HITACHI
Inspire the Next

Open Source Summit Japan 2019

# Implementing security and availability requirements for banking API system using Open Source Software (OSS)

Yoshiyuki Tabata
OSS Solution Center
Hitachi, Ltd.

© Hitachi, Ltd. 2019. All rights reserved.

# Self introduction

HITACHI
Inspire the Next

## Yoshiyuki Tabata :
## OSS Solution Center, Hitachi, Ltd. @ Yokohama, Japan

- Software engineer
- API Management & Identity Management
  - 3scale, Keycloak
  - Contributor of 3scale
    - Developed “Edge Limiting policy”, “Keycloak Role Check policy”

© Hitachi, Ltd. 2019. All rights reserved.

HITACHI
Inspire the Next

# Contents



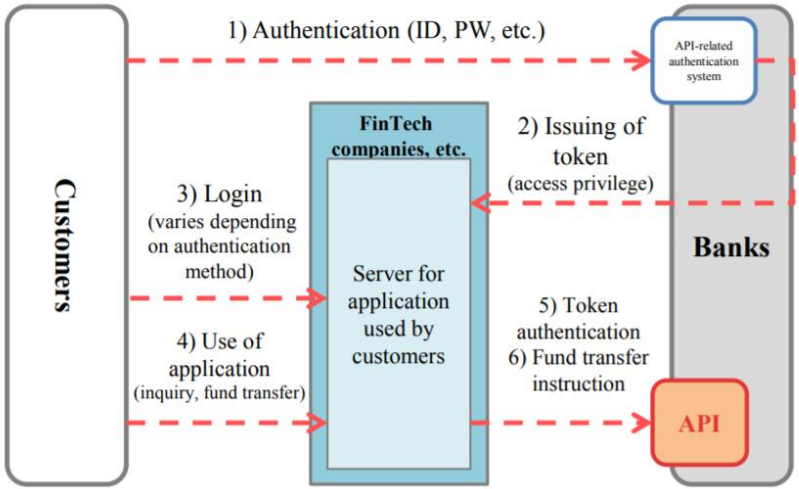
© Hitachi, Ltd. 2019. All rights reserved.

# Background: Banking API and its security in Japan

HITACHI
Inspire the Next

- The revised banking act was published in Jun 2017 to promote API.
  - Similar to PSD2 in EU
- 83% of banks (114 banks) answered they will open API by 2020/6(*).
  (*) Based on survey of Japanese Bankers Association as of Dec 2017
- Security : OAuth 2.0 is recognized as a key technology to secure API

Figure 1: Basic Framework of Open APIs (OAuth 2.0)

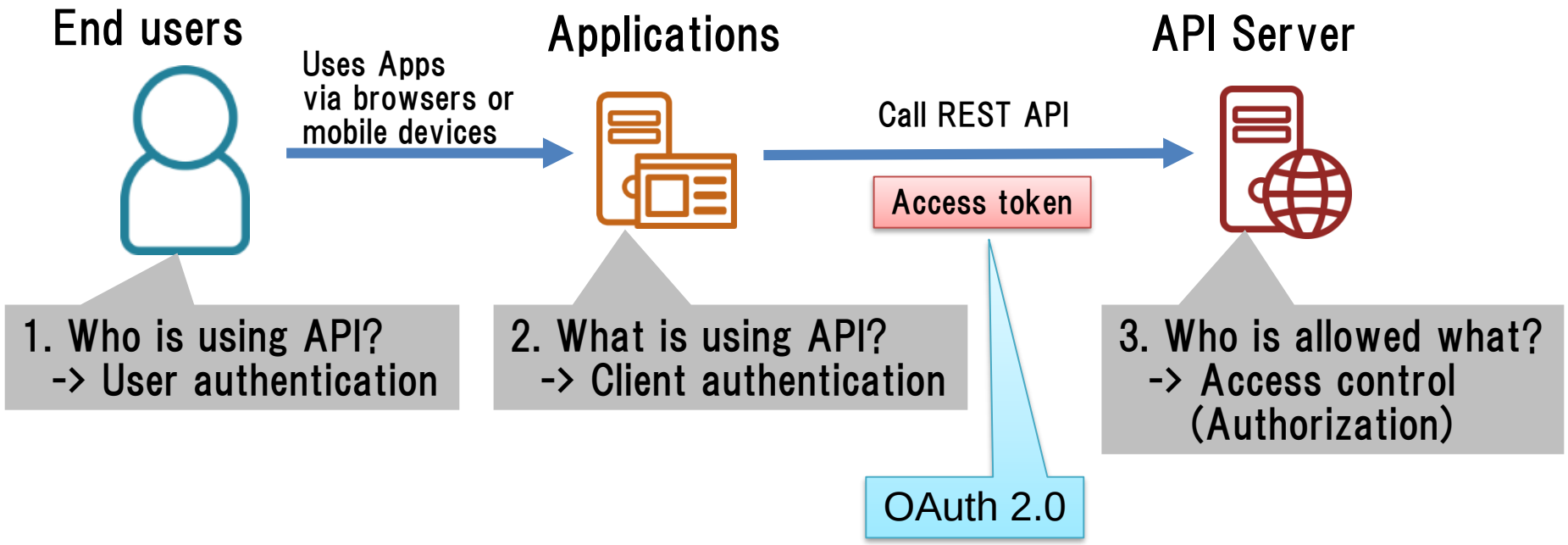



Quoted from Report about open API by the Japanese Bankers Association
https://www.zenginkyo.or.jp/fileadmin/res/news/news290713_3.pdf

© Hitachi, Ltd. 2019. All rights reserved.

# Usage of OAuth 2.0: Authentication, Authorization

HITACHI
Inspire the Next




* OAuth 2.0 (RFC 6749) only describes how tokens are issued.
We have to use other standards or create something outside of standards.

© Hitachi, Ltd. 2019. All rights reserved.

