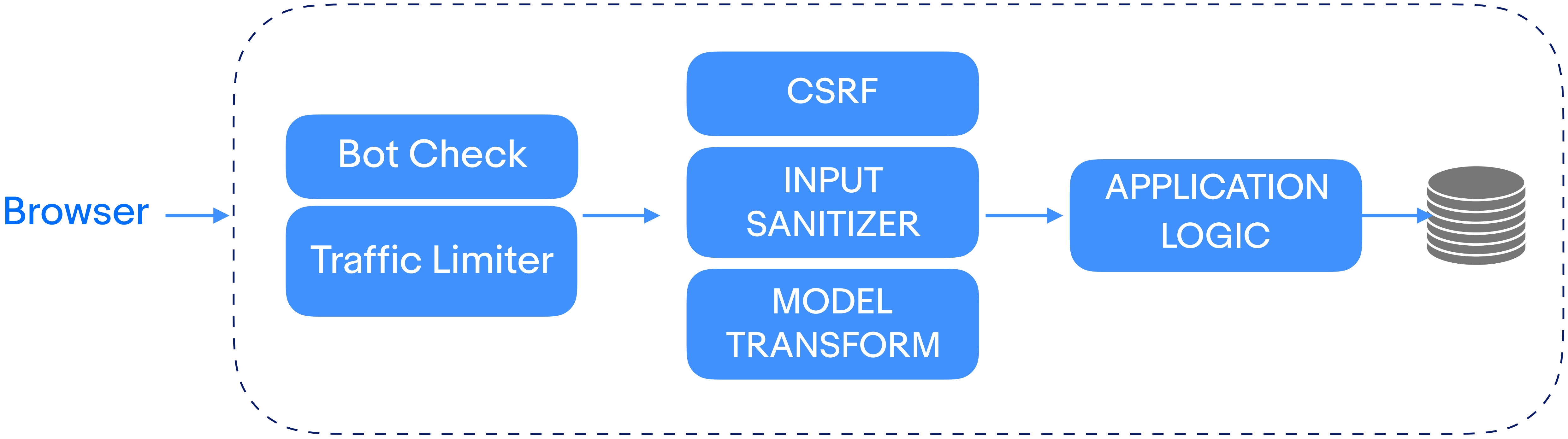
# Token based API Security in **TEN** steps

Senthilkumar Gopal

ebay

# ACME Fort Knox Web Application



Browser

Bot Check

Traffic Limiter

CSRF

INPUT SANITIZER

MODEL TRANSFORM

APPLICATION LOGIC

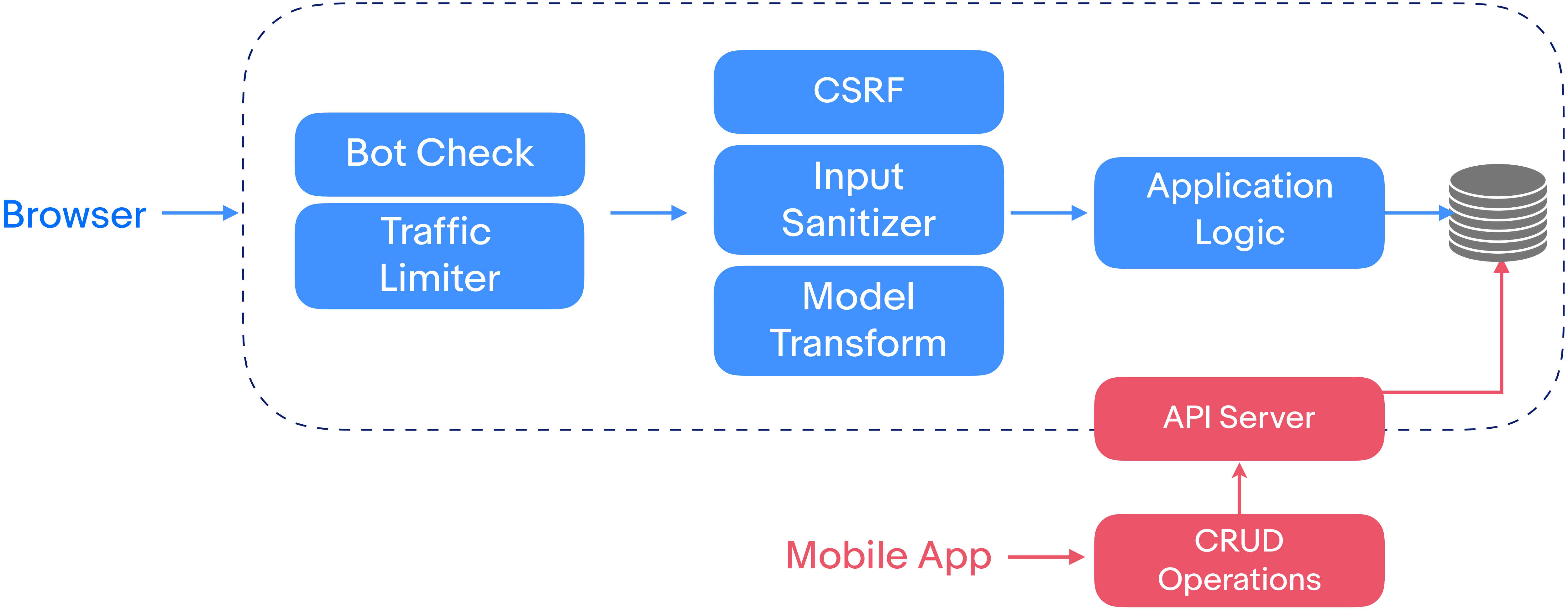
# A Hero’s (‘real’) story

Build an

*Awesome*

Mobile App

# ACME (Not) Fort Knox Web Application



# Web Application vs. APIs

> “But no one else knew about the API server”

ebay

@sengopal

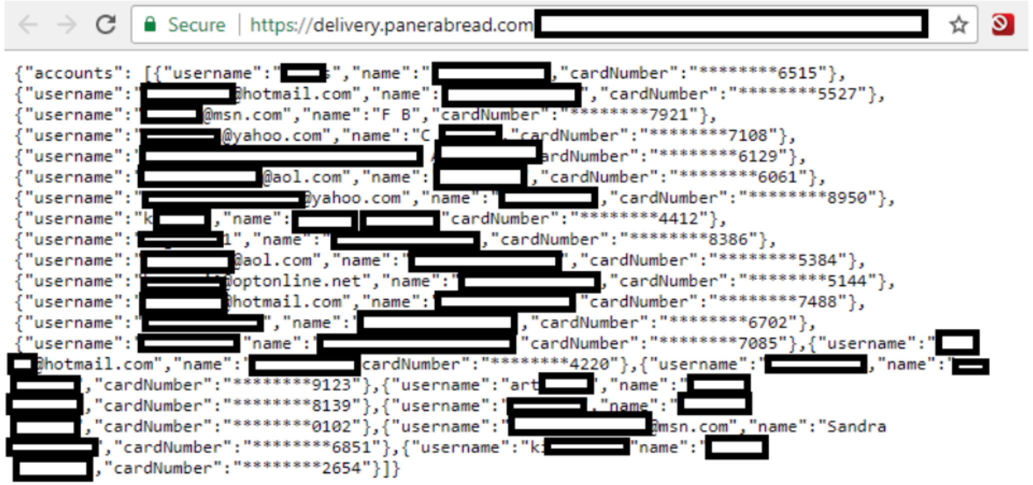
# Web Application vs. APIs

02
APR 18

## Panerabread.com Leaks Millions of Customer Records

**Panerabread.com**, the Web site for the American chain of bakery-cafe fast casual restaurants by the same name, leaked millions of customer records — including names, email and physical addresses, birthdays and the last four digits of the customer's credit card number — for at least eight months before it was yanked offline earlier today, KrebsOnSecurity has learned.

The data available in plain text from Panera's site appeared to include records for any customer who has signed up for an account to order food online via panerabread.com. The St. Louis-based company, which has more than 2,100 retail locations in the United States and Canada, allows customers to order food online for pickup in stores or for delivery.



Secure | https://delivery.panerabread.com[redacted]

```
{"accounts": [{"username":"[redacted]","name":"[redacted],"cardNumber":"********6515"},
{"username":"[redacted]@hotmail.com","name":[redacted]","cardNumber":"********5527"},
{"username":"[redacted]@msn.com","name":"F B","cardNumber":"********7921"},
{"username":"[redacted]@yahoo.com","name":"C [redacted],"cardNumber":"********7108"},
{"username":"[redacted] [redacted]rdNumber":"********6129"},
{"username":"[redacted]@aol.com","name":[redacted],"cardNumber":"********6061"},
{"username":"[redacted]@yahoo.com","name":"[redacted],"cardNumber":"********8950"},
{"username":"k[redacted],"name":[redacted] [redacted]"cardNumber":"********4412"},
{"username":"[redacted]1","name":"[redacted],"cardNumber":"********8386"},
{"username":"[redacted]@aol.com","name":"[redacted]","cardNumber":"********5384"},
{"username":"[redacted]@optonline.net","name":"[redacted],"cardNumber":"********5144"},
{"username":"[redacted]@hotmail.com","name":[redacted]"cardNumber":"********7488"},
{"username":"[redacted]","name":[redacted],"cardNumber":"********6702"},
{"username":"[redacted]"name":[redacted]"cardNumber":"********7085"},{"username":"[redacted]
[redacted]@hotmail.com","name":"[redacted]cardNumber":"********4220"},{"username":"[redacted],"name":"[redacted]
[redacted],"cardNumber":"********9123"},{"username":"art[redacted]","name":"[redacted]
[redacted],"cardNumber":"********8139"},{"username":"[redacted],"name":"[redacted]
[redacted],"cardNumber":"********0102"},{"username":"[redacted]@msn.com","name":"Sandra
[redacted]","cardNumber":"********6851"},{"username":"k[redacted]"name":"[redacted]
[redacted],"cardNumber":"********2654"}]}
```

