Ten steps for
Token based API Security

Senthilkumar Gopal

ebay

ACME Fort Knox Web Application
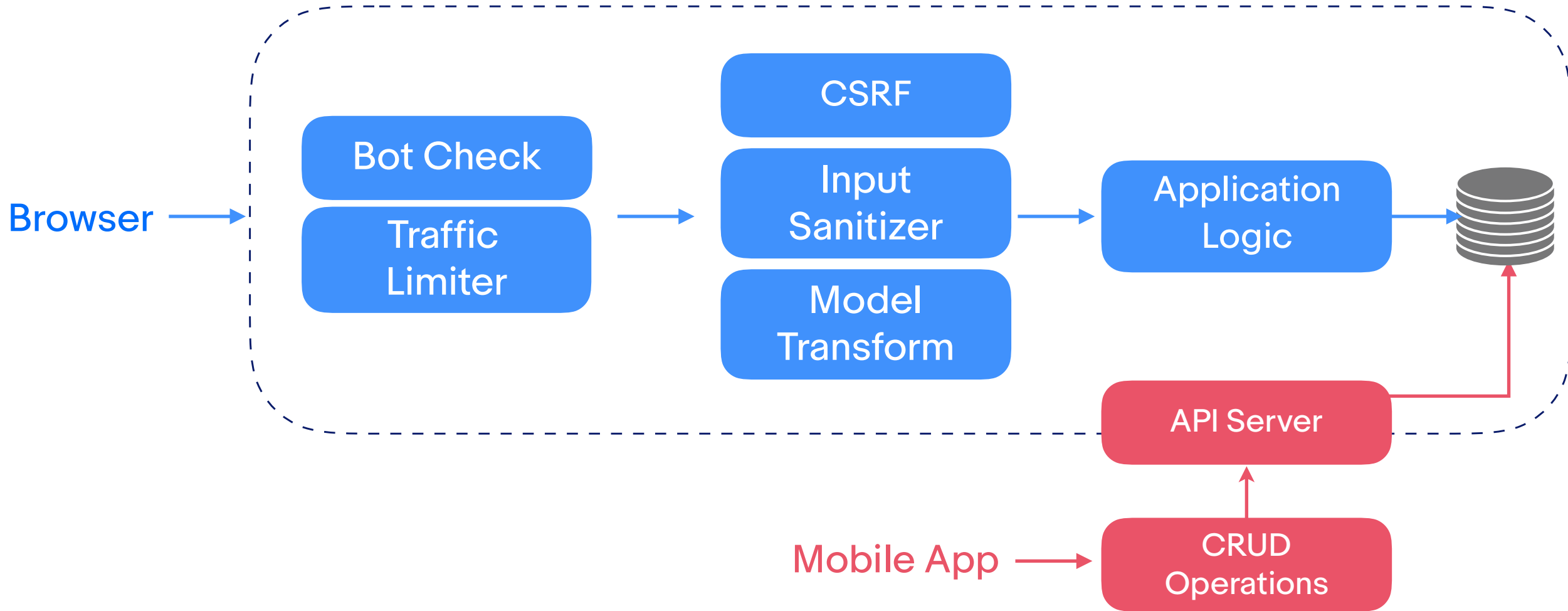
Browser →

- Bot Check
- Traffic Limiter

→

- CSRF
- INPUT SANITIZER
- MODEL TRANSFORM
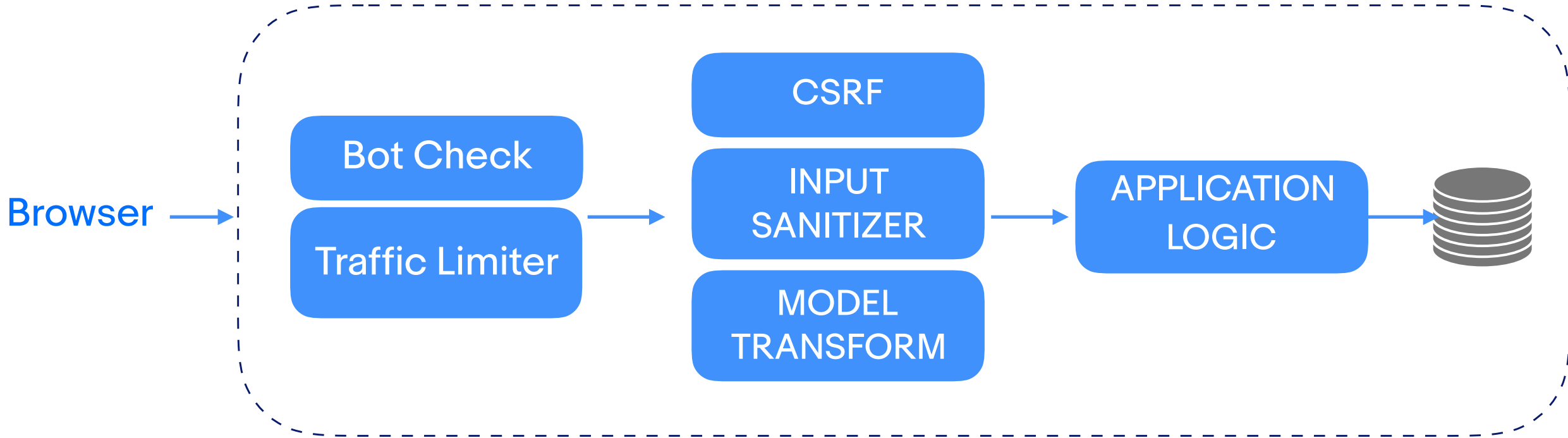
→ APPLICATION LOGIC →

A Hero's ('real') story

Build an
Awesome
Mobile App

ACME (Not) Fort Knox Web Application

Browser →

Bot Check
Traffic Limiter

→

CSRF
Input Sanitizer
Model Transform

→

Application Logic

→ (database)

API Server

Mobile App →

CRUD Operations

ebay

Web Application vs. APIs

" But no one else knew about the API server "

Web Application vs. APIs

ebay

02 **Panerabread.com Leaks Millions of Customer**
APR 18 **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.

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.

A Hero's ('real') story

I need an 'expert'

First Principles

APIs are ...

