

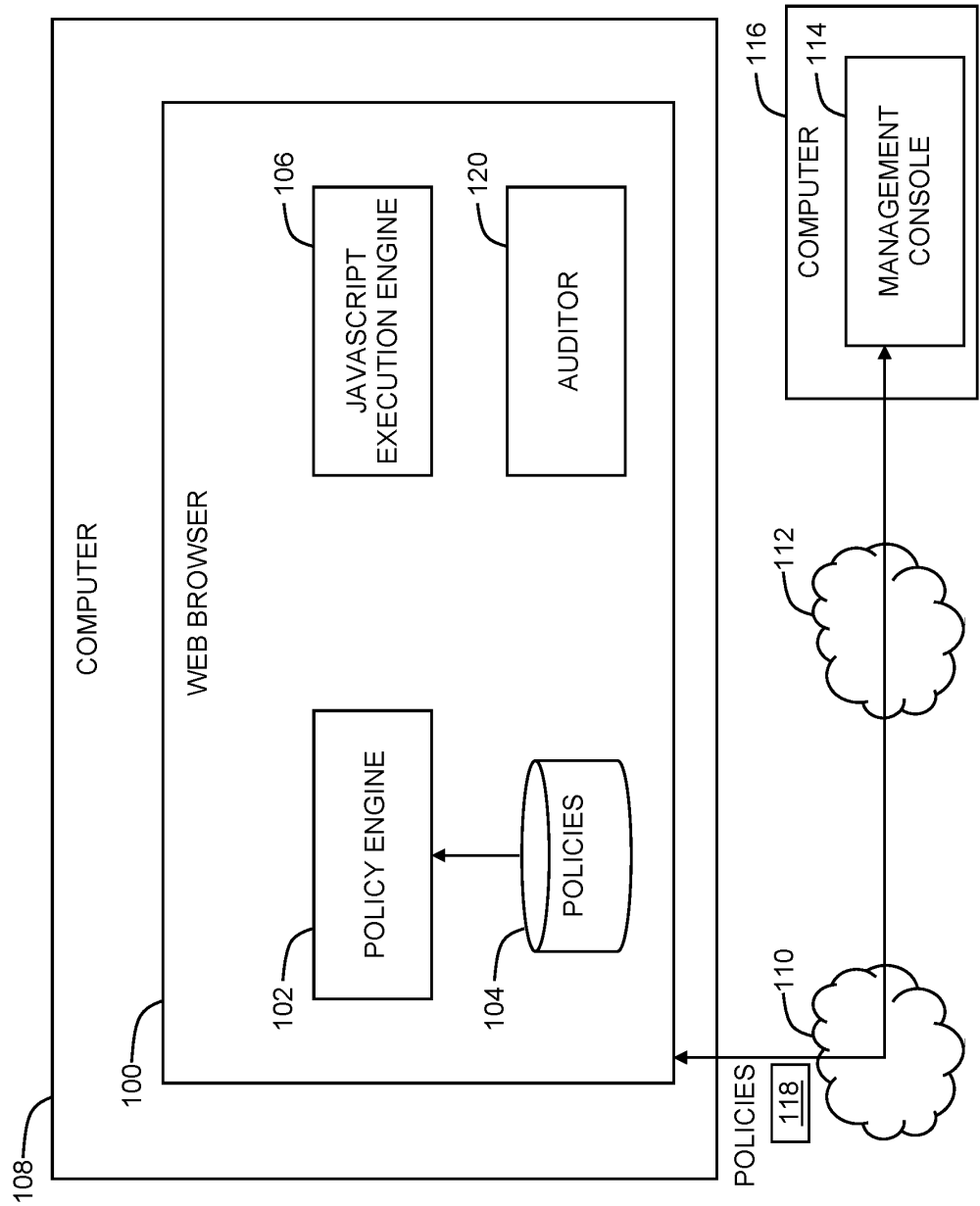


Fig. 1

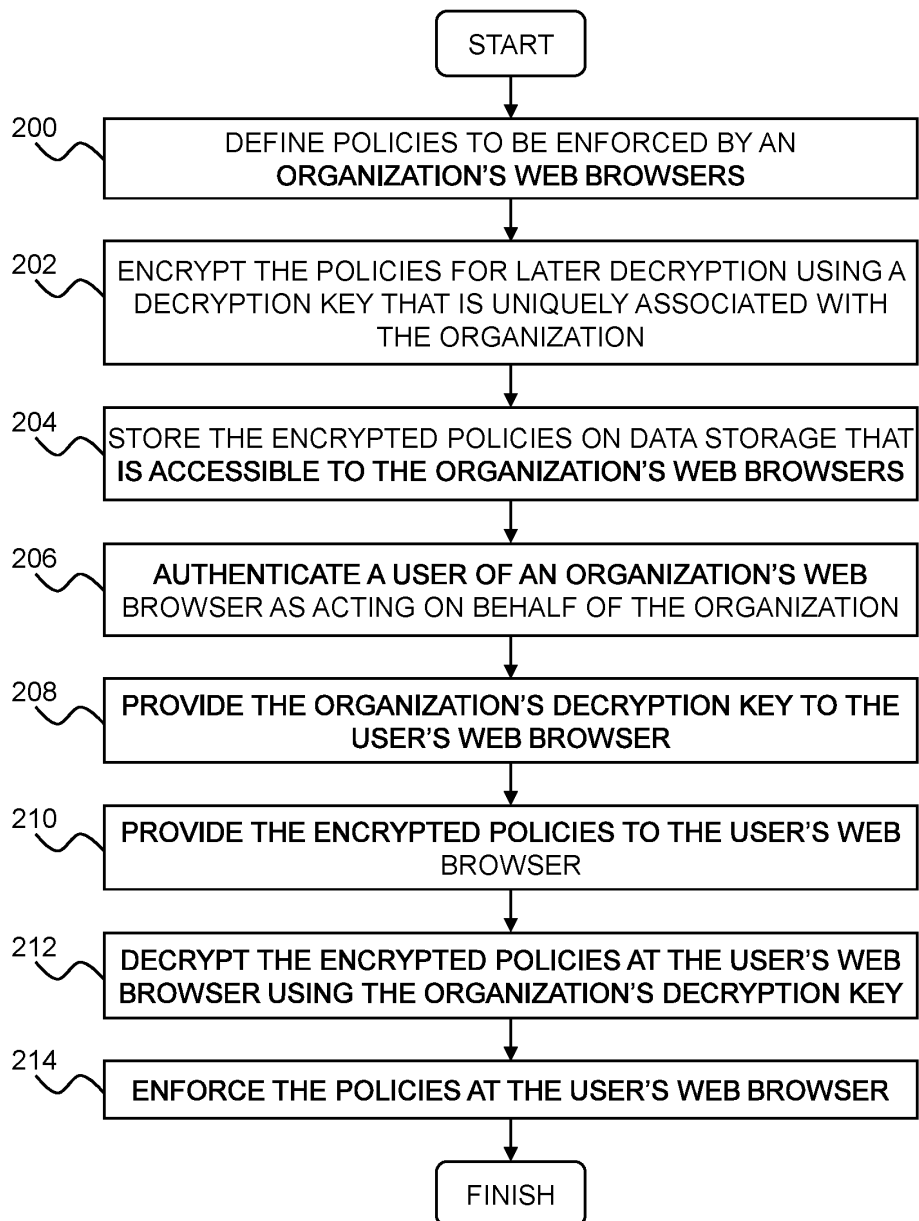


Fig. 2

300

Source

Object Condition

AND

OR

Objects

Identity

Any

☒

Device

Any

☒

Network

Any

☒

Location

Any

☒

Time

Any

☒

Fig. 3A

302

Identity Object

Marketing Team

All users in the Browser product marketing team

MATCH ATTRIBUTES

AND OR

Conditions

▼ Include +

User ▼ adam@za.com

Group ▼ Accounting

Custom ▼ NAME: Employment

VALUE: Vendor

▼ Exclude +

User ▼ moty@za.com

CANCEL

SAVE CHANGES

Fig. 3B

304

Device Object

...

