

# Virtualization + Cloud & Security

Mgr. Michael Grafnetter



# Agenda

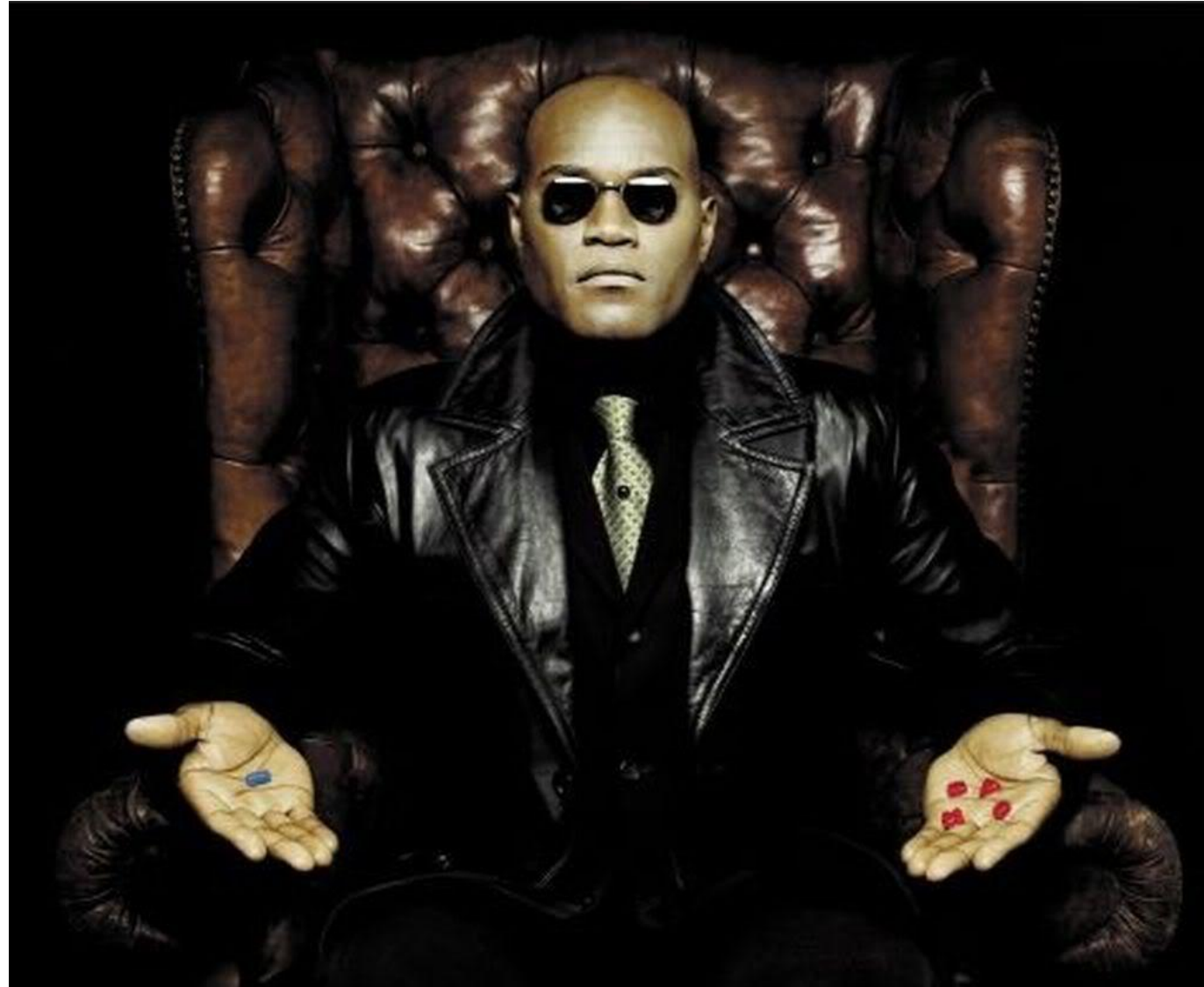
- Virtualization Security Risks and Solutions
- Cloud Computing Security Risks
- Authentication in Cloud Applications

[dsinternals.com/nswi150](https://dsinternals.com/nswi150)

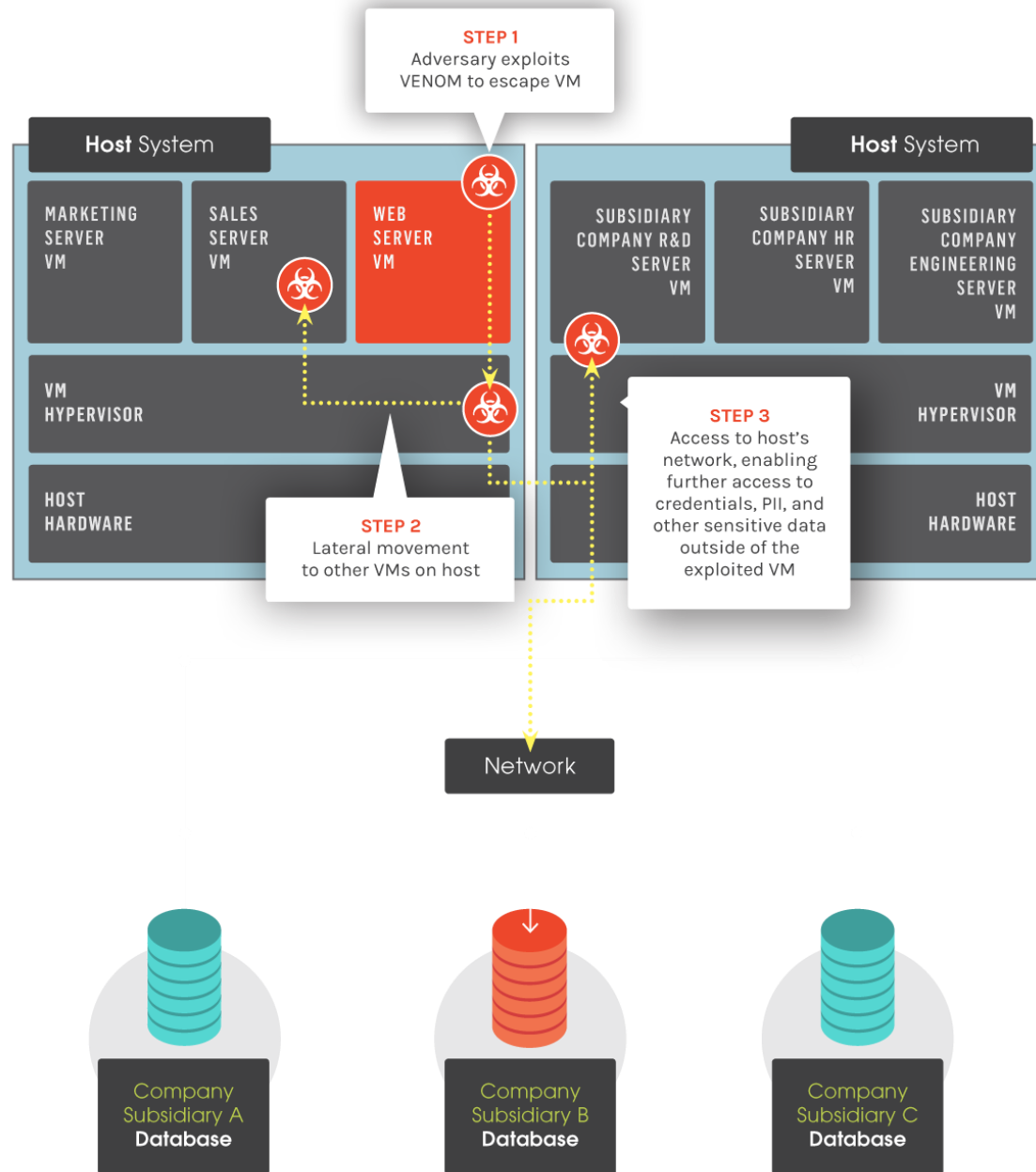
# Virtualization Security Risks



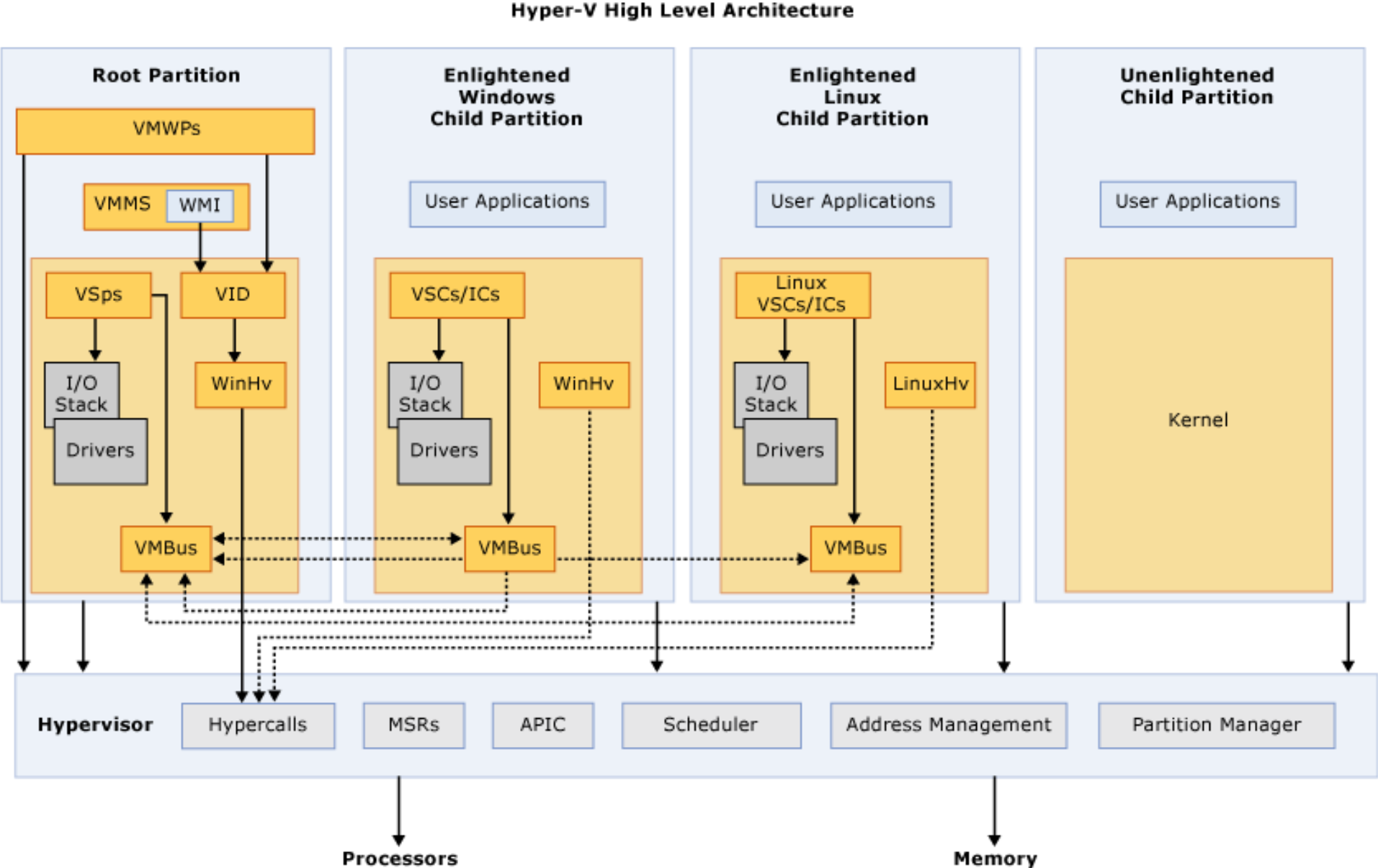
# Blue Pill Attack



# VM Escape Vulnerabilities



# VM Escape Vulnerabilities



# Known VM Escape Vulnerabilities

CVE-2007-1744	VMware Workstation
CVE-2008-0923	VMware Workstation
CVE-2009-1244	VMware ESXi, Workstation, Fusion
CVE-2012-0217	Xen Hypervisor
CVE-2014-0983	Oracle VirtualBox
CVE-2015-3456	QEMU
CVE-2015-7835	Xen Hypervisor
CVE-2016-6258	Xen Hypervisor
CVE-2016-7092	Xen Hypervisor
CVE-2017-0075	Microsoft Hyper-V
CVE-2017-0109	Microsoft Hyper-V
CVE-2017-4903	VMware ESXi, Workstation, Fusion
CVE-2017-4934	VMware Workstation, Fusion
CVE-2017-4936	VMware Workstation, Horizon View
CVE-2018-2698	Oracle VirtualBox

# CPU Vulnerabilities





# VLAN Hopping

**VLANs** are essential for segmenting traffic and **enhancing network security**. ▶

**However,**

**misconfigurations** can lead to **VLAN hopping attacks**, where **attackers exploit vulnerabilities** to gain **unauthorized access** to different **VLANs**.

## How VLAN Hopping Works

▼  
**Switch Spoofing:** Attackers configure their device to act like a switch, gaining access to multiple VLANs.

▶ **Double Tagging:** Attackers embed two VLAN tags in packets, bypassing VLAN boundaries to reach unauthorized segments.

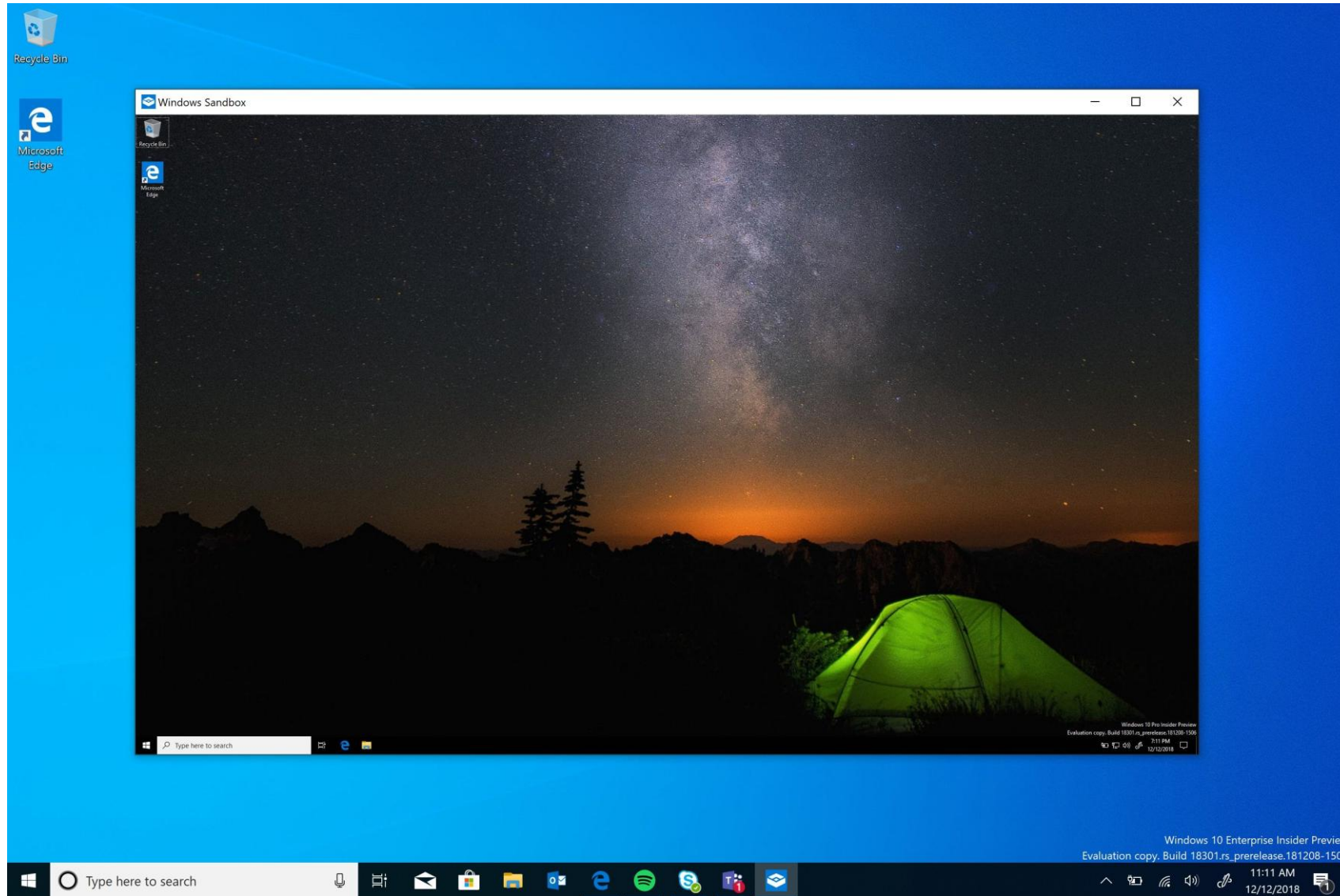
# Other Risks of Virtualization

- Introduction of yet another OS
- Reliance on traditional barriers
- Accelerated provisioning
- Security left to non-traditional security staff
- Audit scope creep