*Redacted records from Panera's site, which let anyone search by a variety of customer attributes, including phone number, email address, physical address or loyalty account number. In this example, the phone number was a main line at an office building where many different employees apparently registered to order food online.*

source

# A Hero’s (‘real’) story

# I need an 'expert'

@sengopal

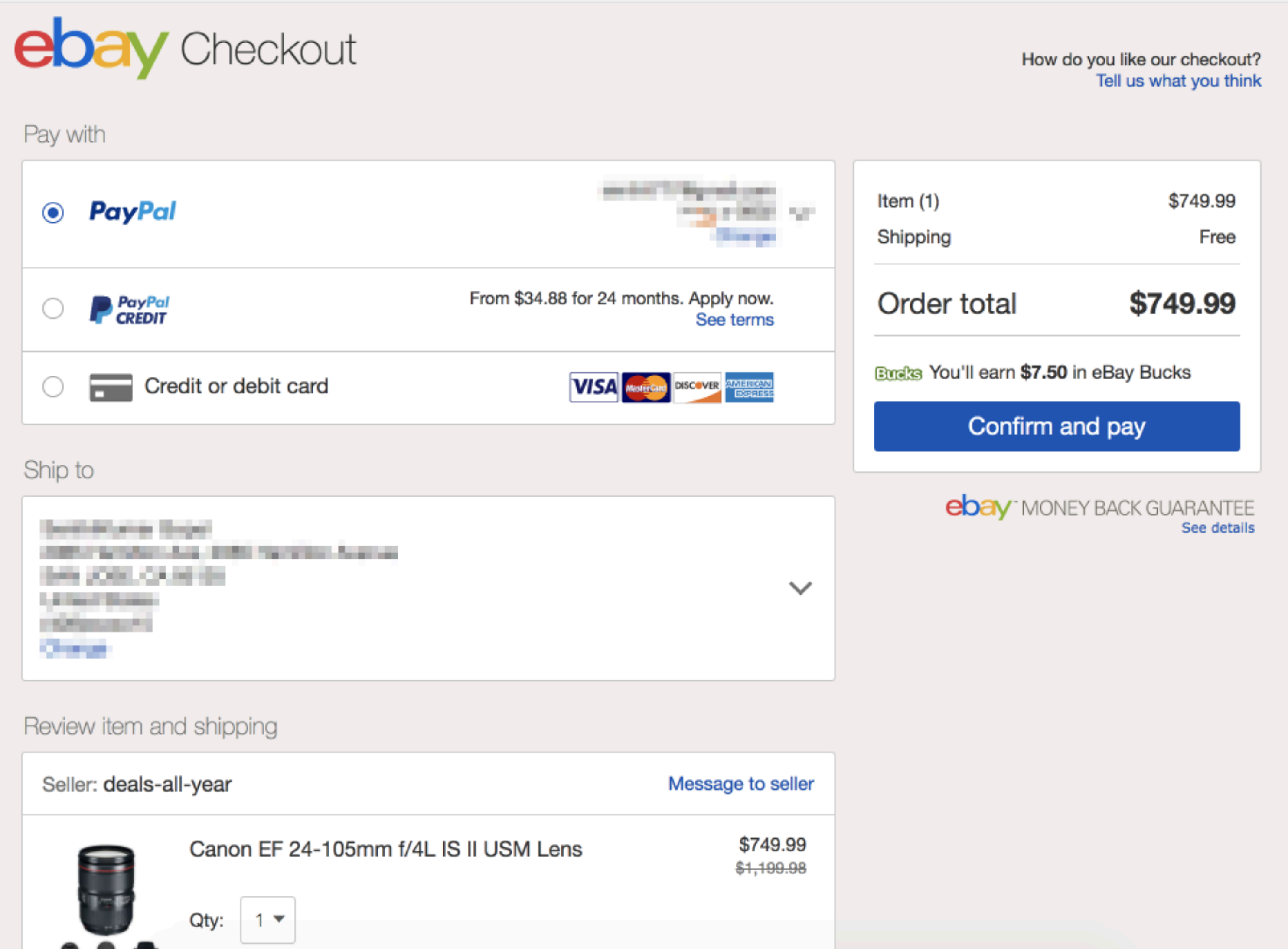
# First Principles

APIs are ...

Closer to Object Data Model

Intended to serve machines instead of real users

# Example of Web Application vs. APIs



Checkout

How do you like our checkout?
Tell us what you think

Pay with

● PayPal [redacted]

○ PayPal CREDIT — From $34.88 for 24 months. Apply now. See terms

○ Credit or debit card — VISA MasterCard DISCOVER AMERICAN EXPRESS

Ship to

[redacted]

Review item and shipping

Seller: deals-all-year — Message to seller

Canon EF 24-105mm f/4L IS II USM Lens $749.99 ~~$1,199.98~~

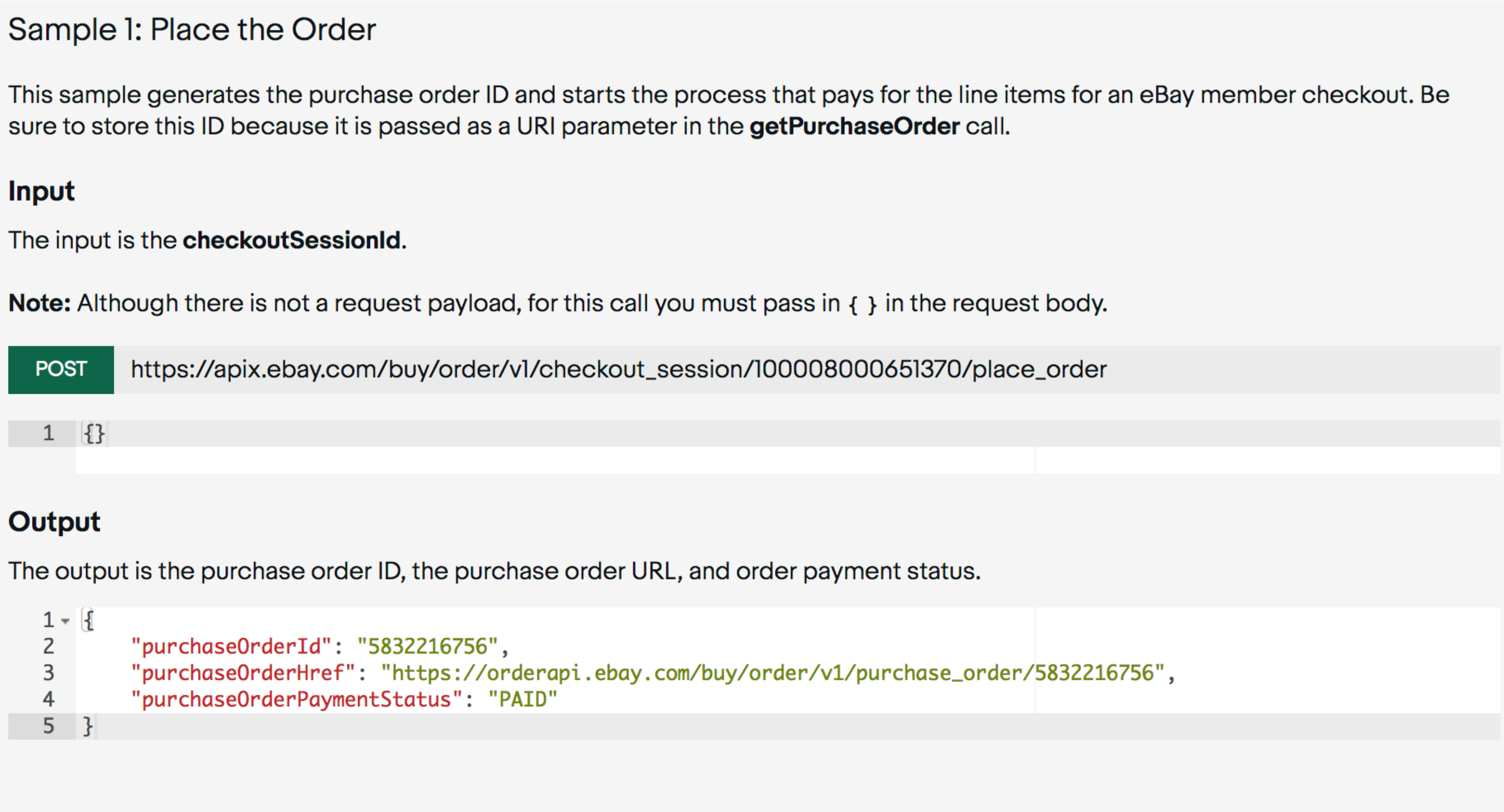
Qty: 1

| | |
|---|---|
| Item (1) | $749.99 |
| Shipping | Free |
| Order total | $749.99 |

Bucks You'll earn **$7.50** in eBay Bucks

ebay MONEY BACK GUARANTEE
See details

# Example of Web Application vs. APIs



## Sample 1: Place the Order

This sample generates the purchase order ID and starts the process that pays for the line items for an eBay member checkout. Be sure to store this ID because it is passed as a URI parameter in the **getPurchaseOrder** call.

### Input

The input is the **checkoutSessionId**.

**Note:** Although there is not a request payload, for this call you must pass in `{ }` in the request body.

POST https://apix.ebay.com/buy/order/v1/checkout_session/100008000651370/place_order

```
{}
```

### Output

The output is the purchase order ID, the purchase order URL, and order payment status.

```
{
    "purchaseOrderId": "5832216756",
    "purchaseOrderHref": "https://orderapi.ebay.com/buy/order/v1/purchase_order/5832216756",
    "purchaseOrderPaymentStatus": "PAID"
}
```

https://developer.ebay.com/api-docs/buy/order/resources/checkout_session/methods/placeOrder#_samples