|Add Object Name

Add description

Windows☒

Mac OS☐

Android☐

MATCH ATTRIBUTES

AND

OR

Posture Types

FULL DISK ENCRYPTION

FIREWALL

▶ DETECT DLP

+

▶ DETECT EPP

+

▶ DOMAIN JOINED

+

▶ PROCESS CHECK

+

▶ REGISTRY KEY

+

▶ FILE PATH

+

▶ CERTIFICATE TRUST

+

▶ CLIENT CERTIFICATE

+

CANCEL

CREATE OBJECT

Fig. 3C

```

> chrome.zaPolicy.wmiQuery("SELECT ProcessId, WorkingSetSize, ExecutablePath FROM Win32_Process", {result} => console.log(result))
< undefined

{queryResult: Array(164), status: "success"}
{queryResult: Array(164)}
  [0 ... 99]
  [100 ... 163]
    # 100: {ExecutablePath: "C:\windows\system32\taskhost.exe", ProcessId: "1928", WorkingSetSize: "28160512"}
    # 101: {ExecutablePath: "C:\windows\system32\svchost.exe", ProcessId: "4584", WorkingSetSize: "16072704"}
    # 102: {ExecutablePath: "C:\windows\system32\svchost.exe", ProcessId: "5296", WorkingSetSize: "8288208"}
    # 103: {ExecutablePath: "C:\windows\system32\ctfmon.exe", ProcessId: "5452", WorkingSetSize: "26801408"}
    # 104: {ExecutablePath: "C:\windows\Explorer.EXE", ProcessId: "6532", WorkingSetSize: "147976192"}
    # 105: {ExecutablePath: "C:\windows\system32\svchost.exe", ProcessId: "6836", WorkingSetSize: "22974464"}

> chrome.zaPolicy.wmiQuery("SELECT DomainRole, Domain FROM Win32_ComputerSystem", {result} => console.log(result))
< undefined

{queryResult: Array(1), status: "success"}
{queryResult: Array(1)}
  # 0: {Domain: "WORKGROUP", DomainRole: "0"}
    length: 1
    __proto__: Array(0)
    status: "success"
  __proto__: Object

```

Fig. 3D

306

Network Object

...

US Datacenter

Add description

CONDITIONS

▼ Include

+

IP▼198.51.100.0

🗑

Subnet▼221.1.1.20 / 26

🗑

IP Range▼FROM: 1.0.0.0

TO: 126.0.0.0

🗑

▼ Exclude

+

IP▼1.2.3.4

🗑

CANCEL

CREATE OBJECT

Fig. 3E

308

Destination

Object Condition

AND

OR

Objects

Application

Any

☒

URL

Any

☒

Web Category

Any

☒

Web Reputation

Any

☒

Network

Any

☒

Location

Any

☒

Fig. 3F

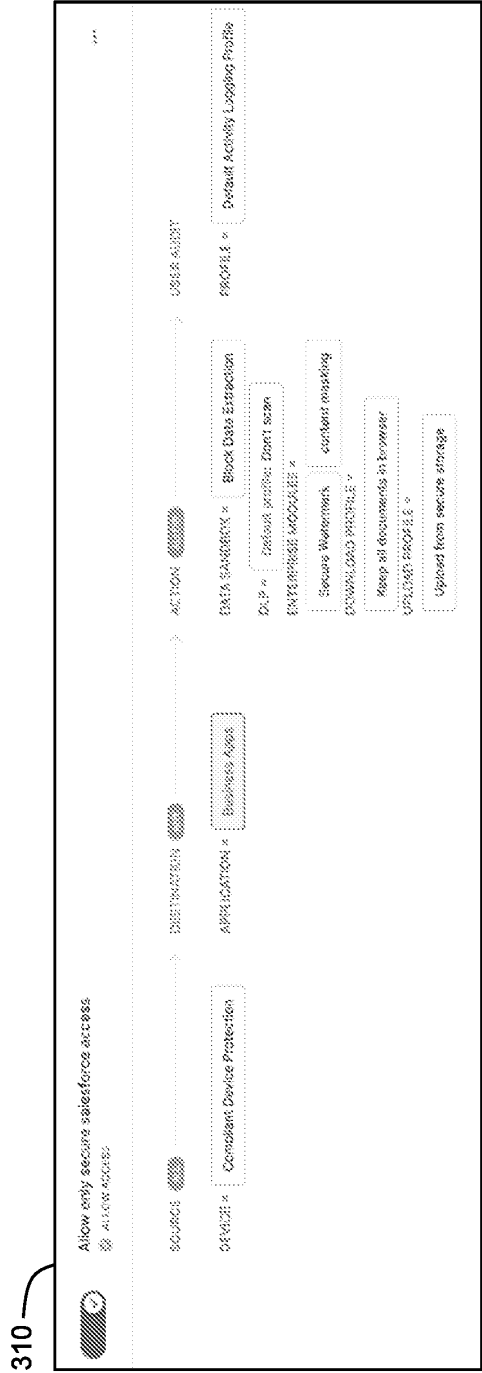


Fig. 3G

312

EDIT DATA LOSS PREVENTION PROFILE		...
Name		
Restrict Credit Cards and PII		
Add Description		
WHEN USERS DOWNLOAD FILES		☐
WHEN USERS UPLOAD FILES		☒
Run this scanner:		
Credit Cards x		
If scan results matched -> ☒ Block		
WHEN USERS COPY TO CLIPBOARD		☐
WHEN USERS FILL DATA IN FORM		☐
CANCEL		SAVE CHANGES

Fig. 3H

314

CHOOSE FILE TYPE

DEFAULT: SCAN

▶ Popular File Types

Scan All ▼

ALL CATEGORIES

▼ Popular File Types

Scan All ▼

Electronic Publication (EPUB)	Scan All ▲
MHTML (.mht, .mhtml)	Allow
S/MIME Compressed Email Message (.p7z)	Block
tar (.tar)	Secure Storage
Gzipped TAR file files (.tgz)	Scan
ZIP (.zip)	View

CANCEL

CHOOSE FILES

Fig. 3I

316

CREATE DOWNLOAD PROTECTION PROFILE

Name

Add Download Protection Profile name

Add description

► Override specific file types

+

▼ Advanced settings

Maximum file size allowed to download:

☒ Unlimited

☐ Custom size

When files can't be scanned:

☒ Block download

☐ Download without scanning

▼ Antimalware Scanner

☒ Webroot Threat Intelligence (URL Based)

☒ Google Safe Browsing (URL Based)

☒ Bitdefender (Static Analysis)

CANCEL

CHOOSE FILES

Fig. 3J

318

 **CREATE ENTERPRISE MODULE**

NAME

Add Enterprise Module name

DESCRIPTION (OPTIONAL)

Add description

 **ENTERPRISE MODULES (17)**

Select a built-in custom module or create your own with a custom Javascript code

 Block Email Attachments On ServiceNow

 HRMS form auto fill

 HTTP Header Addition

 Launch Darkly Enforce Client SDK Availability

 Launch Darkly Remove Wrong Tenant Attribute

 Protect idle tabs

 Require MFA For Configured Action

 Require MFA on page load

 Sensitive Content Watermarking

 Sensitive Data Masking

 Warn Before Leaving Page

 Warn Or Block When Sending Sensitive...

 WebRTC Usage Control

 Whatsapp block CCN message

CANCEL

CREATE ENTERPRISE MODULE

Fig. 3K

CREATE ENTERPRISE MODULE

CREATE A CUSTOM ENTERPRISE MODULE

Add a custom Javascript code to create custom behaviours on top of your business apps. This module will be added to the pool of custom Enterprise Modules.