Closer to Object Data Model

Intended to serve machines instead of real users

ebay

Example of Web Application vs. APIs

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

```
1  {}
```

Output

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

```
1 ▾ {
2       "purchaseOrderId": "5832216756",
3       "purchaseOrderHref": "https://orderapi.ebay.com/buy/order/v1/purchase_order/5832216756",
4       "purchaseOrderPaymentStatus": "PAID"
5   }
```

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

ebay

@sengopal

STEP 1

Embrace the standards

How to protect them?

Delegated Authentication

Delegated Authorization

Client Revocability

User Control

Code @ http://bit.ly/ebay-oauth

ebay

@sengopal

How to protect them?

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

ebay
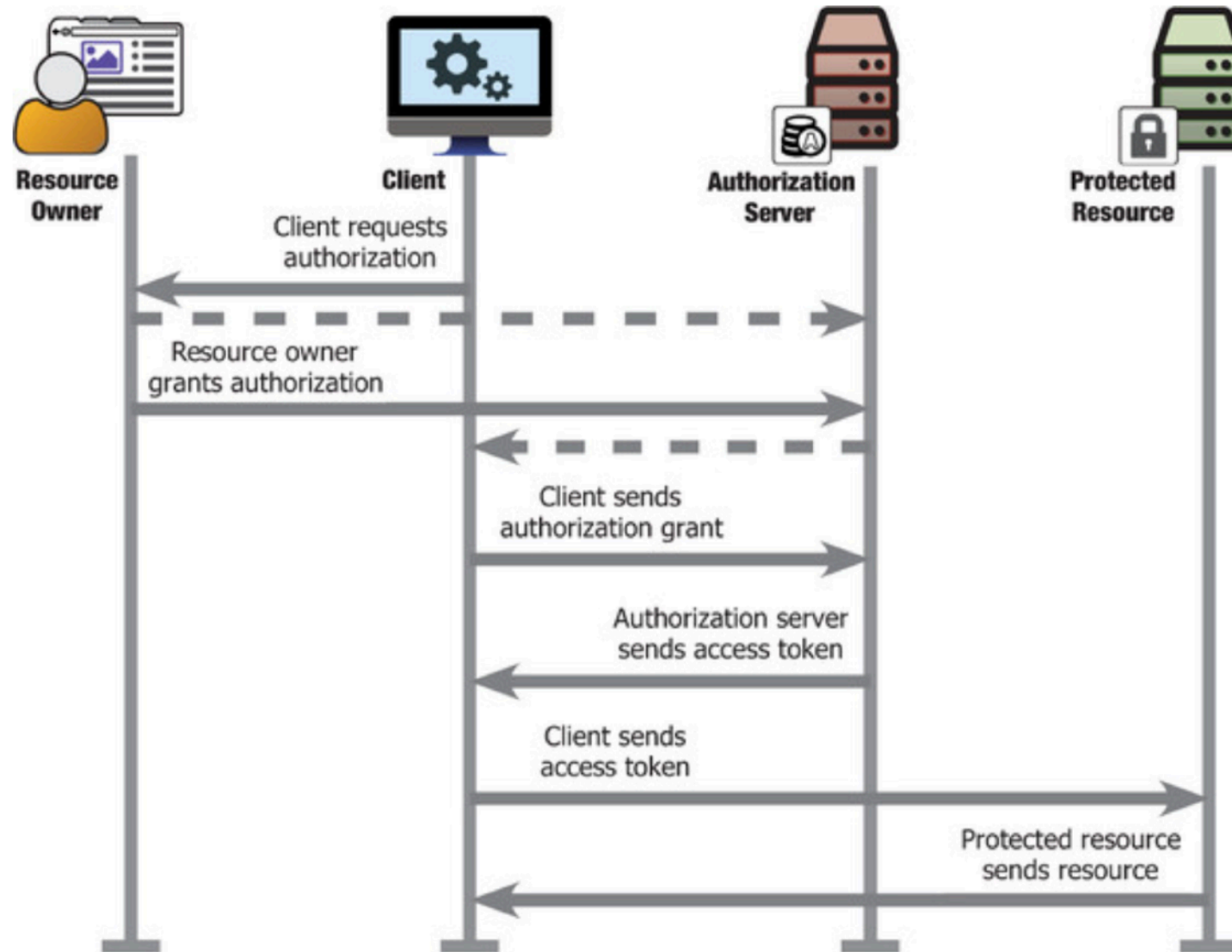
Typical API Security Workflow

ebay
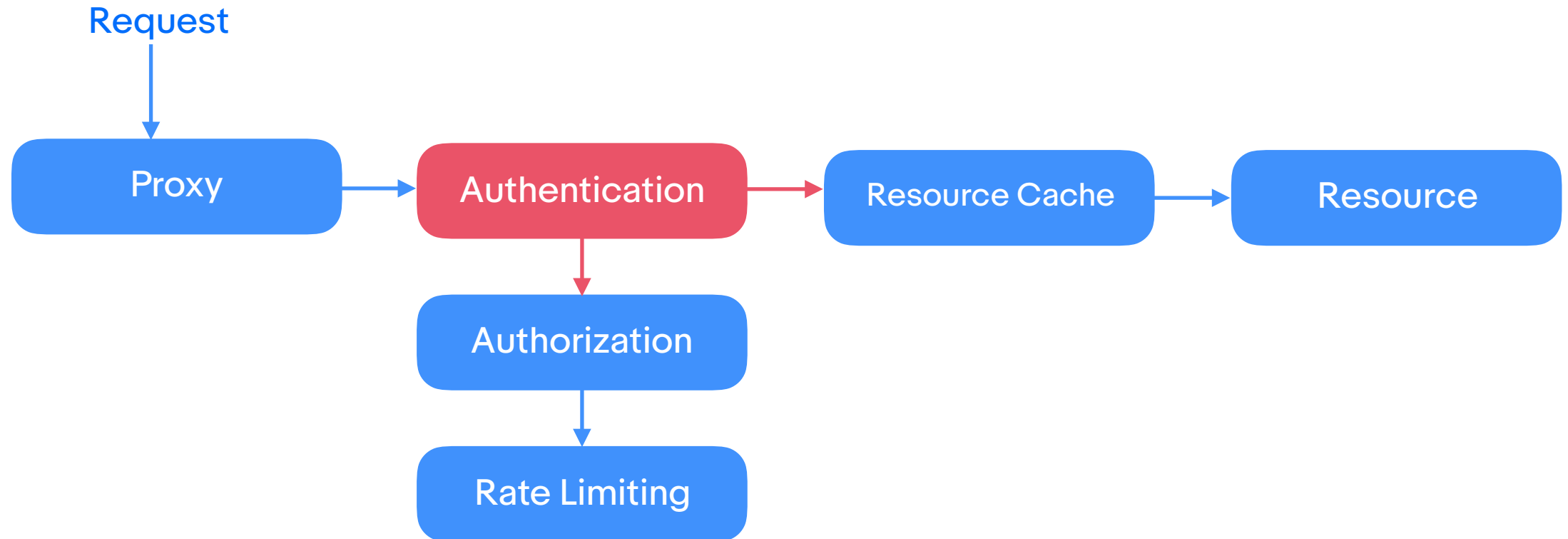
Why "Authentication" is important?

