AARHUS UNIVERSITET

# Microservices and DevOps

## DevOps and Container Technology

### Security 101 - Authorization

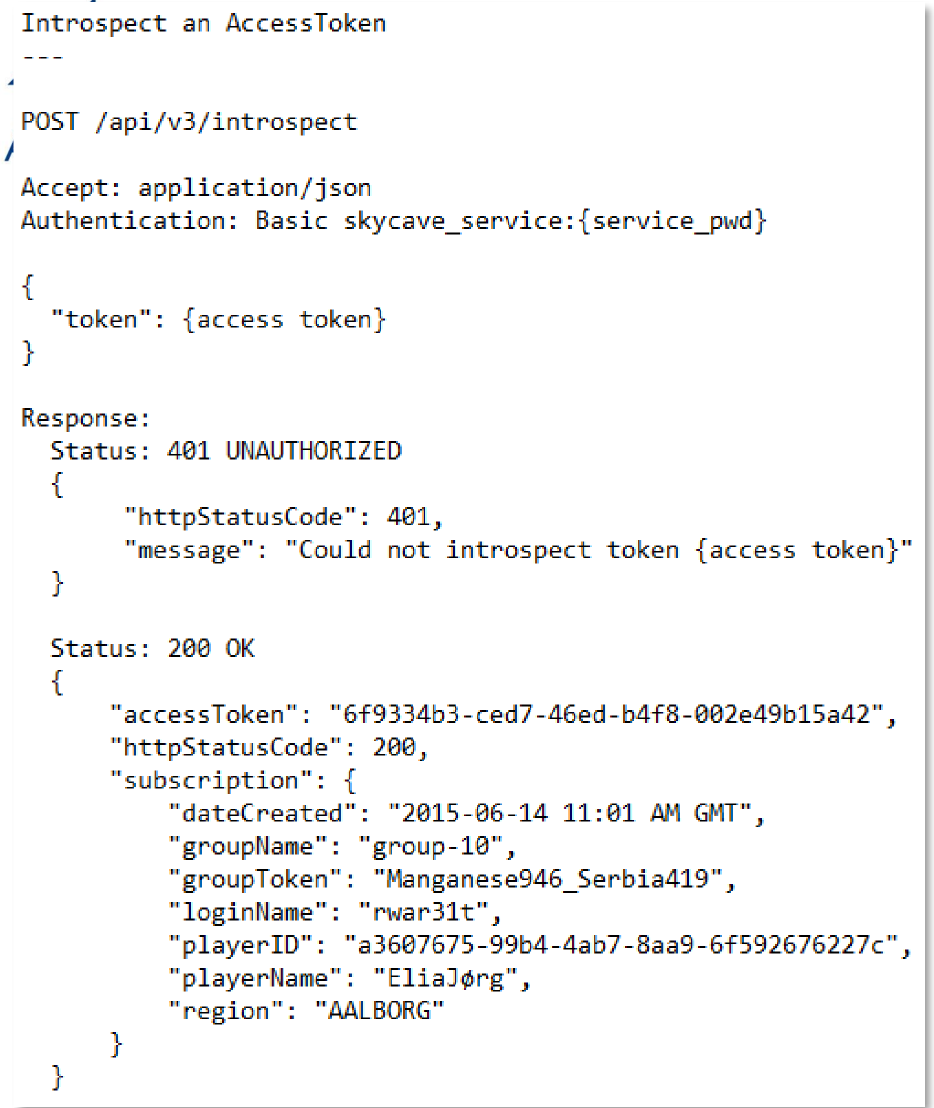
Henrik Bærbak Christensen

# /introspect

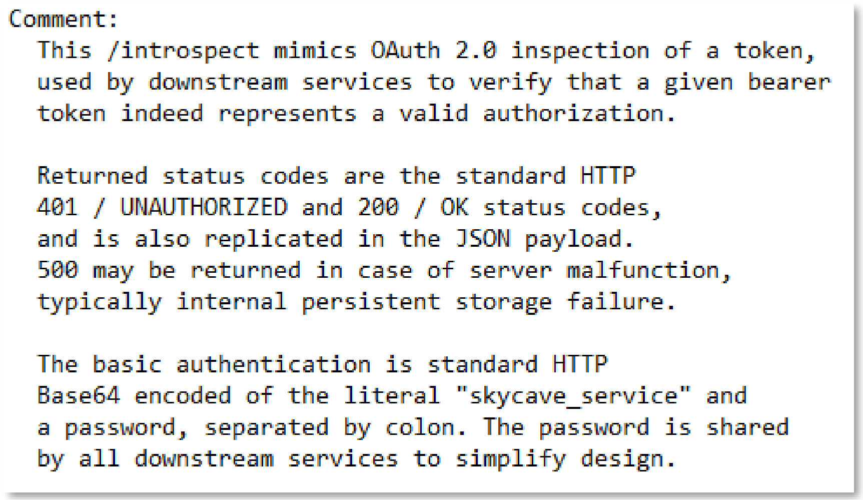
- In Mandatory 2.1 you will be *strangling the daemon…*
  - Migrate from a monolith to a set of microservices
- Example
  - Daemon does *not handle messages in the room*
    - It asks a MessageService to do that on behalf of it

- That is, *A down stream service that needs to validate token*
  - *Because it is only a authorized player that may interact with the message service, right?*

# API



```
Introspect an AccessToken
---

POST /api/v3/introspect

Accept: application/json
Authentication: Basic skycave_service:{service_pwd}

{
  "token": {access token}
}

Response:
  Status: 401 UNAUTHORIZED
  {
       "httpStatusCode": 401,
       "message": "Could not introspect token {access token}"
  }

  Status: 200 OK
  {
      "accessToken": "6f9334b3-ced7-46ed-b4f8-002e49b15a42",
      "httpStatusCode": 200,
      "subscription": {
          "dateCreated": "2015-06-14 11:01 AM GMT",
          "groupName": "group-10",
          "groupToken": "Manganese946_Serbia419",
          "loginName": "rwar31t",
          "playerID": "a3607675-99b4-4ab7-8aa9-6f592676227c",
          "playerName": "EliaJørg",
          "region": "AALBORG"
      }
  }
```



```
Comment:
  This /introspect mimics OAuth 2.0 inspection of a token,
  used by downstream services to verify that a given bearer
  token indeed represents a valid authorization.

  Returned status codes are the standard HTTP
  401 / UNAUTHORIZED and 200 / OK status codes,
  and is also replicated in the JSON payload.
  500 may be returned in case of server malfunction,
  typically internal persistent storage failure.

  The basic authentication is standard HTTP
  Base64 encoded of the literal "skycave_service" and
  a password, separated by colon. The password is shared
  by all downstream services to simplify design.
```

A single token for all services

skycave_service:~~56utxf8tq~~

56utxf8tqv

## Note of Caution

- It is rather tedious, so to lower implementation burden on you and on the consuming groups…

- Do not implement it or make it optional

  - If there is not Authorization header in REST payload, then ignore it and allow access!