CODE WILL AFFECT: Top frame only

```

1
2. /* *****
3  ***Use the following editor to insert scripts into the browser***
4  1. You can use island.sdk .to access different libraries.
5  for example island.sdk.$ is similar to using Cash J5 library
6  2. Context can be referenced to get specific details about the
7  current browsing session. For example: alert(context.userinfo.email)
8  will show an alert with the logged in user's email.
9  *context.url
10 *context.tabId
11 *context.userInfo.userId
12 *context.userInfo.tenantId
13 *context.userInfo.picture
14 *context.userInfo.name
15 *context.userInfo.email
16 3. You can send custom audit events by adding island.auditAsync( 'name_of_event') to the c
17 4. You can show customized user notification by adding the following in the code:
18 rpa.sdk.userNotification.show(context, {
19   type: 'Warning',//Warning {Info - will set the notification icon
20   action: 'Some Action',//free text
21   verdict: 'Warned',// 'Blocked' | 'Warned' | 'Monitored' - will be shown as the verdict in
22   tooltip: 'Some tooltip message',
23   link: 'https://some.link' // optional
24   closeNotificationAfterMS: 2000 //optional
25 }
26 *****
27 */
28
29. /* *****Mask CCN*****
30 should be configured to hide CCN shown in a webpage
31 */
32 // var targetNode = document.documentElement;
33 // Options for the observer (which mutations to observe)
34 // var config = { childList: true, subtree: true };
35 // var textNodesUnder = function(el){
36
    
```

CANCEL

CHOOSE FILES

Fig. 3L

```
if (await this.matchAlwaysAllow(context)) {
    cacheEntry.ruleMatchType = RuleCacheResultType.AllowAllRule
    this.policyMatchCacheProvider.setCache(cacheEntry.key, cacheEntry)
    return AlwaysAllowRule
}

for (let i = 0; i < policy.rules.length; i++) {
    if (await this.ruleMatcher.match(policy.rules[i], context)) {
        cacheEntry.ruleMatchType = RuleCacheResultType.UserRule
        cacheEntry.ruleIndex = i
        this.policyMatchCacheProvider.setCache(cacheEntry.key, cacheEntry)
        return policy.rules[i]
    }
}

this.policyMatchCacheProvider.setCache(cacheEntry.key, cacheEntry)
return policy.defaultRule
```

Fig. 4A

```
match(categoryConditions: CategoriesGroupDto[], urlCategories: Array<Category>): boolean {  
    if (categoryConditions || categoryConditions.length === 0) {  
        return true  
    }  
    return categoryConditions.some((cond : CategoriesGroupDto ) => {  
        return (  
            UrlCategoriesMatcher.isCategoryConditionValid(cond) &&  
            urlCategories.some((cat : Category ) => {  
                return this.doesCategoryMatch(cat, cond)  
            })  
        )  
    })  
}
```

Fig. 4B

```
const profile = roleData?.uploadProfile
return Promise.all(
  this.uploadContext.filePaths.map(async (filePath :string) => {
    const allowed = !this.fileTaggingMatcher.match(profile?.fileTaggingProperties, filePath)
    this.logger.info( msg `ZA upload file ${filePath} allowed=${allowed} url=${this.context.destinationUrl}` )
    if (!allowed) {
      this.SendFeedback(filePath)
    }
    await this.auditingProvider.auditAsync(this.context, new AuditFileUploadDetails(allowed, this.uploadContext), roleData)
    return allowed
  })
),
```