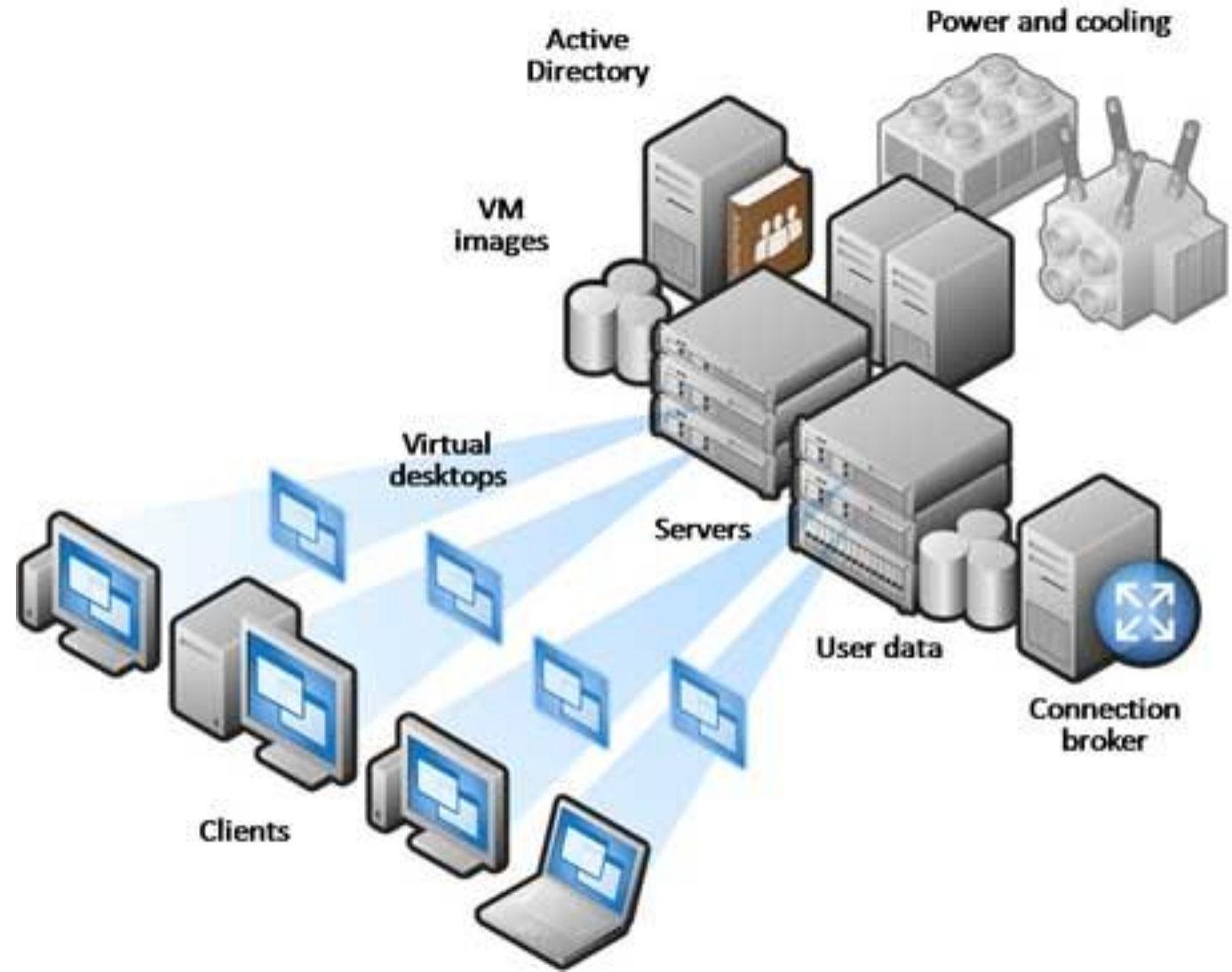
# Virtualization-Based Security Solutions



# Sandboxing



# Virtual Desktop Infrastructure



# Windows 10+ Virtual Secure Mode: Motivation

```
Authentication Id : 0 ; 2594251 (00000000:002795cb)
Session          : Service from 0
User Name       : svc-SQLAnalysis
Domain         : ADSECLAB
SID            : S-1-5-21-1473643419-774954089-2222329127-1608
```

```
msv :
```

```
0000000021 Primary
```

```
* Username : svc-SQLAnalysis
* Domain   : ADSECLAB
* NTLM     : 3c917b61c58c4cba165396aad7d140a2
* SHA1    : f089edb437e1f455ac1ab65886ed51959df7dc30
```

```
tspkg :
```

```
* Username : svc-SQLAnalysis
* Domain   : ADSECLAB
* Password : ThisIsAnOKPassword99!
```

```
wdigest :
```

```
* Username : svc-SQLAnalysis
* Domain   : ADSECLAB
* Password : ThisIsAnOKPassword99!
```

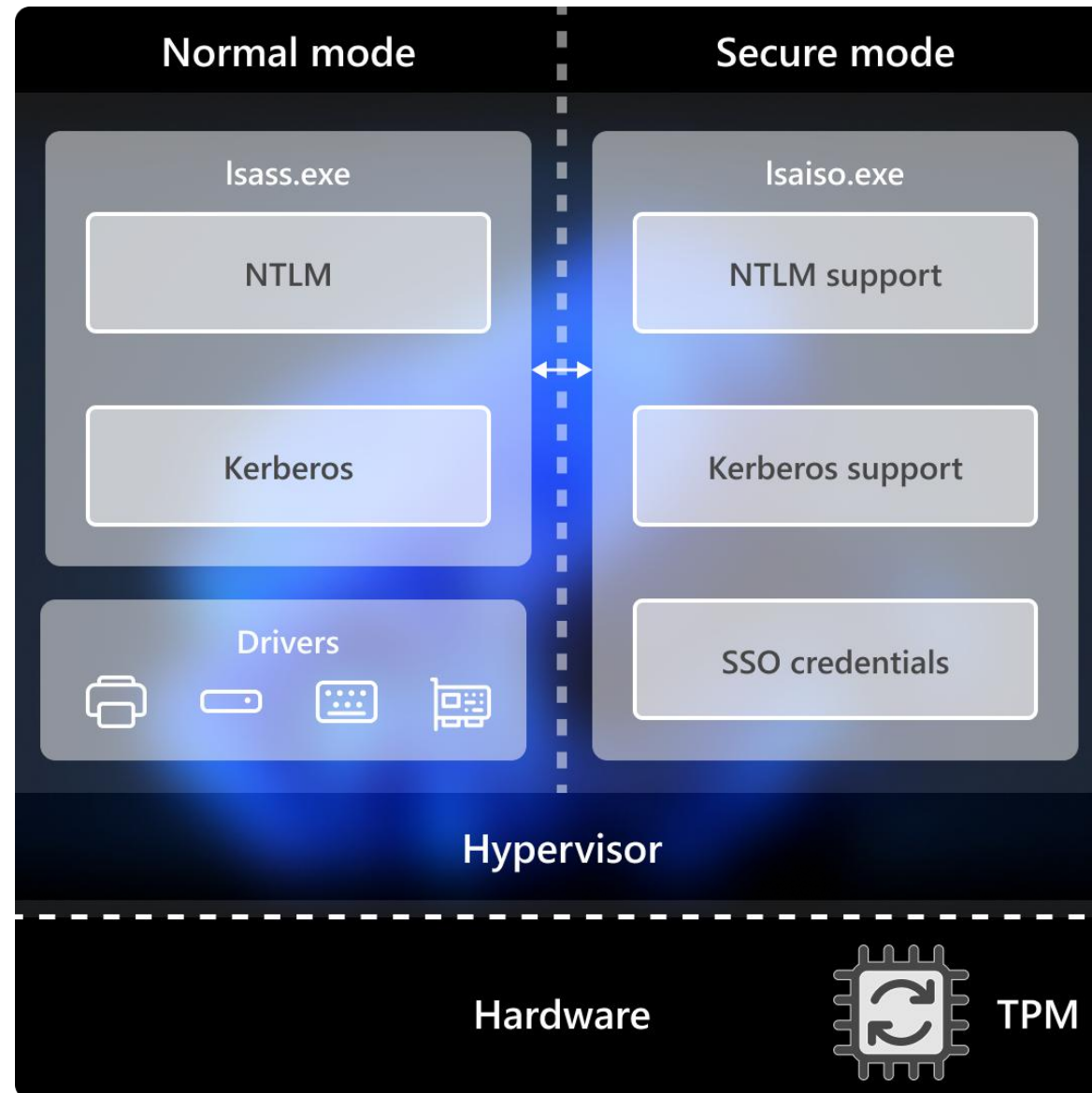
```
kerberos :
```

```
* Username : svc-SQLAnalysis
* Domain   : LAB.ADSECURITY.ORG
* Password : ThisIsAnOKPassword99!
```

```
ssp :
```

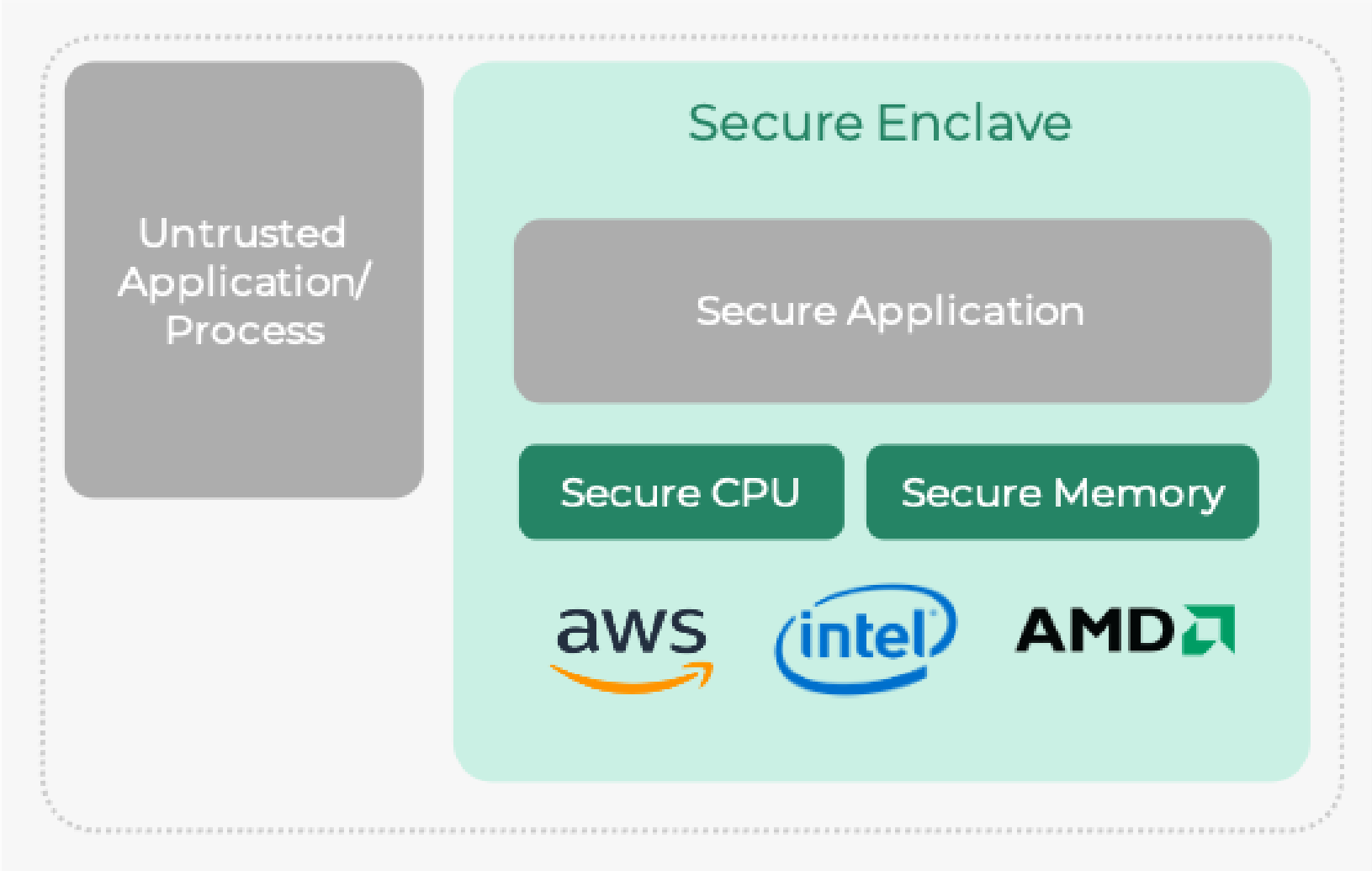
```
credman :
```

# Windows 10+ Virtual Secure Mode



# Secure Enclaves / Trusted Computing

## Cloud Instance/On-Prem Server





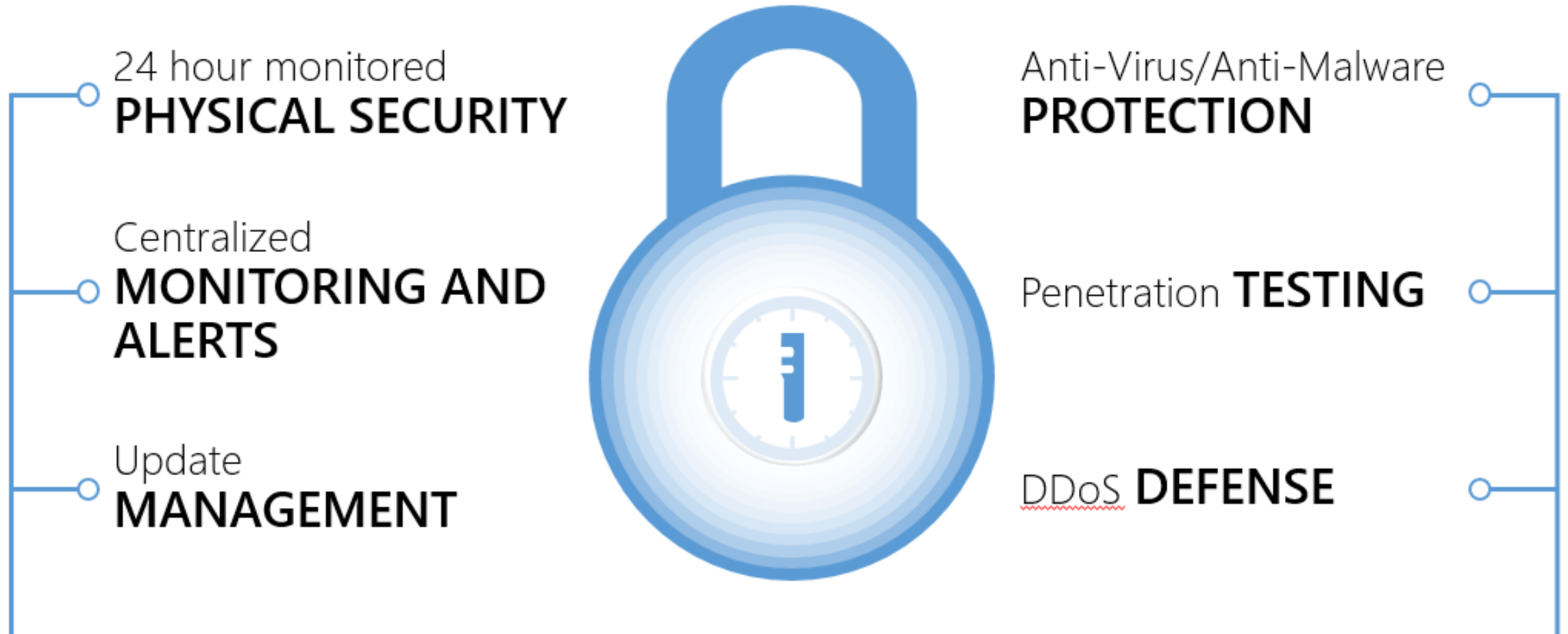
# Cloud Computing Security Risks



Who has access to our data?



