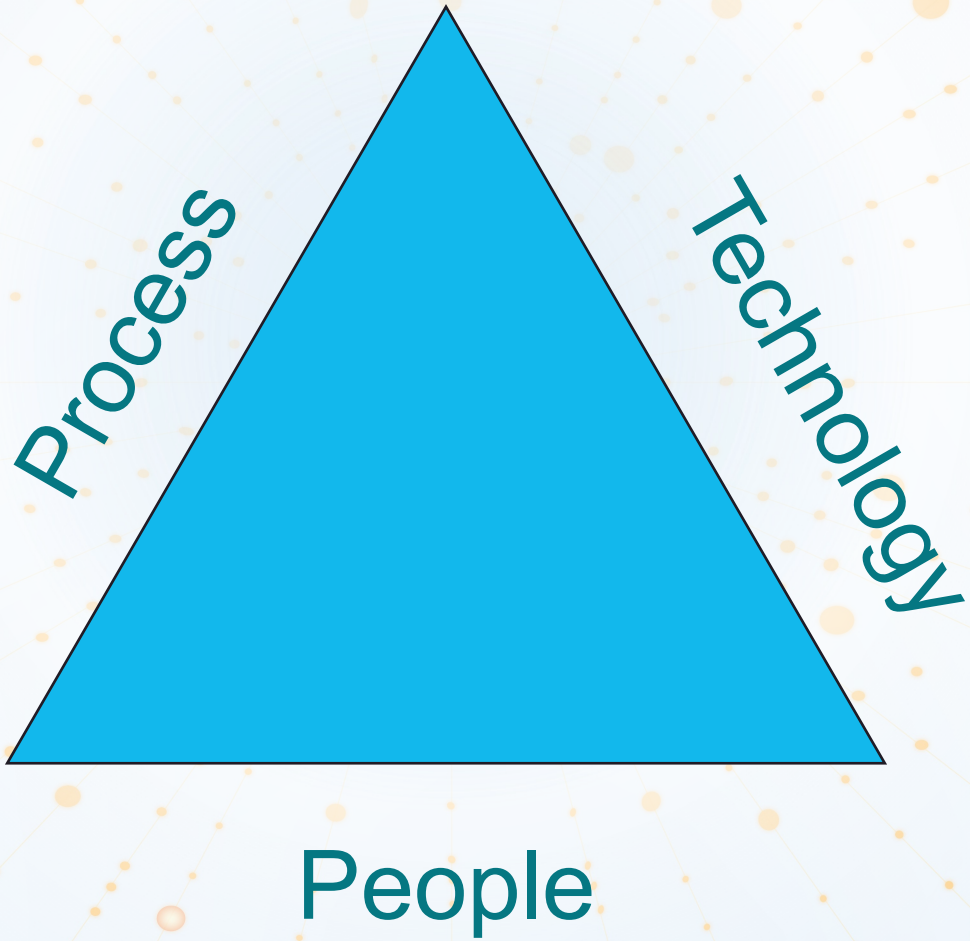
2v0

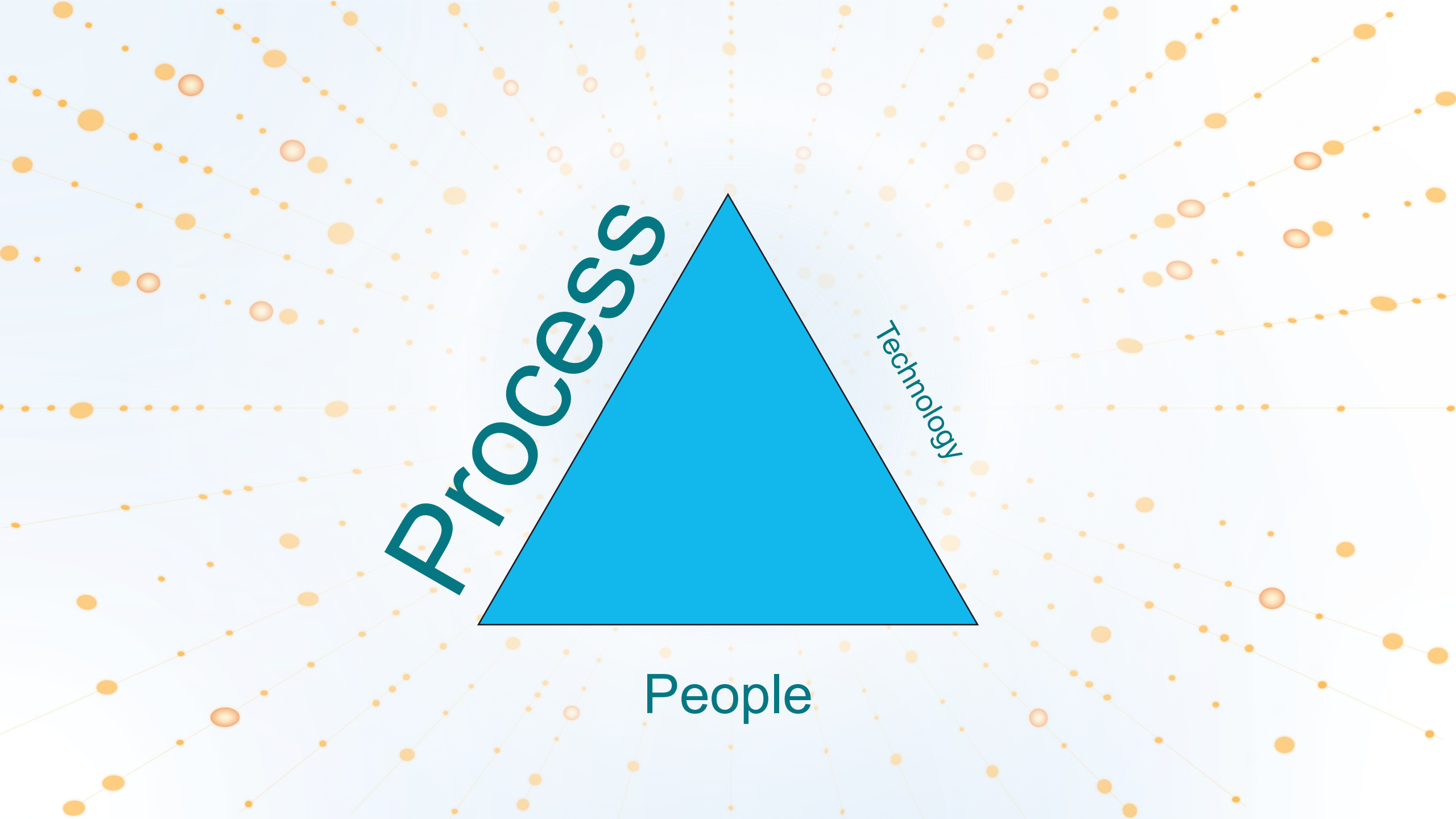
About Me


QSA
Brand representative
Merchant Information Security
Brand representative
PPO Vendor
2010
2024
PCI DSS v 2
PCI DSS v 3
PCI DSS v4.0

Important Disclaimers

1. I have not been a QSA since 2012.
2. I am not your assessor.
3. Although I was part of the working group that developed PCI DSS v4.0, you should not consider anything I say as a definitive interpretation of the standard.
4. These are all my own views, not Jscrambler's, or the card brand where I used to work.
5. I have no "inside knowledge" of the thoughts of the PCI SSC or the card brands.
6. You should:
    a) Make your own judgement based on the words and objectives in the requirements.
    b) Ask your assessor.
    c) Ask the PCI SSC.
    d) Ask the card brands (FAQ 1142).

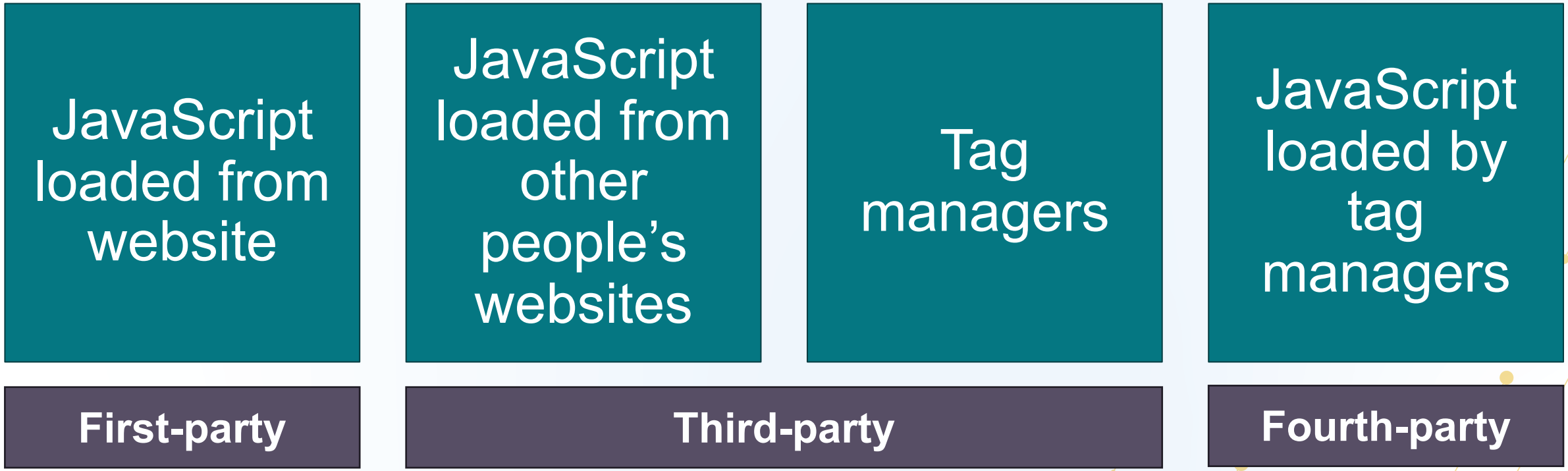

Process
Technology
People


Process
Technology
People

The Problem

Technical	Every bit of JavaScript has access to any page element.
Practical	The web is now built by assembling JavaScript in the browser
Vulnerability	The attack surface is all first- and third-party sources. Tag managers, and JavaScript loaded by tag managers
Threat	This is how most data is lost from e-commerce environments