# STEP 1

# Embrace the standards

# How to protect them?

**Delegated Authorization**

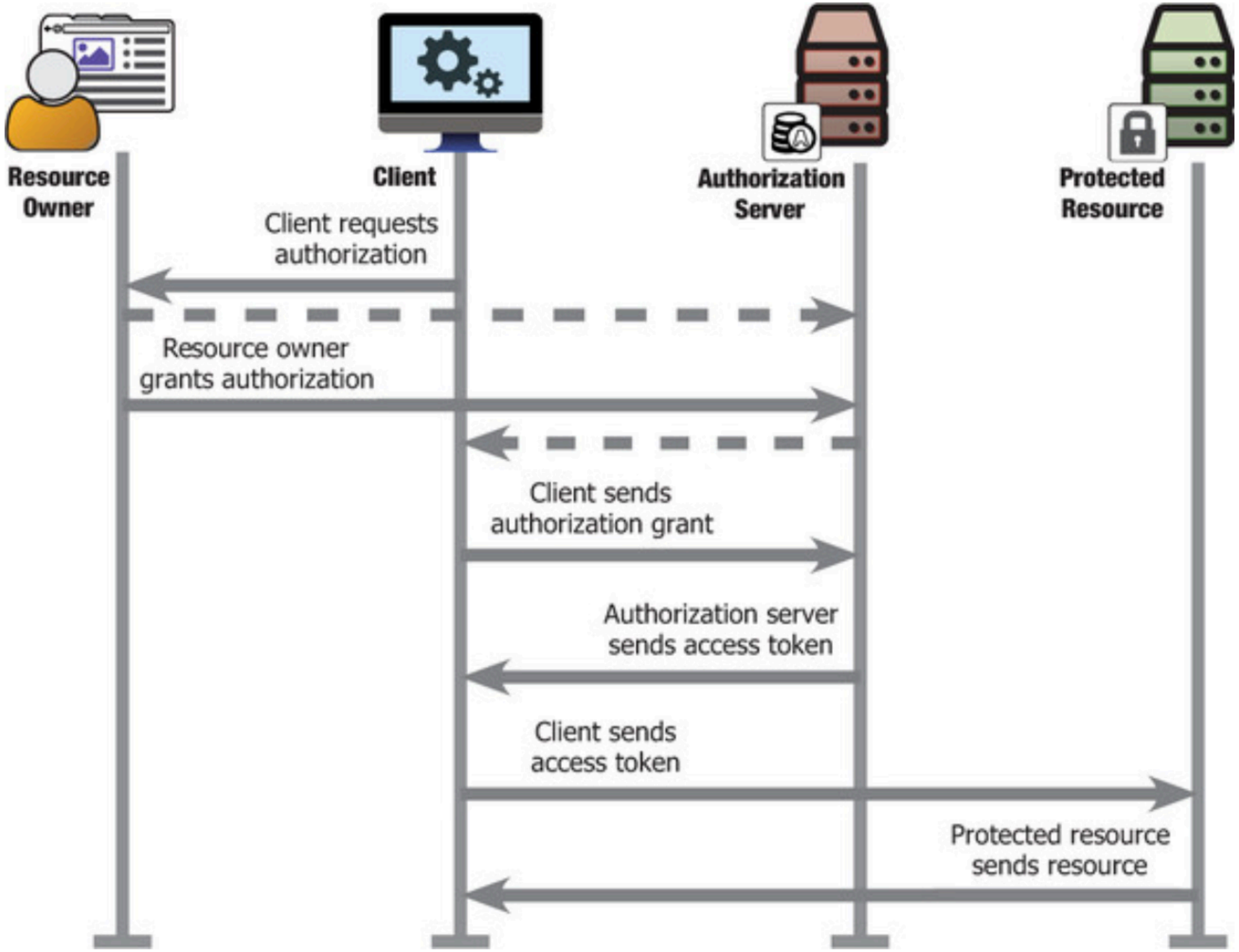
Delegated Authentication

Client Revocability

User Control

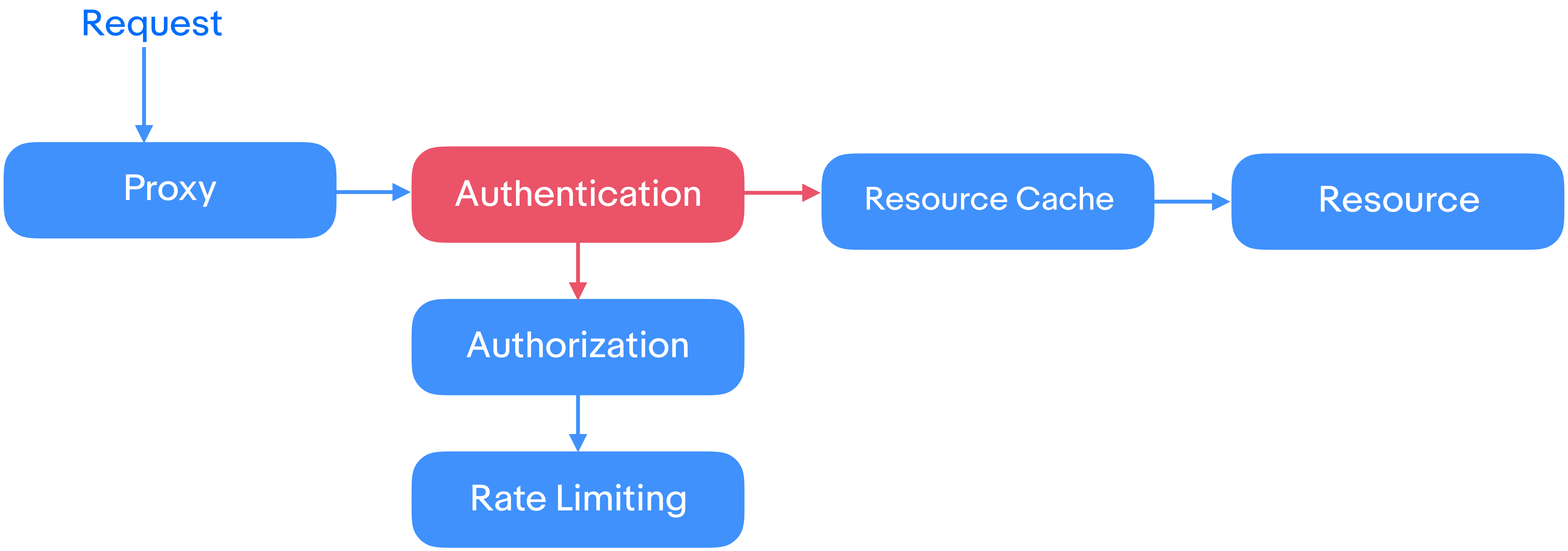
By Chris Messina, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=7188066

# How to protect them?


Resource Owner
Client
Authorization Server
Protected Resource
Client requests authorization
Resource owner grants authorization
Client sends authorization grant
Authorization server sends access token
Client sends access token
Protected resource sends resource

Source: OAuth2 in Action - By Justin Richer & Antonio Sanso

# Typical API Security Workflow



# Why “Authentication" is important?

Authorization

```
@PreAuthorize("hasPermission(#contact, 'admin')")
  public void deletePermission(Contact contact, Sid
recipient, Permission permission);
```

Rate Limiting

```
fs.setPath(“/hi")
 .requestRateLimiter(MyRL.args(2, 4,AppKeyResolver))
```

https://docs.spring.io/spring-security/site/docs/3.0.x/reference/el-access.html

## STEP 2

# Maintain an extensible token architecture

“If you decide to go and create your own token system,
you had best be really smart.”

*- Stack Overflow*

source

# What is a token?

“A token is a piece of data which only a **specific authentication server** could possibly have created & contains enough information to **identify a particular entity or entities.** They are created using various techniques from the field of **cryptography**.”

# What is a token?

"A token is a piece of data which only a specific authentication server could possibly have created & contains enough information to **identify a particular entity or entities**. They are created using various techniques from the field of cryptography."

# Entities

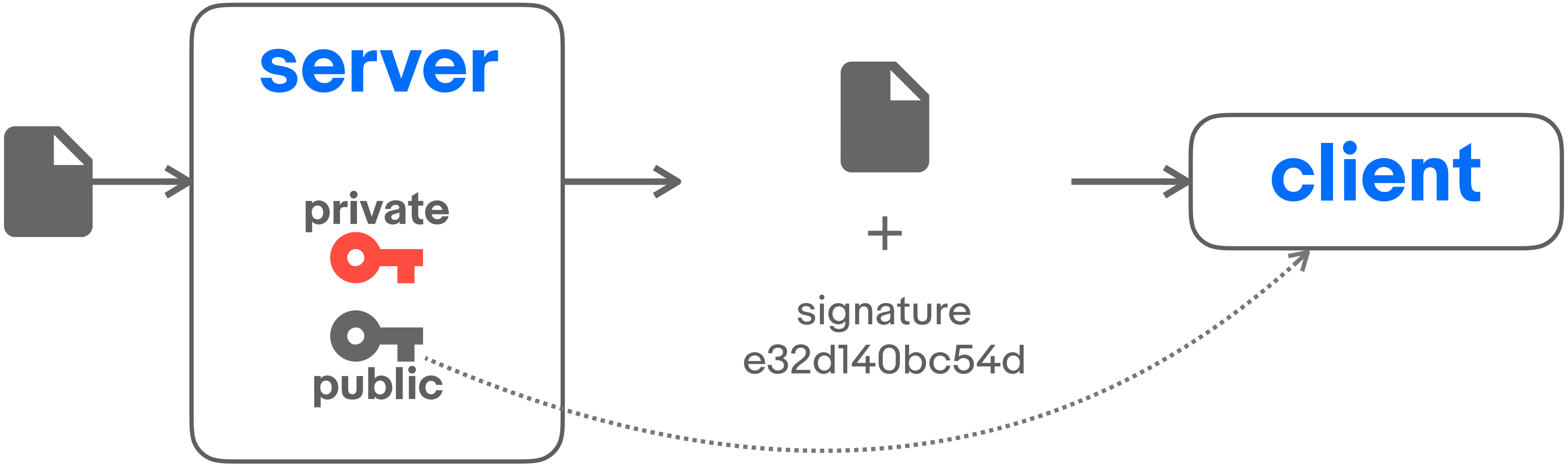
User Entity

Application Entity

# What is a token?

"A token is a piece of data which only a specific authentication server could possibly have created & contains enough information to identify a particular entity/entities. They are created using various techniques from the field of **cryptography**."

# Cryptography 101



signature
e32d140bc54d

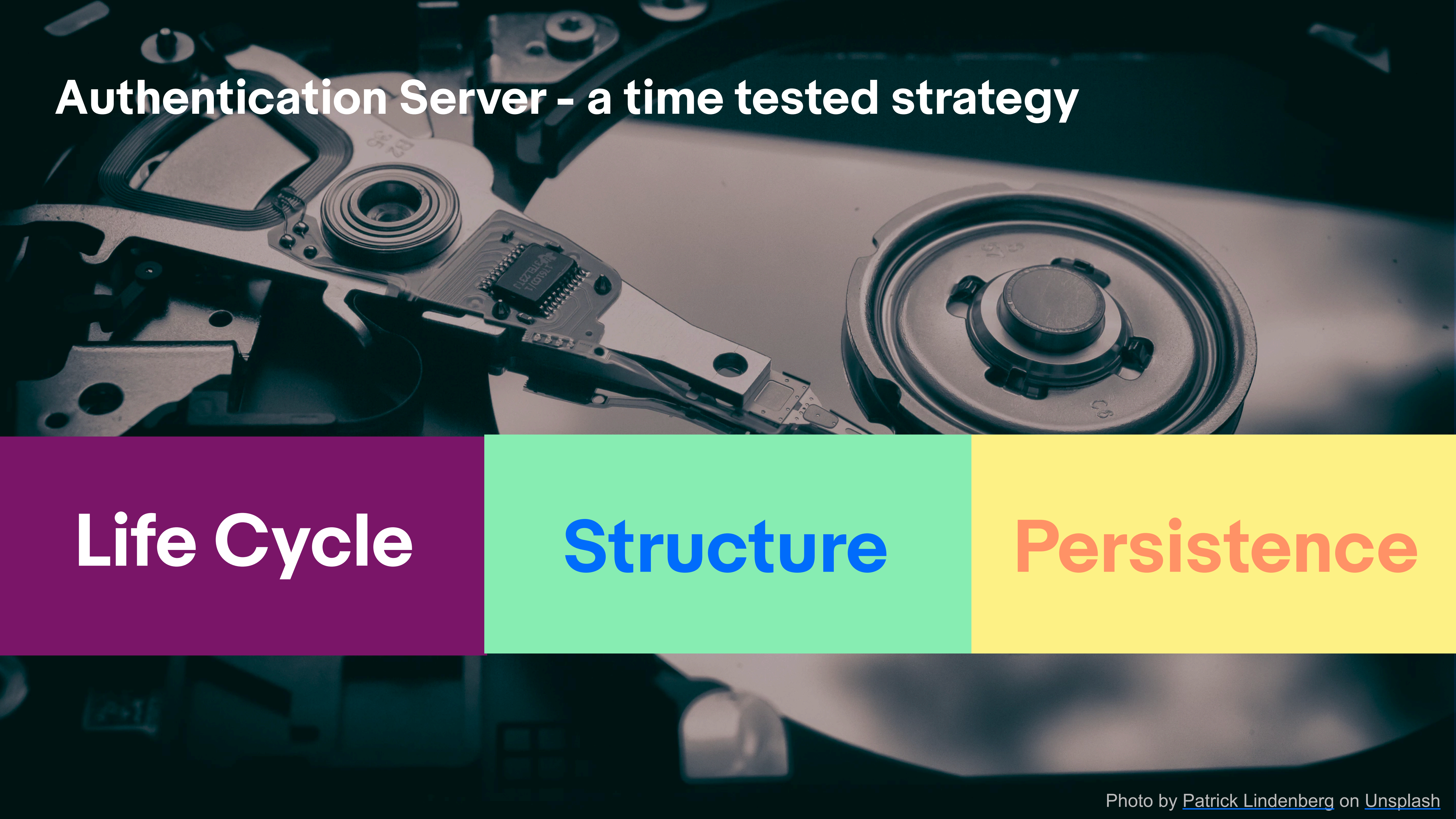
## STEP 3

# Learn the nuances of Cryptography

# What is a token?

"A token is a piece of data which only a **specific authentication server** could possibly have created & contains enough data to identify a particular entity. They are created using various techniques from the field of cryptography."

