User & Device Identity
For Microservices
@ Netflix Scale

Satyajit Thadeshwar
QCon San Francisco 2019
Logged out?
#$%&!
Logged out?
#$%&!
User & Device Identity for Microservices @ Netflix Scale
Satyajit Thadeshwar
Satyajit Thadeshwar
Product Edge Access Systems
sthadeshwar@netflix.com
Complicated
User & Device Identity for Microservices @ Netflix Scale

Satyajit Thadeswar
User & Device Identity for Microservices @ Netflix Scale
Satyajit Thadeshwar
Netflix **subscribers** and the **devices** that they use
Where we were

What we did

Wins
Where we were
User Login

Email: jsmith@gmail.com
Password: *******
ESN: LGTV20165-193456G568
User Login

Email: jsmith@gmail.com
Password: ********
ESN: LGTV20165-193456G568
User Login

User Login Diagram:

- **EDGE**
  - Email: jsmith@gmail.com
  - Password: *******
  - ESN: LGTV20165-193456G568

- **ZUUL**
  - /login

- **ORIGIN**
  - API

- **MID-TIER SERVICES**
  - Netflix Microservices
  - AUTH SERVICE
  - SUCCESS

- **Database**
  - Cassandra
User Login

Email: jsmith@gmail.com
Password: *******
ESN: LGTV20165-193456G568

/customerId: 10192378
ESN: LGTV20165-193456G568
Expires: In 8 hours
User Login

Email: jsmith@gmail.com
Password: ********
ESN: LGTV20165-193456G568

/Set-Cookie

customerID: 10192378
ESN: LGTV20165-193456G568
Expires: In 8 hours
Authenticate Request

EDGE

/browse

Zuul
Authenticate Request

EDGE

|/browse |

ZUUL

|/browse |

ORIGIN

API
Authenticate Request

EDGE

ZUUL

API

ORIGIN

KEY MANAGEMENT SERVICE

Authenticate Request

ZUUL

API

EDGE

ORIGIN

KEY MANAGEMENT SERVICE

Authenticate Request

ZUUL

API

EDGE

ORIGIN

KEY MANAGEMENT SERVICE

Authenticate Request

ZUUL

API

EDGE

ORIGIN

KEY MANAGEMENT SERVICE
Authenticate Request

EDGE

ZUUL

ORIGIN

KEY MANAGEMENT SERVICE

SUCCESS

customerId: 10192378
ESN: LGTV20165-193456G568

MID-TIER SERVICES

Netflix Microservices
Authenticate Request
More than one service consuming cookies
User & Device Identity for Microservices @ Netflix Scale
Satyajit Thadeshwar
User & Device Identity for Microservices @ Netflix Scale
Satyajit Thadeshwar
At massive scale
Netflix

158M+ subscribers
Netflix

158M+ subscribers
1B+ devices
2M peak RPS
Authenticate Request / Extract Identity

= 2 million Requests Per Second
More than one token type
Cookies 🍪

- Signup

---

**User & Device Identity for Microservices @ Netflix Scale**

Satyajit Thadeshwar
Cookies 🍪

- Signup
- Login
Cookies 🍪

- Signup
- Login
- Discovery
MSL Tokens

- Device authentication
- Encryption

Message Security Layer (MSL)
MSL Tokens

- License
- Playback
CTicket

- Legacy devices
Partner Tokens

- JWS, JWE
- Non-member experiences
Cookies
- Signup
- Sign-in
- Discovery

MSL Tokens
- License
- Playback

CTicket
- Legacy devices

Partner Tokens
(JWS, JWE)
- Non-member experience

User & Device Identity for Microservices @ Netflix Scale
Satyajit Thadeshwar
Where we were

- Multiple services consuming auth tokens
- Multiple types of auth tokens
- Massive scale
- Inefficient, insecure & complicated
User & Device Identity for Microservices @ Netflix Scale
Satyajit Thadeshwar
What we did
Moved authentication to the edge
User & Device Identity for Microservices @ Netflix Scale

Satyajit Thadeshwar

Edge

Zuul

EAS

RENEWAL / DEVICE AUTH / KEY EXCHANGE

95%

VALID AND NOT EXPIRED

5%

_COOKIE SERVICE

_MSL SERVICE

_PARTNER SERVICE
User & Device Identity for Microservices @ Netflix Scale
Satyajit Thadeshwar
VALID BUT EXPIRED

Zuul

EAS

RENEWAL CALL FAILED

Cookie Service
User & Device Identity for Microservices @ Netflix Scale
Satyajit Thadeshwar

EDGE

VALID BUT EXPIRED

Zuul

RENEWAL CALL RESCHEDULED

EAS

RESOLVED IDENTITY

Cookie Service
User & Device Identity for Microservices @ Netflix Scale
Satyajit Thadeshwar
User & Device Identity for Microservices @ Netflix Scale
Satyajit Thadeshwar