# Requirements for Authentication/Authorization for banking API

HITACHI
Inspire the Next

| # | Category | Description |
|---|---|---|
| 1 | Authentication | • Can support various (customized) authentication in OAuth flow<br>• Compliance to OpenID Connect on top of OAuth |
| 2 | Access control | • Deny/Allow accesses based on claims in token<br>• Can be combined with rate limit to protect backend |
| 3 | Manage tokens | • Revoke tokens triggered by users, administrators<br>• Revoke tokens based on policy |
| 4 | Compliance to the latest standards | • Financial-grade API (FAPI) of OpenID Foundation |

© Hitachi, Ltd. 2019. All rights reserved.

# Background: Banking API and its availability

HITACHI
Inspire the Next

- Non-Functional Requirements 2018(*) was published in Apr 2018 to construct appropriate information systems, and enable stable provision of services.
  (*) Reported by Information-Technology Promotion Agency, Japan
- Information systems are categorized into 5 levels according to characteristics.
- In our experience, almost all banking API systems belong to over Level 3.

Banking API

| Level | 5 | 4 | 3 | 2 |
|---|---|---|---|---|
| Operating Rate | > 99.999% | > 99.99% | > 99.9% | > 99% |
| Total recovery time per year (MTTR) | < 5.26 min | < 52.6 min | < 8.76 h | < 87.6 h |
| Recovery time per failure | < 1 min | < 10 min | < 1 h | < 2 h |

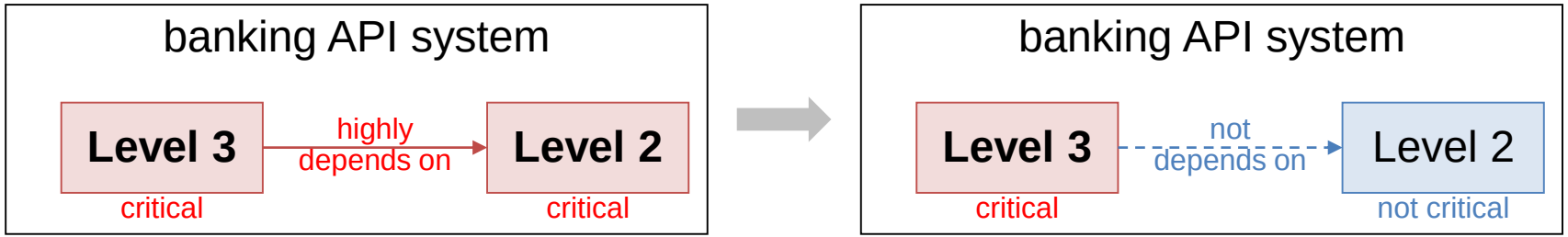
System Infrastructure Non-Functional Requirements Related Grade Table

© Hitachi, Ltd. 2019. All rights reserved.

# Achieve Level 3

How to minimize *MTTR ( < 8h )* and *the recovery time per failure ( < 1 h )*.

- Generally, to construct HA configuration and failover the system when a failure has occurred.
- To configure the system to be recovered automatically.
  -> **Fault Tolerance**
  * This takes a high cost for preparing more resources than usual.
- To reduce dependencies of each component.
  -> **Fail Soft / Fault Avoidance**




© Hitachi, Ltd. 2019. All rights reserved.

# Requirements for Availability for banking API

HITACHI
Inspire the Next

| # | Category | Description |
|---|---|---|
| 1 | Fault Tolerance | • HA configuration<br>• Can be recovered automatically |
| 2 | Fail Soft/<br>Fault Avoidance | • Reduce dependencies of each component |

© Hitachi, Ltd. 2019. All rights reserved.

HITACHI
Inspire the Next

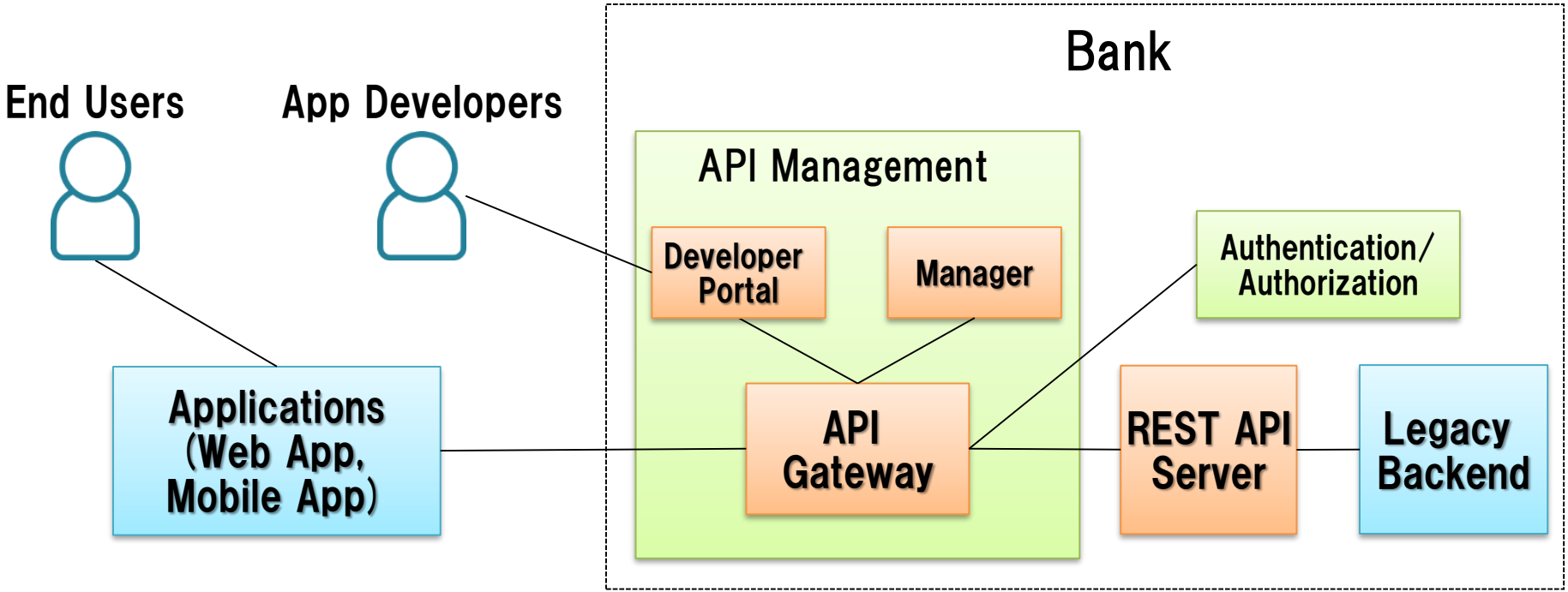
# Contents

1. Introduction: background and requirements

2. Usage of OSS to meet requirements
   - Which OSS should be used?
   - Security requirements
   - Availability requirements

© Hitachi, Ltd. 2019. All rights reserved.