Fig. 4C

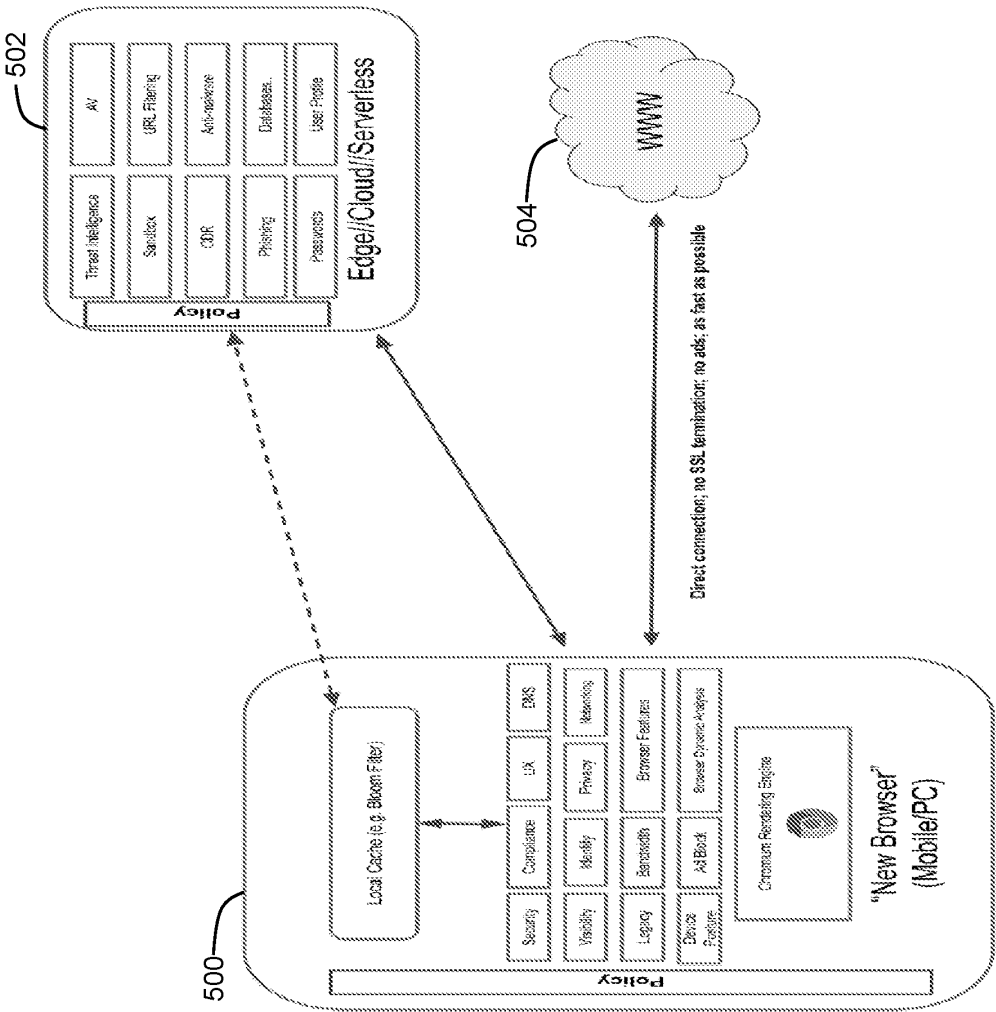


Fig. 5A

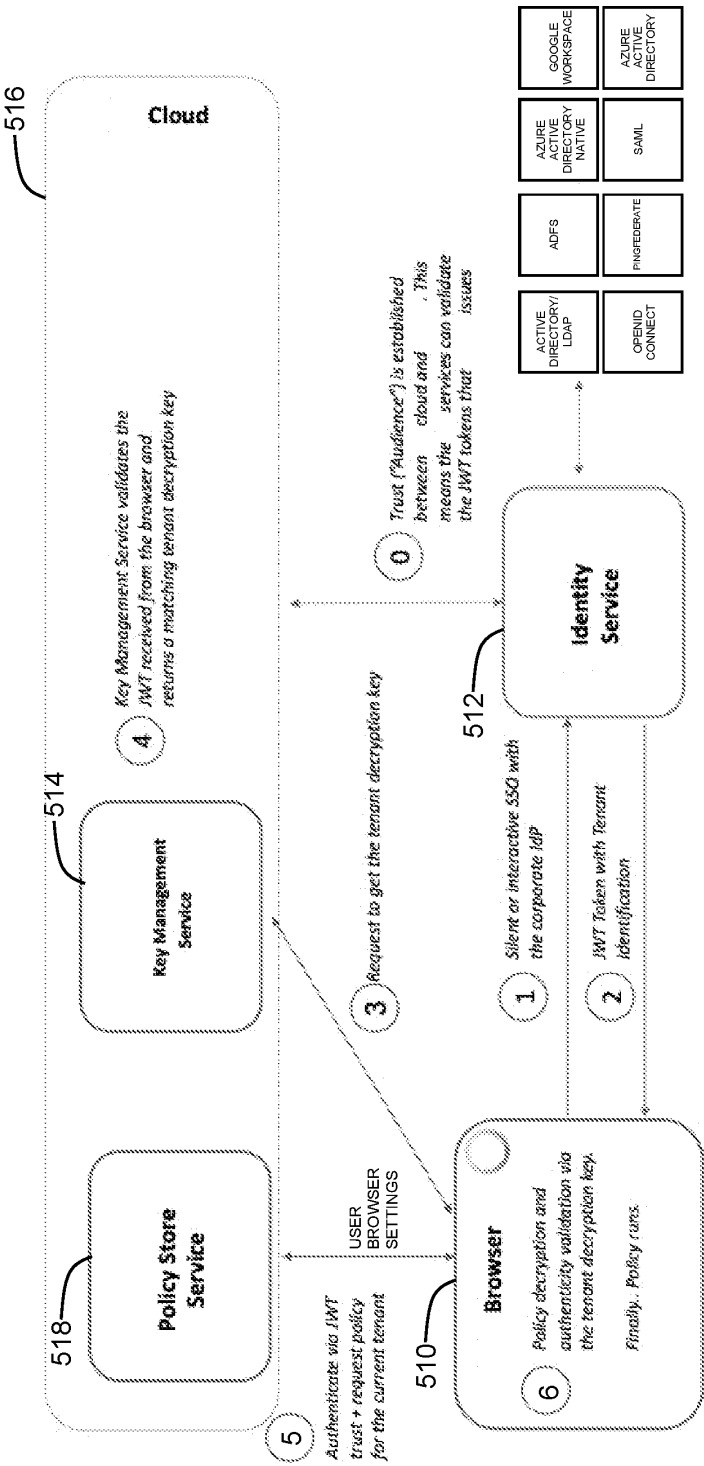


Fig. 5B

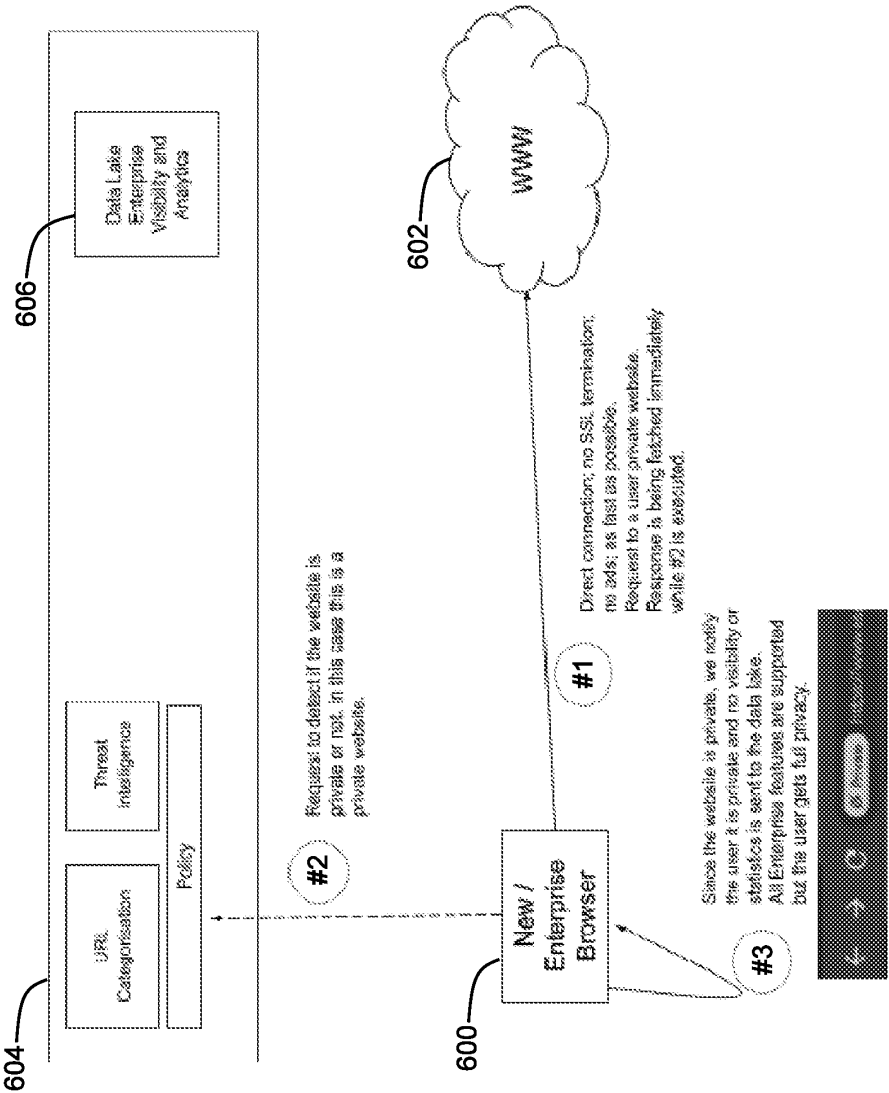


Fig. 6

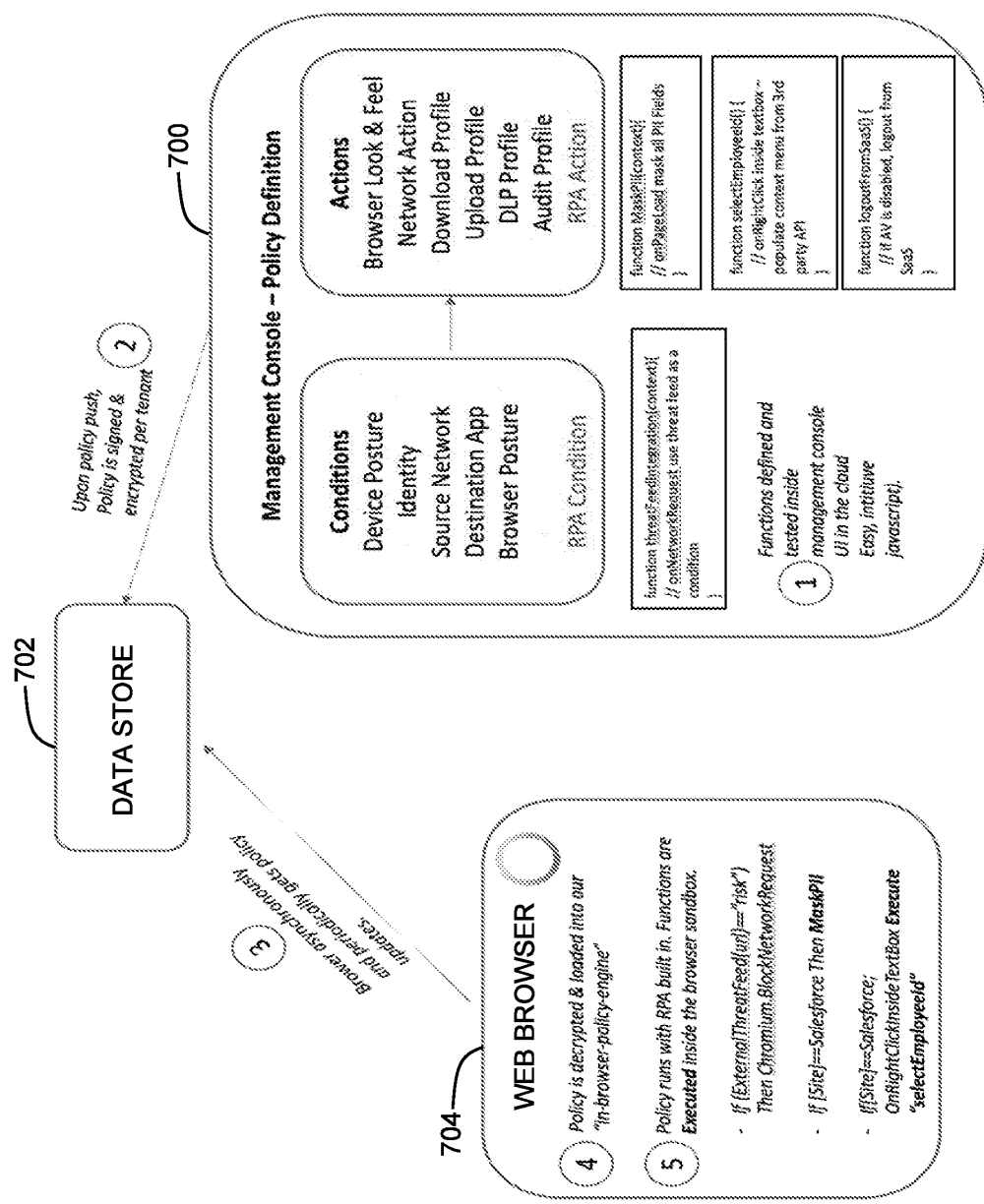


Fig. 7A



All

Search Accounts and more...

Service Console

Accounts

Accounts

Recently Viewed

New

Import

5 items • Updated a minute ago

Search this list...