# Authentication Server - a time tested strategy



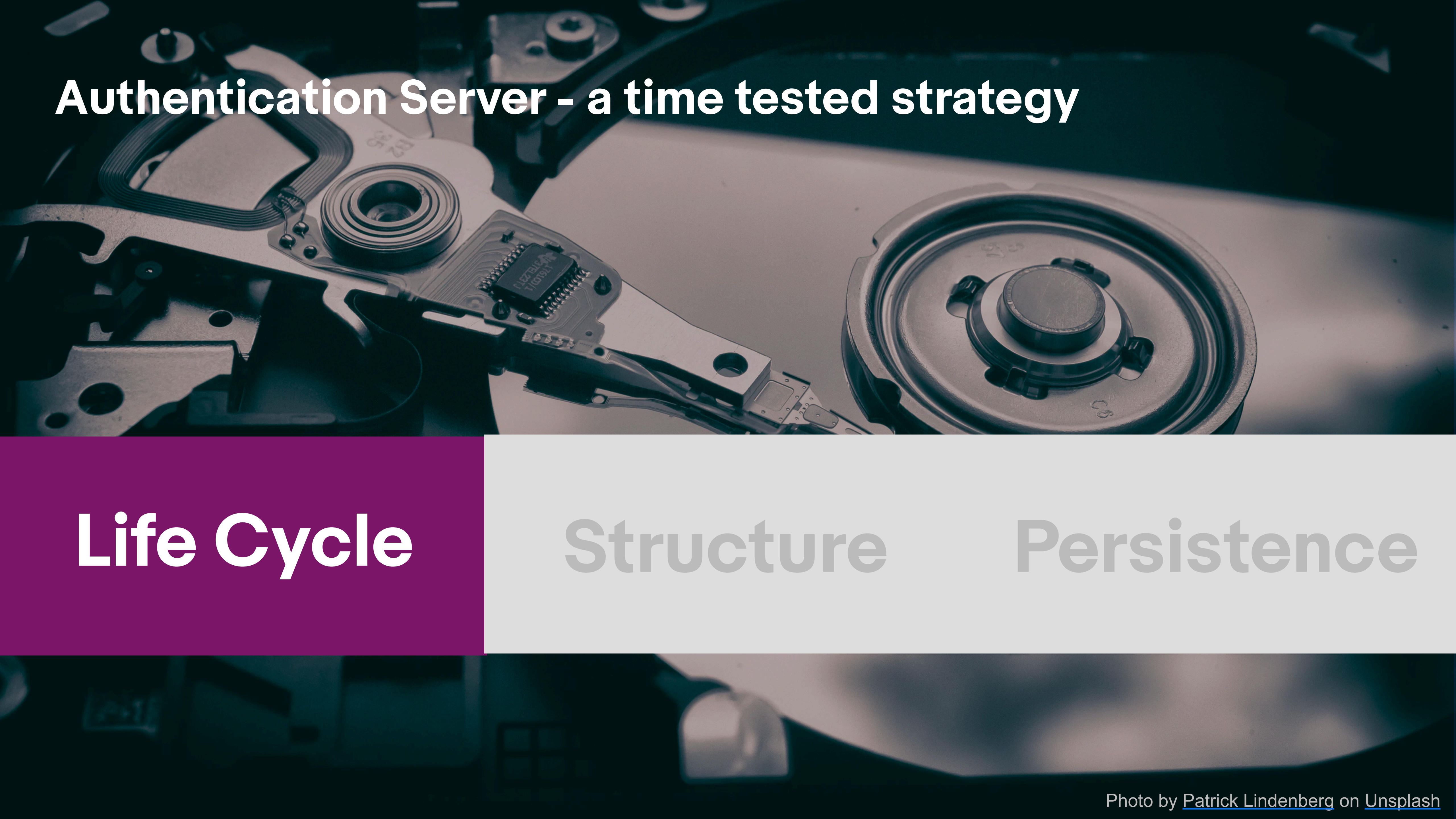
Life Cycle

Structure

Persistence

Photo by Patrick Lindenberg on Unsplash

# Authentication Server - a time tested strategy



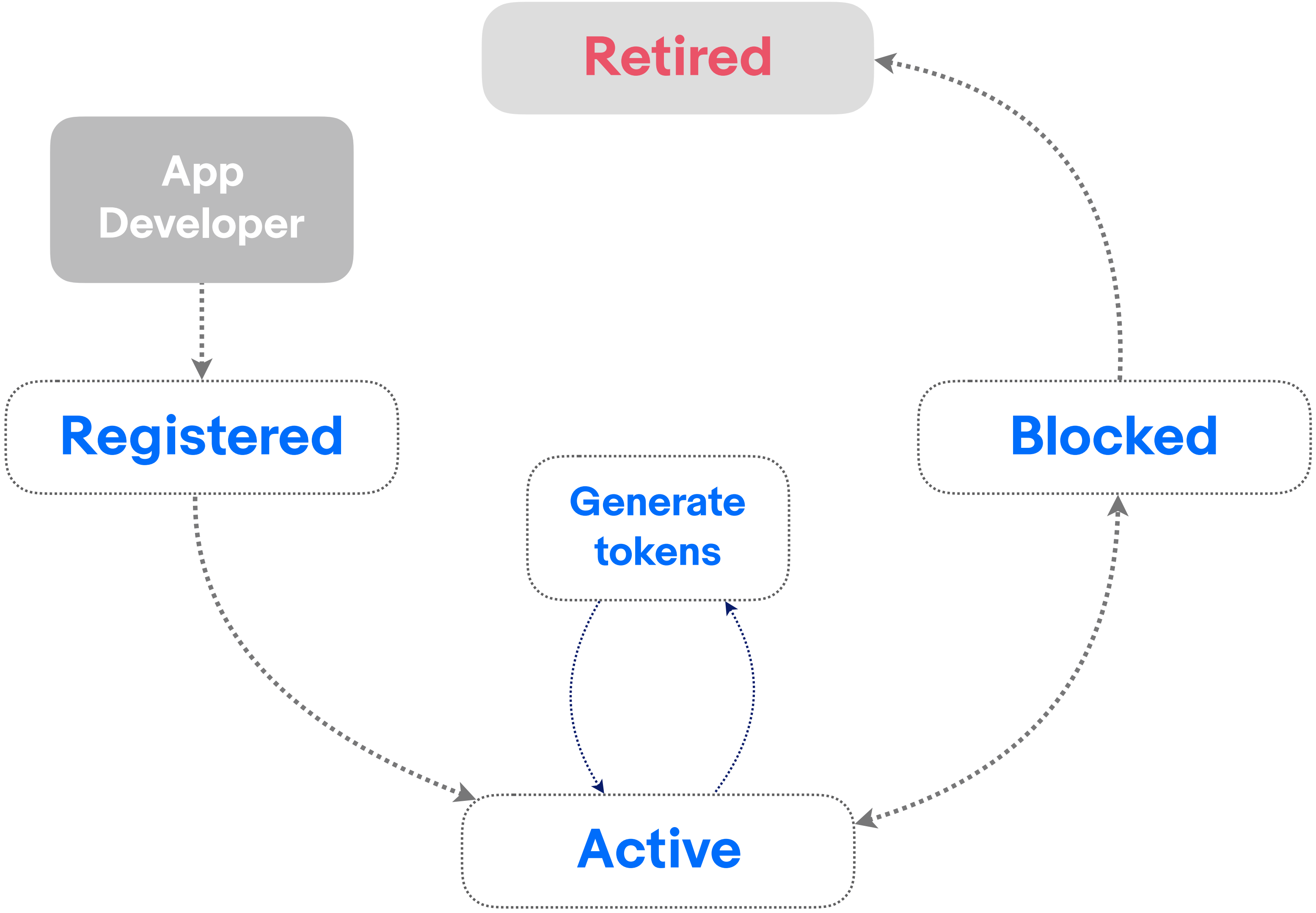
**Life Cycle**

Structure

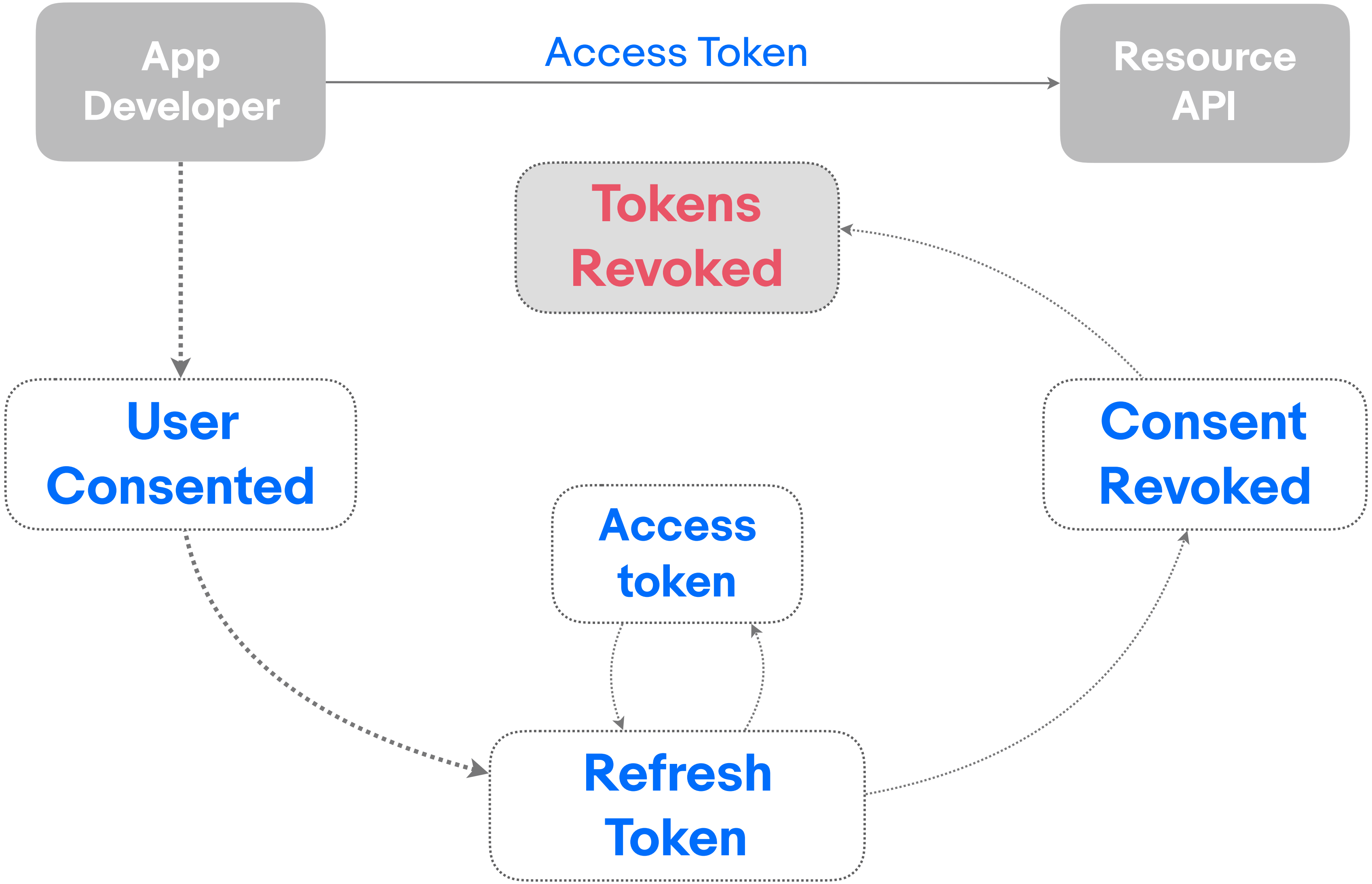
Persistence

Photo by Patrick Lindenberg on Unsplash

# LifeCycle - Application



# LifeCycle - Tokens



App Developer

Access Token

Resource API

Tokens Revoked

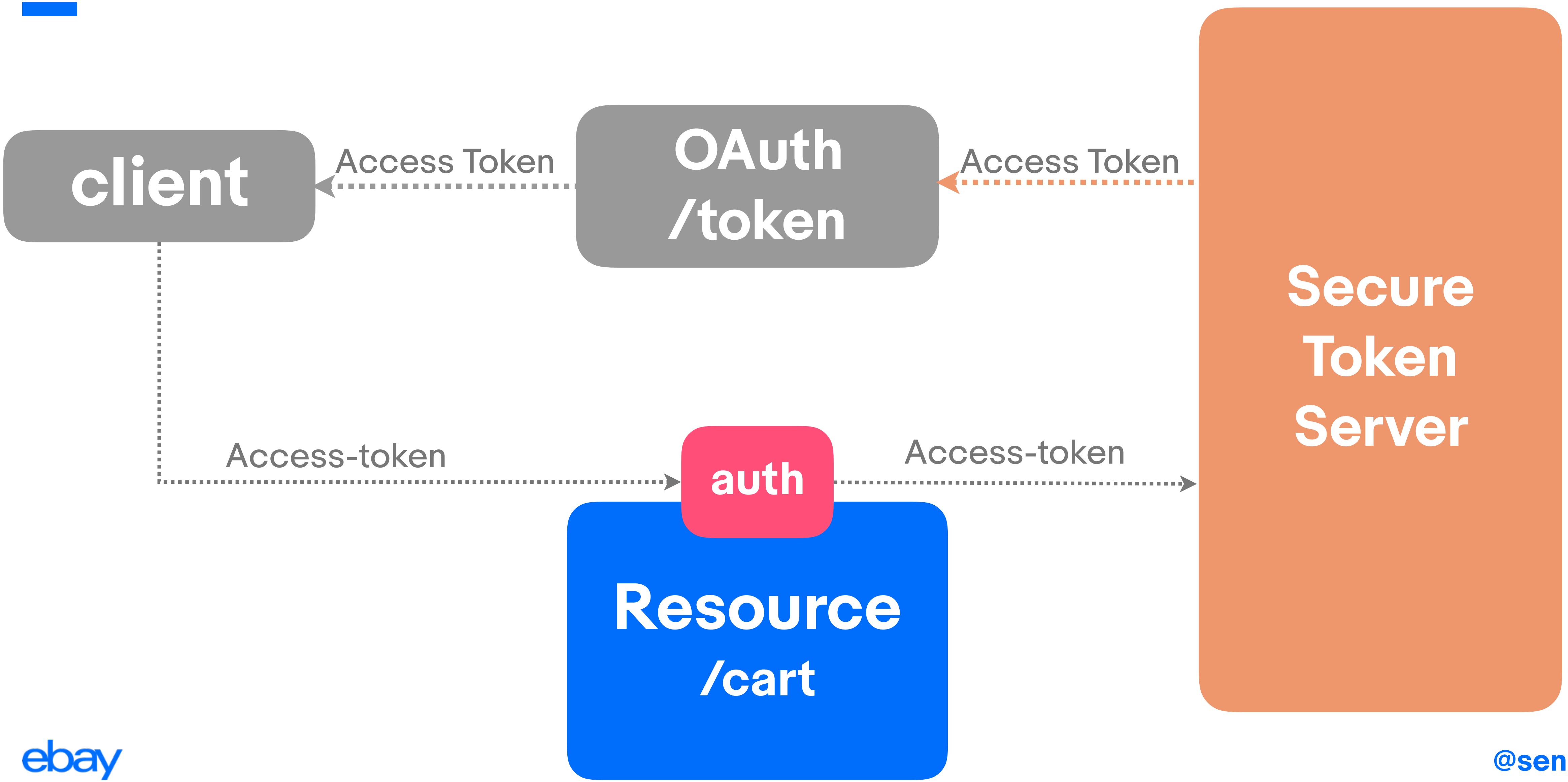
User Consented

Access token

Consent Revoked

Refresh Token

# Fitting it all together



# LifeCycle - Purpose

## Refresh Token

**Long Lived**

To Generate new Access Token

## Access Token

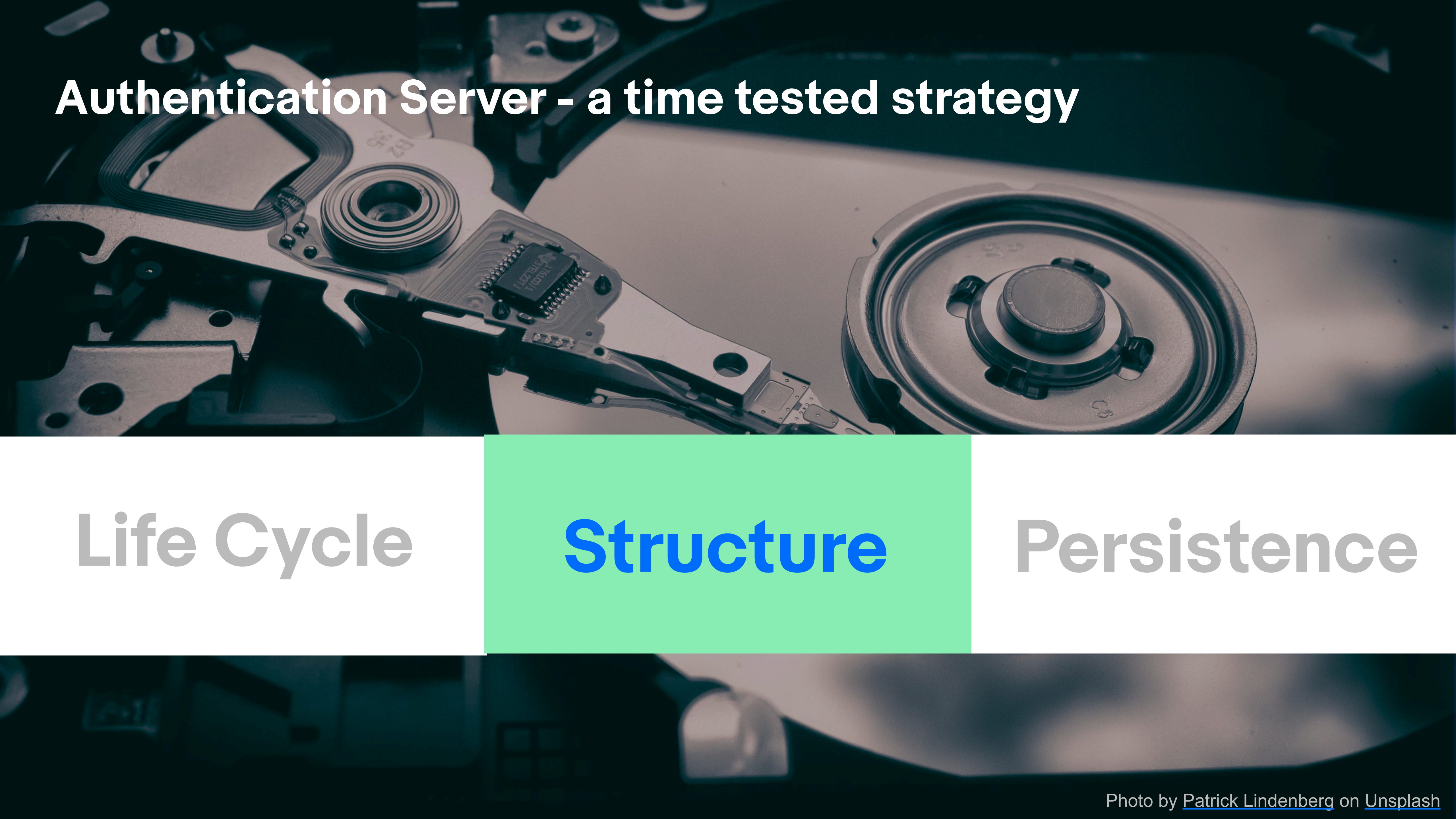
**Short Lived**

To Access protected Resource