# Open API system




- API Management product is usually used for common functions to open APIs
  - Rate limit, dev portal, analytics etc.
- It is desirable authentication/authorization are integrated into API management

© Hitachi, Ltd. 2019. All rights reserved.

# Open Source Software (OSS) for open API

HITACHI
Inspire the Next

| | OSS | |
|---|---|---|
| API Management | 3scale | WSO2 |
| | Kong | tyk |
| Authentication/ Authorization | Keycloak | OpenAM |
| | Gluu | |

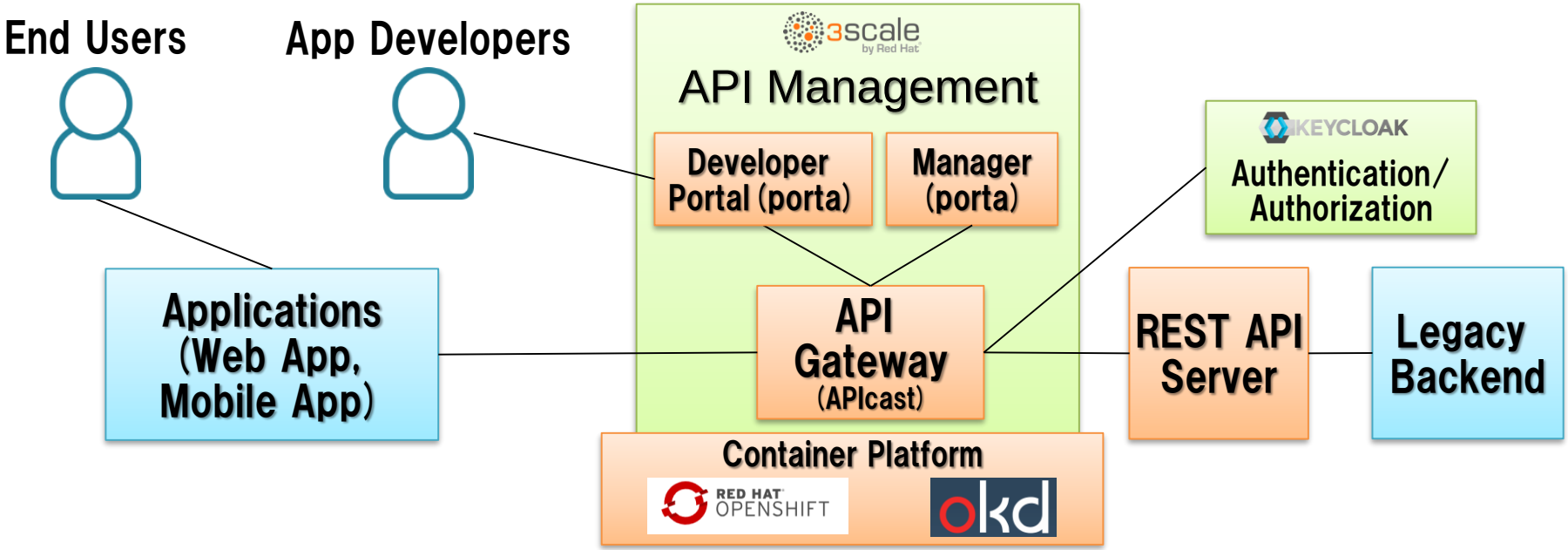
- There are various OSSs
- We chose “3scale” and “Keycloak”
  - Completeness of feature
  - Activity and future of community

© Hitachi, Ltd. 2019. All rights reserved.