The Attack Surface



Tag Manager

https://business.adobe.com/blog/basics/tag-manager

- “A tag is a piece of JavaScript code that is placed on a website.”
- “Tag managers help marketers manage tags from a user interface instead of directly manipulating the website’s source code.”
- “A tag manager shortens web development cycles. It frees up developer time so they can do other important work instead, and allows marketers to gather, organize, and manage website data better.”

A system designed to get information security and change control out of the way so that marketing and digital departments could add any random JavaScript to an organization’s website.

100
Scripts

52%
First party

48%
Third party

Source: Jscrambler research

48

Third-party scripts

13

Domains (mean)

35

Domains (max)

Source: Jscrambler research

How Does PCI DSS v4.0 Aim to Reduce the Risk?

6.4.3 Defined Approach

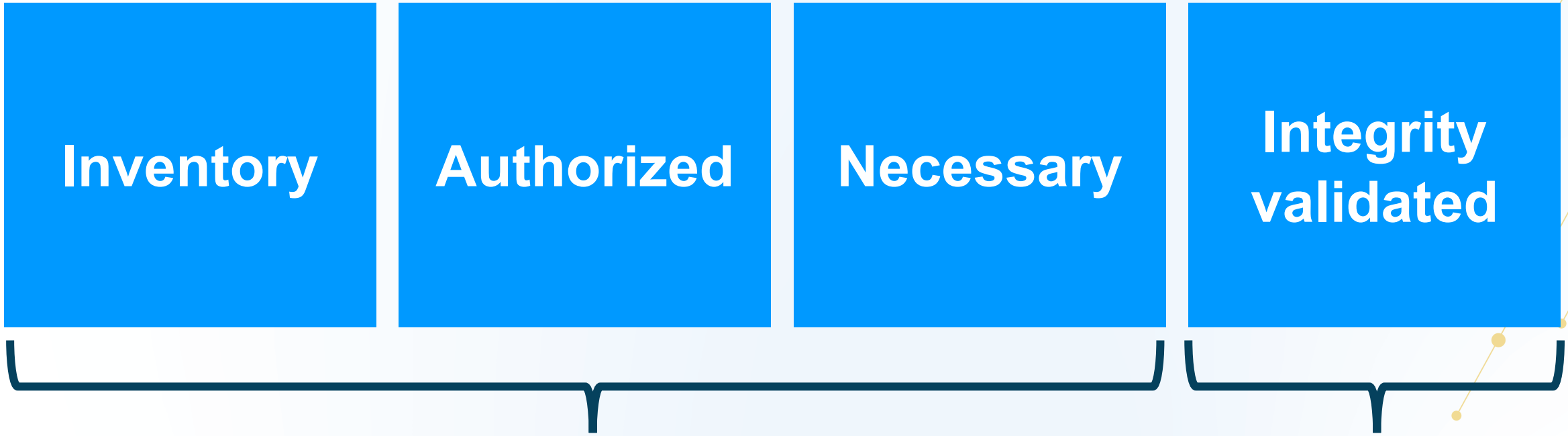
All payment page scripts that are loaded and executed in the consumer's browser are managed as follows:

- 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

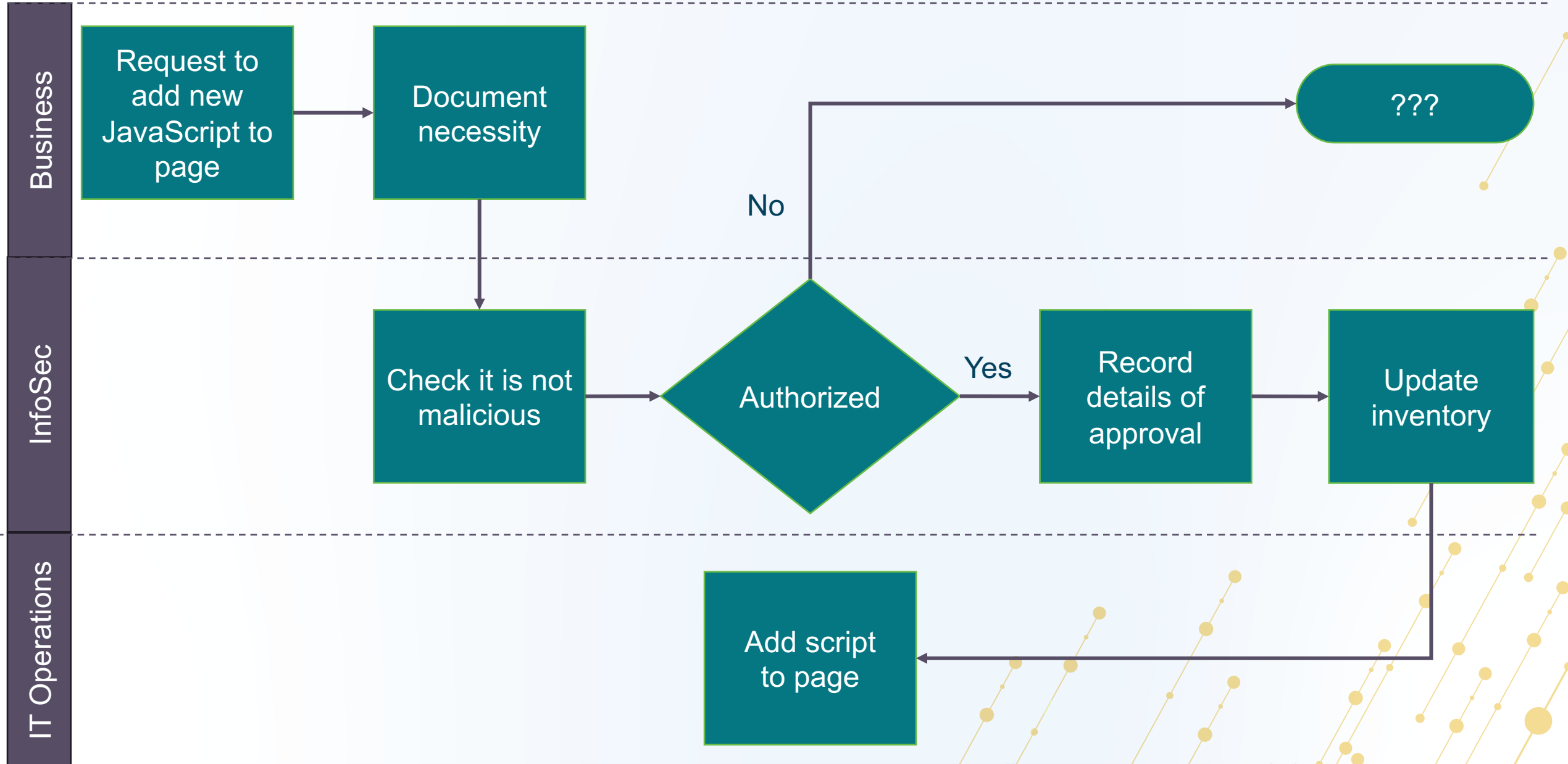
Customized Approach

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

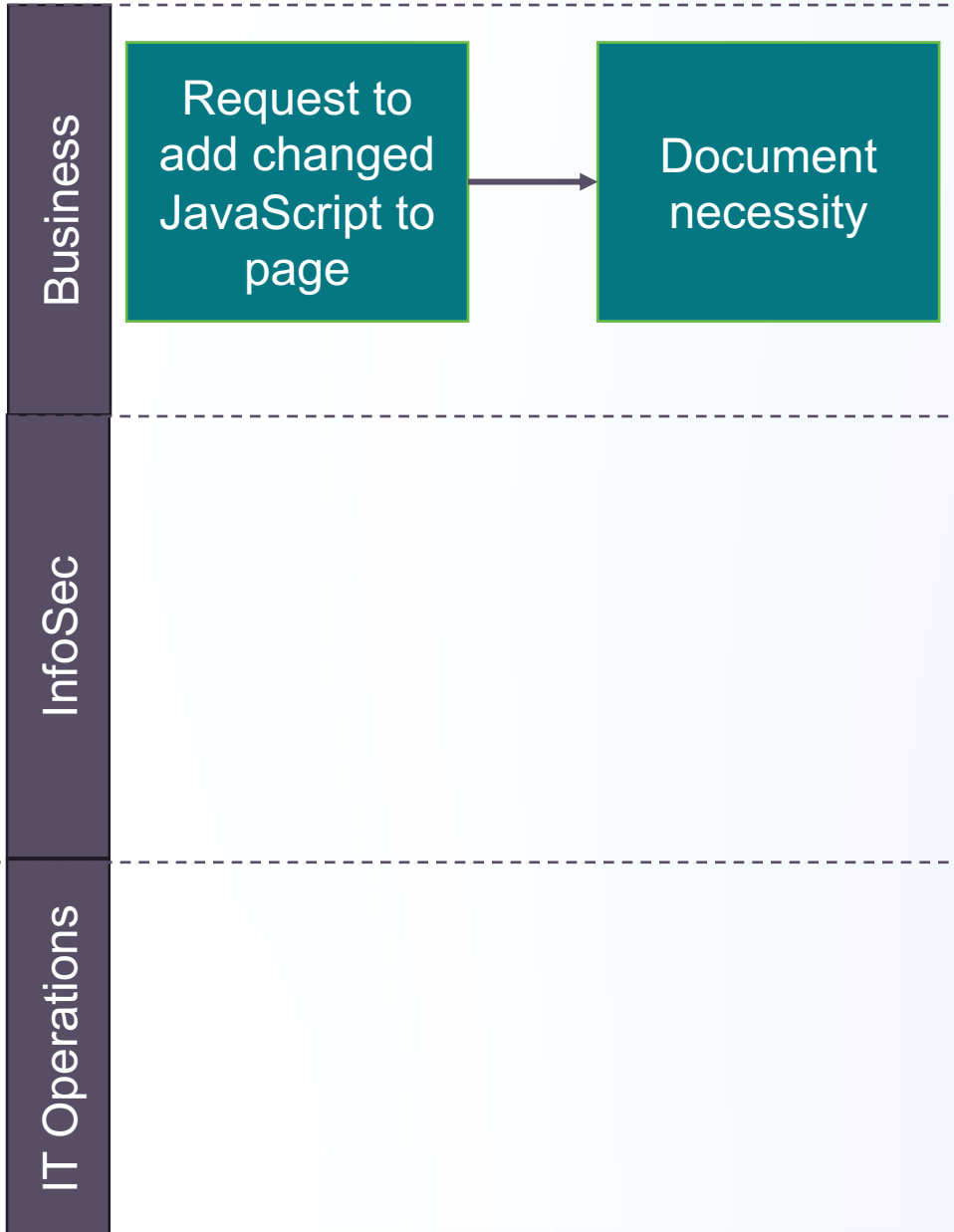
Requirement 6.4.3



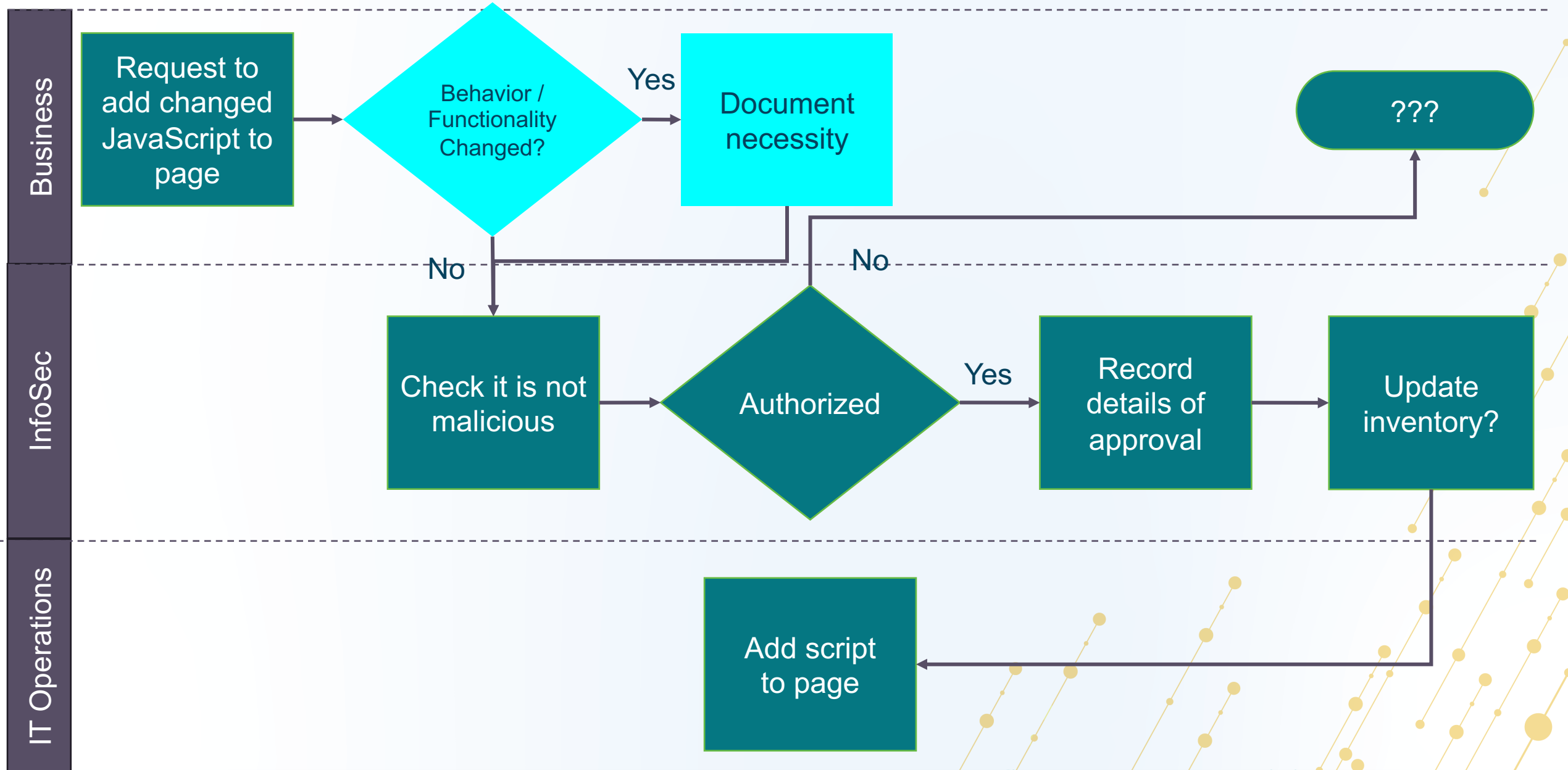
In an Ideal World: New Script



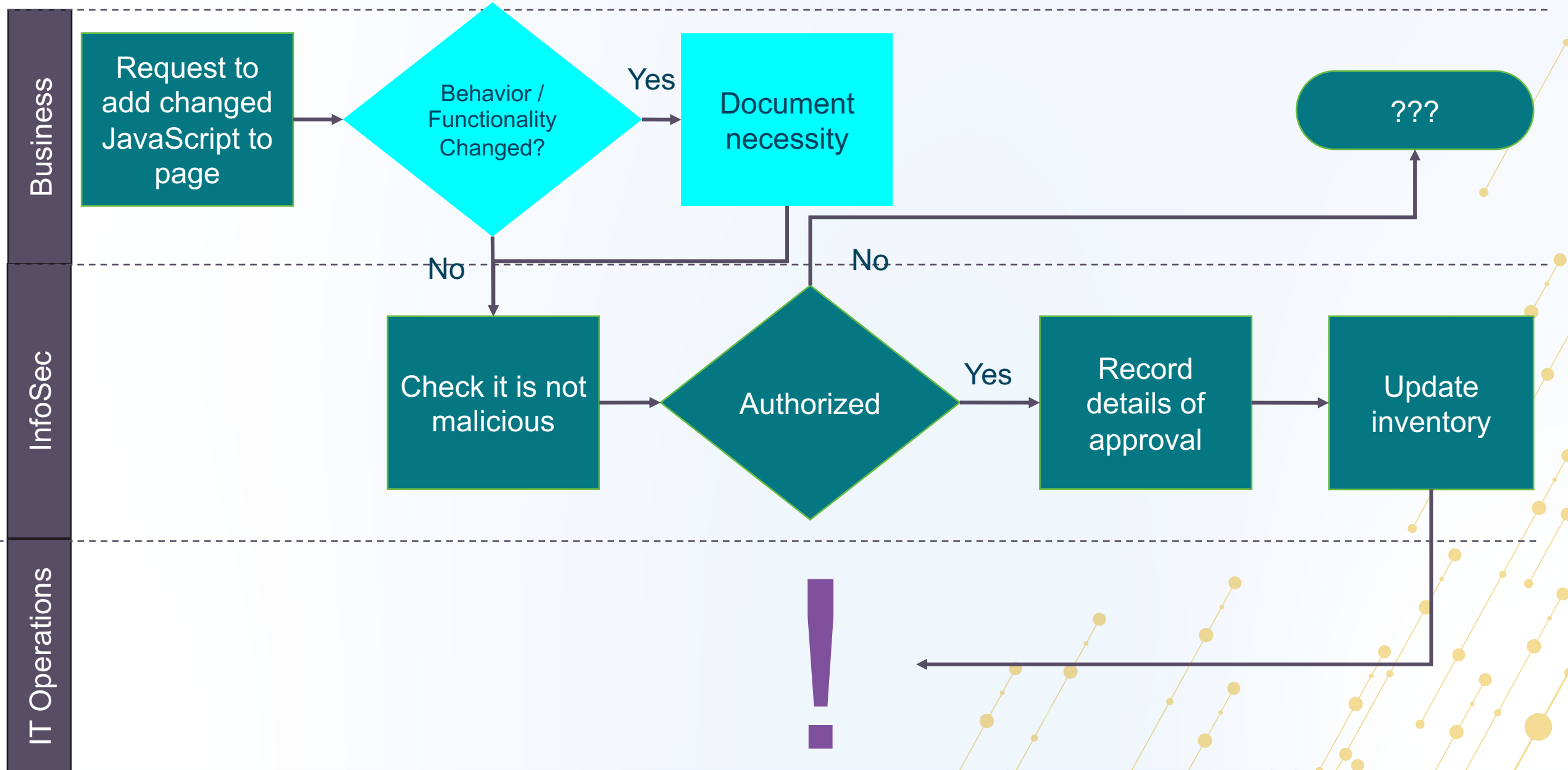
In an Ideal World: Changed Script



In an Ideal World: Changed Script