# STEP 4

## ~~Learn~~ Live the nomenclature

# Authentication Server - a time tested strategy



Life Cycle

**Structure**

Persistence

Photo by Patrick Lindenberg on Unsplash

# Structure

## ebay

AgAAAA**AQAAAA**aAAAAA**E6+EWg**nY+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6wMkIGkCJCGoA
2dj6x9nY+seQ+/5wK1dskM5/3EOEY7BDg7VHK/CmDimCvVPbtJankHhzJUF8rU876Qzjs

## google

ya29.GltiBRICgroWhf0XJ-
e4nYpzc9UG0Fn_Ghq06_yg3BDZ4EHM_X8rlirEnFUJVb9uawqW2tE9yqfT0KwcaEXLKp7VFpde5v

## facebook

EAACEdEose0cBAJyrAOqIWCAPVobbylB7mZB7X3L0x5BLBosAAm2BDdUnhYKSp7VM9Tpyi8Ehr
AD6ZBYZBtymYC5ZBxNv1XrCBngEi0gEWLejezZb0gkArZBkJWcFiVjGcKYy44EY8ZD

** Tokens edited for brewity*

https://developer.ebay.com https://developers.facebook.com/tools/explorer/ https://developers.google.com/oauthplayground

# Structure

Is it just a random string?

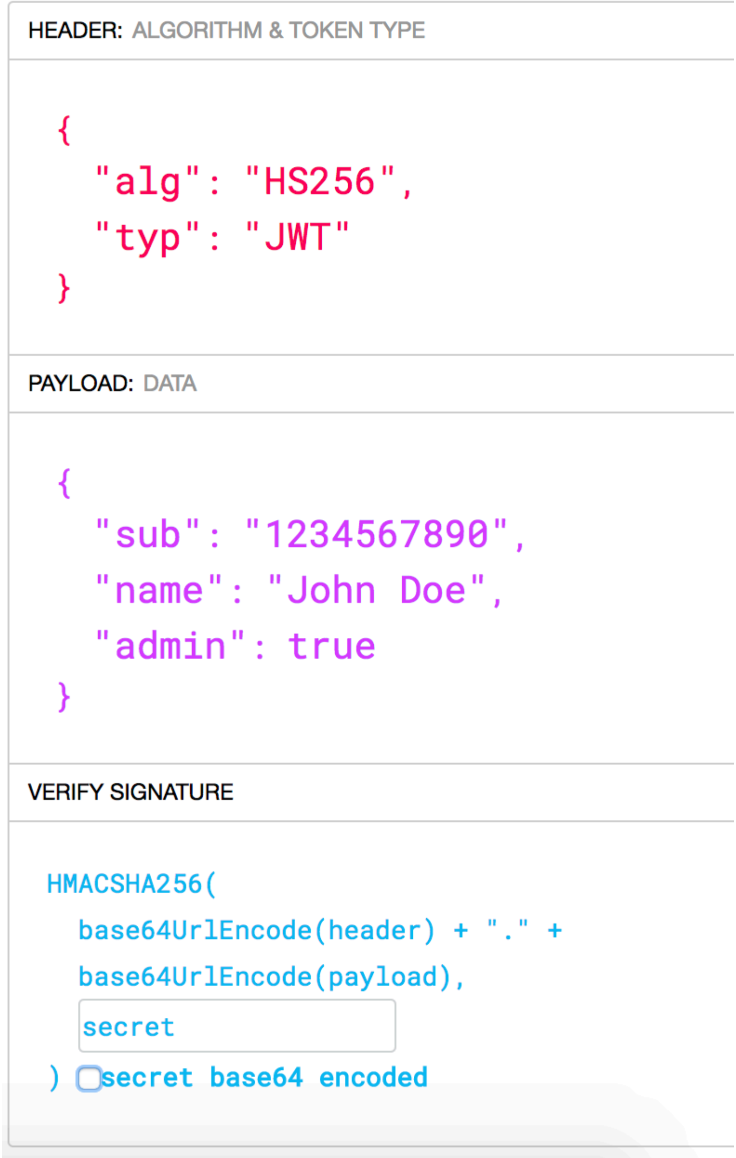
Are there any standards?