Fig. 7B

```
function telephoneCheck(str) {  
  var isphone = /^(1\s|1)?((\d{3}\)|\d{3})?(\d{3})?(\d{4})?$/; // test(str);  
  return isphone;  
}  
  
function MaskPII(context){  
  document.getElementsByTagName('span').forEach(function(el) {  
    if ( ! telephoneCheck(el.innerText)) {  
      el.innerText = '[MASKED]';  
    }  
  });  
}
```

Fig. 7C

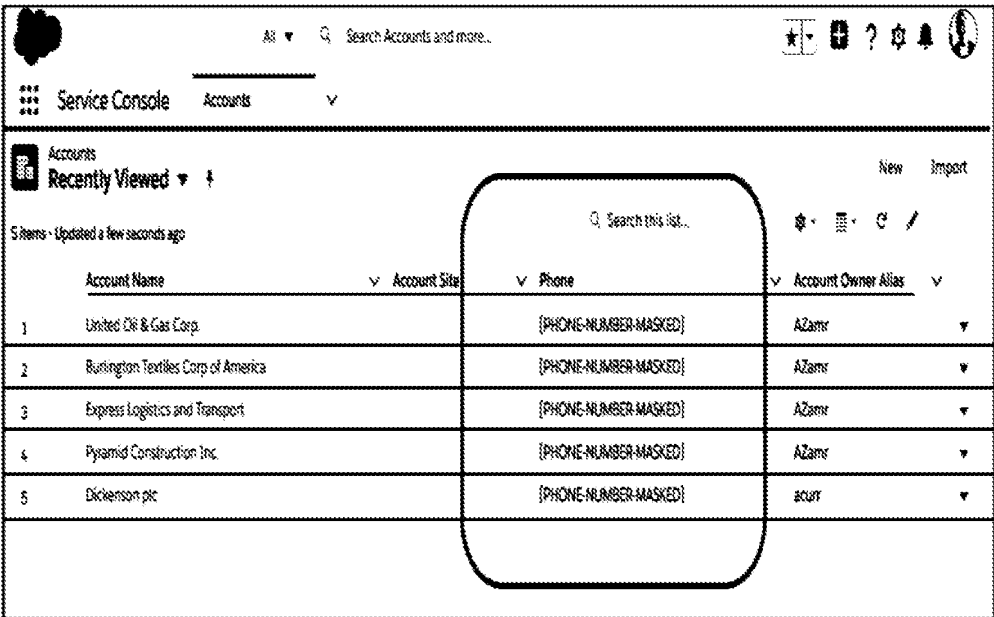


Fig. 7D

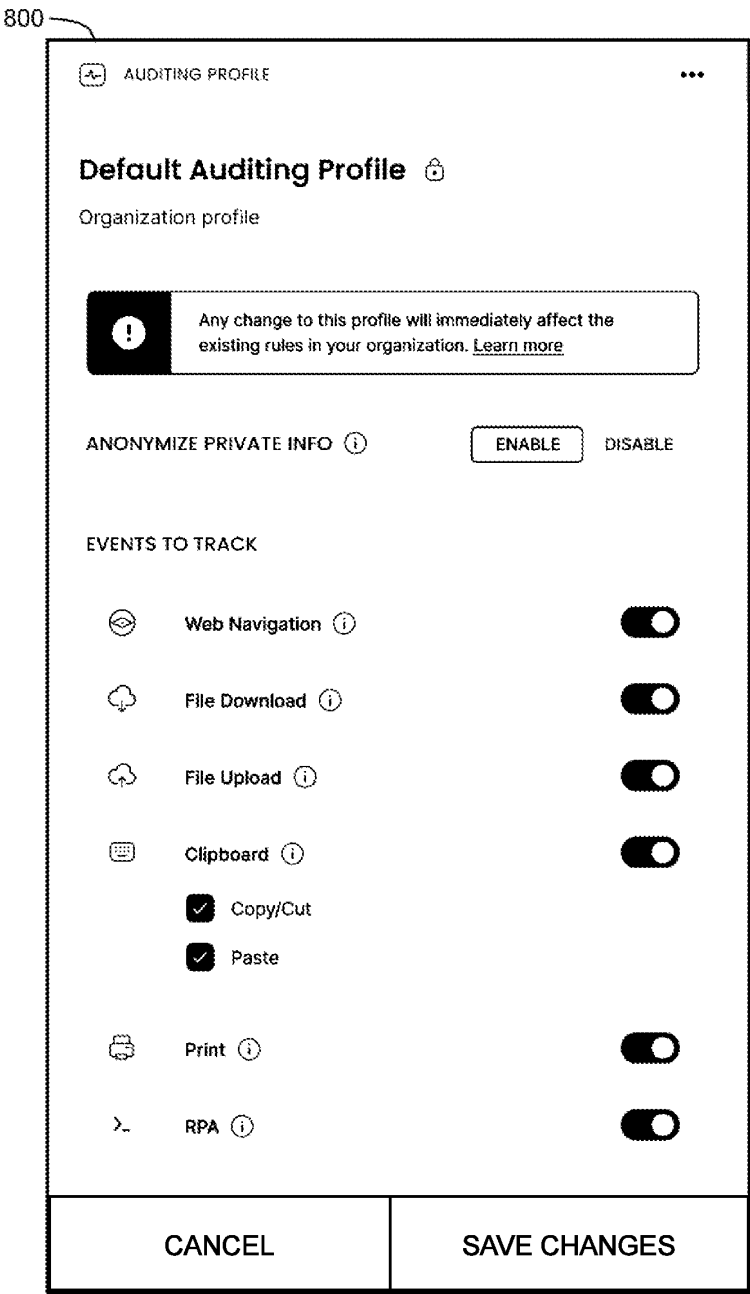


Fig. 8A

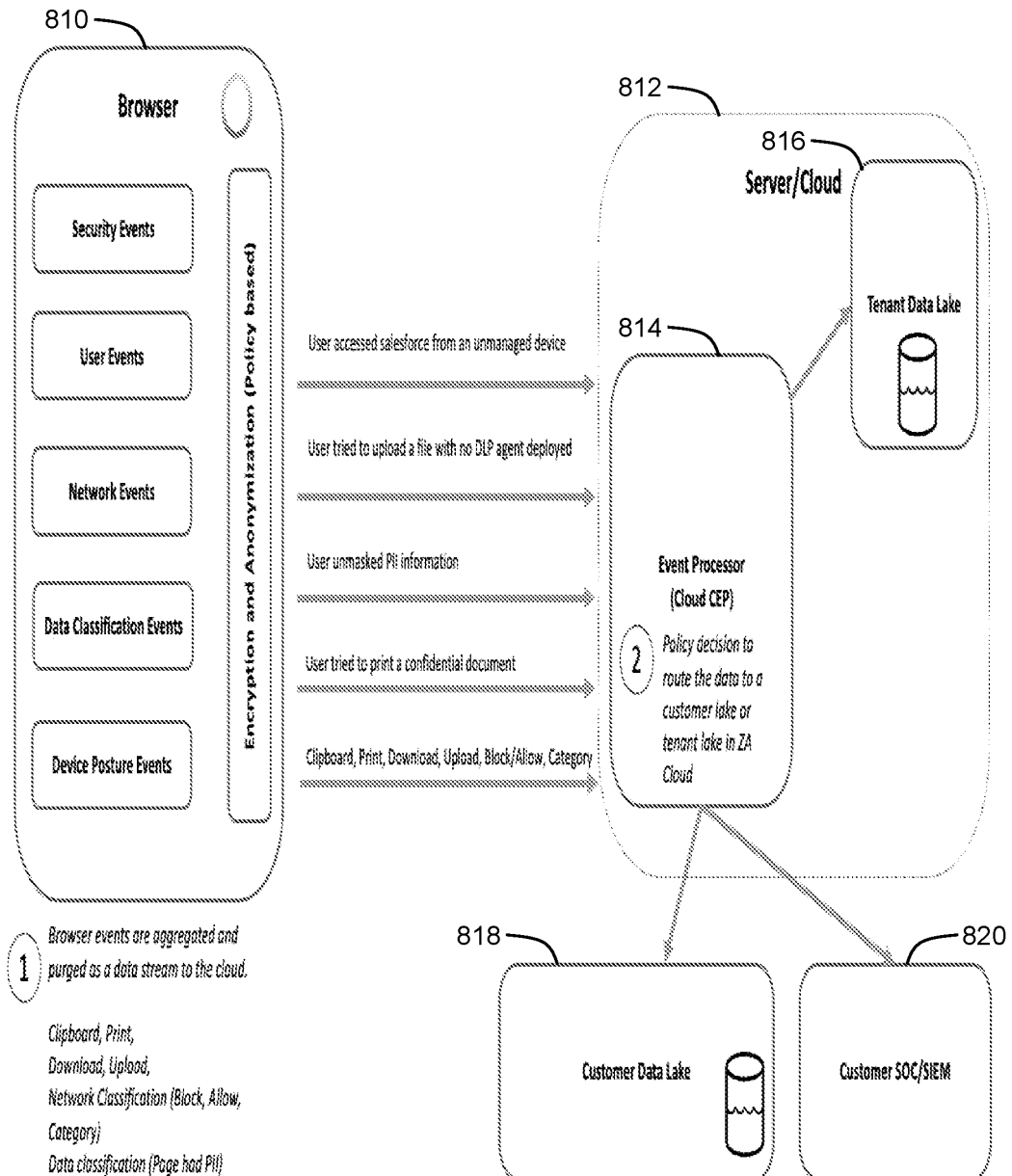


Fig. 8B

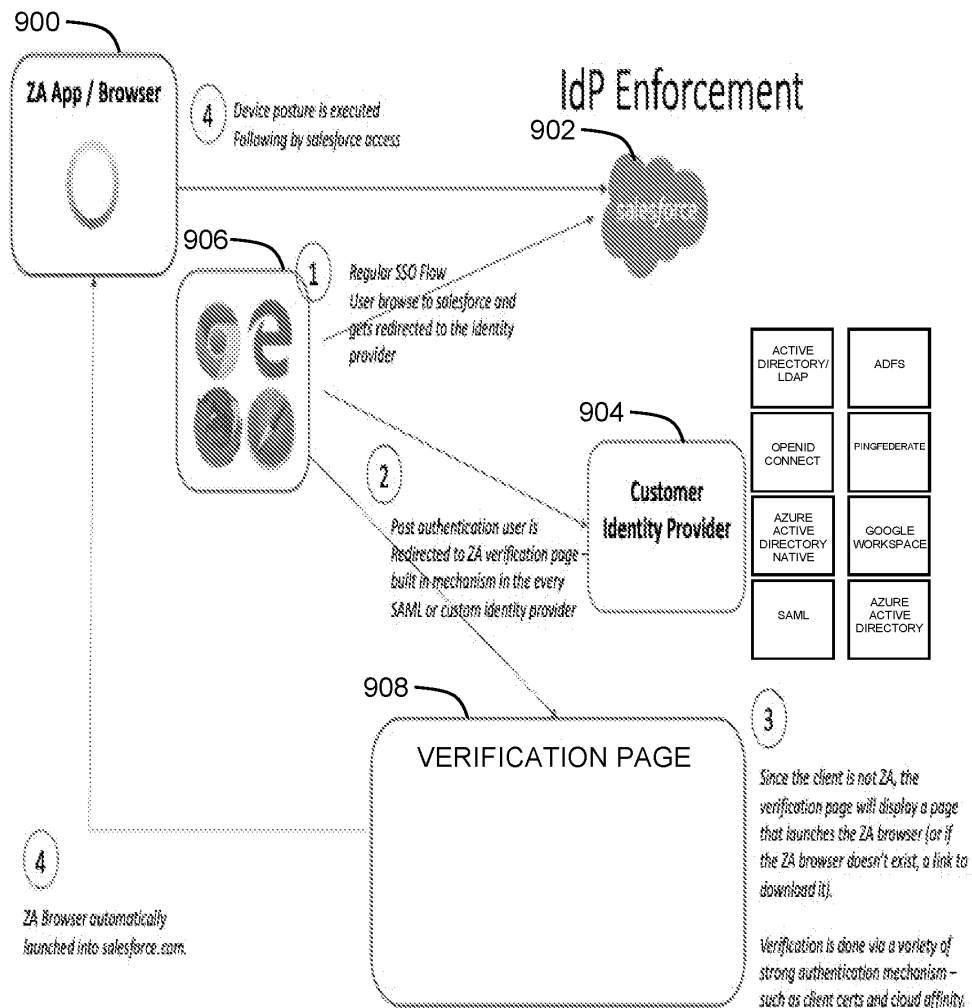


Fig. 9A

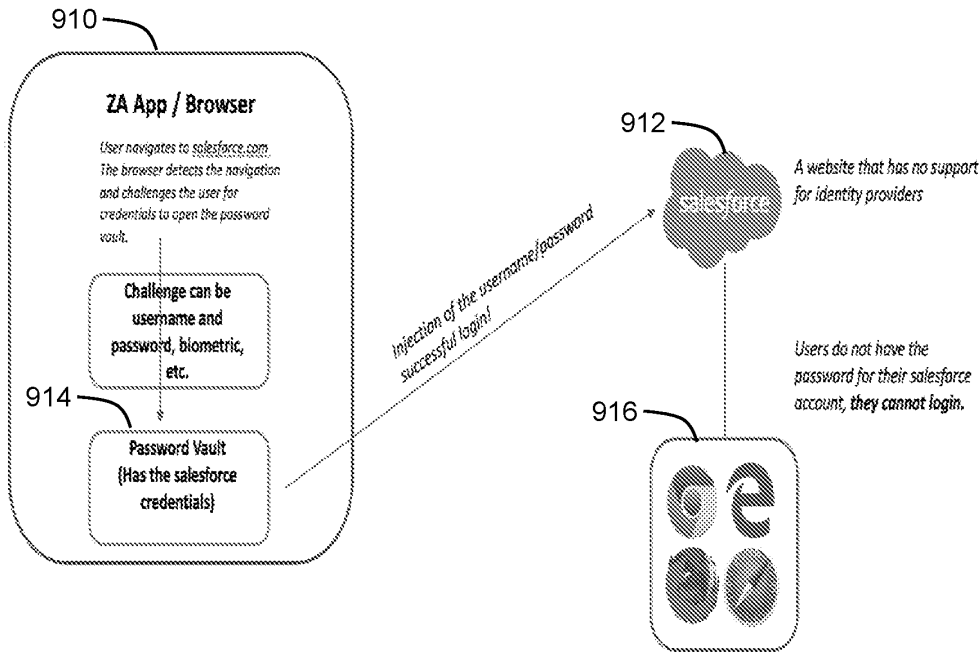


Fig. 9B

29/36

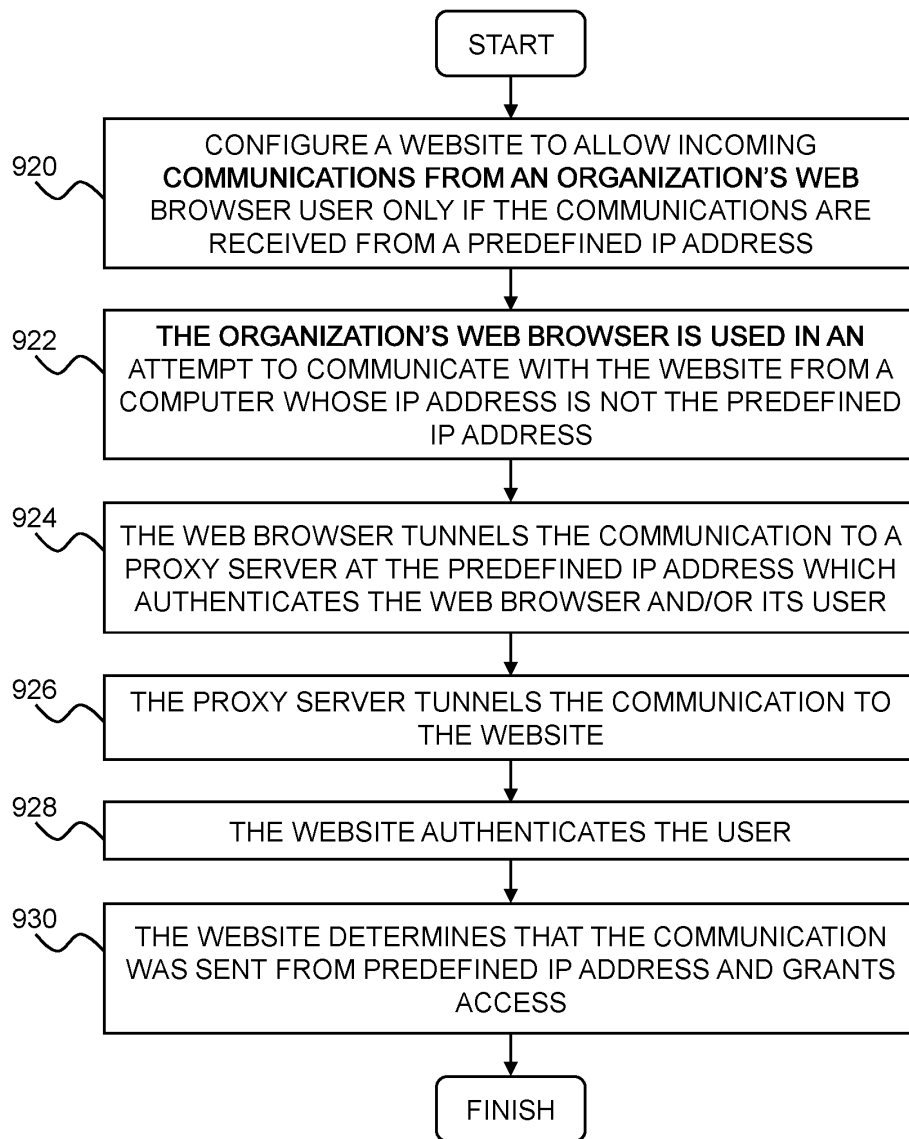


Fig. 9C

```
#include "extensions/common/extension.h"

namespace extensions {

ExtensionFunction::ResponseAction ZaPolicyPathExistsFunction::Run() {
    auto params = api::za_policy::PathExists::Params::Create(*args_);
    EXTENSION_FUNCTION_VALIDATE(params.get());

    const base::FilePath file_path = base::FilePath::FromUTF8Unsafe(params->path);
    const bool path_exists = base::PathExists(file_path);

    return RespondNow(
        ArgumentList(api::za_policy::PathExists::Results::Create(path_exists)))
    }

} // namespace extensions
```

Fig. 10A

```

void ZapPolicyAPIImpl::DispatchIsClipboardPasteAllowed(
    const GURL& url,
    const GURL& main_frame_url,
    IsPasteAllowedCallback callback) {
    EventRouter* event_router = EventRouter::Get(browser_context_);

    for (auto& loaded_extension : is_paste_allowed_requests_) {