In an Ideal World: Changed Script



PCI Security Standards Council

The Easy (Free) Technology Solution

Content Security Policy (CSP)

Sub-resource Integrity (SRI)

Content Security Policy (CSP)

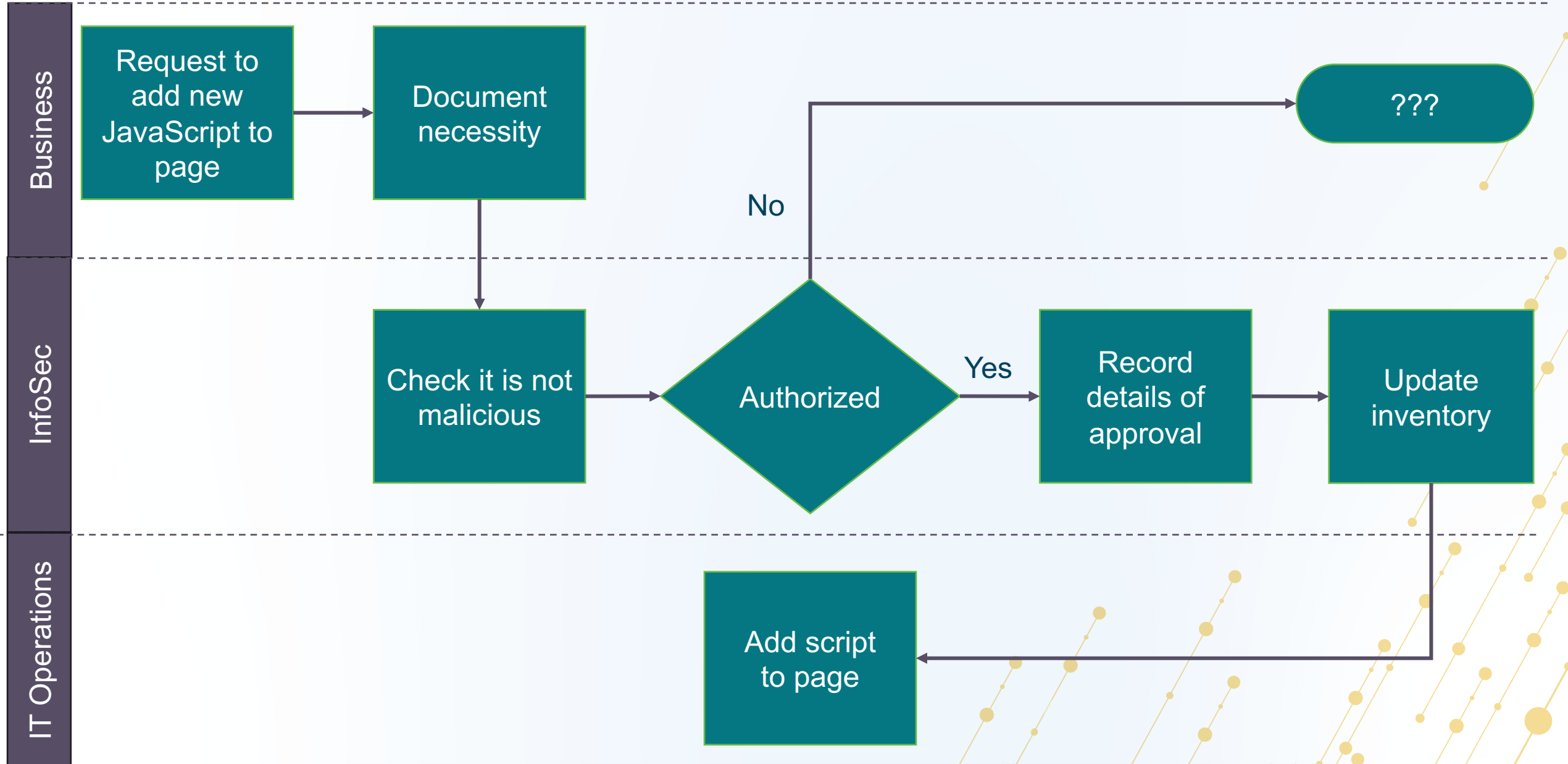
Ideally set in the HTTP header

Restricts where JavaScript can be loaded

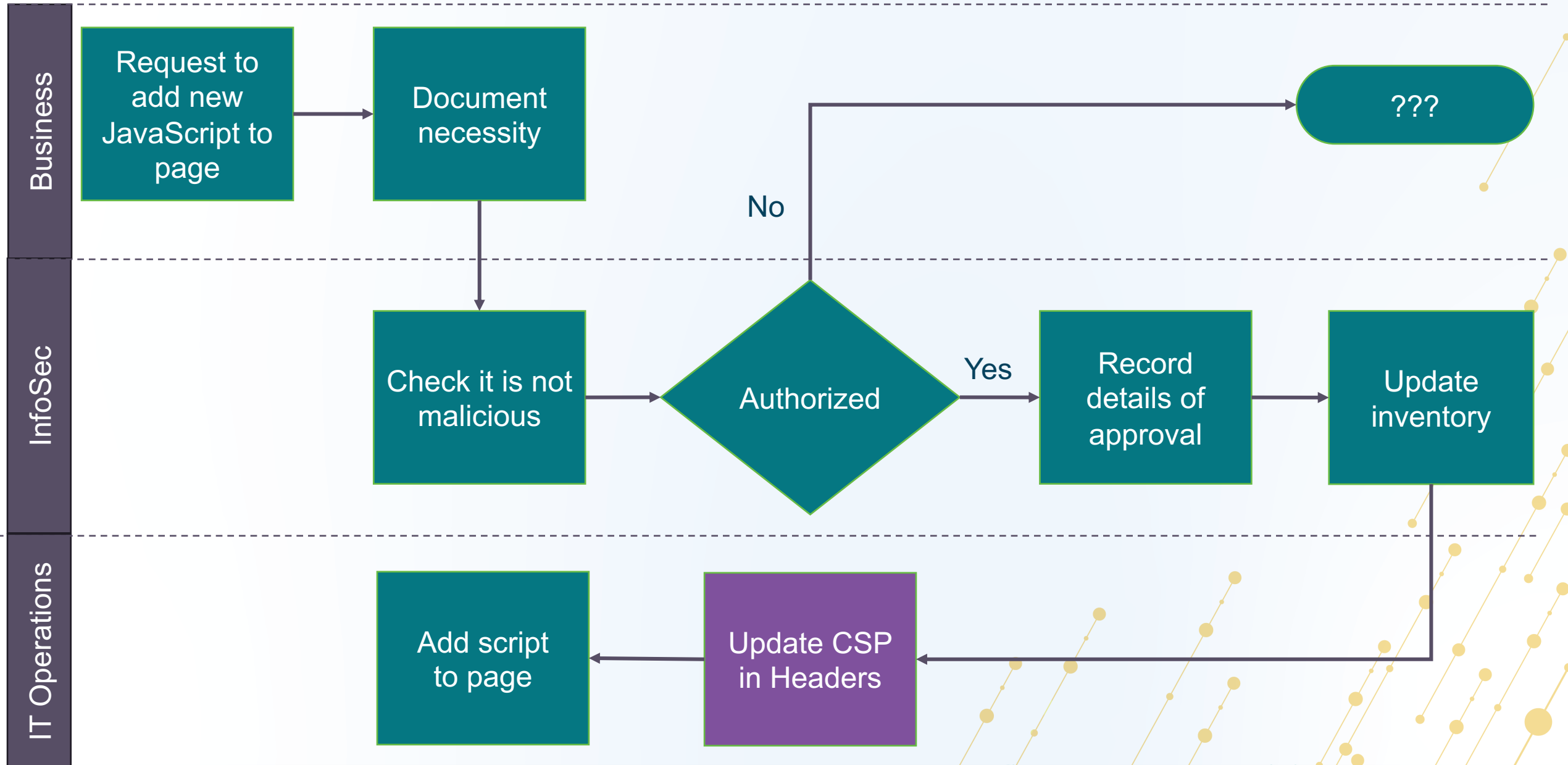
```
Content-Security-Policy: default-src 'self’;
script-src safescripts.foo.com
```