# Example: Azure Datacenter Security



# Physical Security



# Physical Security



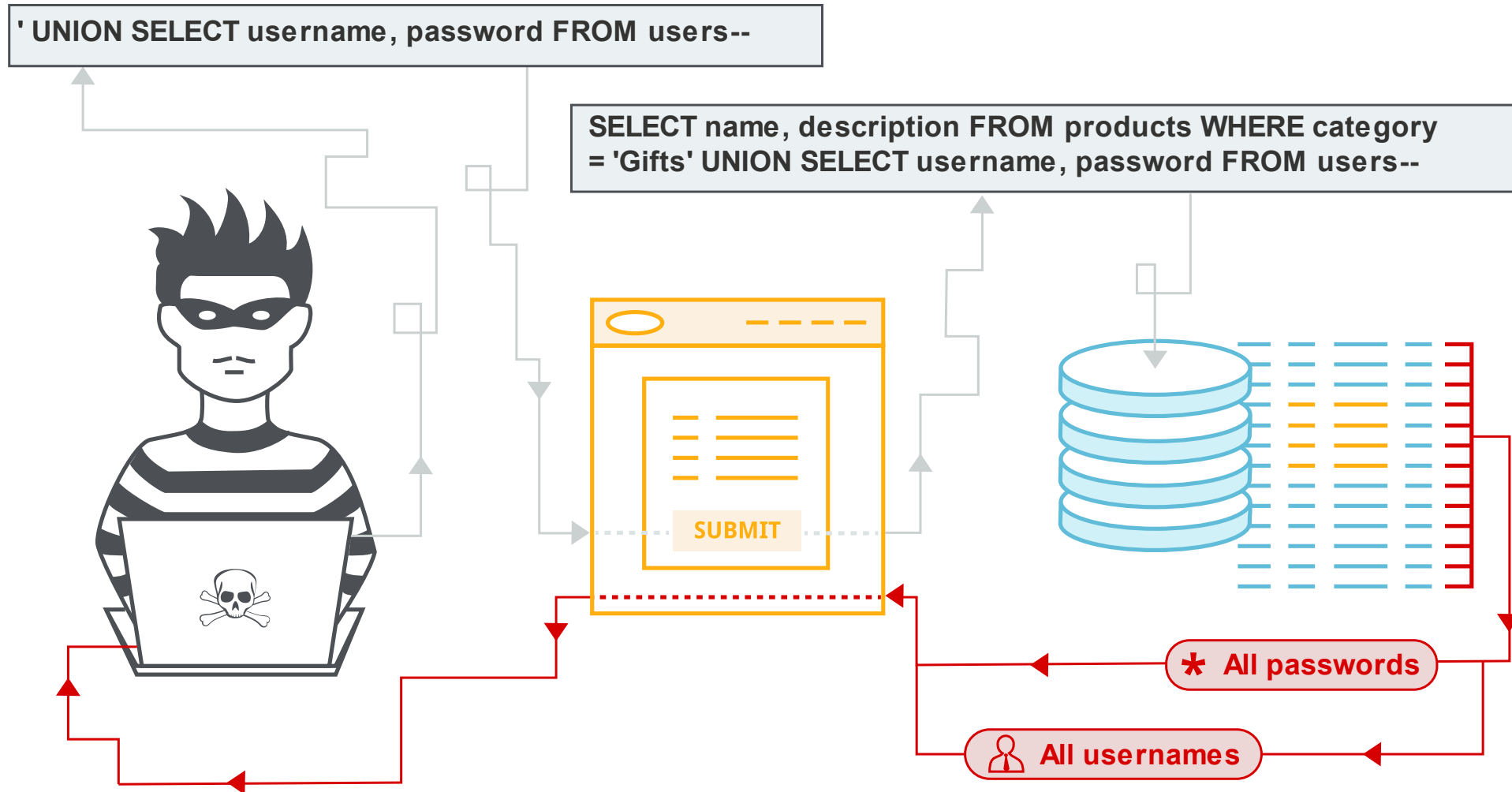
# Shared Responsibility (Misconfigurations)



# OWASP Top 10 (2021) ~ IaaS, PaaS, Serverless

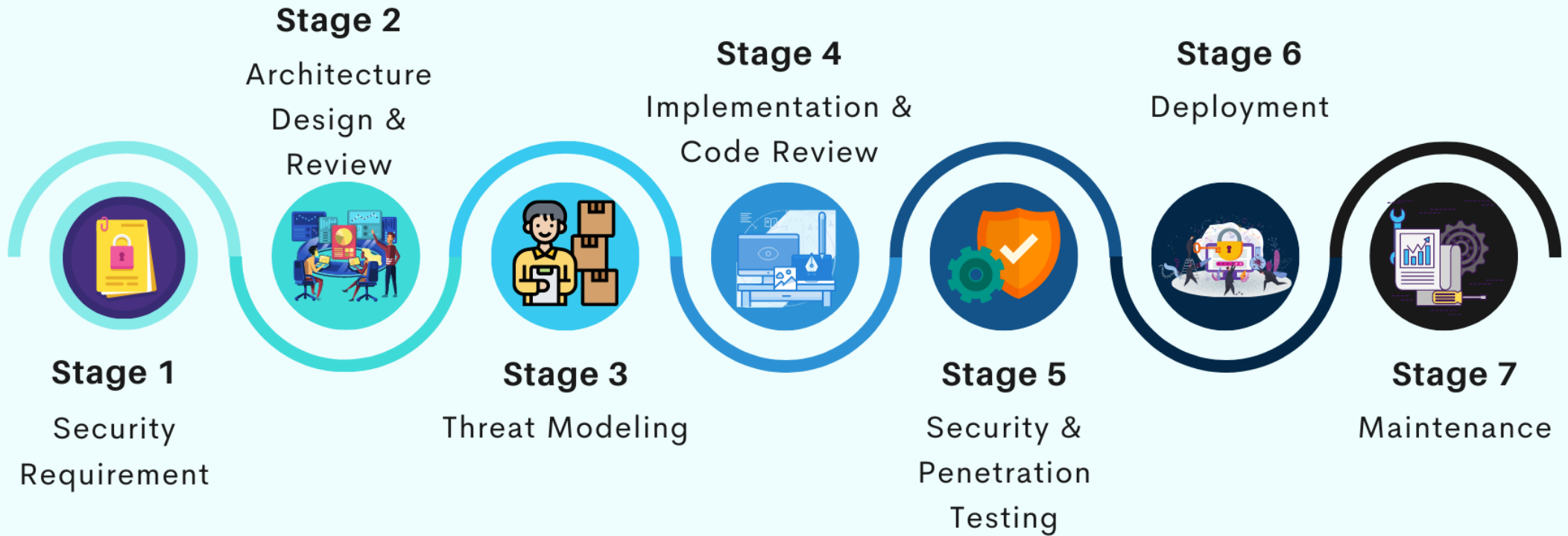
- A01: Broken Access Control
- A02: Cryptographic Failures
- A03: Injection
- A04: Insecure Design
- A05: Security Misconfiguration
- A06: Vulnerable and Outdated Components
- A07: Identification and Authentication Failures
- A08: Software and Data Integrity Failures
- A09: Security Logging and Monitoring Failures
- A10: Server-Side Request Forgery


# Example: SQL Injection





# Secure SDLC



 +1 (888) 958-0554  
contact@iosentrix.com

**ioSENTRIX**  
ioSENTRIX LLC, All right reserved

# OWASP Cloud-Native Application Security Top 10 (2022)

- CNAS-1: Insecure cloud, container or orchestration configuration
  - Publicly open S3 bucket
  - Container runs as root
  - Container shares resources with the host (network interface, etc.)
  - Insecure Infrastructure-as-Code (IaC) configuration
- CNAS-2: Injection flaws (app layer, cloud events, cloud services)
  - SQL injection
  - XXE
  - NoSQL injection
  - OS command injection
  - Serverless event data injection

# Default Credentials

the **INQUIRER**

[al Intelligence](#) [Internet of Things](#) [Open Source](#) [Hardware](#) [Software](#) [Security](#)

[Action >](#)

[Huawei sues FCC >](#)

[Xerox vs HP >](#)

[Galaxy S11 >](#)

[McAfee 2020 >](#)

## **Equifax used default 'admin' password to secure hacked portal**

Lawsuit claims firm failed to take even 'the most basic precautions'

# PII in Public AWS S3 Buckets

☰ SPIEGEL International

**We Know Where You Parked**

## Massive Data Breach at VW Raises Questions about Vehicle Privacy

Already facing significant headwinds, VW has now been hit by a data protection nightmare. Location data from 800,000 electric vehicles and contact info from owners was accessible unprotected on the internet. And the company didn't even know about it.

By [Patrick Beuth](#), [Flüpke](#), [Max Hoppenstedt](#), [Michael Kreil](#), [Marcel Rosenbach](#) und [Rina Wilkin](#)

03.01.2025, 16.19 Uhr



# OWASP Cloud-Native Application Security Top 10 (2022)

- CNAS-3: Improper authentication & authorization
  - Unauthenticated API access on a microservice
  - Over-permissive cloud IAM role
  - Lack of orchestrator node trust rules (e.g. unauthorized hosts joining the cluster)
  - Unauthenticated orchestrator console access
  - Unauthorized or overly-permissive orchestrator access

# Sample Bug: Azure Cosmos DB



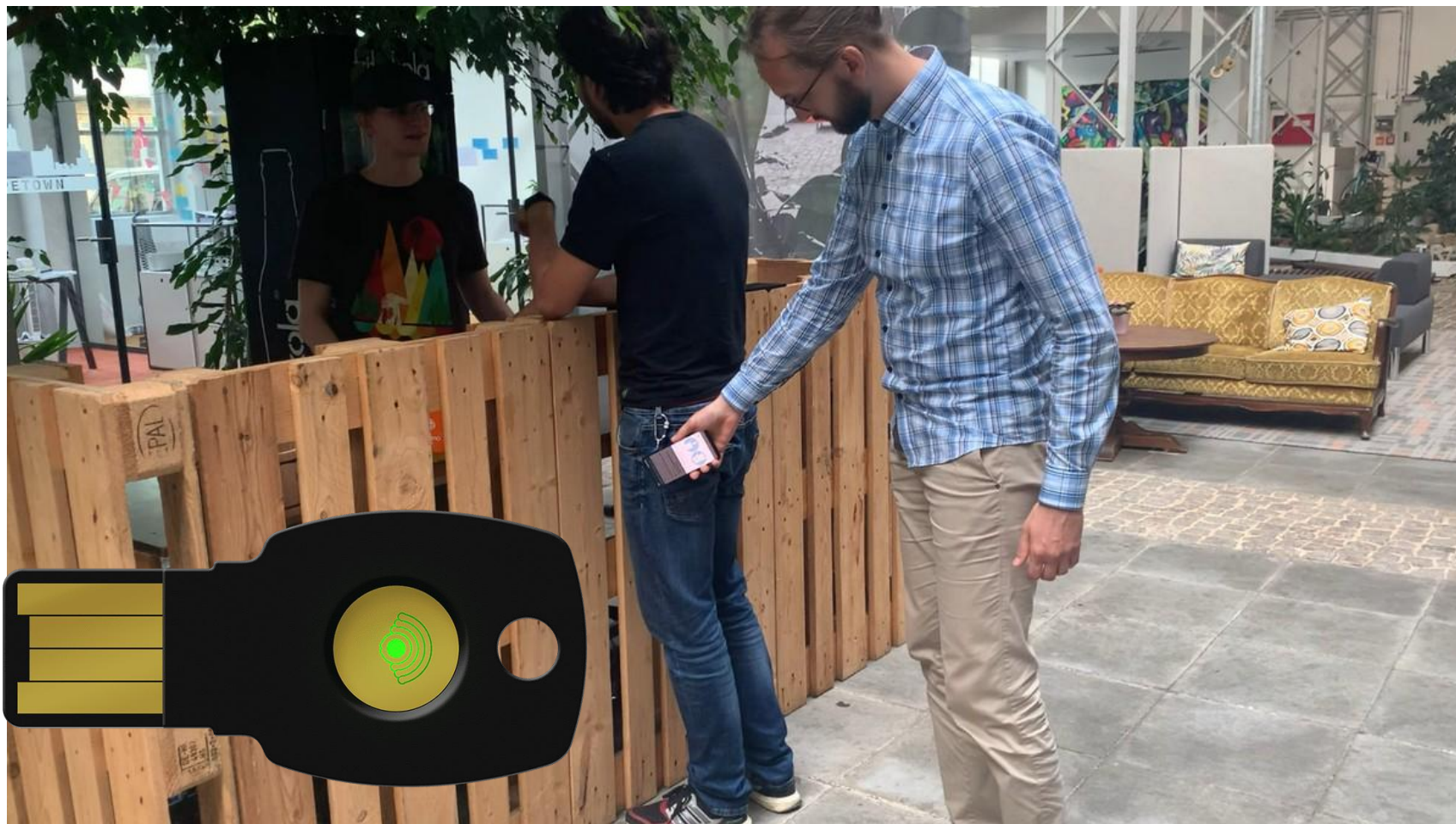
World ▾ Business ▾ Legal ▾ Markets ▾ Breakingviews ▾ Technology ▾ Investigations Sports ▾ More ▾

August 28, 2021  
12:06 AM GMT+2  
Last Updated 7 months ago

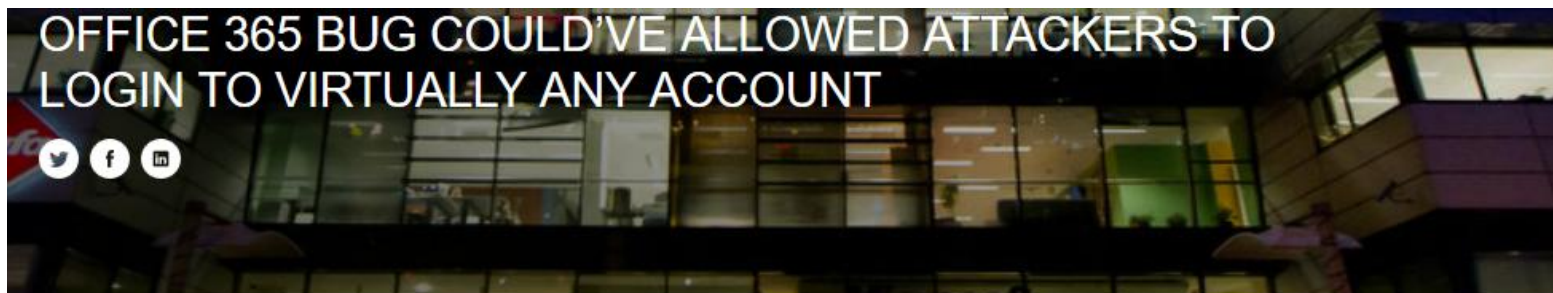
Disrupted

## **EXCLUSIVE Microsoft warns thousands of cloud customers of exposed databases**

# Sample Azure AD Bug: PIN Bypass in Passkey Authentication



# Sample Azure AD Bug: Missing UPN Suffix Validation in SAML



[Pindrop](#) > [Blog](#) > [Office 365 Bug Could've Allowed Attackers to Login to Virtually Any Account](#)

Security researchers in January discovered a critical vulnerability in the SAML implementation in Microsoft's Office 365 service that could allow an attacker to log in to a victim's account and gain full access to email, contacts, and other sensitive data.

The vulnerability was present in Office 365 for an unknown amount of time, and there is a long list of prominent companies that use the Web-based email and productivity suite, including Telefonika, British Telecom, Verizon, Microsoft, Cisco, Intel, and many others. The researchers who discovered the flaw said that an attacker targeting the vulnerability would be able to get to a wide range of important information on a victim's account.

## RELATED POSTS

[Google Adds New Anti-Phishing Feature to G Suite](#)

[Europol Dismantles International Fraud Ring](#)

[Phishing Attacks Using SSL Spike](#)

[Apple to Switch Users to 2FA on iOS 11, macOS High Sierra](#)

[Preventing Forgery With Paper Fingerprinting](#)



# Sample Azure AD Bug: Missing UPN Suffix Validation in SAML

```
<AttributeStatement>  
  <Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn">  
    <AttributeValue>michael@dsinternals.com</AttributeValue>  
  </Attribute>  
</AttributeStatement>  
<AuthnStatement AuthnInstant="2021-01-11T06:26:33.297Z">  
  <AuthnContext>  
    <AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProt
```

# Sample Azure AD Bug: Exposed DPAPI Encryption Keys

```
Windows PowerShell
PS C:\> $uri = 'https://graph.windows.net/myorganization/users?&api-version=1.6-internal'
PS C:\> Invoke-RestMethod -Uri $uri -Headers $authHeader -UseBasicParsing |
>>   foreach value |
>>   select userPrincipalName,@{ n = 'DPAPIKey'
>>                               e = { $PSItem.searchableDeviceKey | where usage -eq DPAPI | foreach keyMaterial } } |
>>   where DPAPIKey -ne $null

userPrincipalName      DPAPIKey
-----
alice@contoso.com      D/V65EQK178KWIBczSeDkQ7k7tj6+4KWgHIhNiBINJo=
bob@contoso.com        Rfa+WvKxqqkrjzjR2Qf6AhDibpAg/7/uVT1fDb993ng=
charlie@contoso.com    9jjEDtfQmqn1tIV4Mgbr3bk0v9g0hZGwq9U16p5K/sU=

PS C:\>
```

# Reporting Bugs to MSRC

Microsoft MSRC | Researcher Portal 🔍 ⚙️ 👤

## My vulnerability reports + Create report

All Pending Reviewing Developing Complete Additional Info Needed

Title/Short description	Status	Bounty	Created On	Last modified	Report ID	Case Number	Security Impact	Reported Products	Points
<a href="#">LsaRetrievePrivateData function vulnerable to known-plaintext...</a>	Complete	-	Jul 31, 2023, 8:03 PM	Jan 13, 2024, 12:50 AM	VULN-106312	81278	Information Disclosure	Windows	—
<a href="#">Orphaned user NGC keys are left in Active Directory after devic...</a>	Complete	Awarded Closed	Sep 29, 2019, 1:08 AM	—	VULN-010399	54282	Elevation Of Privilege	Windows	—
<a href="#">Application of KB4046462 might introduce a backdoor into Act...</a>	Complete	Out of Scope Closed	Sep 28, 2019, 9:19 PM	—	VULN-010397	54250	Security Feature Bypass	Windows	—
<a href="#">Azure AD Graph API exposes all DPAPI encryption keys to auth...</a>	Complete	Out of Scope Closed	Feb 16, 2020, 1:50 PM	—	VULN-016808	56666	Security Feature Bypass	Graph Services	—