# What is 3scale

HITACHI
Inspire the Next

OSS for API Management, community is led by Red Hat: https://github.com/3scale



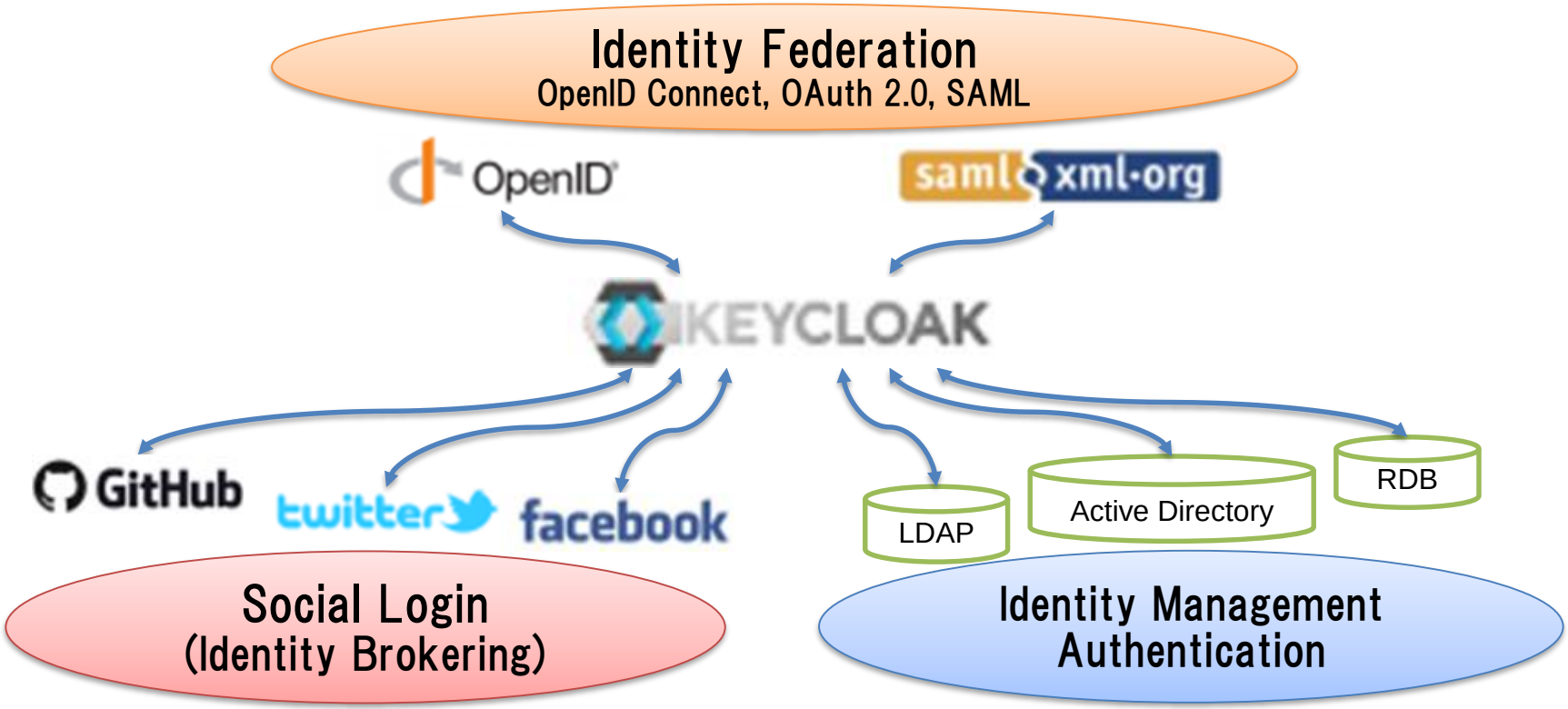

- Include full functions of API management (not only API GW)
- Cloud native : Works on OpenShift or okd
- OAuth2, OIDC in combination with Keycloak

© Hitachi, Ltd. 2019. All rights reserved.

# What is Keycloak

HITACHI
Inspire the Next

**OSS for Identity Management, community is led by Red Hat: https://www.keycloak.org**




© Hitachi, Ltd. 2019. All rights reserved.

HITACHI
Inspire the Next

# Contents

1. Introduction: background and requirements

2. Usage of OSS to meet requirements
    - Which OSS should be used?
    - Security requirements
    - Availability requirements

© Hitachi, Ltd. 2019. All rights reserved.

# <Recap> Requirements for Authentication/Authorization for banking API

HITACHI
Inspire the Next

| # | Category | Description |
|---|---|---|
| 1 | Authentication | • Can support various (customized) authentication in OAuth flow<br>• Compliance to OpenID Connect on top of OAuth |
| 2 | Access control | • Deny/Allow accesses based on claims in token<br>• Can be combined with rate limit to protect backend |
| 3 | Manage tokens | • Revoke tokens triggered by users, administrators<br>• Revoke tokens based on policy |
| 4 | Compliance to the latest standards | • Financial-grade API (FAPI) of OpenID Foundation |

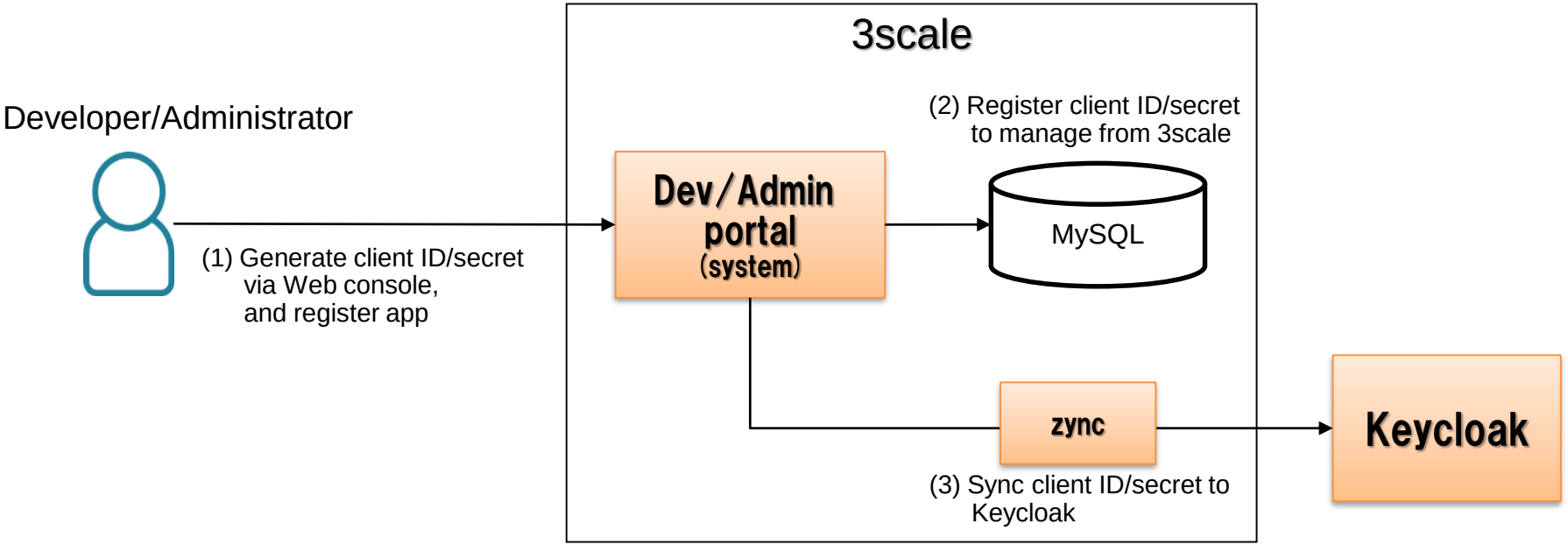
Implemented these requirements using 3scale + Keycloak, collaborating with OSS community

© Hitachi, Ltd. 2019. All rights reserved.