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))
```

ebay

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

@sengopal

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

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**."

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."

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

server

private

public

+

signature
e32d140bc54d

client

STEP 3

Learn the nuances of Cryptography

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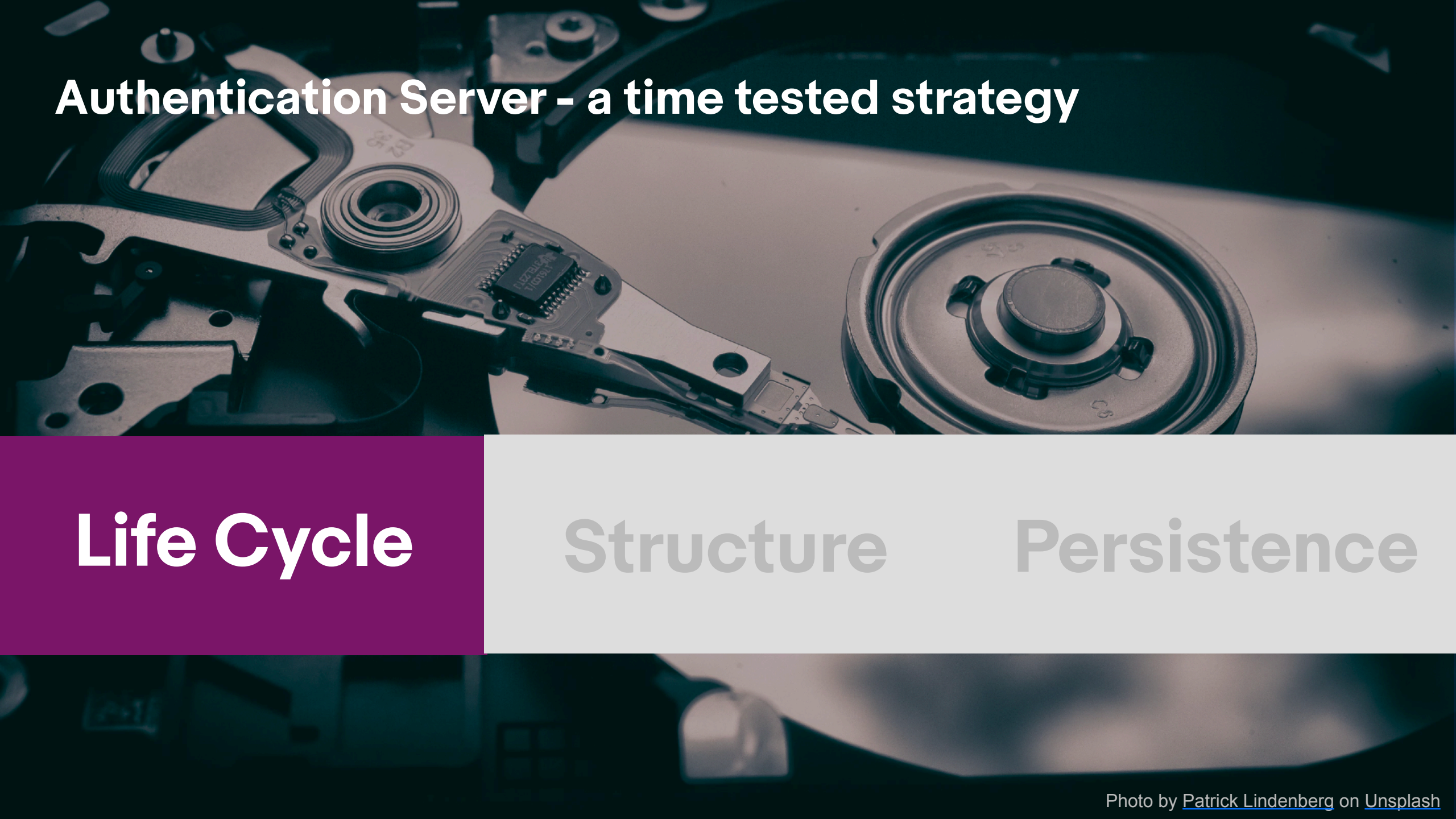
"A token is a piece of data which only a **specific authentication server** could possibly have created & contains enough data to identify a particular entity. They are created using various techniques from the field of cryptography."