JWT

SAML

# Structure - JWT

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdW
IiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lI
iwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrH
DcEfxjoYZgeFONFh7HgQ
```



HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret
) □secret base64 encoded
```

https://jwt.io/

@sengopal

## STEP 5

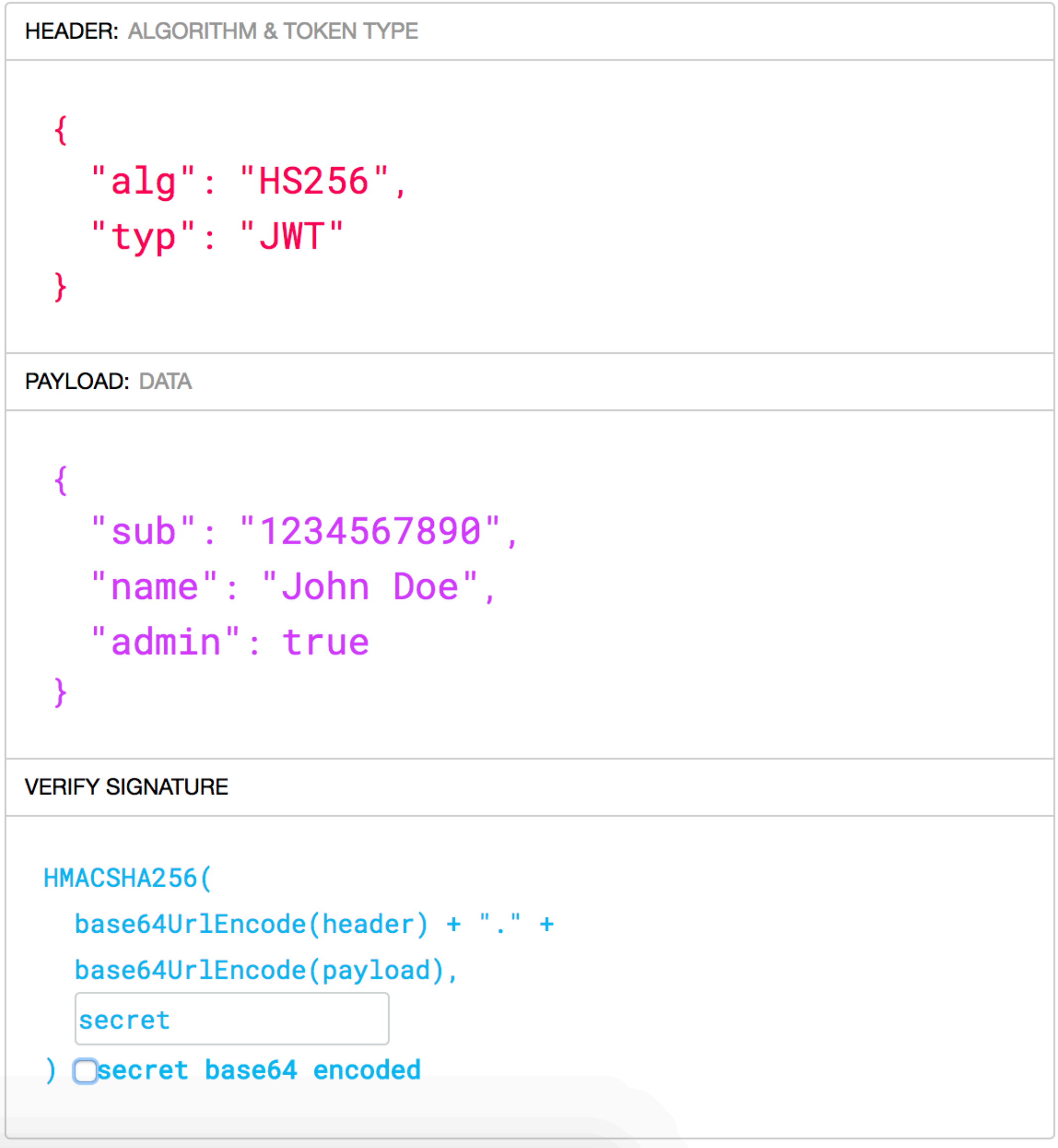
# Choose the token format wisely

(*standards*)

# Structure - JWT

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdW
IiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lI
iwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrH
DcEfxjoYZgeFONFh7HgQ