# Microsoft Identity Bounty Program

Security Impact	Report Quality	Severity Critical	Important	Moderate	Low
<b>Elevation of Privilege</b> <i>(Involving Multi-factor Authentication Bypass)</i>	High	\$100,000	\$50,000		
	Medium	\$75,000	\$25,000	\$0	\$0
	Low	\$45,000	\$10,000		
<b>Elevation of Privilege</b> <i>(e.g Authentication Bypass or authentication flaw)</i>	High	\$40,000	\$20,000		
	Medium	\$20,000	\$10,000	\$0	\$0
	Low	\$8,000	\$2,500		
<b>Spoofing</b> <i>(e.g Cross-Site Scripting (XSS) or Cross-Site Request Forgery CSRF)</i>	High	\$20,000	\$10,000		
	Medium	\$10,000	\$5,000	\$0	\$0
	Low	\$6,000	\$1,500		
<b>Information Disclosure</b> <i>(e.g. Sensitive Data Exposure)</i>	High	\$12,000	\$7,500		
	Medium	\$6,000	\$3,000	\$0	\$0
	Low	\$4,500	\$1,500		
<b>Standards Design Vulnerabilities</b> <i>(Some limitations apply, see Out of Scope section below)</i>	High	\$100,000	\$30,000		
	Medium	\$60,000	\$20,000	\$0	\$0
	Low	\$25,000	\$2,500		
<b>Standards-based implementation vulnerabilities</b> <i>(Some limitations apply, see Out of Scope section below)</i>	High	\$75,000	\$25,000		
	Medium	\$50,000	\$10,000	\$0	\$0
	Low	\$20,000	\$2,500		

# OWASP Cloud-Native Application Security Top 10 (2022)

- CNAS-4: CI/CD pipeline & software supply chain flaws
  - Insufficient authentication on CI/CD pipeline systems
  - Use of untrusted images
  - Use of stale images
  - Insecure communication channels to registries
  - Overly-permissive registry access
  - Using a single environment to run CI/CD tasks for projects requiring different levels of security
- CNAS-5: Insecure secrets storage
  - Orchestrator secrets stored unencrypted
  - API keys or passwords stored unencrypted inside containers
  - Hardcoded application secrets
  - Poorly encrypted secrets (e.g., use of obsolete encryption methods, use of encoding instead of encryption)
  - Mounting of storage containing sensitive information

# Passwords in Public AWS S3 Buckets



Cloud Security / Malware / Vulnerabilities / InfoSec Insider / Podcasts

## Verizon Wireless Internal Credentials, Infrastructure Details Exposed in Amazon S3 Bucket



# Secrets in Source Code Repositories

```
def ftp_file():
    ftp_host='[REDACTED]'
    ftp_user='[REDACTED]'
    ftp_password='[REDACTED]'
    ftp_destination='/Test/test/'
    source_file_name='/home/airflow/airflow/dags/ftpupload/weeklyrefresh.txt'

    with ftputil.FTPHost(ftp_host, ftp_user, ftp_password) as ftp_host:

        buf_size = 4 * 1024
        print("FTP Host: " + str(ftp_host))
        with open(source_file_name, 'rb') as source_file:
            # print("Opened source file: " + source_file.name)
            # ftp_host.makedirs(args.ftp_destination)
            print("Made FTP directory: " + ftp_destination)
            target_file_name = ftp_destination + get_name(source_file_name)
            with ftp_host.open(target_file_name, "wb") as target_file:
                # print("Opened target file: " + str(target_file_name))
                while True:
                    data=source_file.read(buf_size)
                    if not data: break
                    target_file.write(data)
                print("Copy complete: " + source_file_name)
```



UNIVERSAL MUSIC GROUP

# Hardware Security Module (HSM)





# OWASP Cloud-Native Application Security Top 10 (2022)

- CNAS-6: Over-permissive or insecure network policies
  - Over-permissive pod to pod communication allowed
  - Internal microservices exposed to the public Internet
  - No network segmentation defined
  - End-to-end communications not encrypted
  - Network traffic to unknown or potentially malicious domains not monitored and blocked
- CNAS-7: Using components with known vulnerabilities
  - Vulnerable 3rd party open-source packages
  - Vulnerable versions of application components
  - Use of known vulnerable container images

# Public RDP Access

welivesecurity™ BY eset®

## First BlueKeep attacks prompt fresh warnings

The infamous vulnerability has been exploited for a cryptocurrency mining campaign, but more damaging attacks may still be in store



Amer Owaida 11 Nov 2019 - 05:16PM

# Bugs in Applications

## The Hacker News

Subscribe to Newsletter

Home

Data Breaches

Cyber Attacks

Vulnerabilities

Malware

Offers

Contact



## Critical Bug Found in WordPress Plugin for Elementor with Over a Million Installations

February 01, 2022 Ravie Lakshmanan

Best Elementor Addon  
to supercharge your Elementor design

1+ Million  
Active Users

2000+ Five-Star  
Rating

23 Million+  
Downloads

4+ Years of  
Excellence

Light & Fast  
Loading

#1 Most Popular  
Addon

80+ Advanced  
Elements

Constant  
Development

Essential Addons for Elementor  
By WPDeveloper

Download

### Popular This Week



Chinese Hackers Target  
VMware Horizon Servers  
with Log4Shell to Deploy  
Rootkit



CISA Warns of Active  
Exploitation of Critical  
Spring4Shell Vulnerability



Apple Issues Patches for 2  
Actively Exploited Zero-Days  
in iPhone, iPad and Mac  
Devices

# OWASP Cloud-Native Application Security Top 10 (2022)

- CNAS-8: Improper assets management
  - Undocumented microservices & APIs
  - Obsolete & unmanaged cloud resources
- CNAS-9: Inadequate 'compute' resource quota limits
  - Resource-unbound containers
  - Over-permissive request quota set on APIs
- CNAS-10: Ineffective logging & monitoring (e.g. runtime activity)
  - No container or host process activity monitoring
  - No network communications monitoring among microservices
  - No resource consumption monitoring to ensure availability of critical resources
  - Lack of monitoring on orchestration configuration propagation and stale configs

# Subdomain Takeovers

**BLEEPINGCOMPUTER** f t YouTube

[NEWS](#) [DOWNLOADS](#) [VIRUS REMOVAL GUIDES](#) [TUTORIALS](#) [DEALS](#)

[Home](#) > [News](#) > [Security](#) > Starbucks Abandons Azure Site, Exposed Subdomain to Hijacking

## Starbucks Abandons Azure Site, Exposed Subdomain to Hijacking

By [Ionut Ilascu](#)

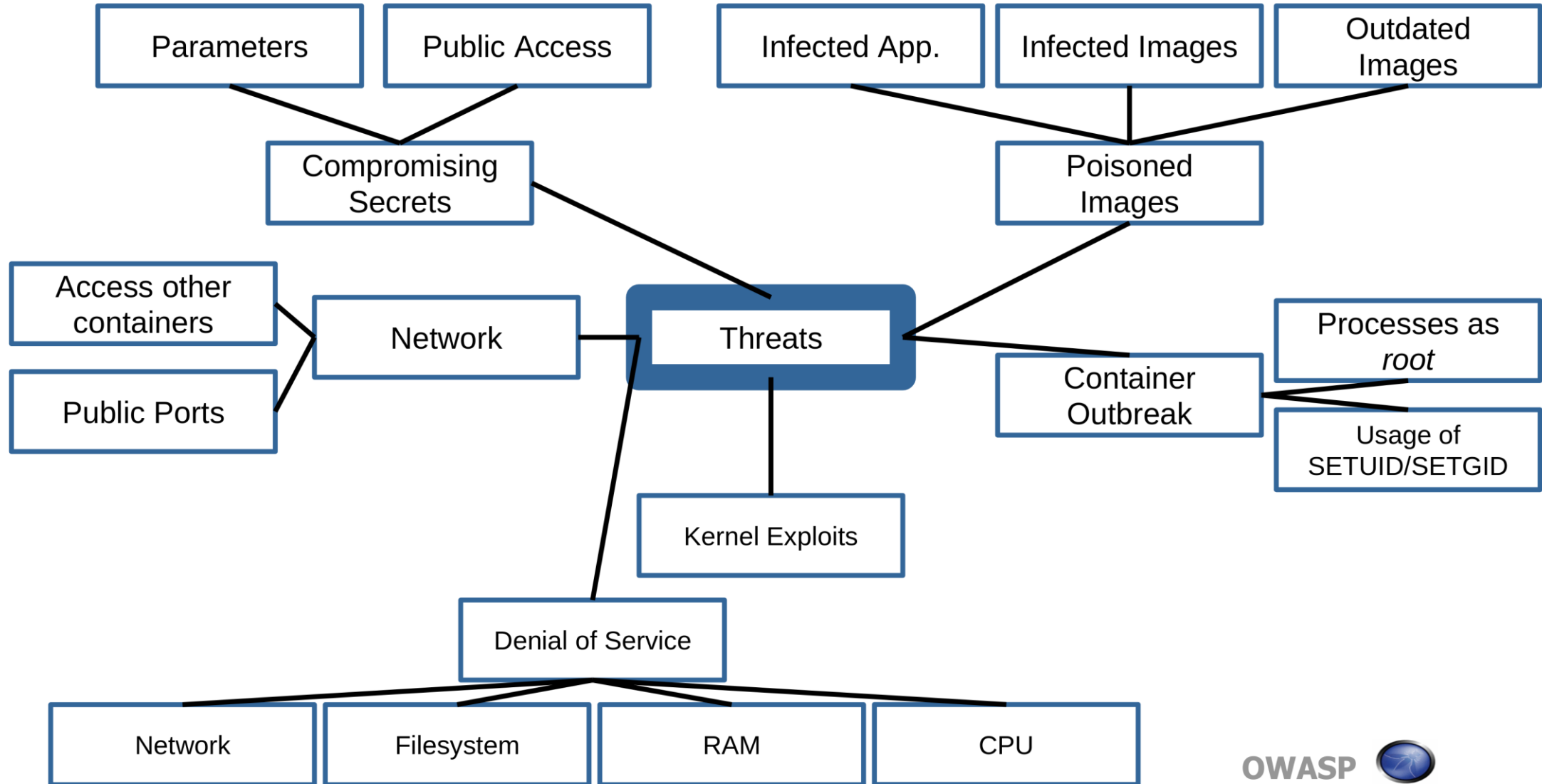
Calendar August 29, 2019 Clock 05:46 AM Comments 2



# OWASP Docker Top 10

- D01 - Secure User Mapping
- D02 - Patch Management Strategy
- D03 - Network Segmentation and Firewalling
- D04 - Secure Defaults and Hardening
- D05 - Maintain Security Contexts
- D06 - Protect Secrets
- D07 - Resource Protection
- D08 - Container Image Integrity and Origin
- D09 - Follow Immutable Paradigm
- D10 - Logging

# OWASP: Docker Threat Modeling



# OWASP Top 10 Low-Code/No-Code Security Risks

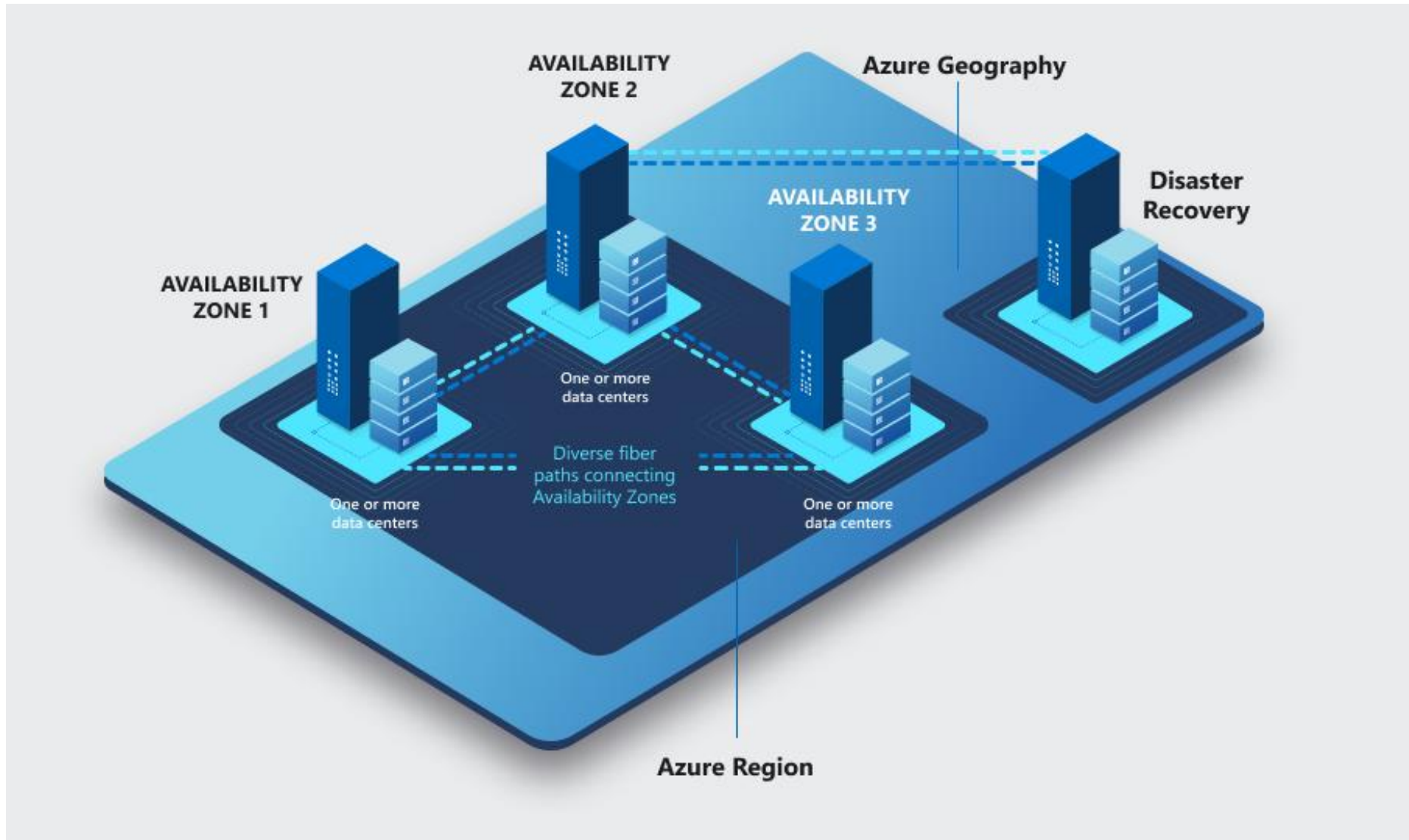
- Risk #1: Privilege Escalation
- Risk #2: Insecure Authentication
- Risk #3: Data Leakage
- Risk #4: Oversharing
- Risk #5: Data and Secret Handling Failures
- Risk #6: Misconfiguration
- Risk #7: Business Continuity, Resiliency and Ownership Failures
- Risk #8: App impersonation
- Risk #9: Dependency Injection
- Risk #10: Unmanaged Custom Code



# OWASP TOP 10 Cloud Security Risks

- R1: Accountability and Data Ownership
- R2: User Identity Federation
- R3: Regulatory Compliance
- R4: Business Continuity and Resiliency
- R5: User Privacy and Secondary Usage of Data
- R6: Service and Data Integration
- R7: Multi Tenancy and Physical Security
- R8: Incident Analysis and Forensic Support
- R9: Infrastructure Security
- R10: Non-Production Environment Exposure

# Business Continuity / Disaster Recovery



# Denial of Service

## **You're Temporarily Blocked for 30 Days**

---

You've been temporarily blocked from using certain features because you violated Facebook's Terms. Please review the Community Standards to learn what's okay to share on Facebook.

This block will be lifted in 30 days, but if you continue to violate Facebook's Terms, your account could be permanently disabled.