Authentication Server – a time tested strategy

Life Cycle   Structure   Persistence

Authentication Server - a time tested strategy

Life Cycle

Structure

Persistence

LifeCycle - Application

LifeCycle - Tokens

App Developer → **Access Token** → Resource API

Tokens Revoked

App Developer → User Consented

User Consented → Refresh Token

Access token

Refresh Token

Consent Revoked

ebay

@sengopal

Fitting it all together

client — Access Token — OAuth /token — Access Token — Secure Token Server

client — Access-token — auth — Access-token — Secure Token Server

Resource /cart

ebay
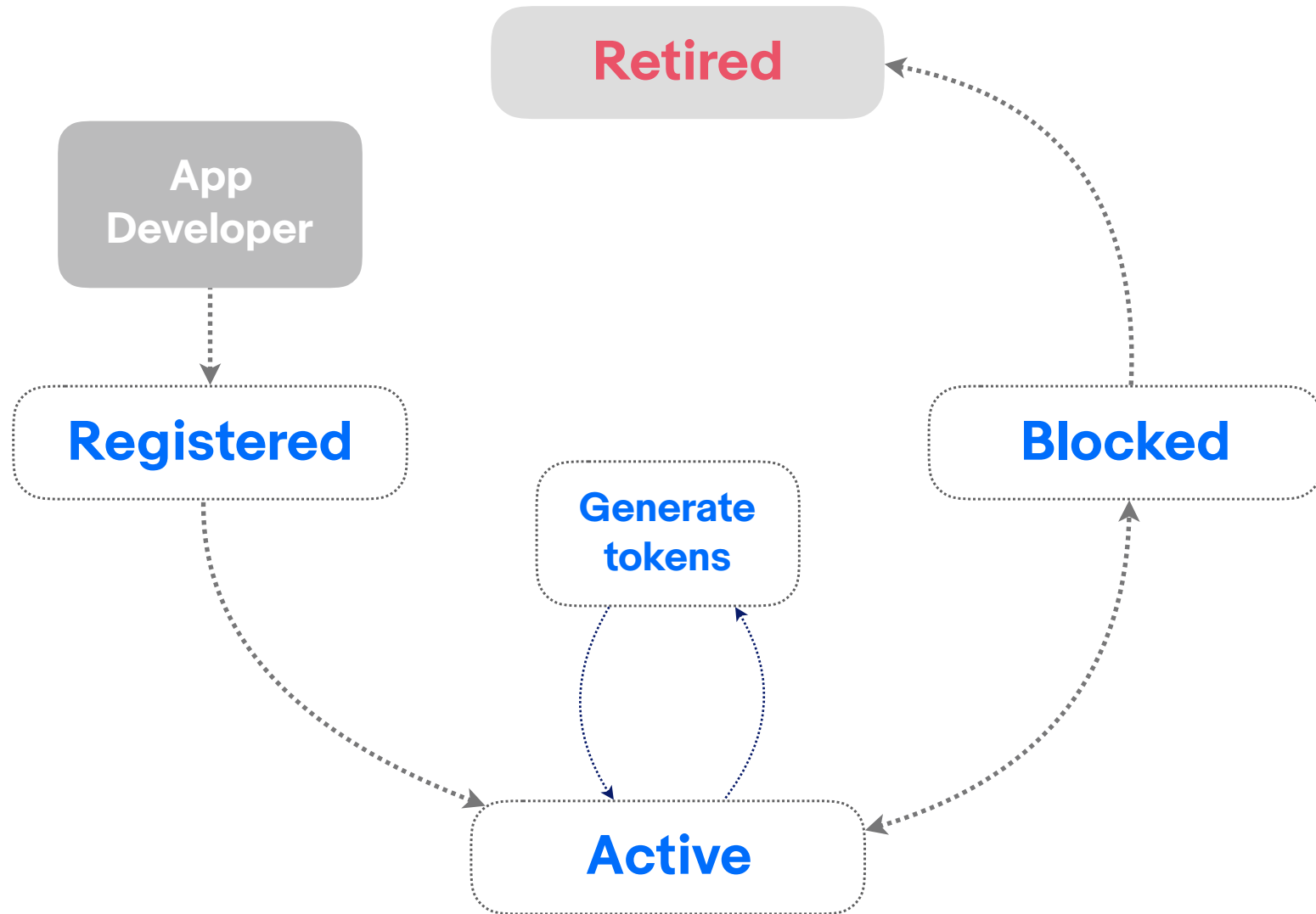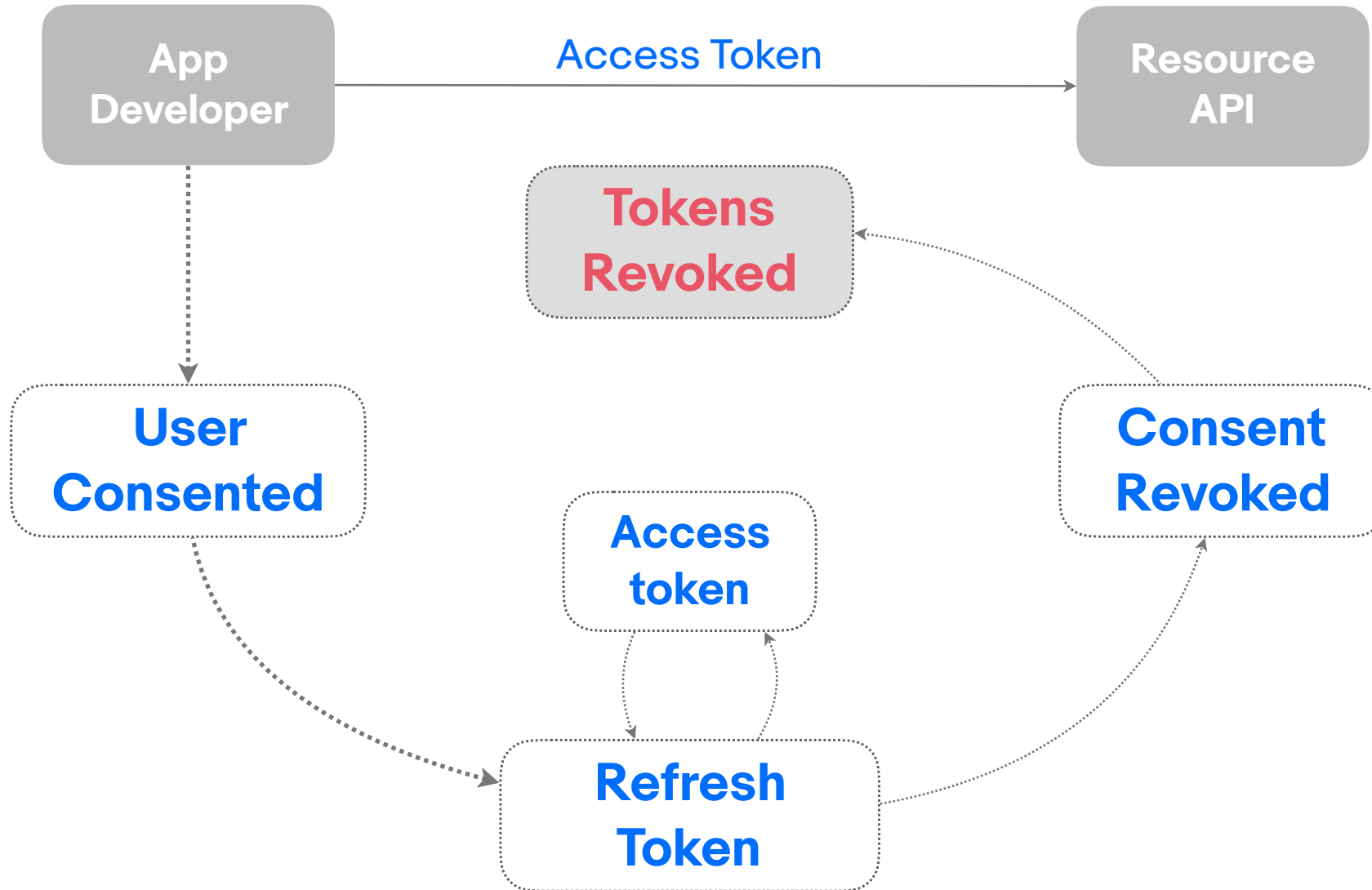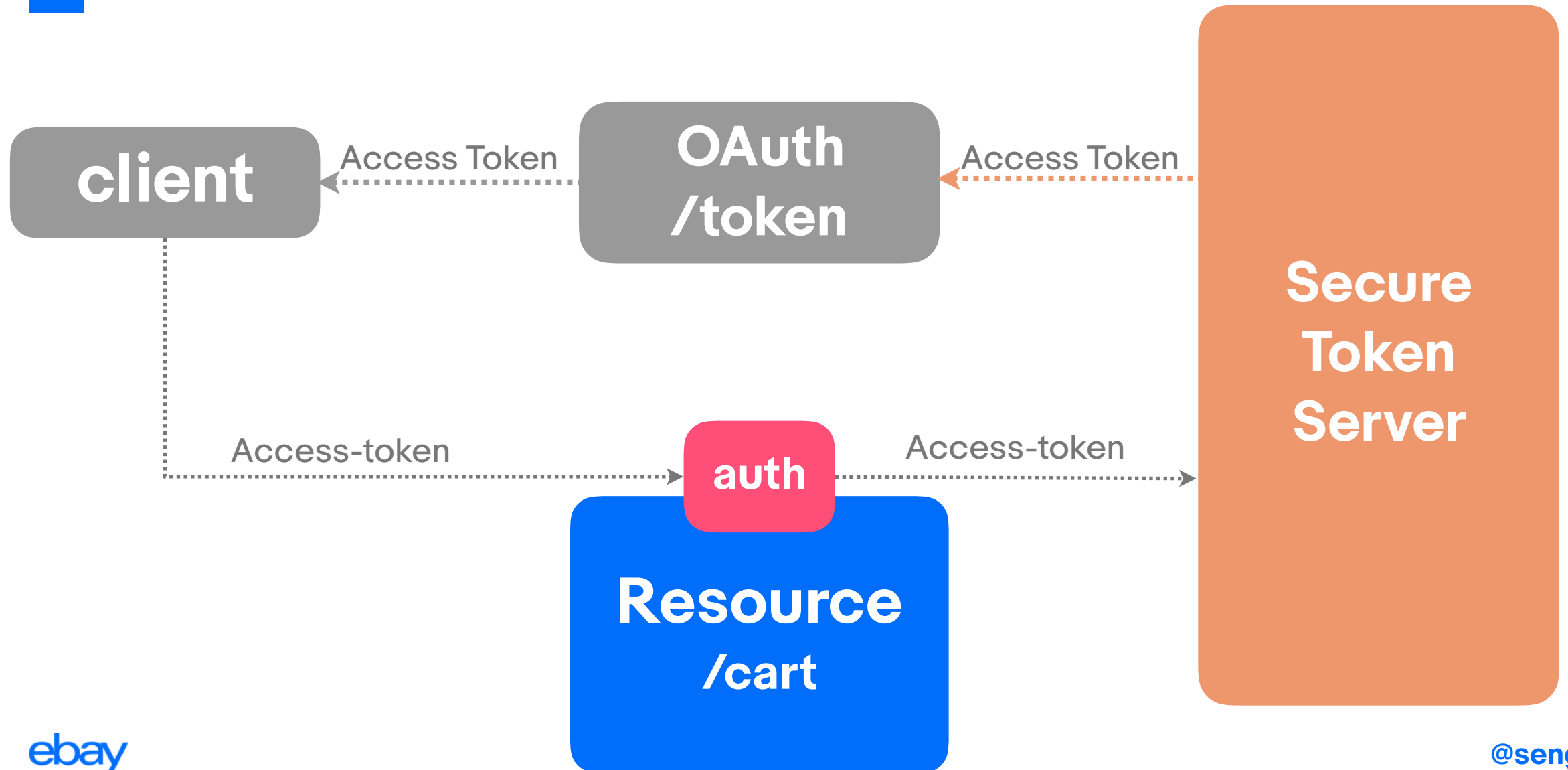
LifeCycle - Purpose

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**

Structure

ebay

AgAAAA**AQAAAA**aAAAAA**E6+EWg**nY+sHZ2PrBmdj6wVnY+sEZ2PrA2dj6wMkIGkCJCGoA
2dj6x9nY+seQ+/5wK1dskM5/3EOEY7BDg7VHK/CmDimCvPbtJankHhzJUF8rU876Qzjs

google

ya29.GltiBRICgroWhf0XJ-
e4nYpzc9UG0Fn_Ghq06_yg3BDZ4EHM_X8rIirEnFUJVb9uawqW2tE9yqfT0KwcaEXLKp7VFpde5v

facebook

EAACEdEose0cBAJyrAOqlWCAPVobbylB7mZB7X3L0x5BLBosAAm2BDdUnhYKSp7VM9Tpyi8Ehr
AD6ZBYZBtymYC5ZBxNv1XrCBngEi0gEWLejezZb0gkArZBkJWcFiVjGcKYy44EY8ZD

Tokens edited for brewity

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
```