# Authentication : Registering Apps

HITACHI
Inspire the Next

## Authentication within OAuth/OIDC flow works well, basically




** OAuth 2.0 Dynamic Client Registration Protocol (RFC 7591)*

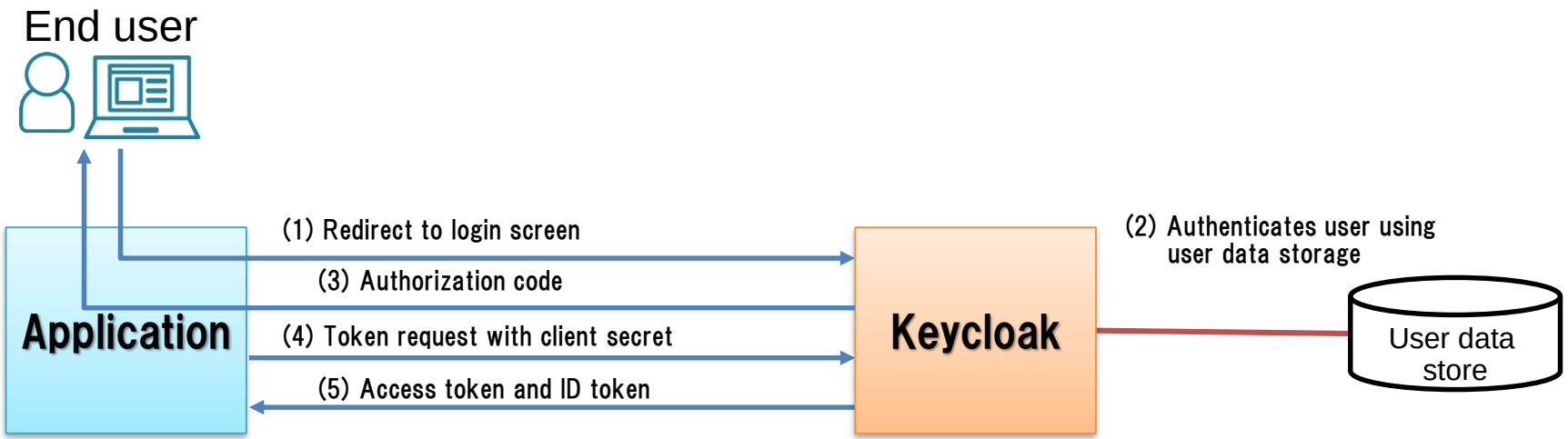
© Hitachi, Ltd. 2019. All rights reserved.

# Authentication : Authentication / Issue token

HITACHI
Inspire the Next

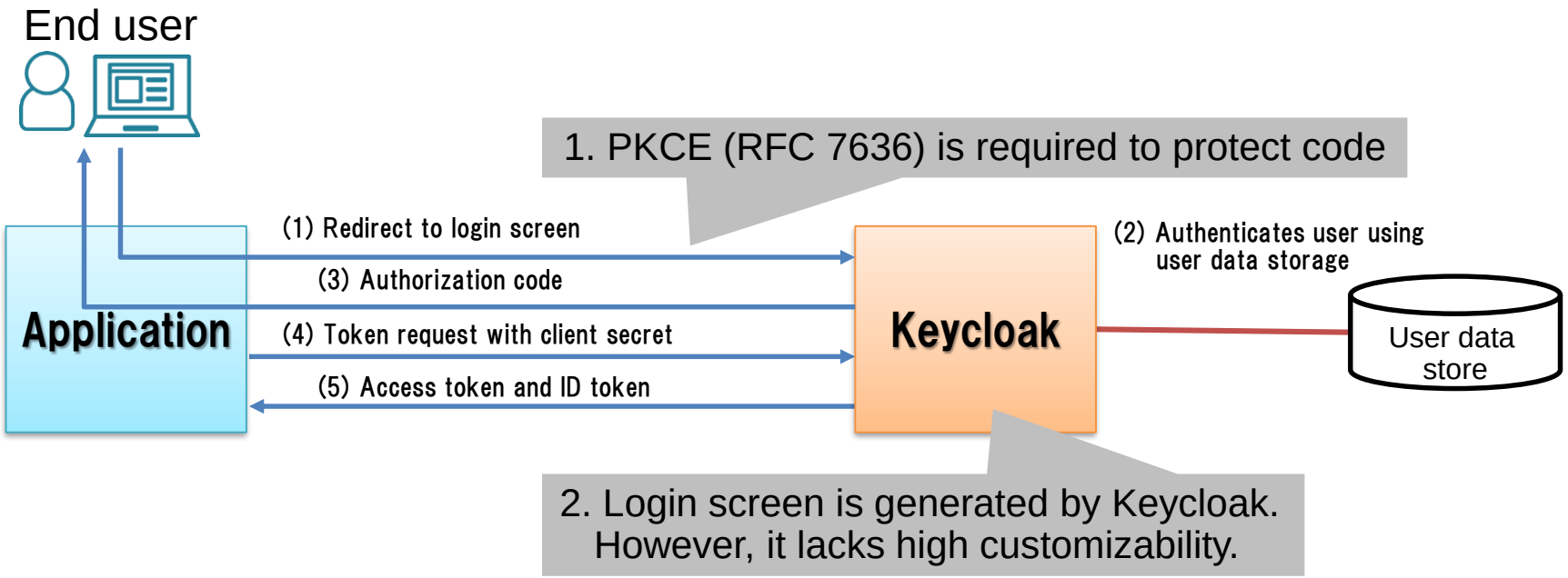
Authentication within OAuth/OIDC flow works well, basically

e.g.) Authorization code grant




© Hitachi, Ltd. 2019. All rights reserved.

# Authentication : Issues




© Hitachi, Ltd. 2019. All rights reserved.