[Facebook Community Standards](#)

下一页

# Long-Term Viability, Vendor Lock-In, Cloud Exit Strategy

## Novinky.cz

Novinky.cz » Internet a PC » Český Jablotron zablokoval datové služby pr... Podrubriky: [Hardware](#) • [Software](#) • [Testy](#) • [Hry a herní systémy](#) • [Mobil](#) • [Bezpečnost](#)

VÁLKA NA UKRAJINĚ

MINUTA PO MINUTĚ

MAPA KONFLIKTU

VÁLKA OBRAZEM

НА РУССКОМ

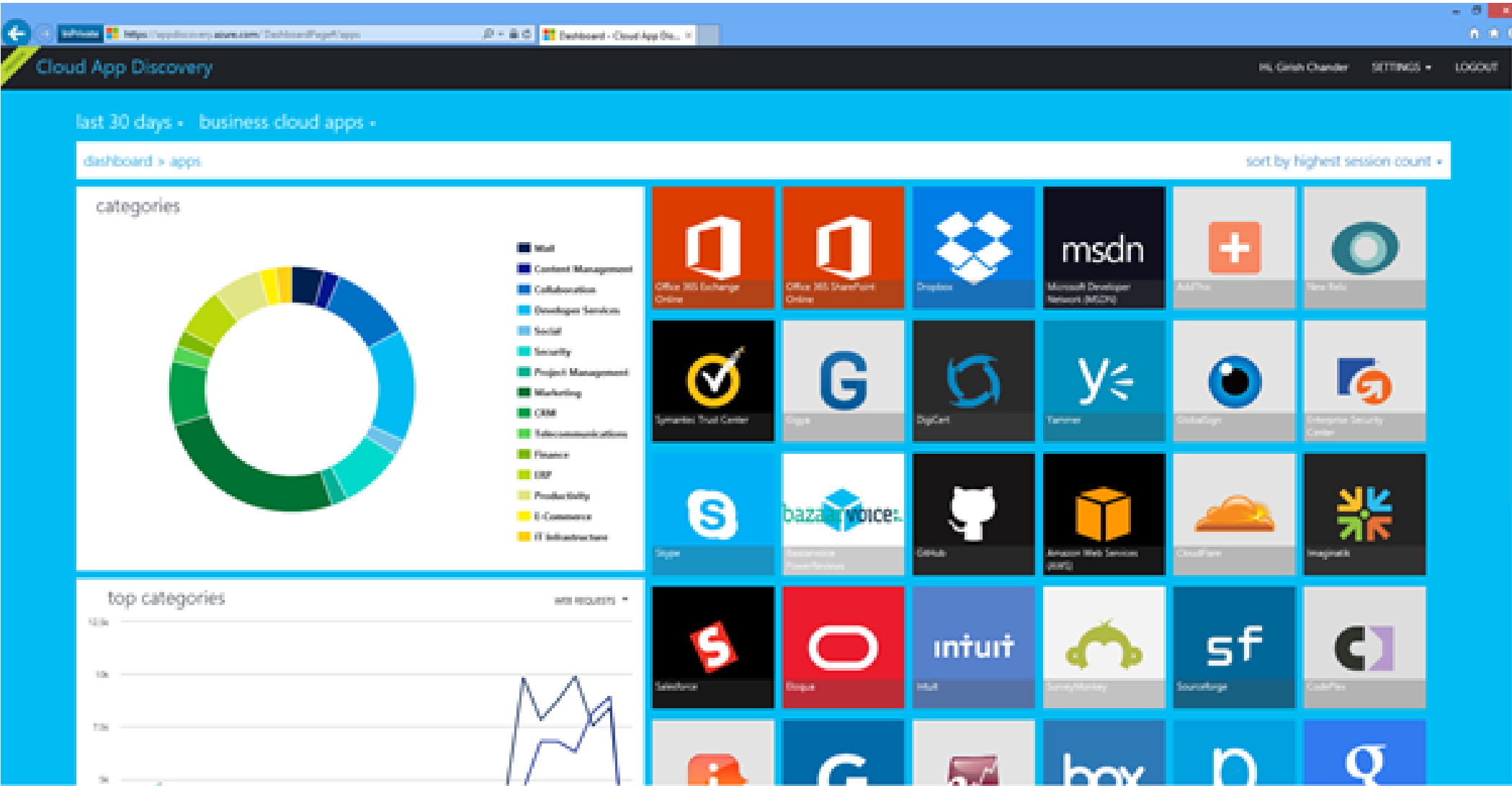
## Český Jablotron zablokoval datové služby pro klienty v Rusku a Bělorusku

2. 3. 2022, 7:30 – Jablonec nad Nisou – [ČTK](#)



Český výrobce zabezpečovacích zařízení Jablotron zablokoval datové služby pro uživatele v Rusku a Bělorusku. Chce tak burcovat ruskou veřejnost proti prezidentovi Vladimíru Putinovi a jeho lidem. Kromě zastavení prodeje do Ruska a Běloruska zablokuje Jablotron datové služby pro své už dříve namontované výrobky v těchto zemích. Uvedl to zakladatel Jablotronu Dalibor Dědek.

# Shadow IT – Anyone can use DropBox, GitHub,etc.



# Authentication in Cloud Applications



# Authentication

- Basic Concepts
  - Two-factor authentication
  - OTPs
- Identity Federation
  - OAuth
  - OpenID Connect
  - SAML

# Passwords Are Dead

- Dictionary Attacks
- Password Leaks
- Phishing
- Keyloggers
- Shoulder Surfing
- ...









# Passwords Are Dead

## ';--have i been pwned?

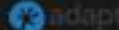

Check if you have an account that has been compromised in a data breach

326	5,575,703,782	83,835	91,310,138
pwned websites	pwned accounts	pastes	paste accounts

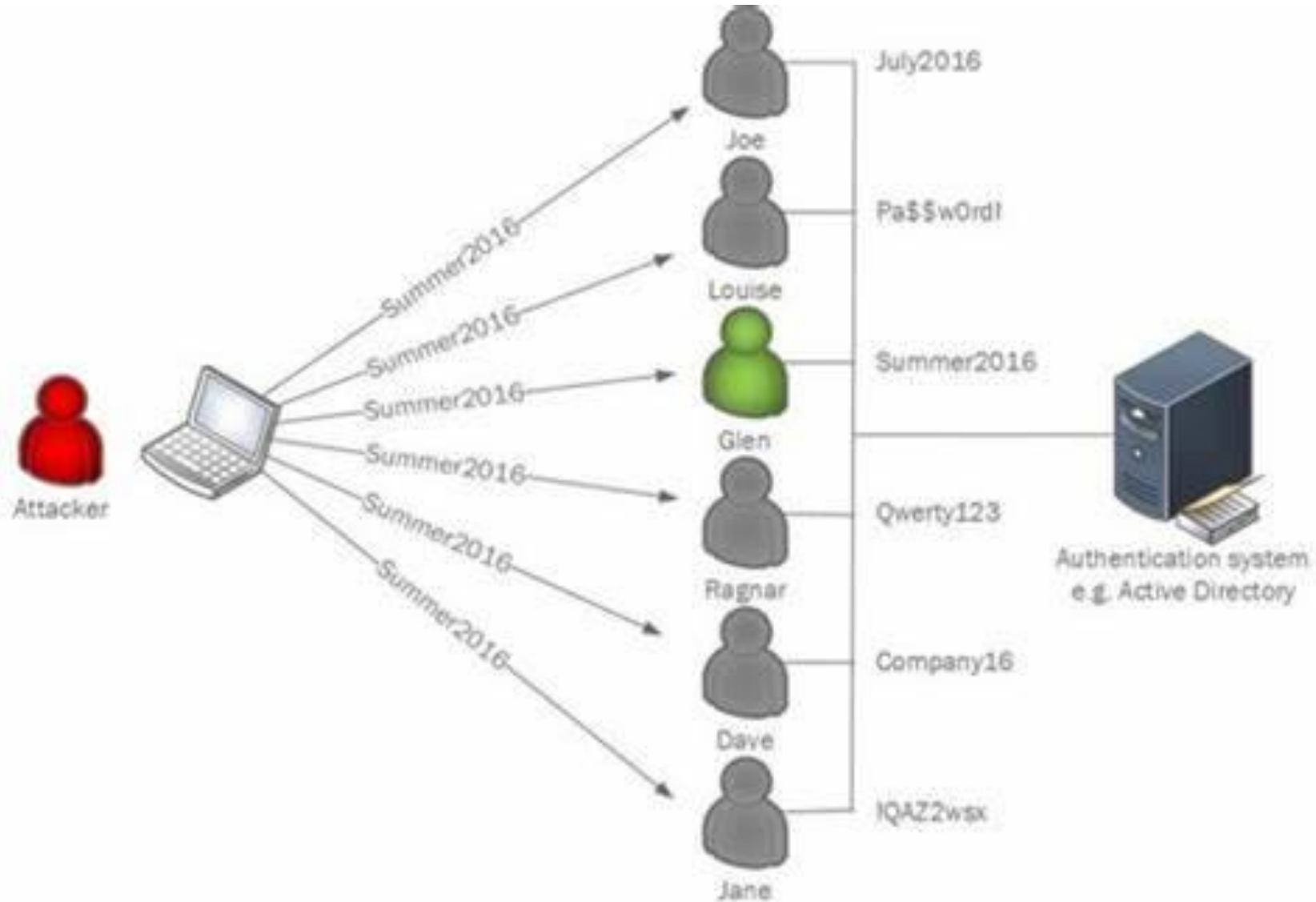
### Largest breaches

-  711,477,622 [Onliner Spambot accounts](#)
-  593,427,119 [Exploit.In accounts](#)
-  457,962,538 [Anti Public Combo List accounts](#)
-  393,430,309 [River City Media Spam List accounts](#)
-  359,420,698 [MySpace accounts](#)
-  234,842,089 [NetEase accounts](#)
-  164,611,595 [LinkedIn accounts](#)
-  152,445,165 [Adobe accounts](#)
-  131,577,763 [Exactis accounts](#)
-  125,929,660 [Apollo accounts](#)

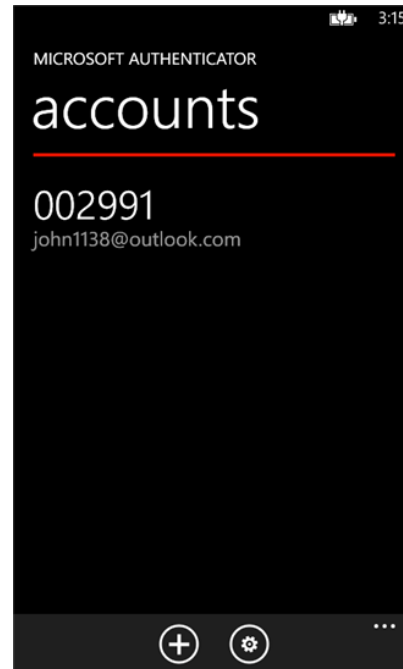
### Recently added breaches

-  9,363,740 [Adapt accounts](#)
-  411,755 [HTH Studios accounts](#)
-  5,788,169 [Elasticsearch Instance of Sales Leads on AWS accounts](#)
-  1,957,600 [KnownCircle accounts](#)
-  24,990 [Rbx.Rocks accounts](#)
-  14,609 [Società Italiana degli Autori ed Editori accounts](#)
-  858 [WPSandbox accounts](#)
-  22,477 [JoomlaArt accounts](#)
-  326,714 [Mac Forums accounts](#)
-  846,742 [Baby Names accounts](#)

# Password Spraying



# Multi-Factor Authentication



# One-time Passwords

- HMAC-based One-time Password (HOTP)
- Time-based One-time Password (TOTP)
- Out-of-Band Transaction Authentication Numbers (TANs)
  - Pre-generated (POTP)
  - SMS
  - Push Notifications

# TOTP

$HMAC(K, C) := SHA1(K \oplus 0x5C5C... \parallel SHA1(K \oplus 0x3636... \parallel C))$

$HOTP(K, C) := Truncate(HMAC(K, C)) \& 0x7FFFFFFF$

K – Secret Key

C – Counter/Clock

# MFA Method Strength



Microsoft Authenticator  
Passwordless



Password  
+ Microsoft Authenticator  
Number match



Password  
+ Hardware Tokens OTP



Password  
+ Software Tokens OTP



Password  
+ Voice



Password  
+ SMS

Phishing resistant



Certificate based  
authentication



FIDO2 security keys

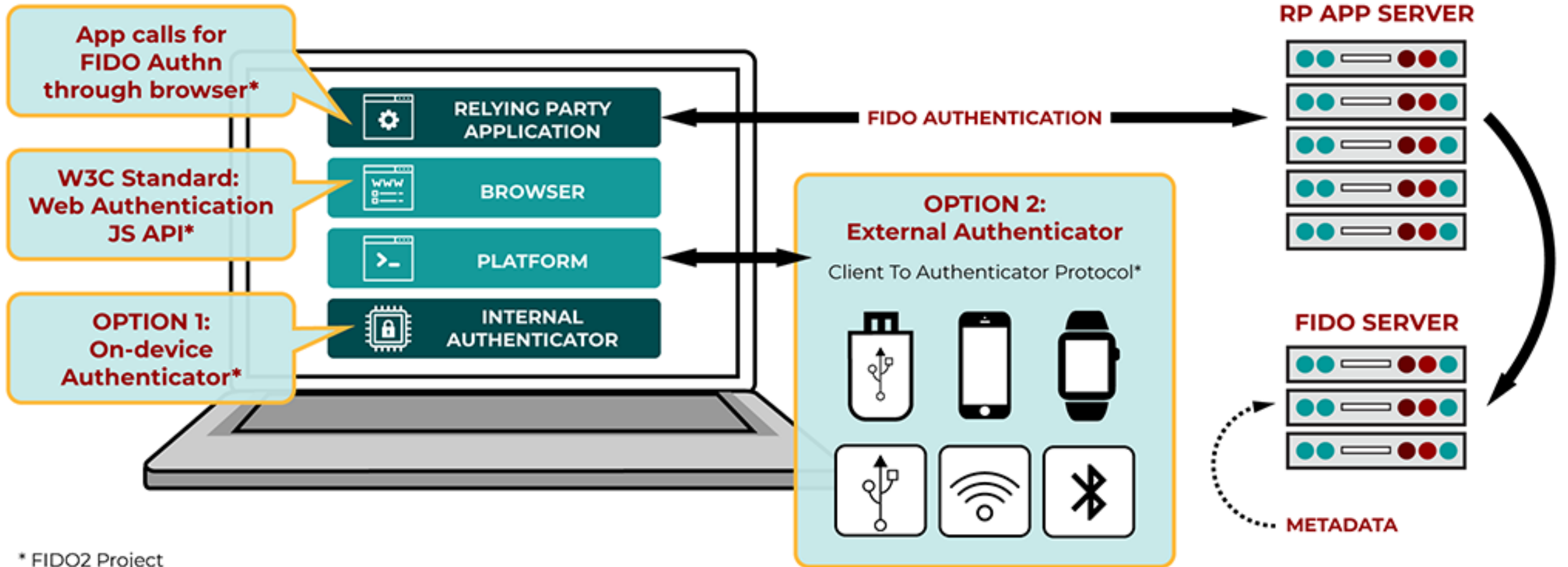


Windows Hello  
for Business



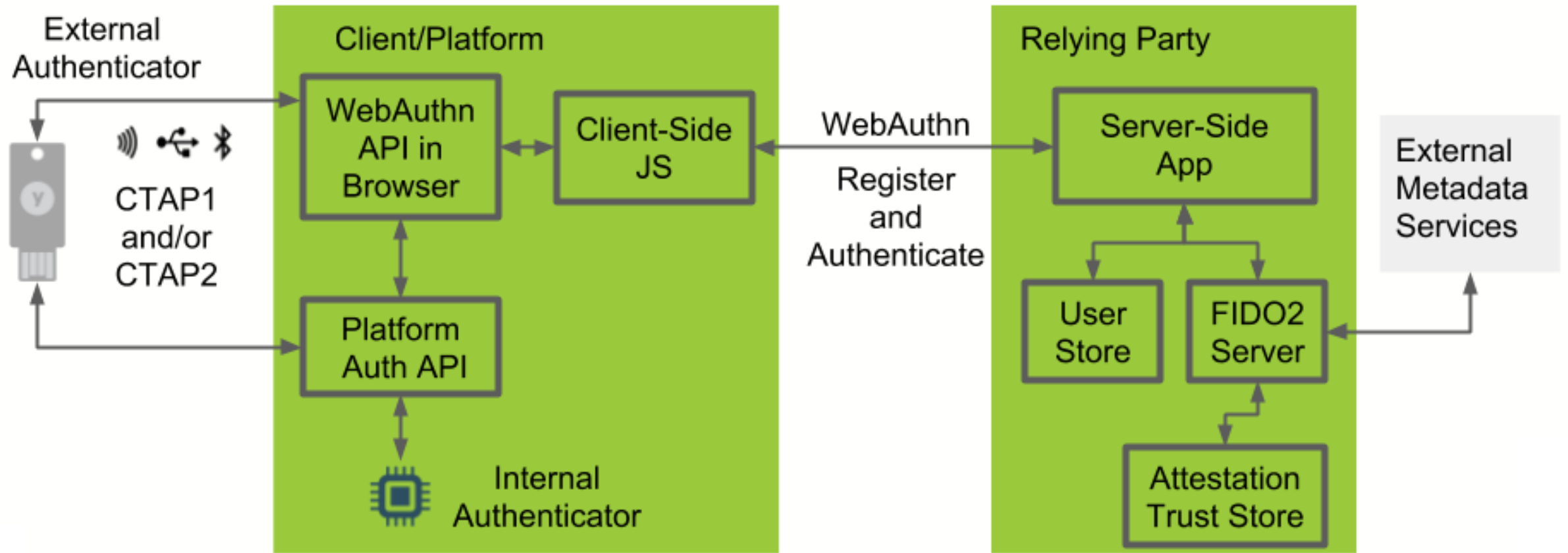
Passkey in Microsoft Authenticator  
(device-bound)