PCI Security Standards Council

In an Ideal World: New Script



In an Ideal World: New Script | CSP



Sub-resource Integrity (SRI)

Generate a hash for every JavaScript on the page

Add the hash to the SCRIPT tag

```
<script src="https://foo.com/myscript.js" integrity="sha384-
oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/uxy9rx7HNQlGYl1kPzQho1wx4JwY8wC">
</script>
```

PCI Security Standards Council

Sub-resource Integrity (SRI)

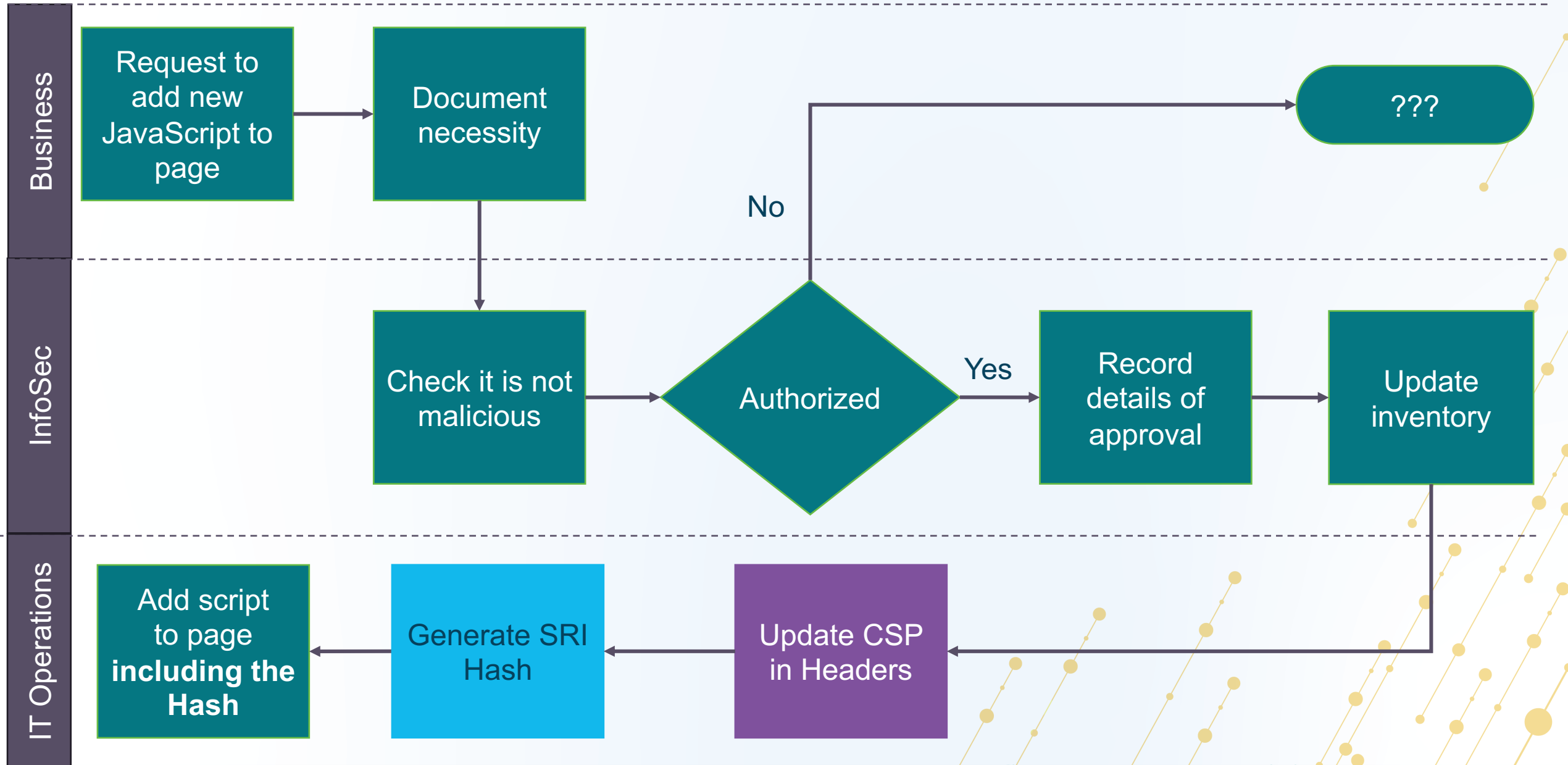
Generate a hash for every JavaScript on the page

Add the hash to the SCRIPT tag

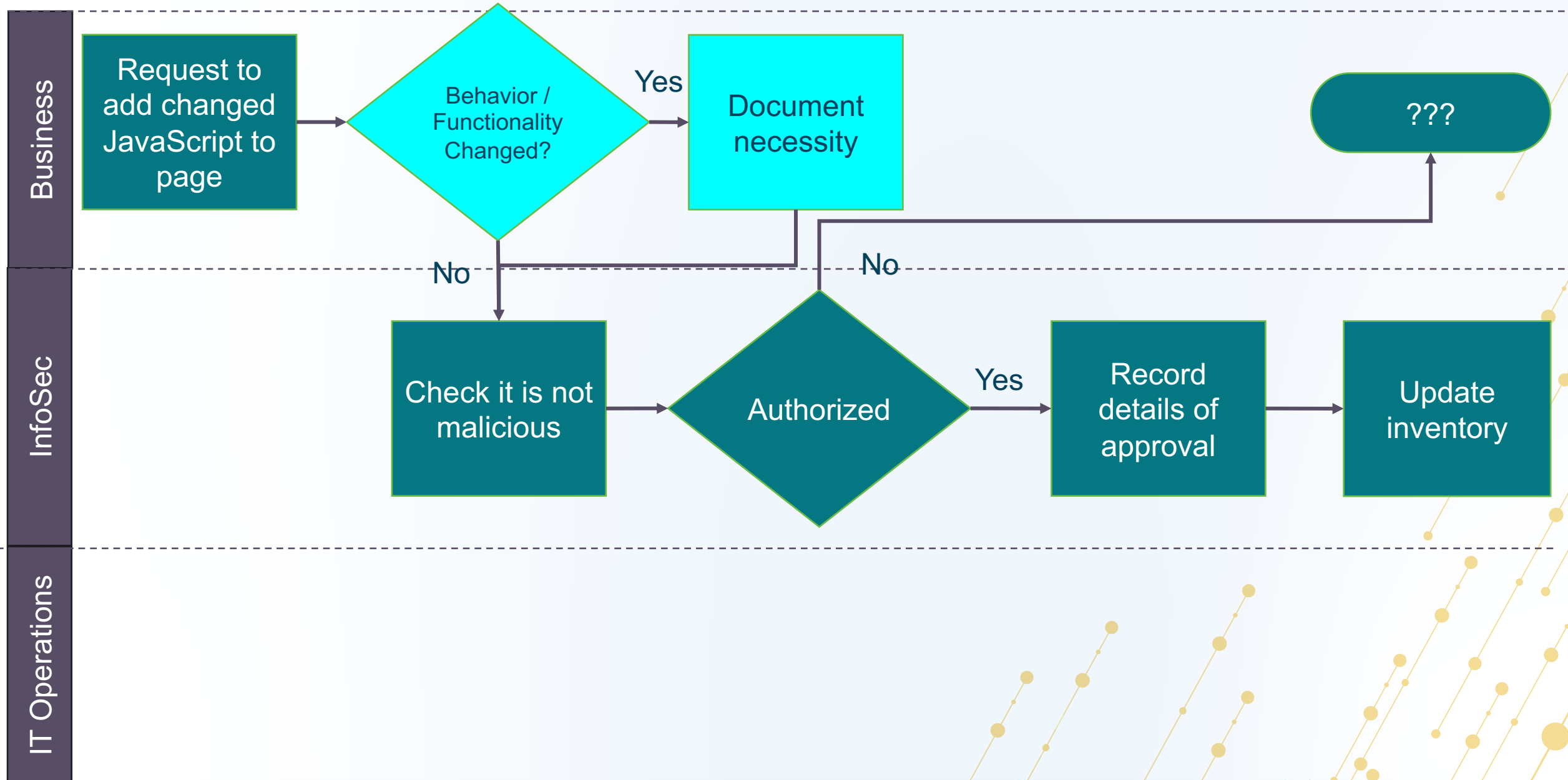
```
<script src="https://foo.com/myscript.js" integrity="sha384-
oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/uxy9rx7HNQlGYl1kPzQho1wx4JwY8wC">
</script>
```