# PKCE support for Keycloak

HITACHI
Inspire the Next

- Keycloak did not support PKCE..
  -> We submitted PR and merged.
  https://github.com/keycloak/keycloak/pull/3831

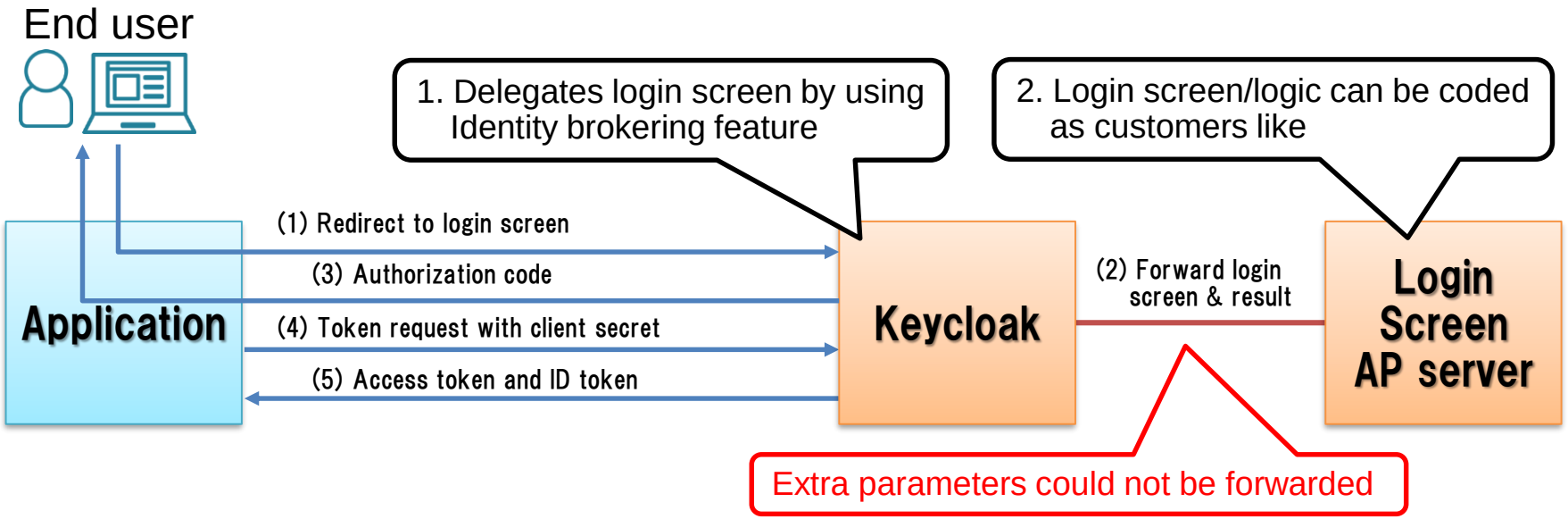
- From Keycloak 3.1.0, PKCE was supported.
  - Enabled by default (no switch)
  - Only when PKCE is requested from a client, it works
  - Included in OIDC server metadata from 4.0.0

© Hitachi, Ltd. 2019. All rights reserved.

# Highly customized login screen

HITACHI
Inspire the Next

Besides the template, login screen can be generated by delegated server



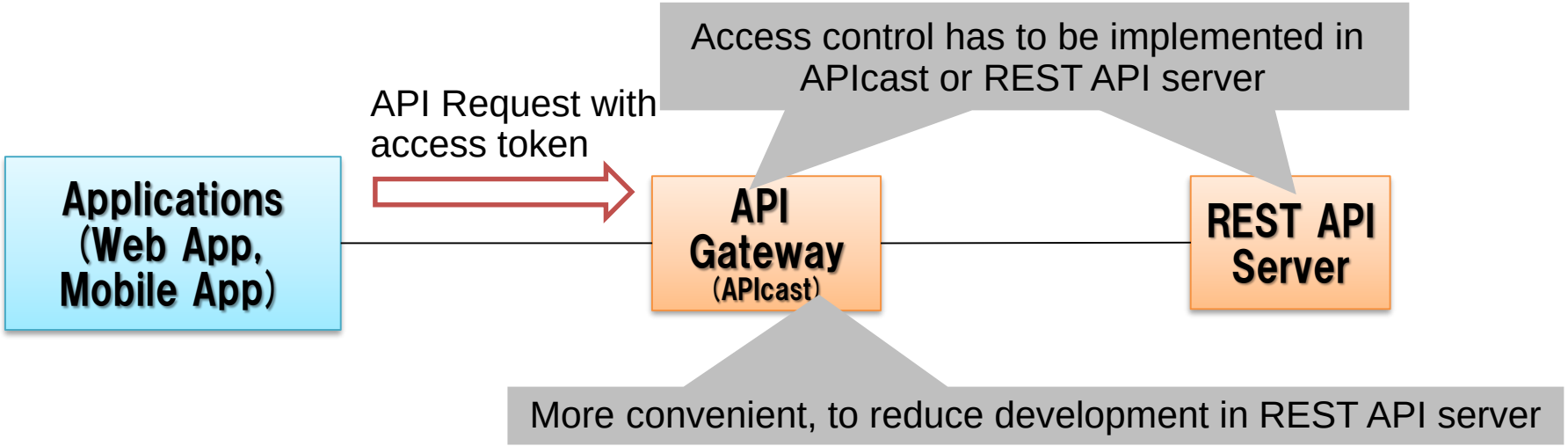

We submitted a patch to enable forward parameters from Keycloak.
https://github.com/keycloak/keycloak/pull/5163

© Hitachi, Ltd. 2019. All rights reserved.

# Access Control

HITACHI
Inspire the Next

Keycloak only issues tokens. Access control is out of scope.




*APIcast did not support access control using tokens* -> We submitted PRs.

© Hitachi, Ltd. 2019. All rights reserved.

# How to access control using tokens

HITACHI
Inspire the Next

- The format of access token is not standardized neither RFC nor OIDC.
  -> It depends on implementation.

- In Keycloak, the format is similar to ID token of OIDC (JWT, claims).

-> We targeted the Keycloak access token, and developed 2 policies(*).