ebay

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
```

ebay

https://jwt.io/

@sengopal

Structure - What goes in the claim?
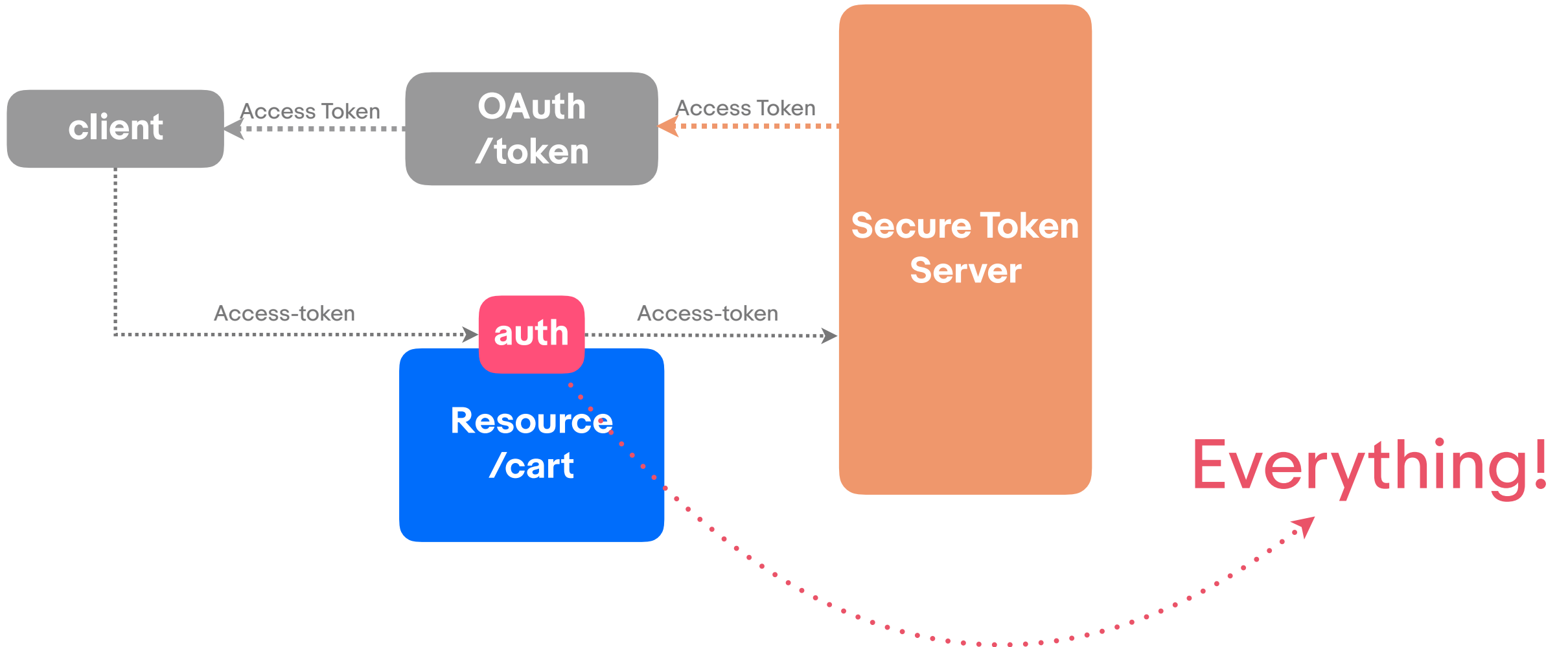
Everything!

Structure - Why everything?
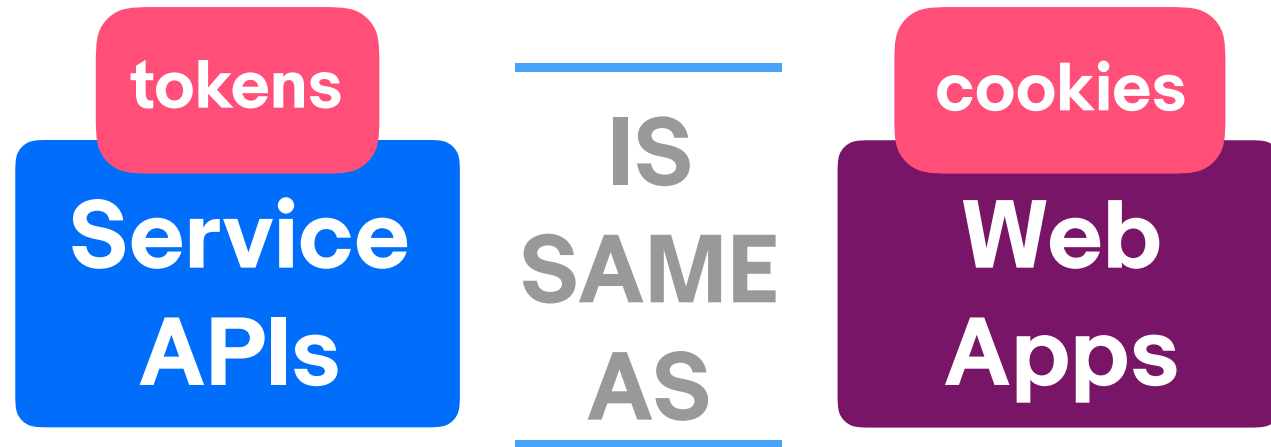
tokens
Service APIs

IS
SAME
AS

cookies
Web Apps

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

ebay

Structure - Versioning

We add new attributes everyday.

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

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

STEP 6

Capture every identifier possible and use versioning

Authentication Server - a time tested strategy

Life Cycle   Structure   Persistence

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
"
}
```

ebay

Security

By Value	By Reference
Integrity Verified	Integrity Verified
	Confidential
	Custom format *
	Persisted