**What goes in the claim?**



HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

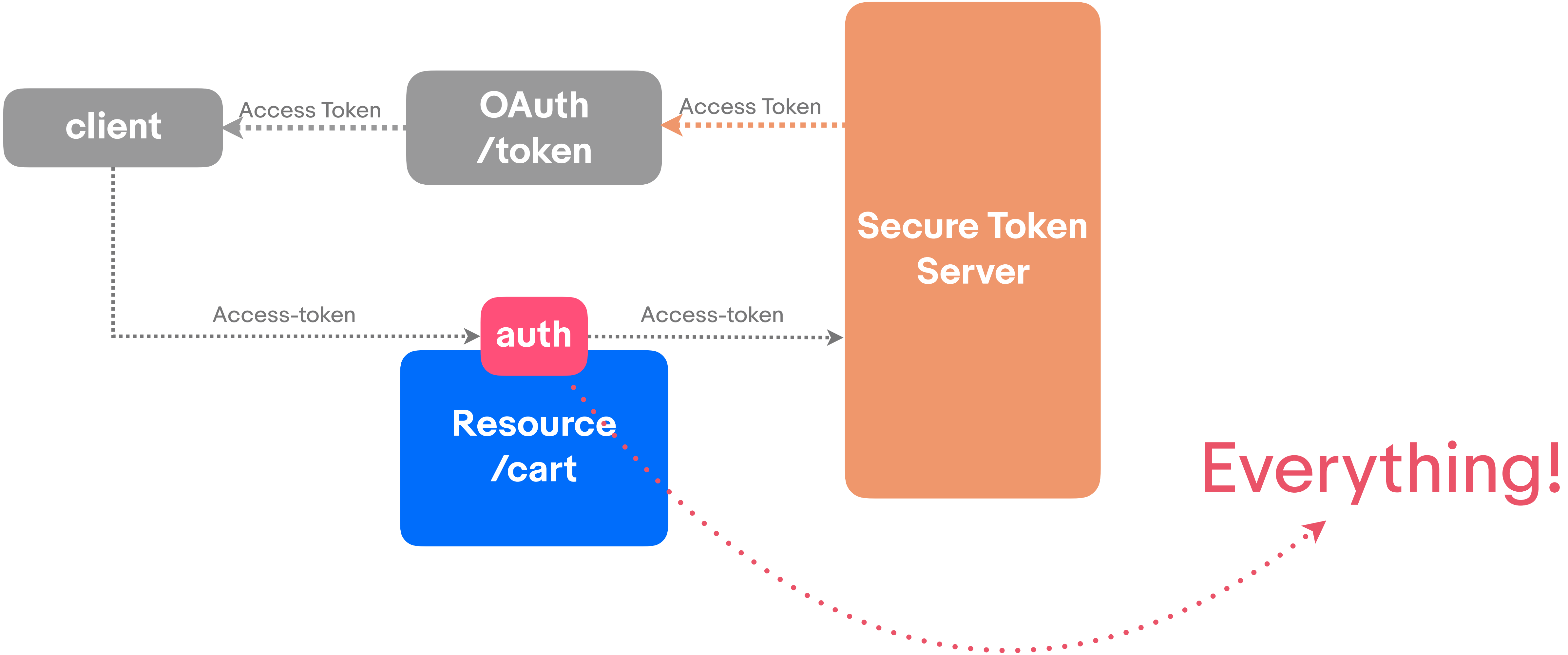
VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret
) □secret base64 encoded
```

https://jwt.io/

@sengopal

# Structure - What goes in the claim?



# Structure - Why everything?

tokens

Service APIs

IS SAME AS

cookies

Web Apps

**User entity**
**App entity**
issuer
issueAt
expiresAt
deviceIdentifier
trackingId
...

Photo by Jennifer Pallian on Unsplash

# Structure - Versioning

## We add new attributes everyday.

## Versioning

v1, v1.1, v1.2, v1.3, v2.0, ....

**User entity**
**App entity**
issuer
issueAt
**version**

expiresAt
deviceIdentifier
trackingId
...

## STEP 6

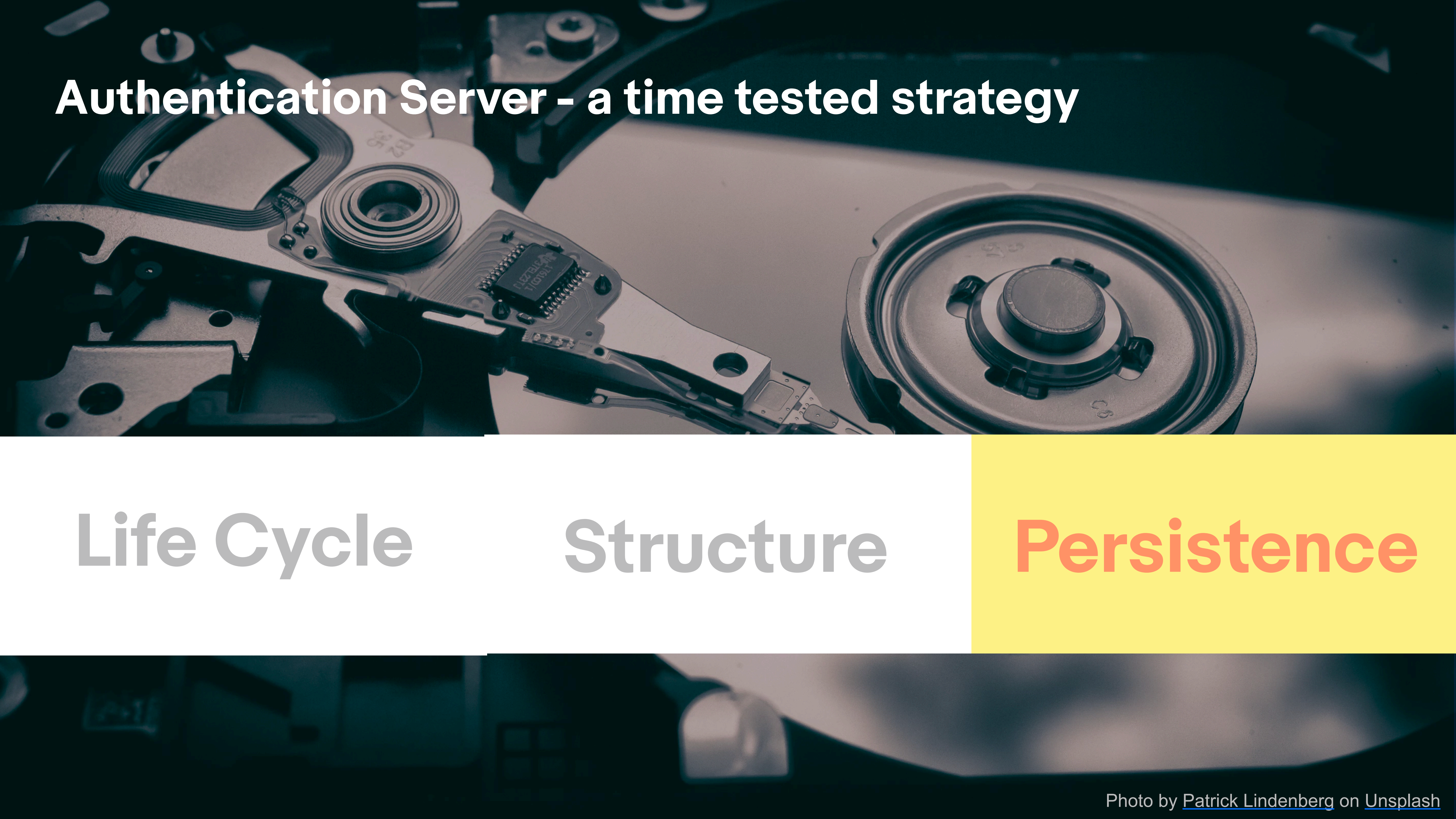
# Capture every identifier possible and use versioning

# No!

## One more important step

Photo by DeviantArt

# Authentication Server - a time tested strategy

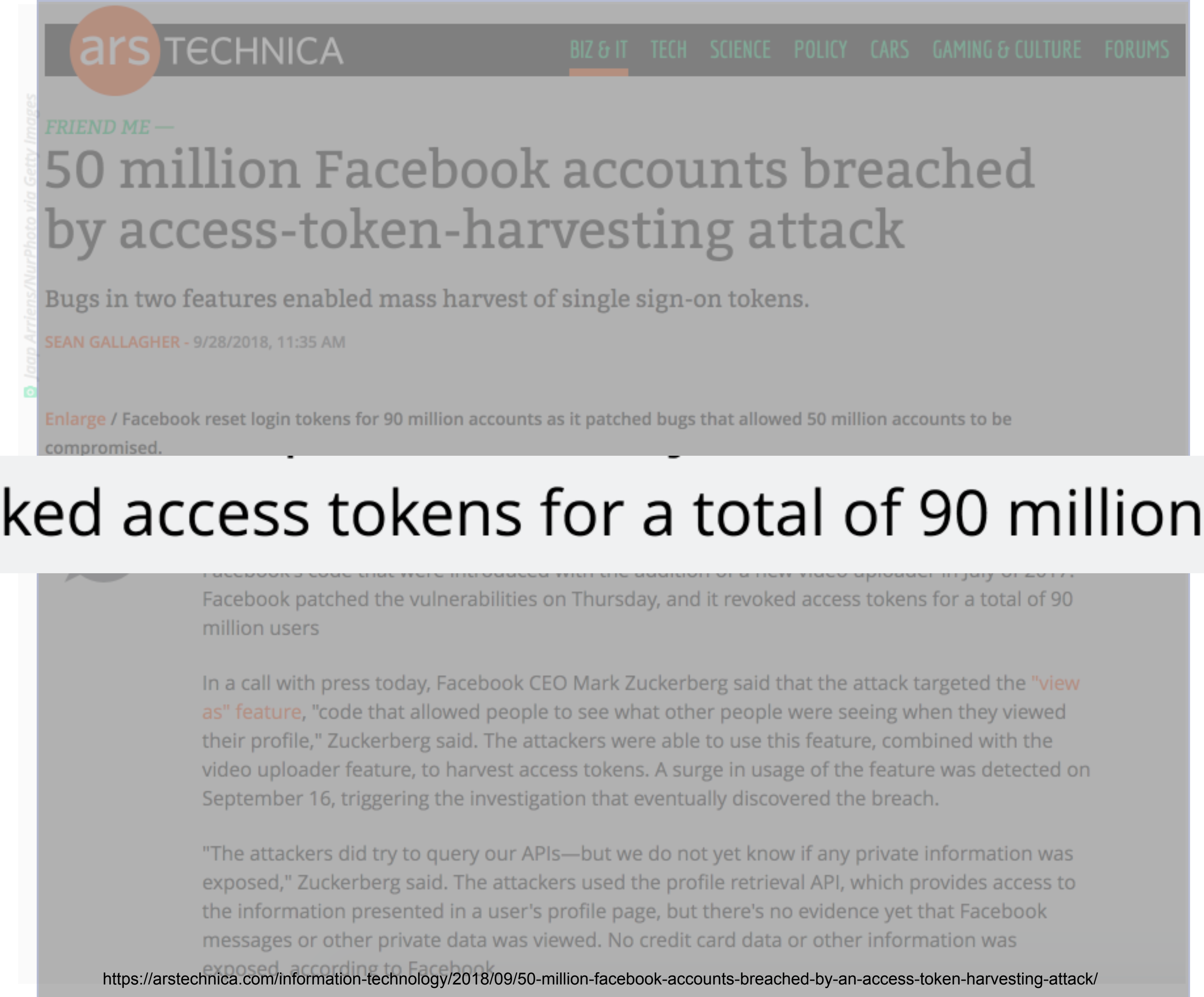


Life Cycle

Structure

**Persistence**

Photo by Patrick Lindenberg on Unsplash



ars TECHNICA

FRIEND ME —

# 50 million Facebook accounts breached by access-token-harvesting attack

Bugs in two features enabled mass harvest of single sign-on tokens.

SEAN GALLAGHER - 9/28/2018, 11:35 AM

Joop Arriens/NurPhoto via Getty Images

Enlarge / Facebook reset login tokens for 90 million accounts as it patched bugs that allowed 50 million accounts to be compromised.

Facebook's code that were introduced with the addition of a new video uploader in July of 2017. Facebook patched the vulnerabilities on Thursday, and it revoked access tokens for a total of 90 million users

In a call with press today, Facebook CEO Mark Zuckerberg said that the attack targeted the "view as" feature, "code that allowed people to see what other people were seeing when they viewed their profile," Zuckerberg said. The attackers were able to use this feature, combined with the video uploader feature, to harvest access tokens. A surge in usage of the feature was detected on September 16, triggering the investigation that eventually discovered the breach.

"The attackers did try to query our APIs—but we do not yet know if any private information was exposed," Zuckerberg said. The attackers used the profile retrieval API, which provides access to the information presented in a user's profile page, but there's no evidence yet that Facebook messages or other private data was viewed. No credit card data or other information was exposed, according to Facebook.

## revoked access tokens for a total of 90 million users

https://arstechnica.com/information-technology/2018/09/50-million-facebook-accounts-breached-by-an-access-token-harvesting-attack/

@sengopal

# Security

```
{
  "sub": "110169484474386276334",
  "name": "John Doe",
  "iss": "https://www.ebay.com",
  "iat": "1433978353",
  "exp": "1433981953",
  "email": "testuser@gmail.com",
  "email_verified": "true",
  "given_name": "Test",
  "family_name": "User",
  "locale": "en"
}
```

## JWT - Claim

**Integrity Verified**

**Missing**

- Confidentiality
- Revocation

# Security

## By Value

```
{
  "sub": "110169484474386276334",
  "name": "John Doe",
  "iss": "https://www.ebay.com",
  "iat": "1433978353",
  "exp": "1433981953",
  "email": "testuser@gmail.com",
  "email_verified": "true",
  "given_name": "Test",
  "family_name": "User",
  "locale": "en"
}
```

## By Reference