PCI Security Standards Council

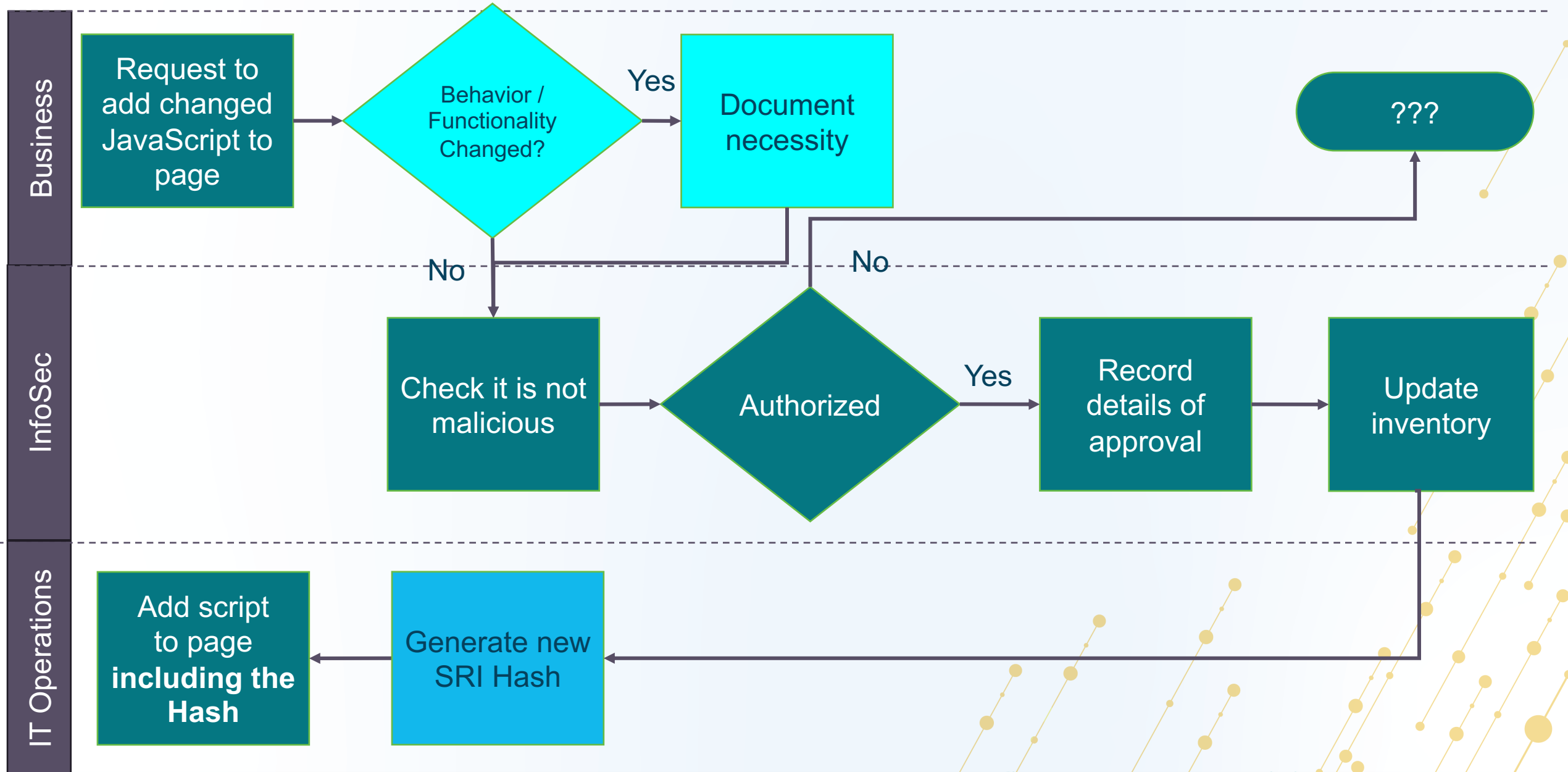
In an Ideal World: New Script | SRI



In an Ideal World: Changed Script | CSP



In an Ideal World: Changed Script | CSP & SRI



So What's The Problem?

New Scripts

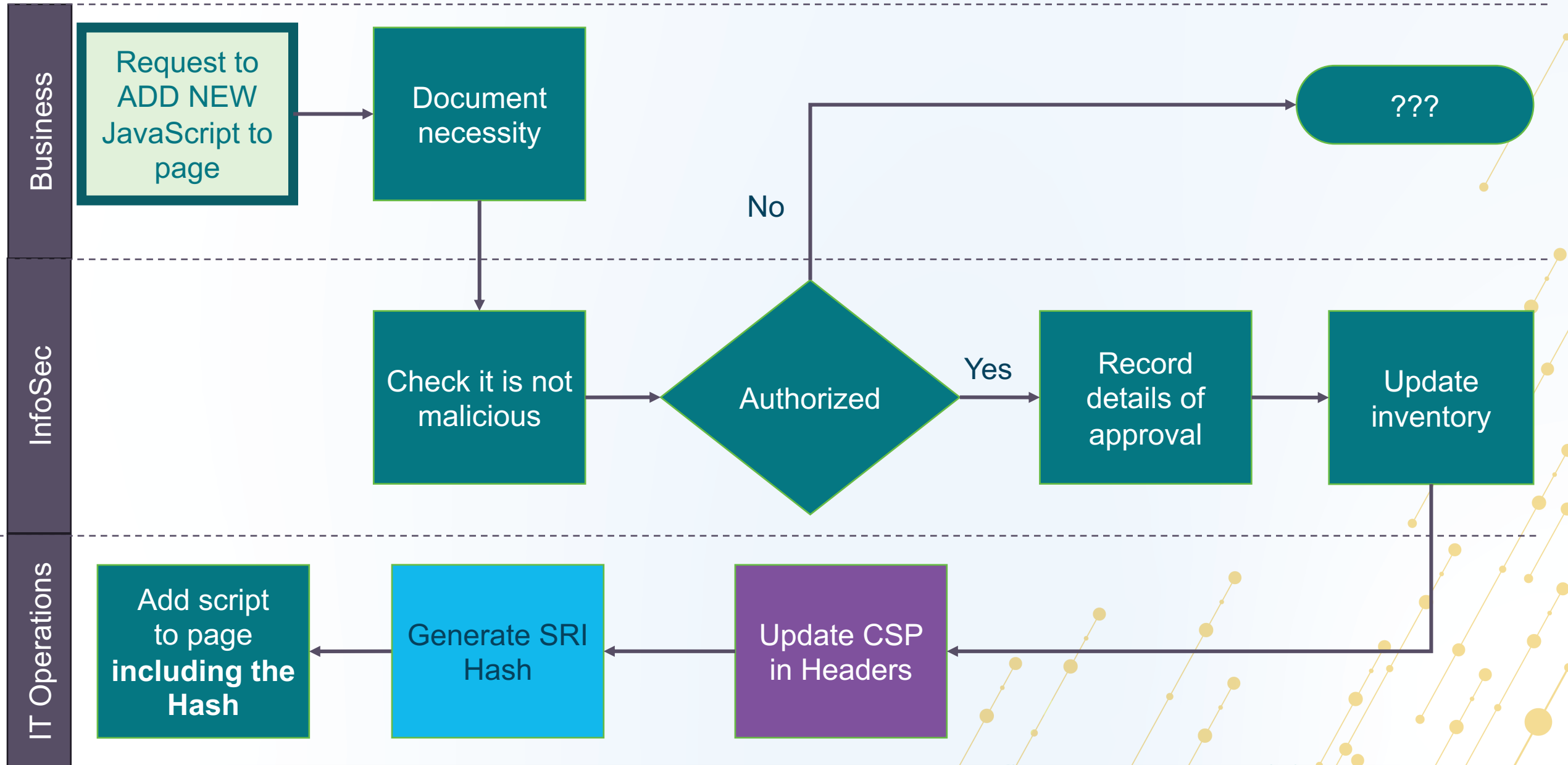
100
Scripts

52%
First party

48%
Third party

Source: Jscrambler research

In an Ideal World: New Script



First Party | New Script

Request to ADD NEW JavaScript to page

- Who wants to add JavaScript?
  - Marketing
    - Via a tag manager
  - Development
  - Digital
  - IT Operations

Action

1. Who in the entity can add a **new JavaScript** to the payment page?
2. How do they do it?
3. Can you enforce a change control process for new first party scripts?