        const std::string& extension_id = loaded_extension.first;

        if (event_router->ExtensionHasEventListener(
            extension_id,
            api::za_policy::IsClipboardPasteAllowed::kEventName)) {
            const int request_id = loaded_extension.second.Add(std::move(callback));
            std::unique_ptr<base::ListValue> internal_args(new base::ListValue);
            // Request id is not part of the public API and it will be massaged out in
            // custom bindings.
            internal_args->AppendInteger(request_id);
            auto details = std::make_unique<base::DictionaryValue>();
            details->SetString("url", url.spec());
            details->SetString("mainFrameUrl", main_frame_url.spec());
            internal_args->Append(std::move(details));

            auto event = std::make_unique<Event>(<
                events::ZA_POLICY_IS_CLIPBOARD_PASTE_ALLOWED,
                api::za_policy::IsClipboardPasteAllowed::kEventName,
                std::move(internal_args));
            event_router->DispatchEventToExtension(extension_id, std::move(event));
            return;
        }
    }

    std::move(callback).Run(true);
}

```

Fig. 10B

```
var config = {
  mode: "pac_script",
  pacScript: {
    data: "function FindProxyForURL(url, host) {\n" +
      "  if (host == 'foobar.com')\n" +
      "    return 'PROXY blackhole:80';\n" +
      "  return 'DIRECT';\n" +
      "}\n"
  }
};
chrome.proxy.settings.set(
  {value: config, scope: 'regular'},
  function() {}
);
```