EDGE

ZUUL

EAS

ORIGINS

API

DEVICE AUTH SERVICE

LEGACY API

SIGNUP FLOW SERVICE

SUBSCRIBER

AUTH SERVICE

NETFLIX MICROSERVICES

MID-TIER SERVICES

DISCOVERY API

LOLOMO / SEARCH

DRM

PLAYBACK API

OTHER SERVICES

NODEJS SERVICES

COOKIE SERVICE

MSL SERVICE

PARTNER SERVICE

EDGE AUTHENTICATION SERVICES
User & Device Identity for Microservices @ Netflix Scale
Satyajit Thadeshwar

Edge Authentication Services:
- **Zuul**
- **EAS**
- **Cookie Service**
- **MSL Service**
- **Partner Service**

Origins:
- **API**
- **Device Auth Service**
- **Legacy API**

Mid-Tier Services:
- **Netflix Microservices**
  - **Signup Flow Service**
  - **Subscriber Auth Service**
  - **Discovery API**
  - **Lolomo/Search**
  - **DRM**
  - **Playback API**
  - **Other Services**

Edge Services:
- **NodeJS Services**
- **Cookie Service**
- **MSL Service**
- **Partner Service**
- **Legacy API**
- **Device Auth Service**
- **API**

Platform Support:
- iOS
- Android
- Windows
- Roku
- 4K

Netflix Infrastructure:
- Casandra
- EVCache
- Hive

Additional Services:
- **Cookie Service**
- **MSL Service**
- **Partner Service**

**Note:** The diagram shows a high-level overview of Netflix's microservices architecture, focusing on user and device identity management and authentication services.
Passport

- Identity structure created at the edge for each request
Passport

- Identity structure created at the edge for each request
- Contains user & device identity
Passport

- Identity structure created at the edge for each request
- Contains user & device identity
- Internal to Netflix ecosystem
Passport

- Identity structure created at the edge for each request
- Contains user & device identity
- Internal to Netflix ecosystem
- Integrity protected by HMAC
Passport

- Identity structure created at the edge for each request
- Contains user & device identity
- Internal to Netflix ecosystem
- Integrity protected by HMAC
- Protobuf format
message Passport {
    Header header = 1;
    UserInfo user_info = 2;
    DeviceInfo device_info = 3;
    Integrity user_integrity = 4;
    Integrity device_integrity = 5;
}
message Passport {
  Header header = 1;
  UserInfo user_info = 2;
  DeviceInfo device_info = 3;
  Integrity user_integrity = 4;
  Integrity device_integrity = 5;
}

message Header {
  string originator = 1;
}
message Passport {
    Header header = 1;
    UserInfo user_info = 2;
    DeviceInfo device_info = 3;
    Integrity user_integrity = 4;
    Integrity device_integrity = 5;
}
message Passport {
    Header header = 1;
    UserInfo user_info = 2;
    DeviceInfo device_info = 3;
    Integrity user_integrity = 4;
    Integrity device_integrity = 5;
}

message UserInfo {
    Source source = 1;
    AuthenticationLevel auth_level = 2;
    { Int64Wrapper customer_id = 3;
    { Int64Wrapper account_owner_id = 4; }
    repeated UserAction actions = ;
}
message Passport {
  Header header = 1;
  UserInfo user_info = 2;
  DeviceInfo device_info = 3;
  Integrity user_integrity = 4;
  Integrity device_integrity = 5;
}

message DeviceInfo {
  Source source = 1;
  AuthenticationLevel auth_level = 2;
  {StringValue esn = 3;
    {Int32Value device_type = 4;
      repeated DeviceAction actions = 5;
    }
  }
}
```java
message UserInfo {
  Source source = 1;
  AuthenticationLevel auth_level = 2;
}

message DeviceInfo {
  Source source = 1;
  AuthenticationLevel auth_level = 2;
}
```
message UserInfo {
  Source source = 1;
  AuthenticationLevel auth_level = 2;
}

message DeviceInfo {
  Source source = 1;
  AuthenticationLevel auth_level = 2;
}

enum Source {
  COOKIE = 1;
  MSL = 2;
  PARTNER_TOKEN = 3;
  CTICKET = 4;
}
Passport

message UserInfo {
    Source source = 1;
    AuthenticationLevel auth_level = 2;
}

messageDeviceInfo {
    Source source = 1;
    AuthenticationLevel auth_level = 2;
}

def AuthenticationLevel {
    LOW = 1; // untrusted transport
    HIGH = 2; // secure tokens over TLS
    HIGHEST = 3; // MSL or user credentials
}
message Passport {
  Header header = 1;
  UserInfo user_info = 2;
  DeviceInfo device_info = 3;
  Integrity user_integrity = 4;
  Integrity device_integrity = 5;
}

message Integrity {
  string key_name = 1;
  bytes hmac = 2;
}
Passport Introspector