ebay

@sengopal

Fitting them together

client — Access Token — OAuth /token — Access Token — async

AUDIT

Access-token — auth — Access-token — Secure Token Server

RDBMS

Resource /cart

App Metadata Server

ebay @sengopal
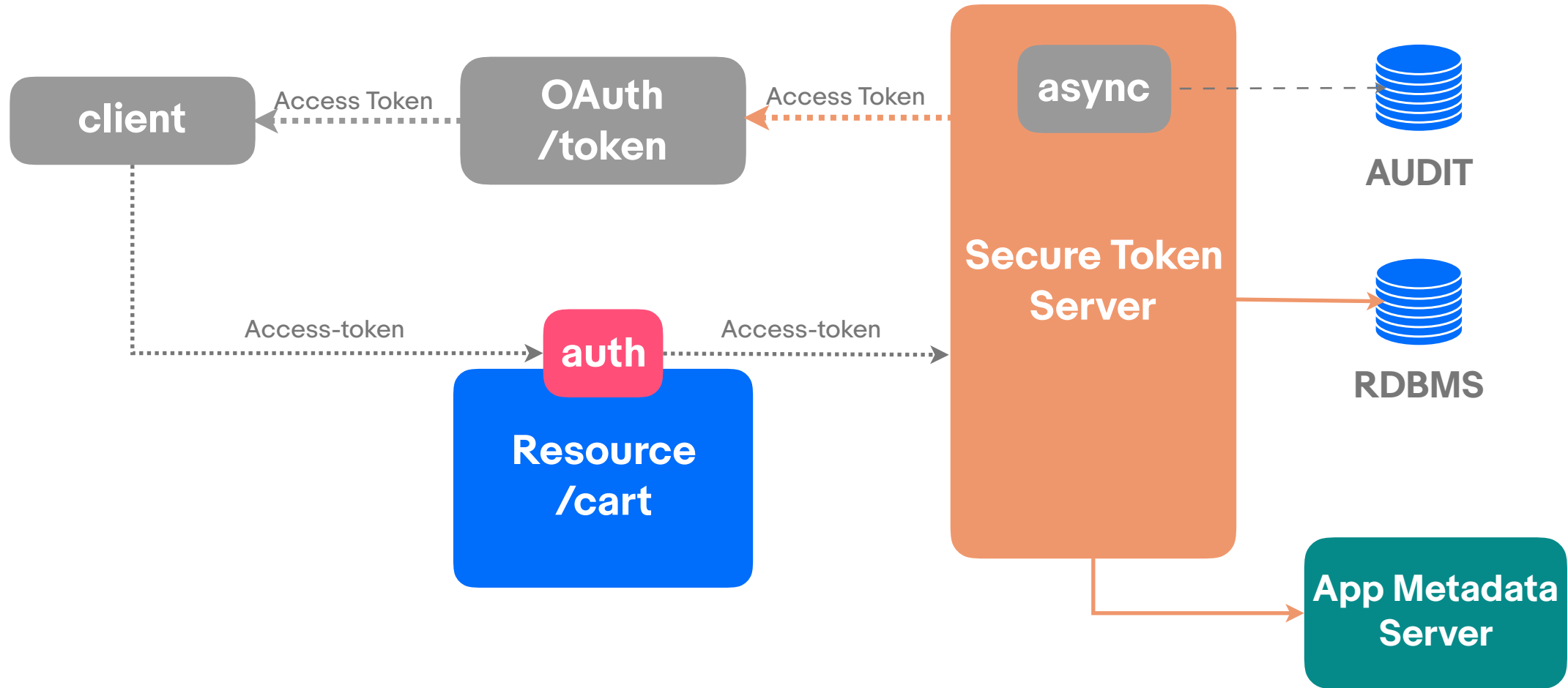
Persistence - Considerations

Atomic & Strong Consistency

Token Generation of new tokens

Token Revocation *

Persistence - Considerations

Eventually Consistent

User - token Auditing

Cache duplication

STEP 7

Identify transactional needs

Performance

"**Premature optimization is the root of all evil**"

- Donald Knuth

Identify Hot spots
Caching in Couchbase

Fitting them together

client

OAuth /token — Access Token → client

Access Token ← async ← OAuth /token

CACHE

Secure Token Server

async

Resource /cart — auth

Access-token (client → Resource /cart)

Access-token (Resource /cart → Secure Token Server)