# Passkeys / FIDO2 / W3C Web Authentication



\* FIDO2 Project

# Passkey Protocols





# WebAuthn API

```
// Detection of WebAuthn support
if (!window.PublicKeyCredential) { /* Client not capable. Handle error. */ }

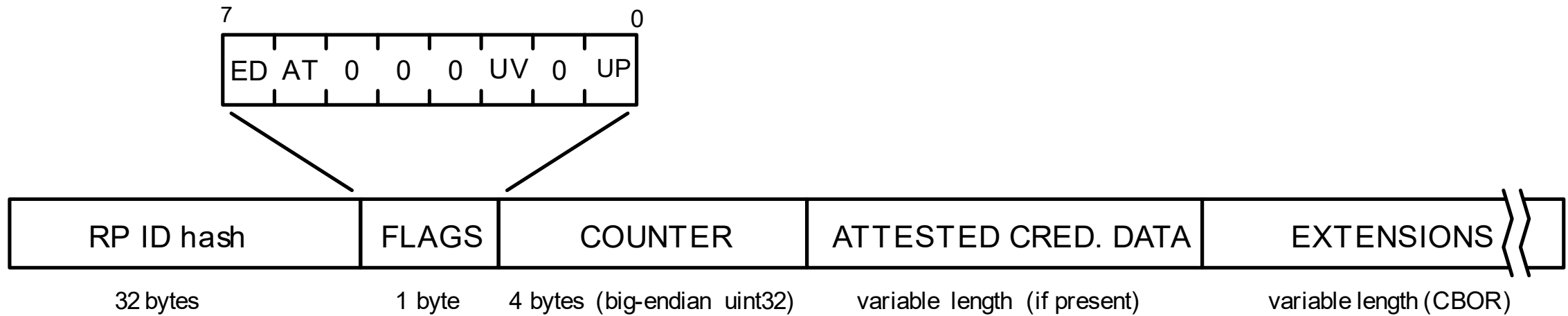
// Registration
const credential = await navigator.credentials.create({
  publicKey: publicKeyCredentialCreationOptions
});

// Assertion (Logon)
const credential = await navigator.credentials.get({
  publicKey: publicKeyCredentialRequestOptions
});
```

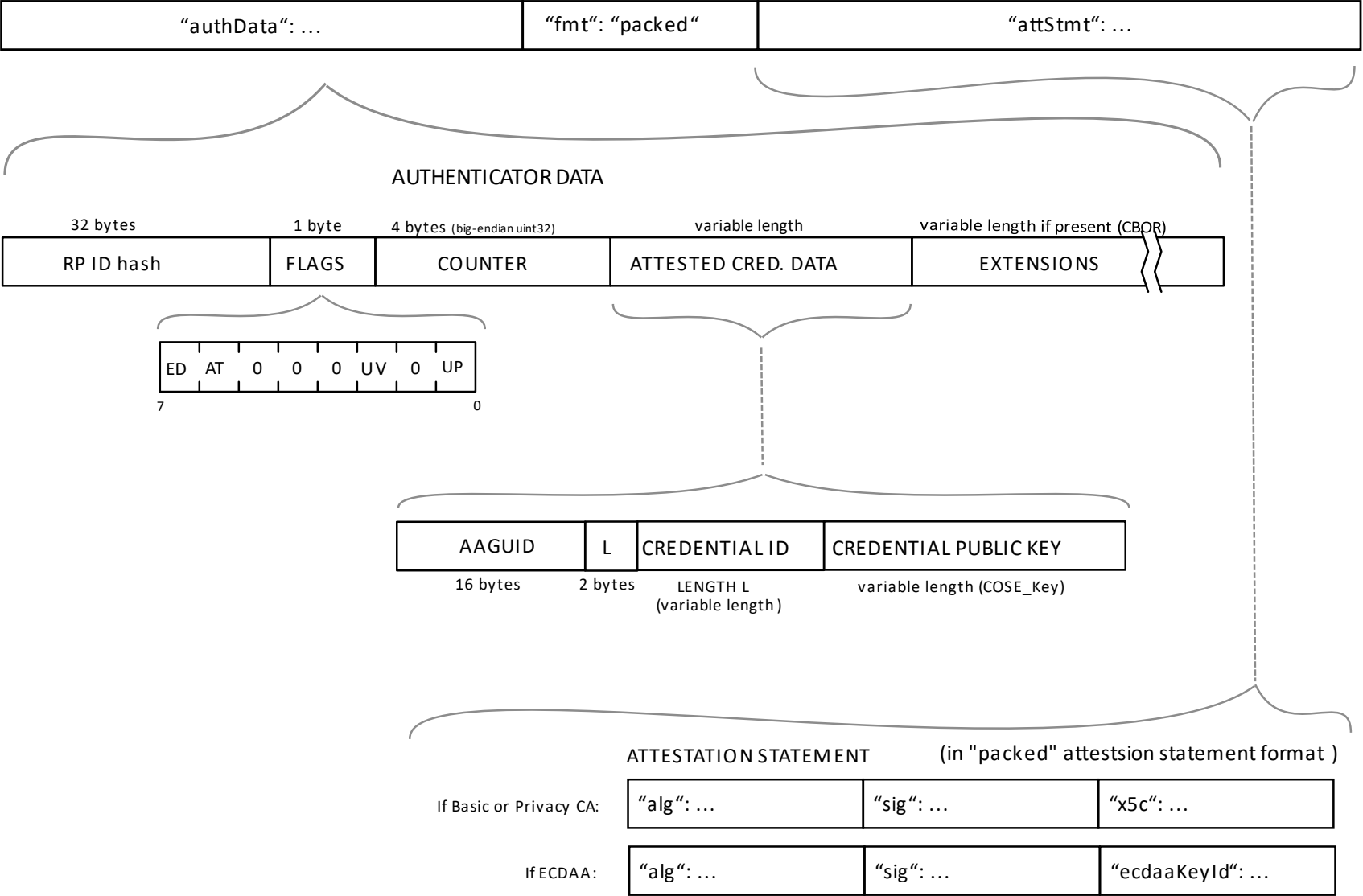
# Public Key Credential Creation Options

```
const publicKeyCredentialCreationOptions = {
  challenge: Uint8Array.from(randomStringFromServer, c => c.charCodeAt(0)),
  rp: {
    name: "Microsoft",
    id: "login.microsoft.com",
  },
  user: {
    id: Uint8Array.from(window.atob("MIIBkzCCATigAwIBAjCCAZMwggE4oAMCAQIwggGTMII="), c=>c.charCodeAt(0)),
    name: "michael@dsinternals.com",
    displayName: "Michael Grafnetter",
  },
  pubKeyCredParams: [{alg: -7, type: "public-key"}],
  authenticatorSelection: {
    authenticatorAttachment: "cross-platform",
    requireResidentKey: true,
    userVerification: "required"
  },
  timeout: 60000,
  attestation: "direct"
};
```

# Authenticator Data





# Attestation Object





# Language-Specific SDKs (webauthn.io)



Go

 duo-labs/webauthn  
 Duo Labs  
</> Server



Go

 duo-labs/webauthn.io  
 Duo Labs  
</> Demo



Go

 koesie10/webauthn  
 Koen Vlaswinkel  
</> Server



Java

 duo-labs/android-webauthn-authenticator  
 Duo Labs  
</> Authenticator



Java

 google/webauthndemo  
 Google  
</> Demo



Java

 webauthn4j/webauthn4j  
 Yoshikazu Nojima  
</> Server



Java

 Yubico/java-webauthn-server  
 Yubico  
</> Server



Javascript

 fido-alliance/webauthn-demo  
 Fido Alliance  
</> Demo



.NET

 abergs/fido2-net-lib  
 Anders Åberg  
</> Server/Demo



Python

 duo-labs/py\_webauthn  
 Duo Labs  
</> Server/Demo



Ruby

 cedarcode/webauthn-ruby  
 Cedarcode  
</> Server

Chrome Extension


 google/virtual-authenticators-tab  
 Nina Satragno  
</> Authenticator

# Problem: Lost Device

- Multiple Devices
- Recovery Questions  
What is the maiden name of your mother? 
- E-Mail Verification 
- POTP
- In-Person Verification (used in enterprise environments, but not usable with cloud services)

# Application-Specific Passwords

Some protocols (or their implementations) like SSH, SFTP or IMAP do not support 2FA







 **brenna**

- Profile
- Account settings
- Emails
- Notification center
- Billing
- Payment history
- SSH keys**
- Security
- Applications
- Repositories
- Organizations

Need help? Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#)

### SSH Keys Add SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

	 <b>info@brennaobrien.com</b> a0:56:bc:a0:ef:06:39:d6:03:cc:9a:7a:0e:d8:92:5c Added 2 years ago — Last used on April 28, 2014	<span>Delete</span>
	 <b>test server</b> 4a:dd:4b:0a:c5:20:55:8c:1a:3c:ac:14:c5:b7:63:04 Added a year ago — Last used on March 20, 2014	<span>Delete</span>
	 <b>hackeryou SSH</b> 93:89:a8:46:ef:46:1b:99:7a:fe:66:c0:ba:28:1c:c9 Added 6 months ago — Last used on April 07, 2014	<span>Delete</span>

# Claims-Based Identity





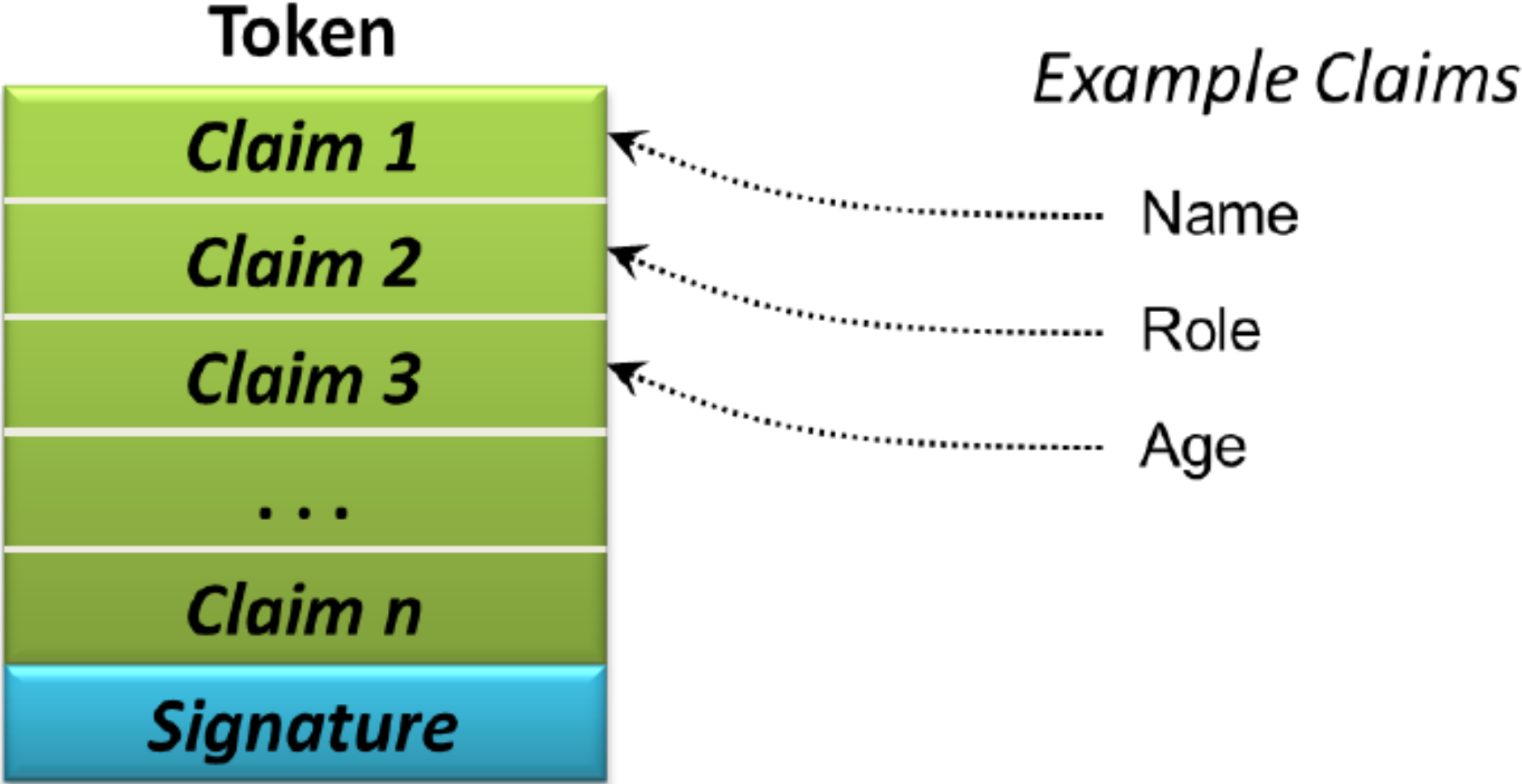
# Most Common Technologies

- OAuth
- OpenID Connect
- SAML
- WS-\*
- JWT

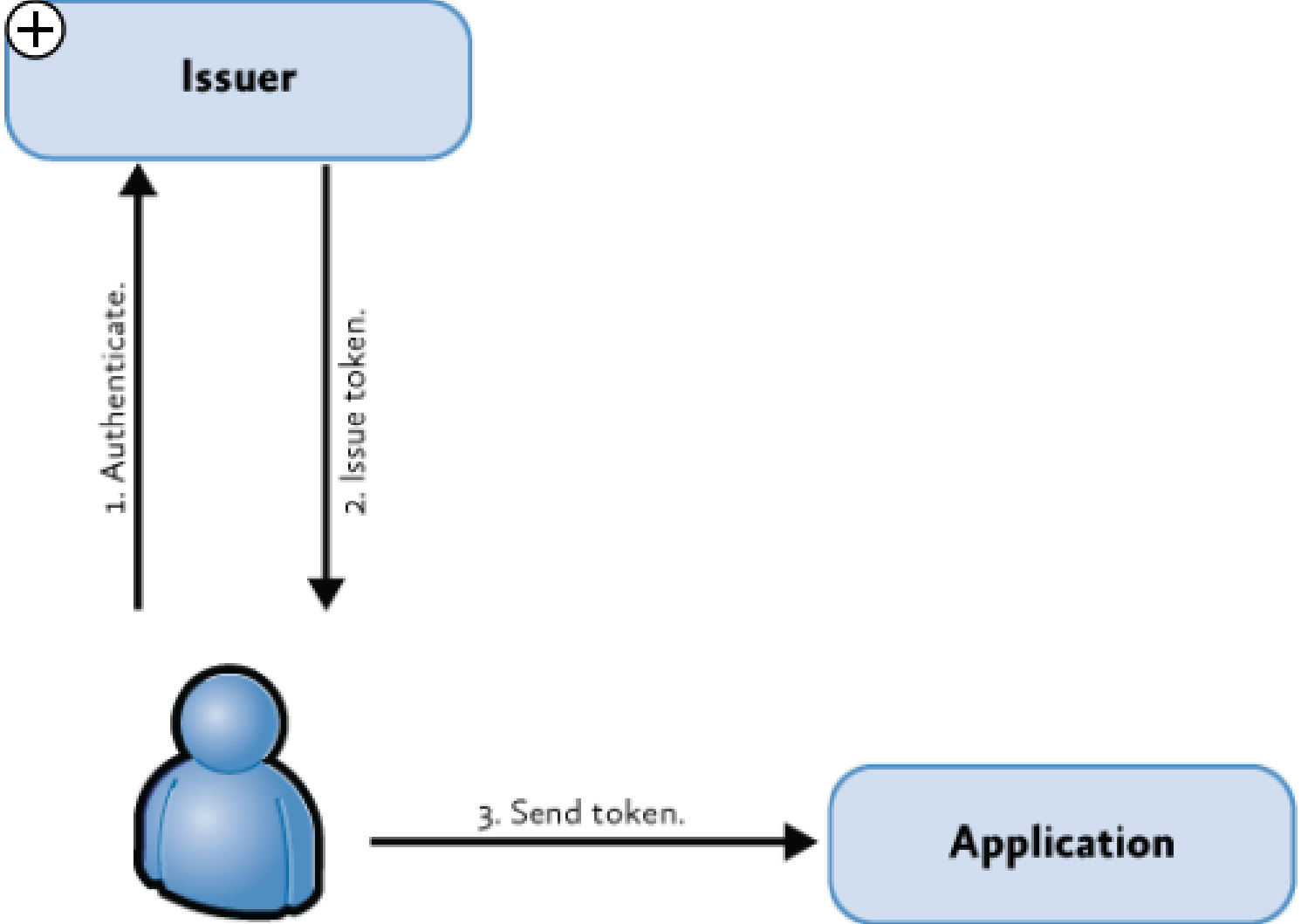
# Claims-based Identity

- First Name: John
- Last Name: Doe
- Login: John
- Mail: john@doe.com
- Role: User
- Role: Administrator