Fig. 11

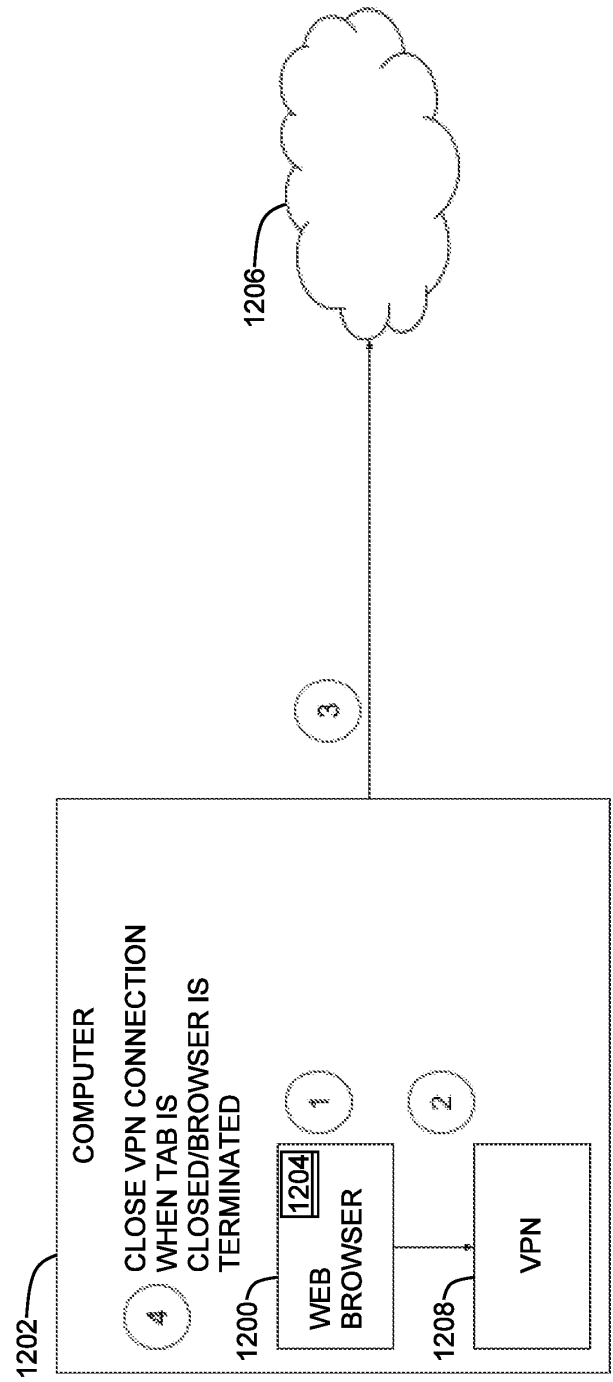


Fig. 12A

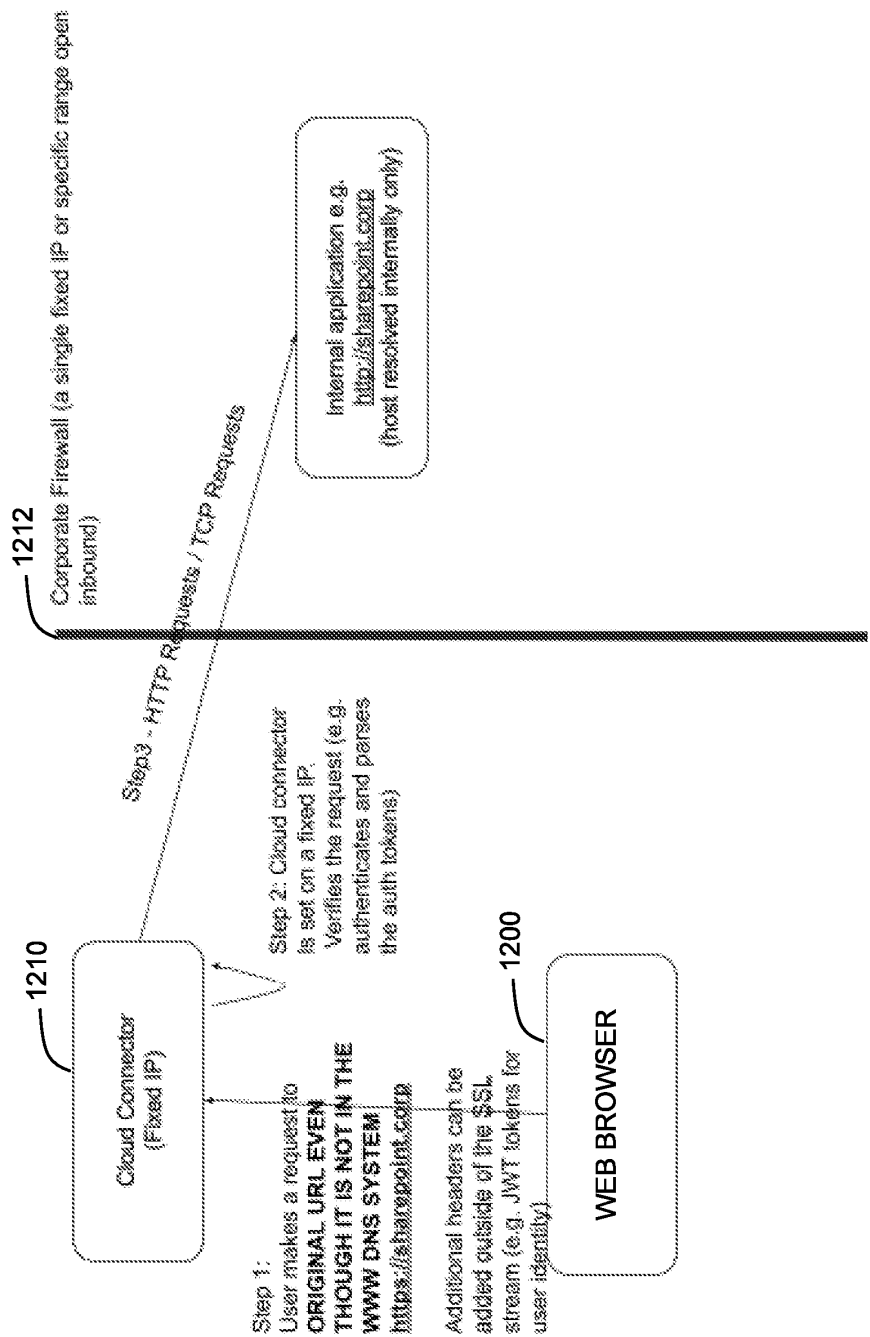


Fig. 12B

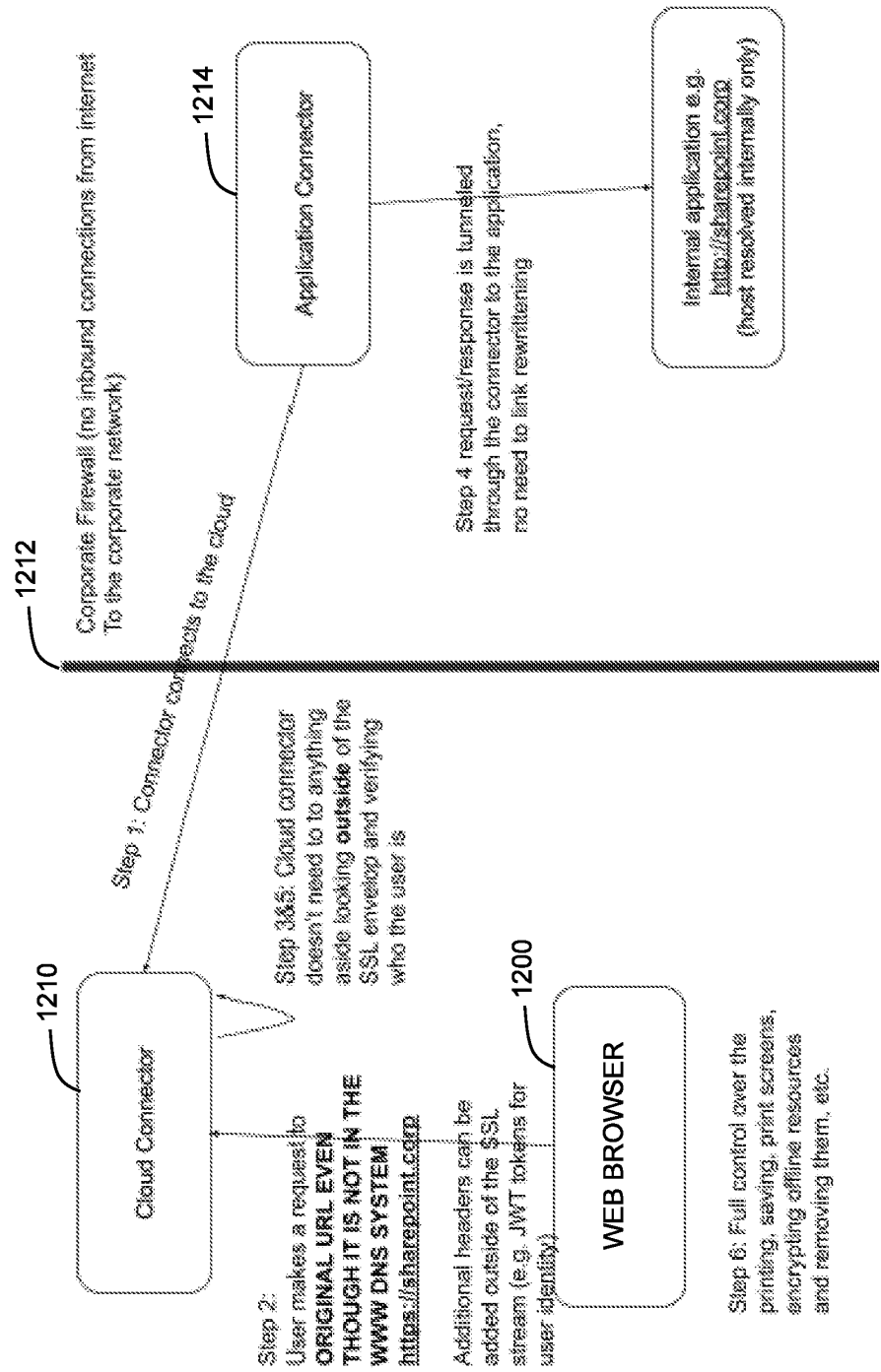


Fig. 12C

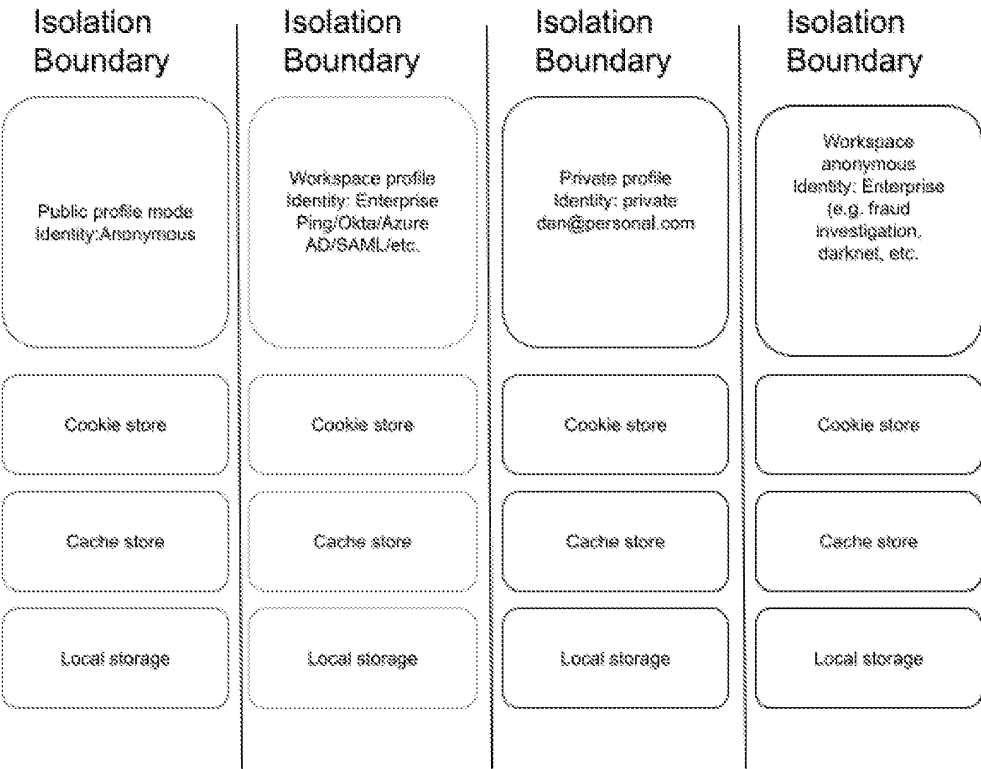


Fig. 13