(*) plugin to extend functions of APIcast

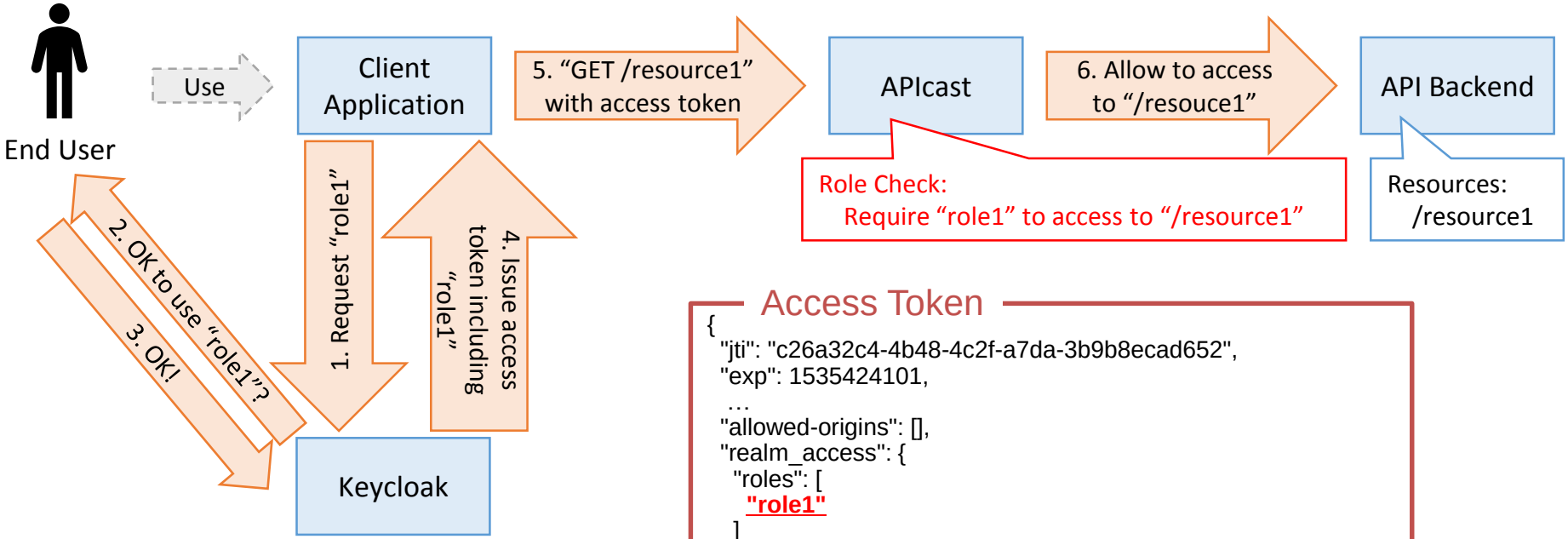

```
{
  "jti": "c26a32c4-4b48-4c2f-a7da-3b9b8ecad652",
  "exp": 1535424101,
  "nbf": 0,
  "iat": 1535423801,
  "iss": "http://localhost:8080/auth/realms/provider",
  "aud": "broker",
  "sub": "e4b11e2e-9136-409b-8720-57463c627c10",
  "typ": "Bearer",
  "azp": "broker",
  "auth_time": 0,
  "session_state": "ac1767e2-2e30-4d44-b6f3-b77935a7a0bc",
  "acr": "1",
  "allowed-origins": [],
  "realm_access": {
    "roles": [
      "read",
      "additional",
      "write"
    ]
  },
 "name": "Takashi Mogi",
  "preferred_username": "mogi",
  "given_name": "Takashi",
  "family_name": "Mogi",
  "email": "mogi@example.com"
}
```


© Hitachi, Ltd. 2019. All rights reserved.

# Keycloak Role Check policy

HITACHI
Inspire the Next

- Checks “role” claims of access token and URL.
- We submitted a patch and included from 3scale 2.3.
  https://github.com/3scale/apicast/pull/773




Access Token


```
{
  "jti": "c26a32c4-4b48-4c2f-a7da-3b9b8ecad652",
  "exp": 1535424101,
  …
  "allowed-origins": [],
  "realm_access": {
    "roles": [
      "role1"
    ]
  },
```

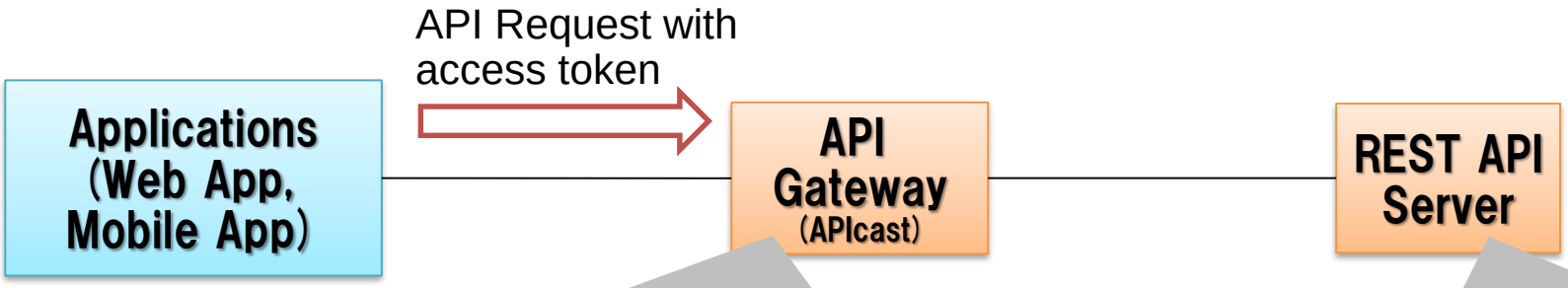

© Hitachi, Ltd. 2019. All rights reserved.

# Edge Limiting policy

HITACHI
Inspire the Next

Rate limiting: A kind of access control, to control the upper limit of traffics.
APIcast did not support **STRICT** rate limiting to protect backend.

-> We implemented patches and “Edge limiting policy” was included in 3scale 2.3.




Any values can be extracted as a key to control access
- header
- body parameter
- JWT claim
- etc.

https://github.com/3scale/apicast/pull/719

Protects backend by rate limit, types of limit:
- leaky bucket
- fixed window
- concurrent connections

https://github.com/3scale/apicast/pull/648

© Hitachi, Ltd. 2019. All rights reserved.

# Manage tokens

HITACHI
Inspire the Next

## Keycloak itself has features to revoke tokens

- Revoke tokens triggered by administrator
  -> Can be revoked from admin console

- Revoke tokens based on policy
  -> Timeout can be configured in admin console

- Revoke tokens triggered by users
  - Keycloak does not support OAuth 2.0 Token Revocation (RFC 7009)
  - Instead, logout endpoint(*) is used.