- Wrapper over passport binary data
Passport Introspector

- Wrapper over passport binary data

```java
public interface PassportIntrospector {
    Long getCustomerId();
    Long getAccountOwnerId();
    String getEsn();
    String getPassportAsString();
    ...
}
```
Passport Introspector

- Wrapper over passport binary data

- Consumers create `passportIntrospector` from binary passport data

```java
public interface PassportIntrospector {
  Long getCustomerId();
  Long getAccountOwnerId();
  String getEsn();
  String getPassportAsString();
  ...
}

factory.createIntrospector(passport);
```
Self-service tool for **teams** to decrypt passport
Passport Actions

```protobuf
message UserInfo {
  repeated UserAction actions = 6;
  ...
}

message DeviceInfo {
  repeated DeviceAction actions = 5;
  ...
}
```
Passport Actions

```protobuf
message UserInfo {
  repeated UserAction actions = 6;
  ...
}

message DeviceInfo {
  repeated DeviceAction actions = 5;
  ...
}
```

- **Explicit signal** sent by the downstream services, when an **update** to user or device identity has been performed
Passport Actions

message UserInfo {
  repeated UserAction actions = 6;
  ...
}

message DeviceInfo {
  repeated DeviceAction actions = 5;
  ...
}

- Explicit signal sent by the downstream services, when an update to user or device identity has been performed

- This "signal" is used by EAS to either create or update the corresponding type of token
Passport Action
Passport Action: User Login
Passport Action: User Login

Email: jsmith@gmail.com
Password: ********
ESN: LGTV20165-193456G568
Passport Action: User Login

Email: jsmith@gmail.com
Password: ********
ESN: LGTV20165-193456G568

Device Bound

Zuul

Origin

/login
Passport Action: User Login

Email: jsmith@gmail.com
Password: *******
ESN: LGTV20165-193456G568
Passport Action: User Login

Email: jsmith@gmail.com
Password: ********
ESN: LGTV20165-193456G568
Passport Action: User Login

Email: jsmith@gmail.com
Password: ******
ESN: LGTV20165-193456G568
Passport Action: Profile Switch
Passport Action: Profile Switch

- Each profile has its own identity
Passport Action: Profile Switch

- Each profile has its own identity
- Switched profile tokens sent back to the device
Passport Actions

Separation Of Concerns

Increased Visibility
What we did

- Moved authentication to the edge
- Streamlined the identity resolution and mutation path
- Making consumption of user & device identity
- Efficient, secure & simple
User & Device Identity for Microservices @ Netflix Scale
Satyajit Thadeshwar
Token Agnostic Identity

Downstream systems don't have to worry about authentication concerns
Simplified Authorization

Downstream services use authentication level for authorization decisions
Simplified Authorization

Before:

```java
long customerId = 2123125603L;
String ESN = "NFXBOX-235F...";
```
Extensible Identity Model

New attributes about user or device can be added
Local cache for up to date subscriber data

message UserInfo {
  BytesValue subscriber_account
  ...
}

Placeholder for local cache of subscriber data
Offloaded & Fine Tuned

Offloaded token processing which resulted into significant gains for
- CPU
- Request Latency
- GC
- Cluster Footprint
Offloaded & Fine Tuned

Offloaded token processing which resulted into significant gains for
- CPU
- Request Latency
- GC
- Cluster Footprint

We were able to fine tune EAS systems based on the token processing profile
Offloaded & Fine Tuned

- 30% reduction in CPU cost per request
- 40% reduction in load average

CPU to RPS ratio for API instance
Offloaded & Fine Tuned

- 30% reduction in average latency
- 99th percentile latency dropping by 20%
Offloaded & Fine Tuned

- Significant reduction in GC pressure and GC pause times

Stop the world GC for API cluster
Increased visibility into identities flowing in and out of Netflix ecosystem

...and into the identity mutations happening in a request
Greatly increased developer velocity for authentication related changes
Team focused on security

Separation of concerns among the teams
Key Takeaways

- Token agnostic identity model
- Simplified authorization
- Extensible identity model
- Offloaded all the token processing from many systems
- Fine tuned individual microservices to suit the token processing profile
- Increased visibility into identities flowing and corresponding mutations
- Increased developer velocity for authentication & identity related changes
- Team focused on security
Thank You.

Satyajit Thadeshwar
sthadeshwar@netflix.com
https://www.linkedin.com/in/satyajit-thadeshwar