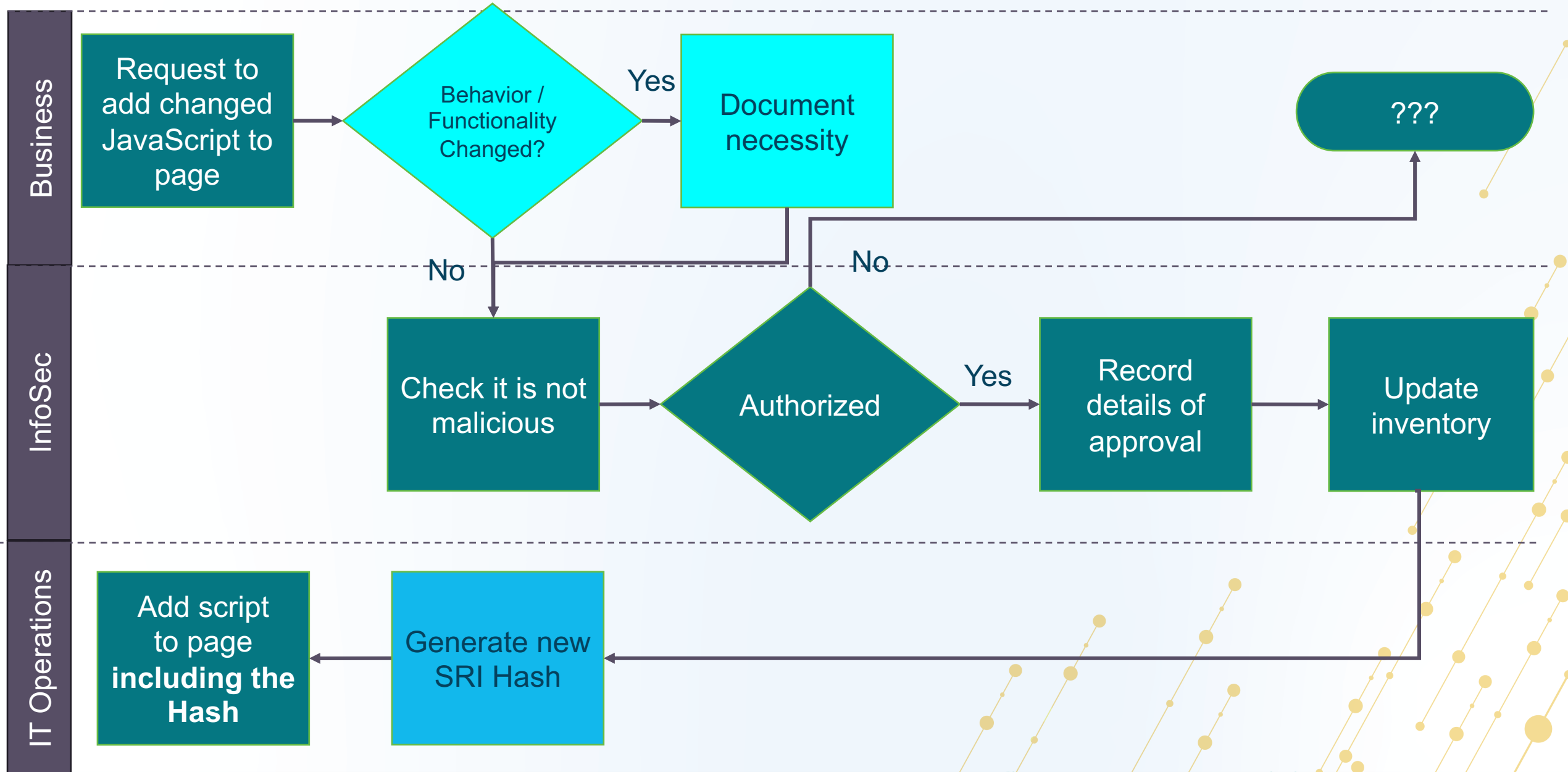
Changed Scripts

100
Scripts

52%
First party

Source: Jscrambler research

In an Ideal World: Changed Script | CSP & SRI



Action. First Parties and SRI

1. Does JavaScript change in its own when requested (i.e. Dynamic)?
2. Who in the entity can **change** existing **JavaScript?**
3. **Would they make a request?**
4. Can you enforce a change control process?
5. How/can will you update the hashes?
6. What automated QA can you include in this process?

Why did he ask “What automated QA can you include in this process”?

SRI fails silently

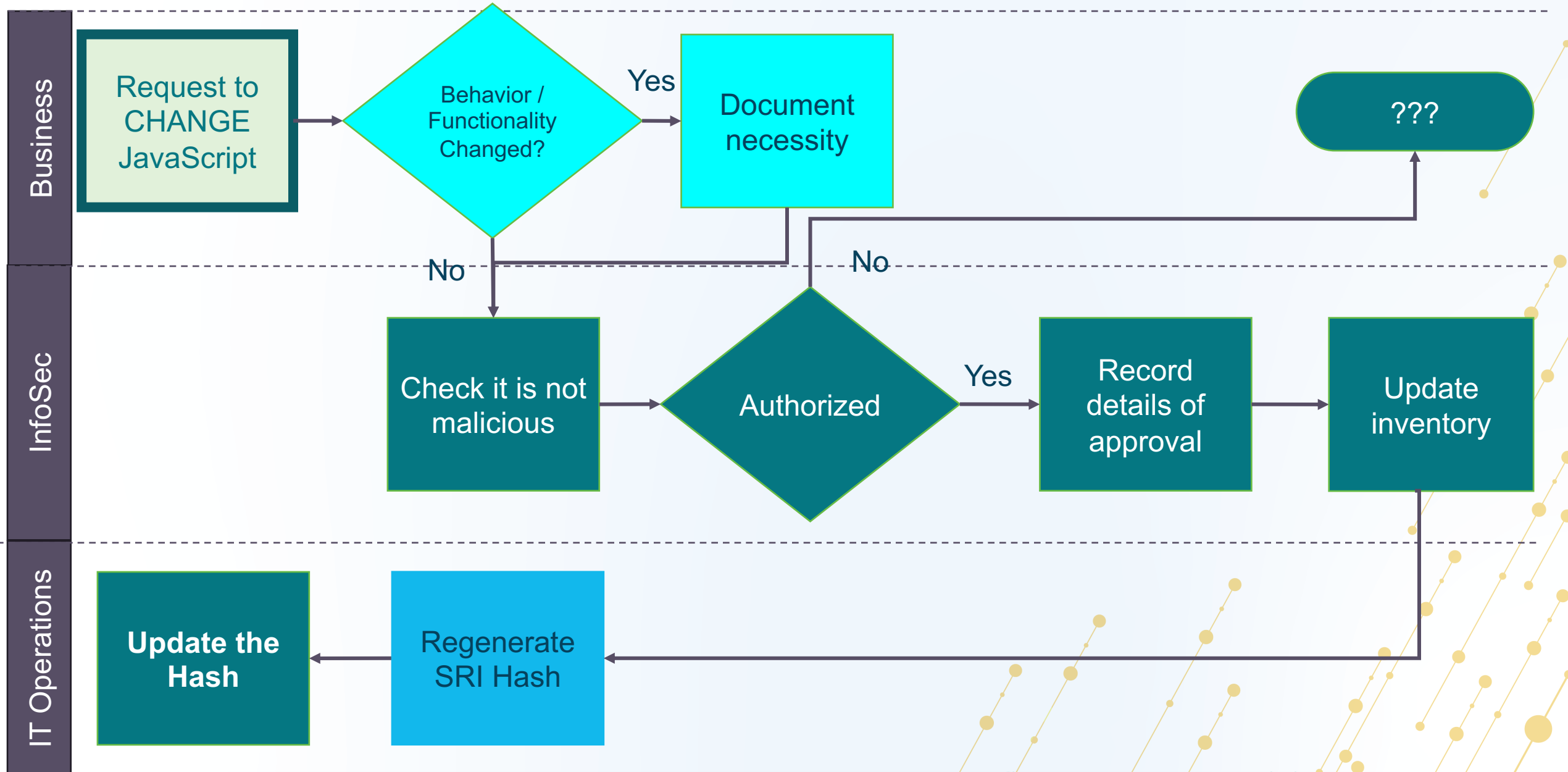
Changed Scripts

100
Scripts

48%
Third party

Source: Jscrambler research

In an Ideal World: Third-party | Changed Script



Thirds Party | Changed Script

Request to CHANGE JavaScript

- This is probably nonsense

Action. Third-parties and SRI

1. Would all third parties notify you via their change control process?
2. Can they send a copy of the JavaScript or give you a hash?
3. Will they wait until all entitles confirm they are ready for the change?
4. How will you update the hashes?
5. Is it dynamic anyway?

Summary


Chart Title
First-party | NEW
Third-party | NEW
First-party | CHANGE
Third-party | CHANGE
Risk
Complexity

Action. Process Mapping

1. What's your current process for **adding** new JavaScript?
2. What's your current process for **changing** JavaScript (if any)?
   a) Any automation?
   b) Any dynamic scripts?
3. It will not be the same for first-party and third-party?

Four Business Processes

New	Change	New	Change
1st Party	1st Party	3rd Party	3rd Party

PCI Security Standards Council

So What Are The Options?

Pragmatic Management and Compliance

Contrast

Entity A	Entity B	Entity C
• 25 First-party scripts • All managed by the dev team • Added via automated CI / CD pipeline • 1 Third-party script (payment processor)	• 6 First-party scripts, part of the framework • Framework updated in strict change control • 35 Third- & fourth-party scripts added by digital and marketing (tag mgr)	• 200 First-party scripts, dynamically generated by website code • 120 Third- and fourth- party scripts added by marketing (tag manger)

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

Focus on Authorization

New scripts need to be authorized

Alert on unauthorized changes

What the Standard Does Not Say

“Unauthorized scripts must be blocked”

What the Standard Does Not Say

~~“Unauthorized scripts must be blocked”~~

Multiple Authorization Processes?