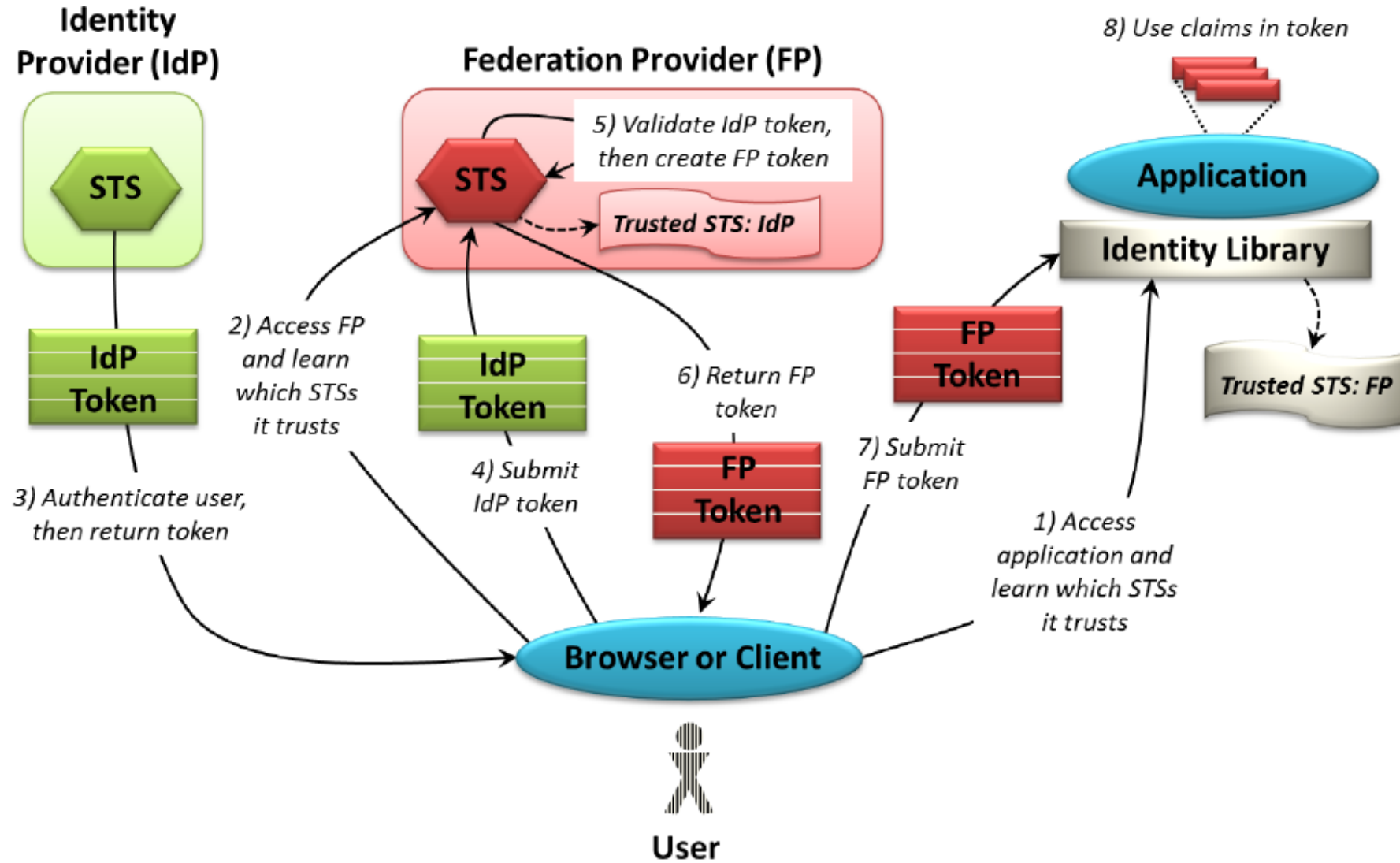
# Tokens



# Basic Communication Pattern



# Identity Federation



SAML



# SAML

- Security Assertion Markup Language
- XML-based
- Supports single sign-on (SSO)
- Requires mutual trust between IdP and SP
- Multiple bindings, not just HTTP
- Supports identity provider-initiated authentication

# Underlying Standards

- Extensible Markup Language (XML)
- XML Schema (XSD)
- XML Signature
- XML Encryption
- Hypertext Transfer Protocol (HTTP)
- Simple Object Access Protocol (SOAP)



# SAML Versions

- SAML 1.0 - 2002
- SAML 1.1 - 2003
- SAML 2.0 - 2005
  - Incompatible with SAML 1.x
  - Renamed XML namespaces, elements and attributes
  - New bindings
  - New protocols

# SAML Assertions

```
<saml:Assertion ID="_3c39bc0fe7b13769cab2f6f45eba801b1245264310738"  
  IssueInstant="2009-06-17T18:45:10.738Z" Version="2.0">  
  <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">  
    https://www.salesforce.com  
  </saml:Issuer>
```

Assertion A was issued at time t by issuer R regarding subject S provided conditions C are valid.

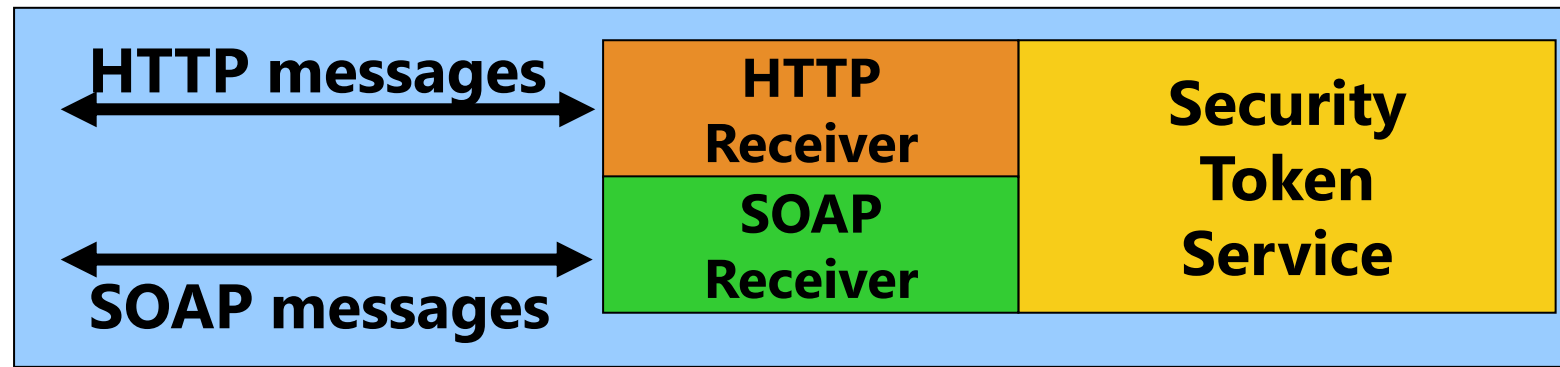
- Authentication statements
- Attribute statements
- Authorization decision statements

# SAML Sign-In Protocol

<https://sts.cloudready.ms/adfs/ls/?SAMLRequest=jZFRT4Mw-FIX%2FCun7KC3OjWaQ4PbgkqlkoA%2B%2BmAKdNCKt9hZ1%2F14Gmk-wfFI%2Fv%0APfc7p6cr4K3qWNq7Ru%2FFWy%2FAeZ%2Bt0sDGRYx6q5nhllFp3gp-grmJ5erdj1A9Y%0AZ40zIVHISwGEddLotdHQQt8Lmwr7LSjzudzFqnOuAYQyNLP1Kmb62gtd-HvwWc%0AD6PSKOEaH8DgE5ni7CEvkLcZokjNT9AzhIM%2FBF4fACvAyNtuYvRSRSI-iZXIN%0AwrIY0CriSxKGhNLDFeXhYjKfZAC92GpwXLSY0YCEM0JnQVQESxaEj-CyekZd9%0AP%2BxG6lrq18stIJMI2G1RZLMp%2FJOwMAYfBChZnbpko7E9a%2Fcylv9UipJ%2FF-bjC%0AZy6TZcfuB%2Bx2kxklq6OXKmU%2B1sOpEzEiCCfTye%2FFt74A%0A&Re-layState=cookie%3A29002348&SigAlg=http%3A%2F%2Fwww.w3.org%2F2000%2F09%2Fxmld-sig%23rsa-sha1&Signature=M0xoWQfcN3Yp94T2HiqlDjzEkxYqGc6hhopqi8xOI%2B2BtPSLufFD-dQIF7z6Xjm6XdLq1MH9Av5xz2QWYs84ZYhIG3fHtZCjjaol2wZqplRszHla%2BjtZoW20NGDe-pDsCRT0AKNkhe%2B4Yj3LshrM6EX5O3obx2Mypy8EcsoURkTF3kf1dwKqsGA3ka7ehbRmUQGJUXD0u4iF-Bog7YgkL4Q9FYMTanZeRo2X4%2FkAeNXT8ormKWJfYnAzg0F4Ku60zDd5N7jYu4XeyOsXDthEFI5H4WYu-cAprREI2hgSUI21J782kKzrslallaJ5BKPIO50NPCib5Sf6Zw4maLpZrFEfrw%3D%3>

# WS-Federation

- OASIS Standard from 2006, drafted by IBM, Microsoft, RSA,...
- Uses SAML Tokens
- Based on WS-Security, WS-Trust and other WS-\* standards
- Supports automatic metadata discovery and certificate roll-over
- Defines common claims
- Two profiles of the model defined
  - Passive – for web browser clients
  - Active – SOAP clients



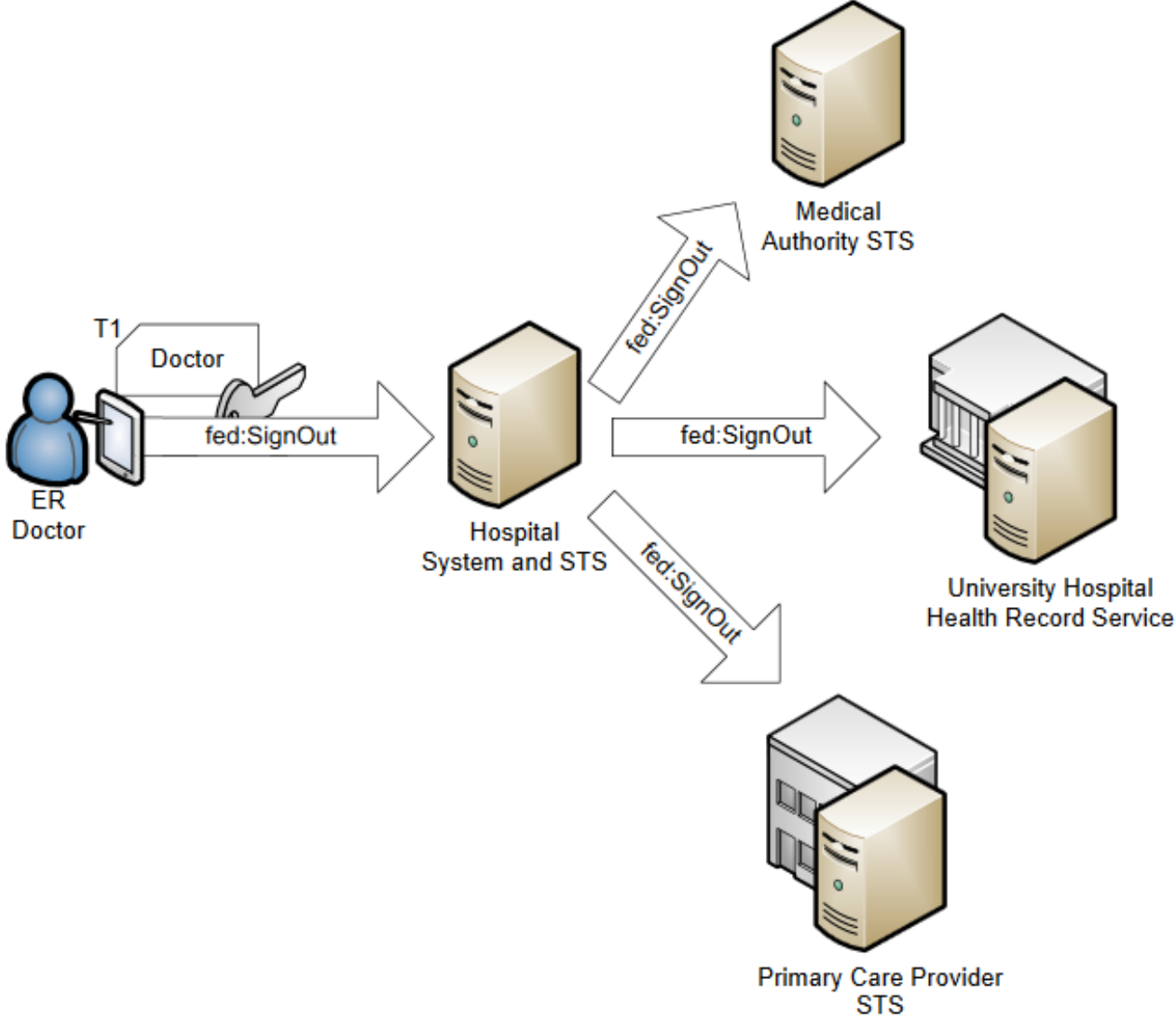
# WS-Federation SignIn

<https://sts.cloudready.ms/adfs/ls/?wa=wsignin1.0&wtrealm=https%3a%2f%2fclaim-sweb.cloudready.ms&wctx=rm%3d0%26id%3dpas-sive%26ru%3d%252f&wct=2014-10-21T22%3a15%3a42Z>

Let's break-out these parameters:

- **Wa=signin1.0:** This tells the ADFS server to invoke a login for the user.
- **Wtrealm:** This tells ADFS what application I was trying to get to. This has to match the identifier of one of the relying party trusts listed in ADFS.
- **Wctx:** This is some session data that the application wants sent back to it after the user authenticates.
- **wct:** This is the exact time I tried to gain access to the application.

# Single SignOut



# FederationMetadata

- Entity (STS) ID
- Token signing certificates
- WS-Federation endpoint URL
- SAML protocol endpoint URL
- ...

# JSON Web Token





# JSON Web Token

- Defined in IETF RFC7519 from May 2015
- Inspired by SWT
- Based on
  - JSON Web Signature (JWS, RFC7515)
  - JSON Web Encryption (JWE, RFC7516)
- Very compact
- Can use different encryption schemes

# Token Structure

Header: { typ: 'JWT', alg: 'HS256' }

Payload/Claims:

```
{  
  user: john,  
  admin: true,  
  exp: 8.10.2016, 15:27  
}
```

Signature

=> BASE64

# Encoded and Signed Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.

eyJzdWIiOiIxMjM0NTY3ODkwIiwiaXNjaWkiOiJkaWVibmFtZSI6IkpvaG4gRG91IiwiaWF0IjoiYXNTb2NpYWwiOnRydWV9.