AUDIT

RDBMS

App Metadata Server

ebay

@sengopal
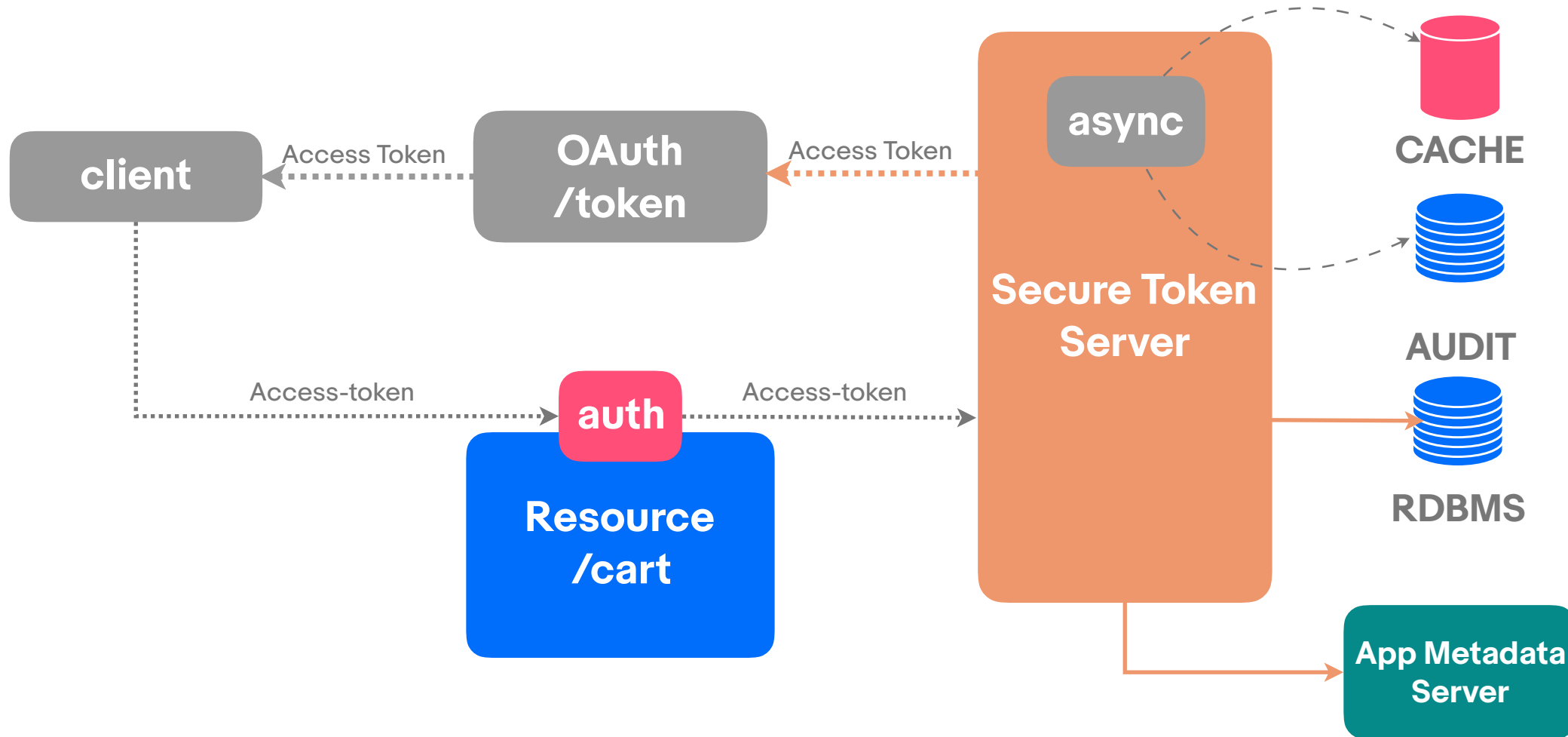
STEP 8

Use caching to get optimal performance

OWASP

Open Web Application Security Project

A2 – Broken Authentication and Session Management

A10 – Underprotected APIs

Reference

ebay

@sengopal

Fitting them together

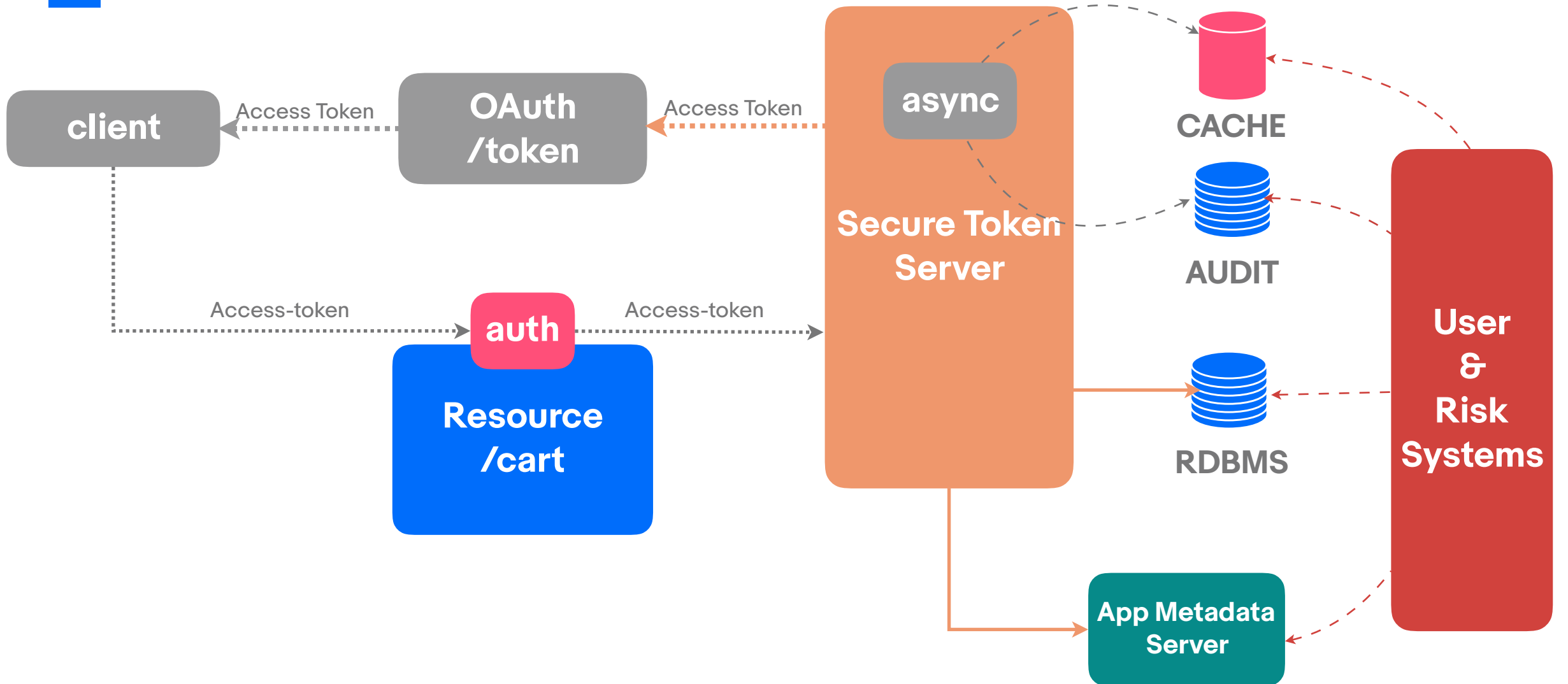

client ← Access Token ← **OAuth /token** ← Access Token ← **Secure Token Server** (async)

client ⋯ Access-token ⋯→ **auth** / **Resource /cart** ⋯ Access-token ⋯→ **Secure Token Server**

CACHE

AUDIT

RDBMS

App Metadata Server

User & Risk Systems

ebay

@sengopal

STEP 9

Audit all access patterns

Managing the whole show

Application Lifecycle

Token lifecycle

Cryptography artifacts rotation

Authorizations registry

....

ebay

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

Use caching

Audit all access patterns

Automate Everything

ebay

@sengopal

Thank You!

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

Tweets **@sengopal**

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