Things to discuss with your assessor

You need to think about multiple, different, processes for authorization:

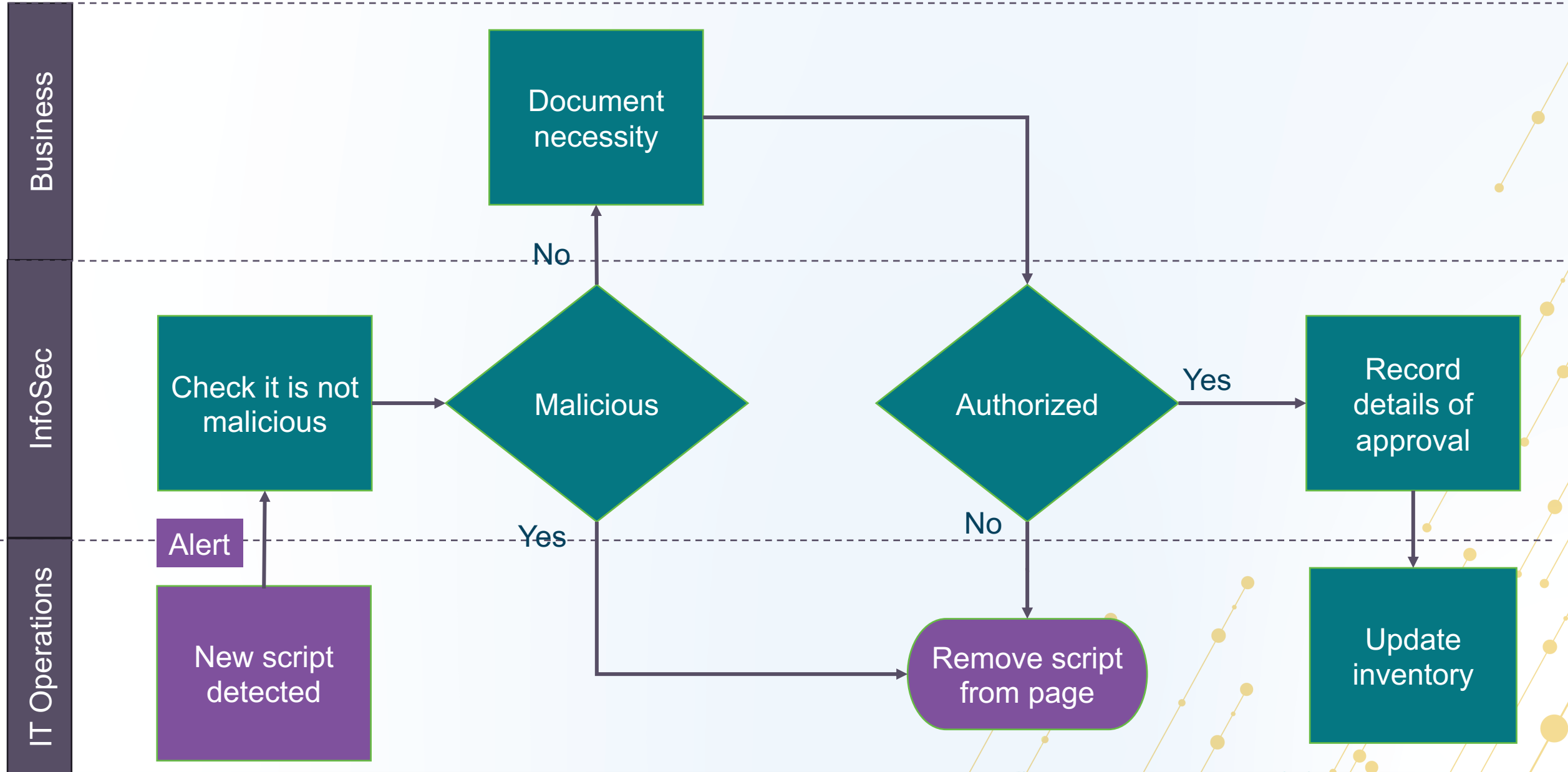
1. New first-party scripts
2. Changes to first-party scripts
3. New third-party scripts
4. Changes to third-party scripts

PCI Security Standards Council

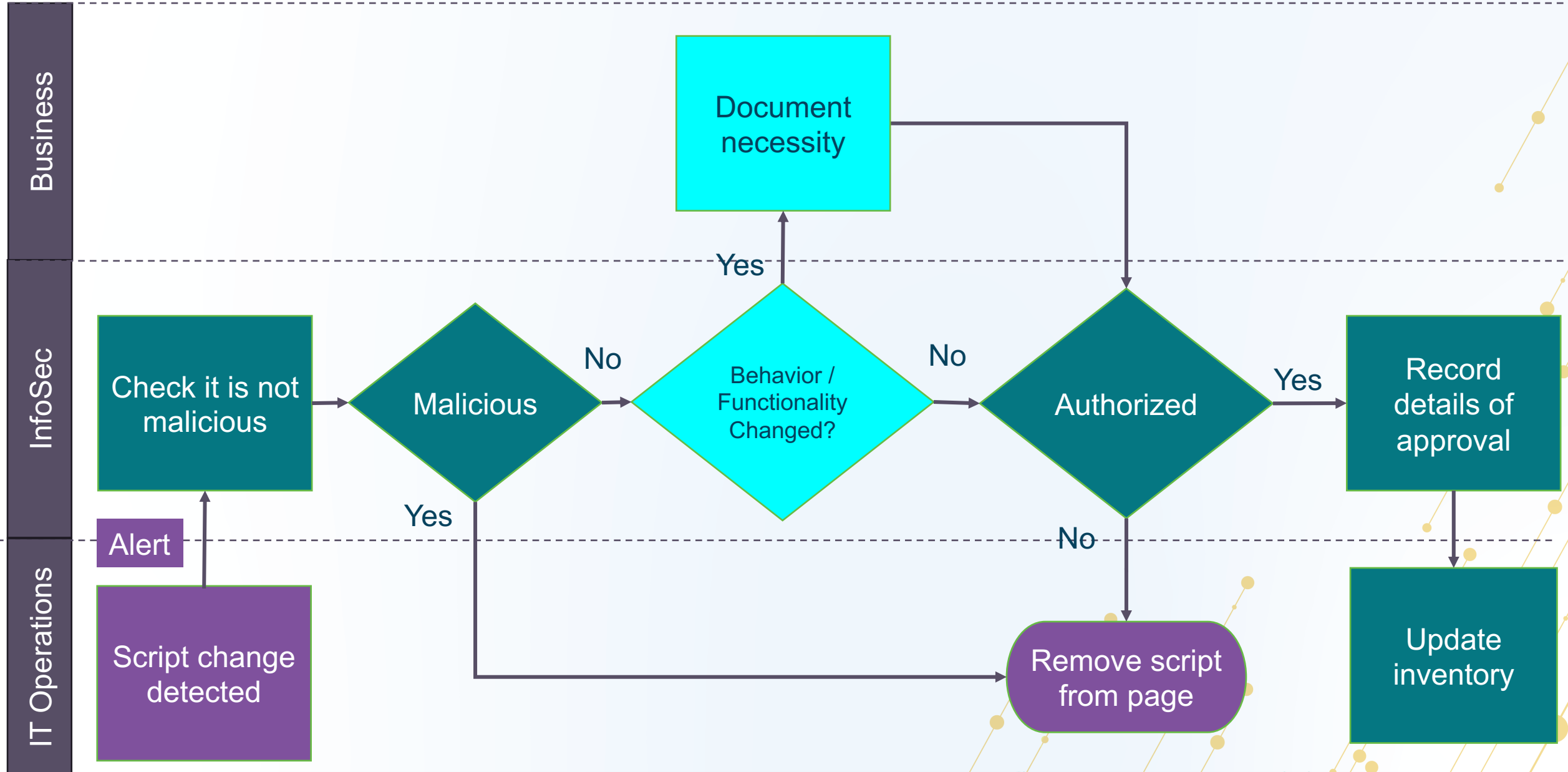
Three Types Of Processes & Authorization

Default authorize	Pre-authorize	Post-authorize
Low- or zero-risk changes	**Before the page is changed**	**After the page is changed**

Post-deployment Authorization: New Script



Post-change Authorization: Changed Script



If you can't use CSP and SRI, you'll probably need a tool

(this is the technology part)

Important take-away.

Don’t buy a tool until you’ve understood your environment and script management processes.

Tool Types

Proxies and CDNs

Scanners

Agents

Reverse Proxies and CDNs

- Sit between the web sever and the browser
- Can detect changes in first-party scripts
- Some will also retrieve third-party scripts and verify them
  - Inventory and authorization workflow
- Can dynamically manage CSP

Scanners

- Synthetic users (bots)
- Retrieve page, all first- and third-party scripts
  - Make inventory, start authorization process
- Look for and alert changes
- Can’t always see everything
  - There is no perfection in this exercise

Agents

- JavaScript that is loaded first into the page
- Can manage any other JavaScript
- Can ”watch” what other JavaScript does.
- Send telemetry back to make:
  - Inventory and kick off the authorization workflow
- Some can block malicious behavior

We've Reached the End

Summary

- Managing JavaScript is hard
  - More = harder
  - Third-party = even harder
- Map processes and discover owners
- Probably need separate processes:
  - First- and third-party
  - New and changed
- Multiple kinds of authorization:
  - Default-, pre-, and post-
- Don’t buy a tool until you understand your business.

Remove all JavaScript from your Payment (and Parent) Pages

Remove all third-party JavaScript from your Payment (and Parent) Pages