```
{
  “ref”:”
AgAAAA**AQAAAA**aAAAAA**E6+EWg*
*nY+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6
wMkIGkCJCGoA2dj6x9nY+seQ+/
5wK1dskM5/3EOEY7BDg7VHK/
CmDimCvVPbtJankHhzJUF8rU876Qzjs
”
}
```

# Security

## By Value

Integrity Verified

## By Reference

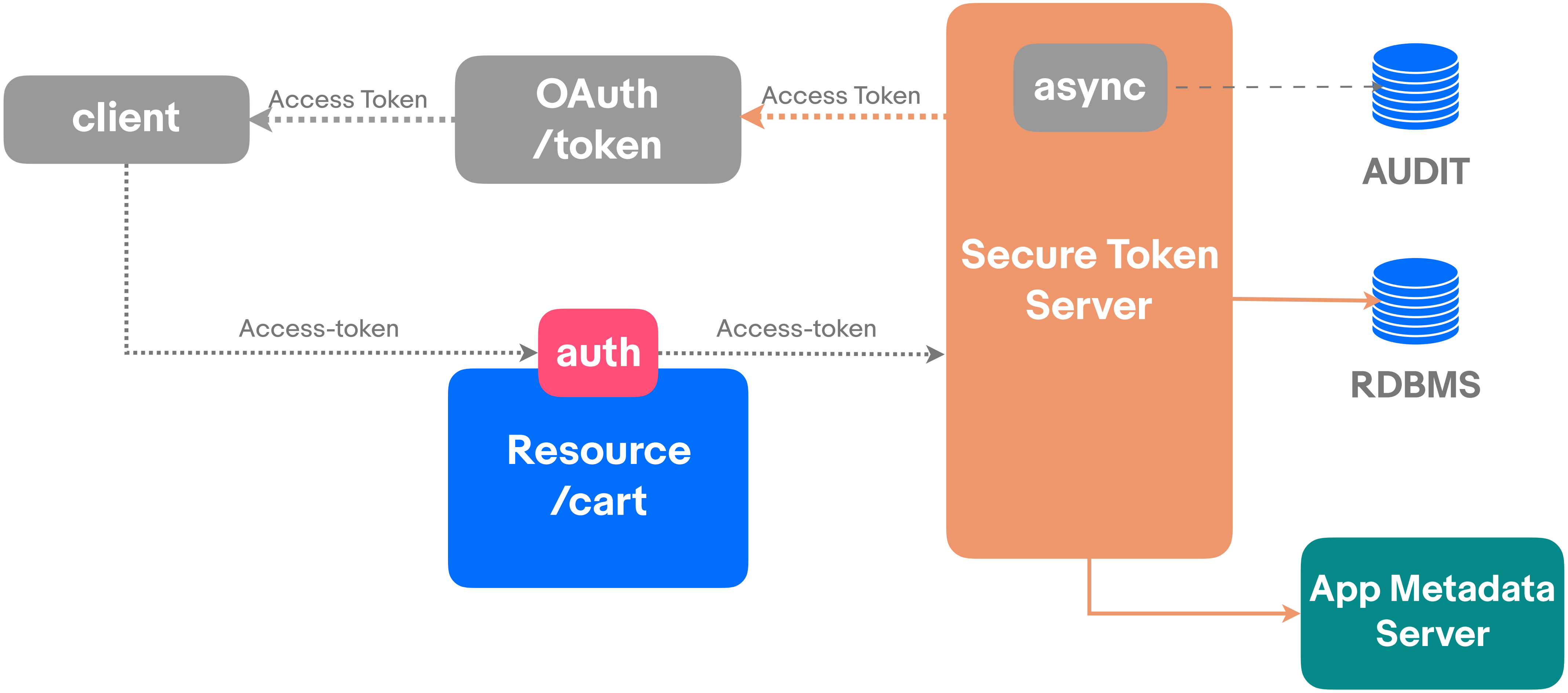
Integrity Verified

Confidential

Custom format *

**Persisted**

# Fitting them together



# Persistence - Considerations

## Atomic & Strong Consistency

Token Generation of new tokens

Token Revocation *

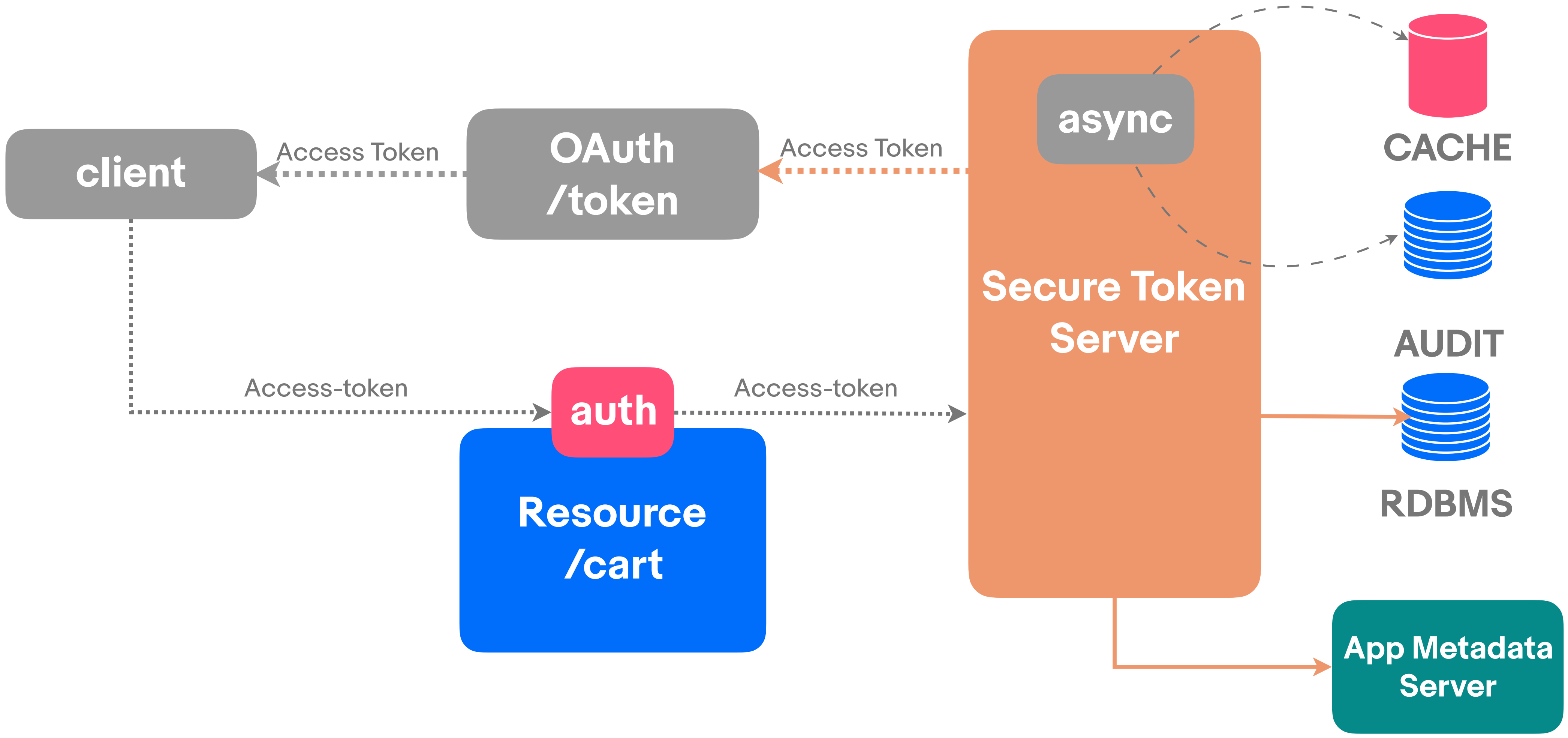
# Persistence - Considerations

Eventually Consistent

User - token Auditing

Cache duplication

# Fitting them together



ebay

@sengopal

# STEP 7

# Identify transactional needs

# Minimal Token Exposure

```
{
  "sub": "110169484474386276334",
  “exp": "14339732223"
   ....
  "given_name": "Test",
  "family_name": “User”,
  "email": “testuser@gmail.com”,
  "iat": "14339732223",
  “scopes": “buy.order item.feed”
}
```

```
@PreAuthorize("hasPermission(#contact, ‘buy.order')")
public void buyOrder(Contact contact);
```

## STEP 8

# Allow only minimal scopes and least expiration time

# OWASP

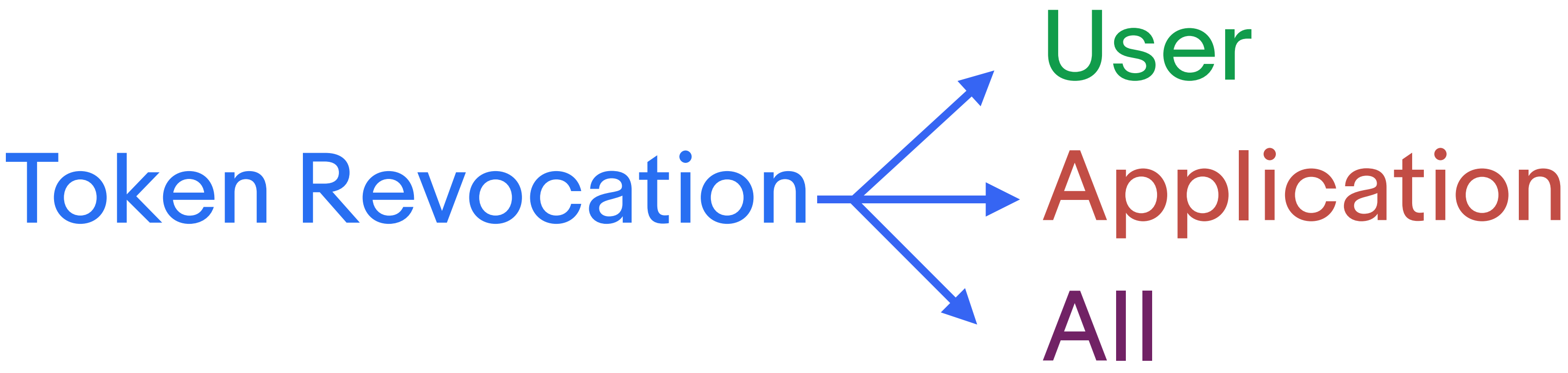
## Open Web Application Security Project

**A2 – Broken Authentication and Session Management**

**A10 – Underprotected APIs**

Reference

# Fire Drill - Revocation Strategy



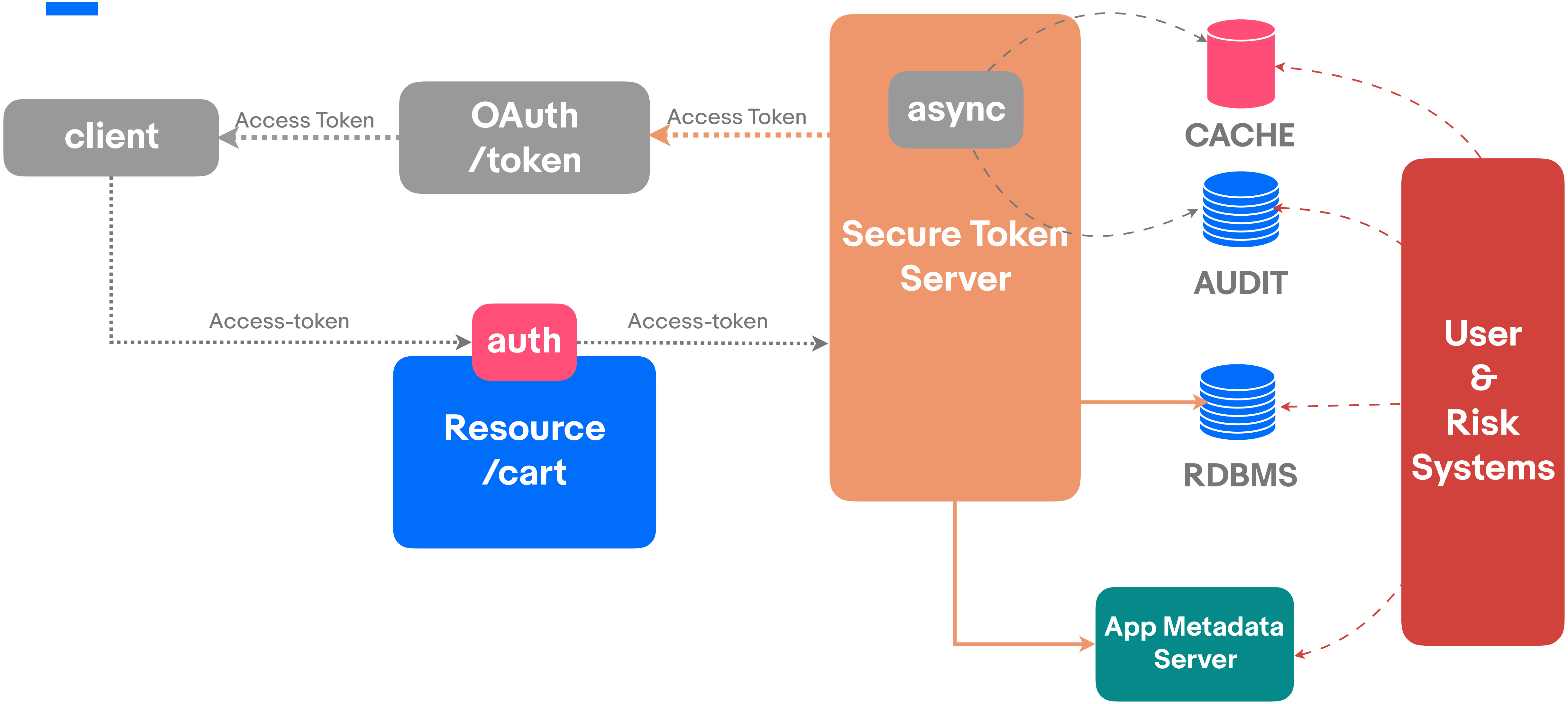
Token Revocation

User

Application

All

# Fitting them together



@sengopal

# STEP 9

## Audit all access patterns and “be prepared”

# Managing the whole show

Application Lifecycle

Token lifecycle

Cryptography artifacts rotation

Authorizations registry

….

## STEP 10

# Automate Everything

# And the 10 steps are ....

Embrace the standards

Extensible token architecture

Nuances of Cryptography

Learn the nomenclature

Correct token format

All identifiers & versioning

Identify transactional needs

Allow only minimal scopes

Audit all access patterns

Automate Everything

# Thank You!

Blogs @ **http://sengopal.me**

Tweets **@sengopal**

Slides and Code @ **http://bit.ly/api-token**