4pcPyMD09o1PSyXnrXCjTwXyr4BsezdI1AVTmud2fU4

---

OAuth

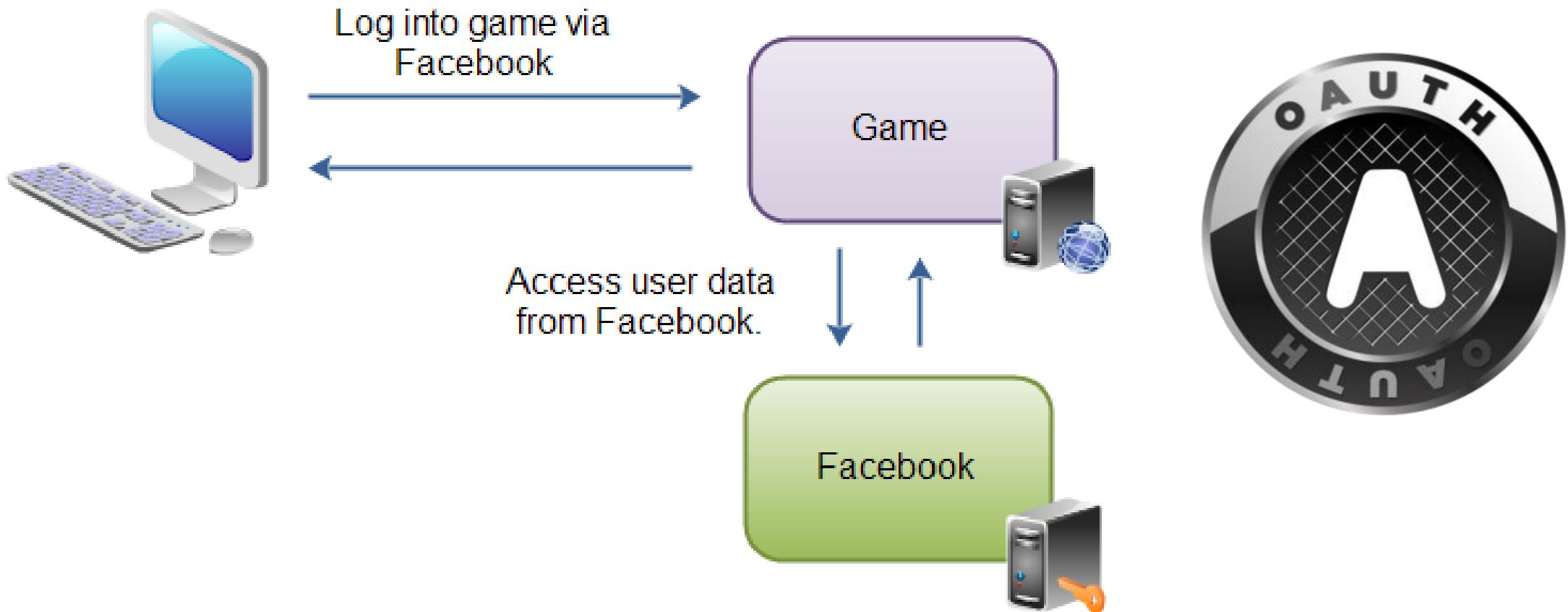


# OAuth / OpenID Connect



# OAuth

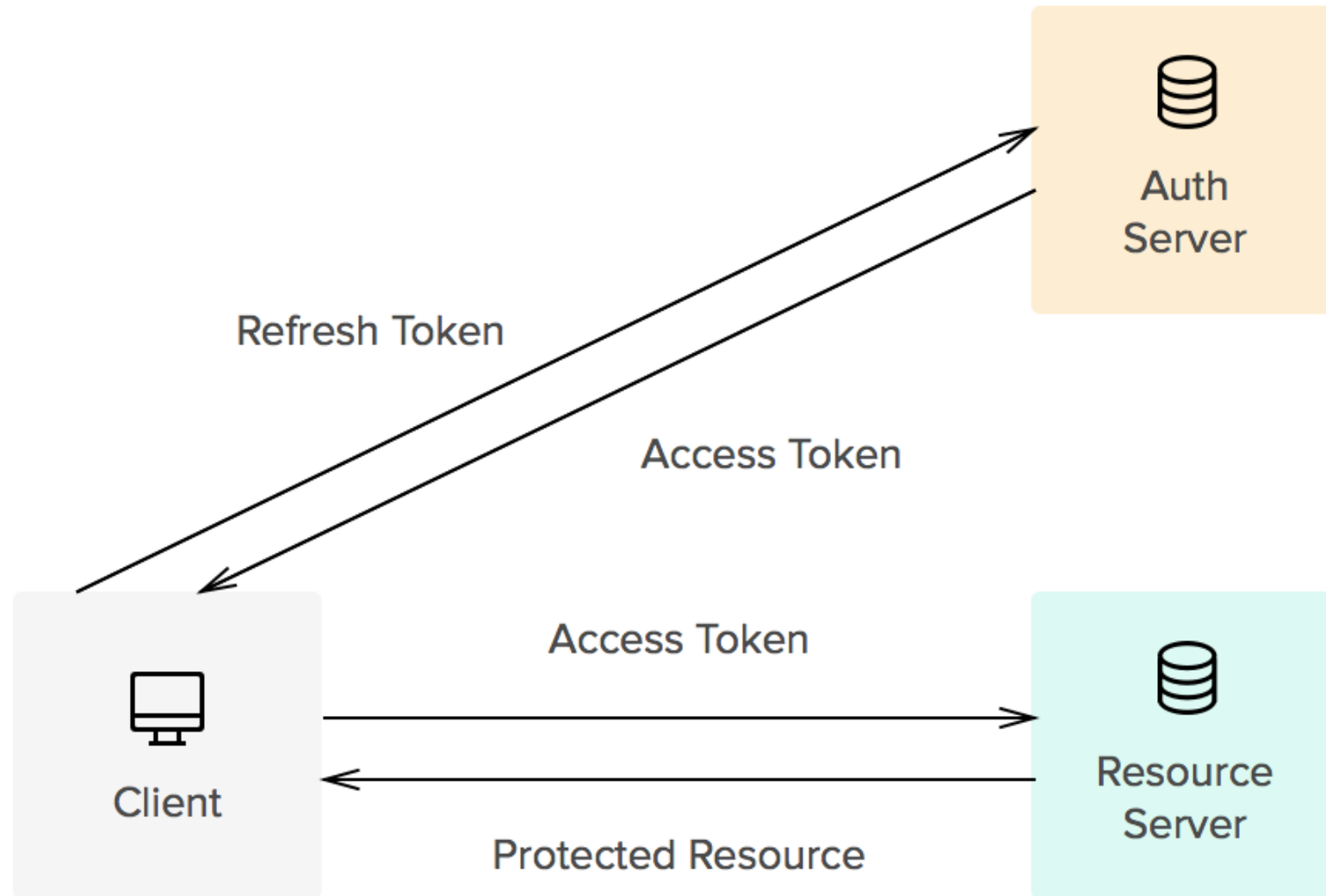
Used to delegate user authorization to a 3<sup>rd</sup>-party service provider



# History

- 2006 – Twitter missing delegation in OpenID
- 2007 – OAuth Core 1.0 (Twitter, Google,...)
- 2010 – OAuth 1.0 (IETF RFC 5849)
- 2012 – OAuth 2.0
  - Framework - RFC 6749
  - Bearer Token Usage - RFC 6750
  - Threat Model and Security Considerations - RFC 6819
  - ...

# OAuth Token Types





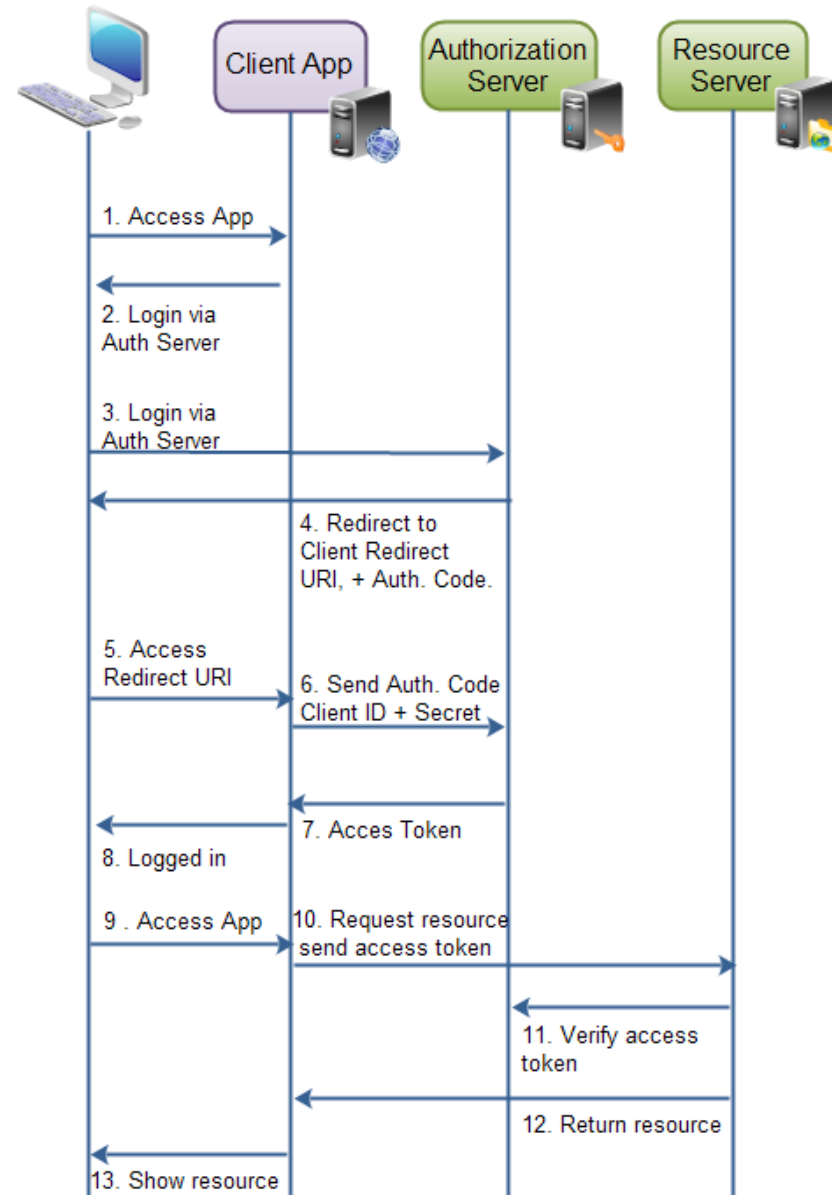
# OAuth Scopes (GitHub)

- read:org
- write:org
- read:discussion
- write:discussion
- read:packages
- write:packages
- delete:packages
- ...

# OAuth 2.0 Grant Types

- Authorization Code
  - Proof Key for Code Exchange (PKCE) Extension
- Client Credentials
- Device Code
- Refresh Token

# Authorization Code Flow



# Authorization Request


[https://sts.cloudready.ms/adfs/oauth2/authorize?response\\_type=code&client\\_id=3fb2a37f-4ced-409c-937c-dddd776f4dfd&redirect\\_uri=https://www.davetestapp.com&resource=https://www.davetestapp.com](https://sts.cloudready.ms/adfs/oauth2/authorize?response_type=code&client_id=3fb2a37f-4ced-409c-937c-dddd776f4dfd&redirect_uri=https://www.davetestapp.com&resource=https://www.davetestapp.com)

Let's break down these parameters:

- **response\_type:** tells that ADFS server that I want to perform OAuth and get an authorization code in return.
- **client\_id:** The ID of the application I'm trying to get to.
- **Resource:** the URL/URI of the application I'm trying to get to.
- **redirect\_uri:** Tells ADFS who to POST the auth code back to

# Authorization Prompt

← → ↻ [https://authorization-server.com/auth?scope=create+delete&client\\_id=2](https://authorization-server.com/auth?scope=create+delete&client_id=2)

**Example Service** Signed in as User Name 

---

**Sample App**

<https://example-app.com> by ACME Corp

This app would like to:

---

Create posts

---

Delete posts

---



# Authorization Response

ms-app://s-1-15-2-1101140336-4090662585-1905587327-  
262951538-2732256205-1306401843-  
4235927180?code=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.lxt7fs590QgJA  
JqMtW9C8KN6tLE.SzahnV- R4lbKXj46nwlbfUm6SLnyryJNE72e  
g3797LwkSfFQsSNpr7E9sUICYBH52xvLZDAbwwu7qXCMIqCuVciQ0j  
3P3-l3ep\_IOSJOD3LnwDnXb3MPM1UUNcGLxxVeJmeYhEr15Basdk  
WqGzTYrCJKf4jbdVT0qb4HKEhpD2aQCDwqjeFF8mNwfne\_KL1Ve6Z  
TNwBWS41SauUnbCTM9qzx-MCDWKEPrLmRR14hCxIsaWfrHmiE  
Ybfl4JXyGcJvhUyffcL-UVwJsQSBjHGlbQXlwrb-ejvvZ6me3YC8CL  
oS2pvXAMzbppBfg8YAJbGzBPNplbkjM10A7OKifLT4yqQ

# OAuth 2.0 Client Credentials Grant (Confidential Clients)

POST /token HTTP/1.1

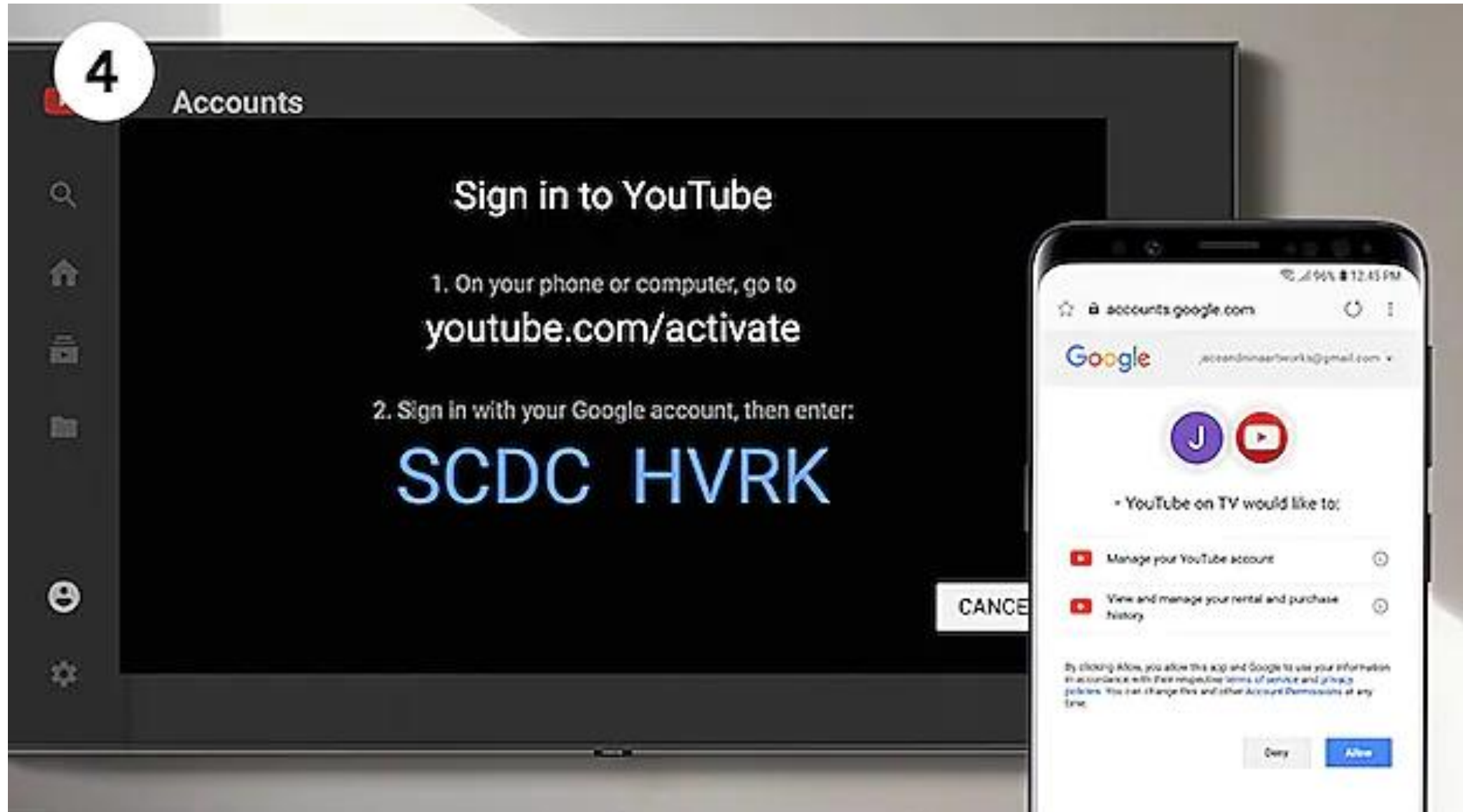
Host: authorization-server.com

grant\_type=client\_credentials

&client\_id=xxxxxxxxxx

&client\_secret=xxxxxxxxxx

# OAuth 2.0 Device Authorization Grant



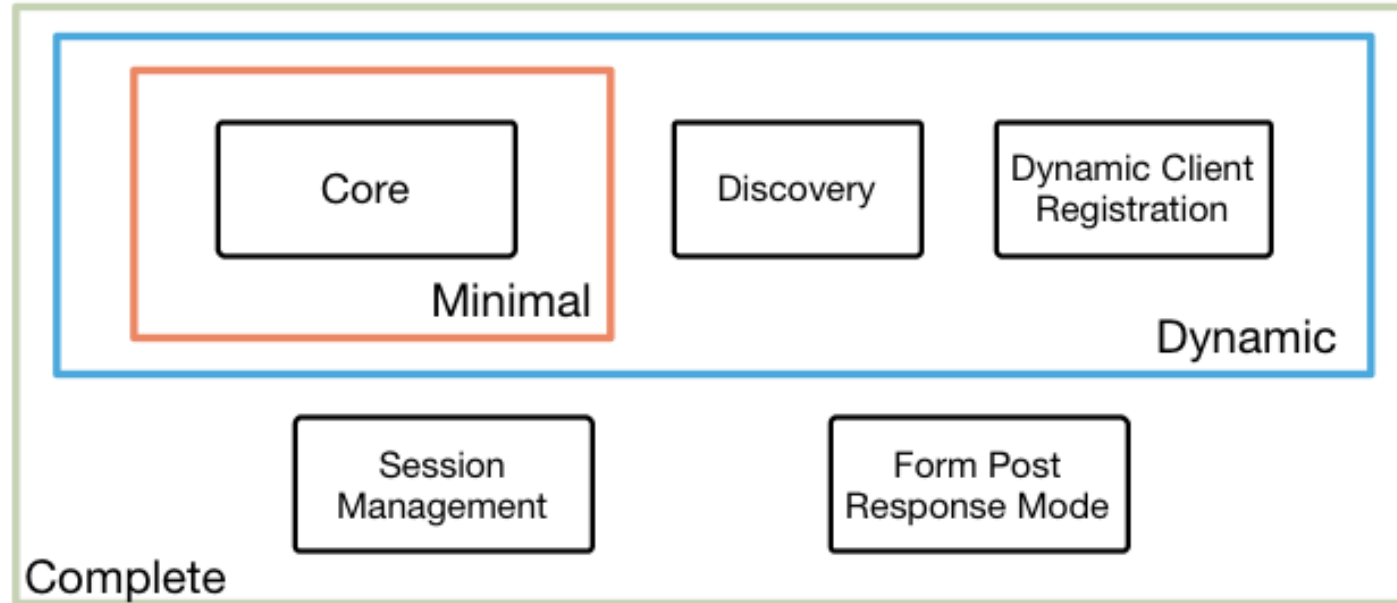


# OpenID Connect

4 Feb 2014

## OpenID Connect Protocol Suite

<http://openid.net/connect>



## Underpinnings