  (*) /auth/realms/<realm>/protocol/openid-connect/logout
  Related access tokens, ID tokens, refresh tokens are revoked.

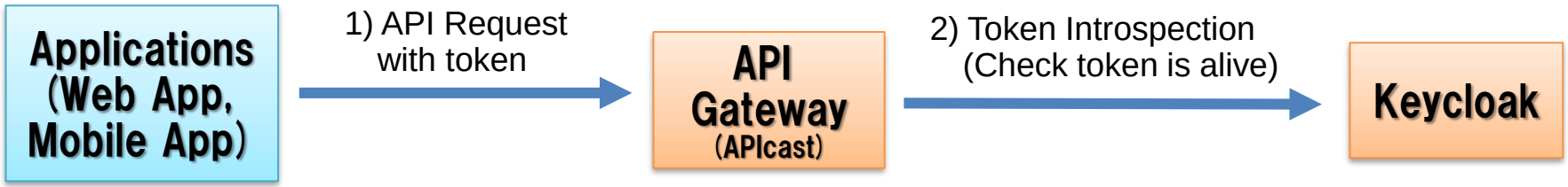
© Hitachi, Ltd. 2019. All rights reserved.

# Manage tokens : Issue

HITACHI
Inspire the Next

- Only authorization server knows that tokens are revoked…
  API gateways couldn't deny API requests even if tokens were revoked.

- API gateways MUST ask the authorization server whether tokens were revoked.
  -> **token introspection (RFC 7662)**

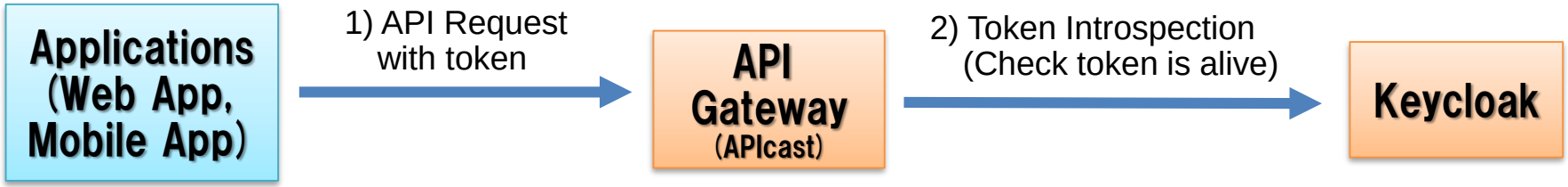



© Hitachi, Ltd. 2019. All rights reserved.

# Token Introspection policy

HITACHI
Inspire the Next

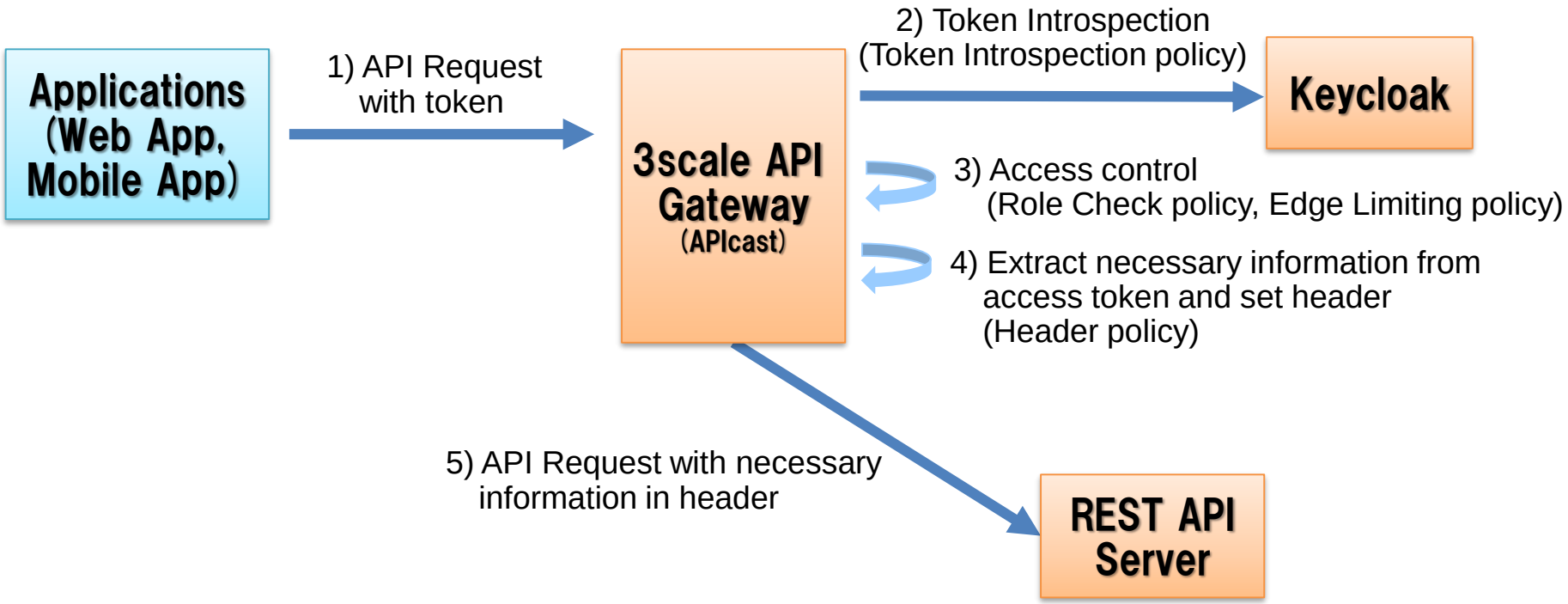
- We implemented patches and “Token Introspection policy” was included in 3scale 2.3.
  https://github.com/3scale/APIcast/pull/619
- This policy can cache the result of token introspection
  for reducing performance impact
  https://github.com/3scale/APIcast/pull/656




© Hitachi, Ltd. 2019. All rights reserved.

# How API is called in 3scale 2.3 + Keycloak

HITACHI
Inspire the Next

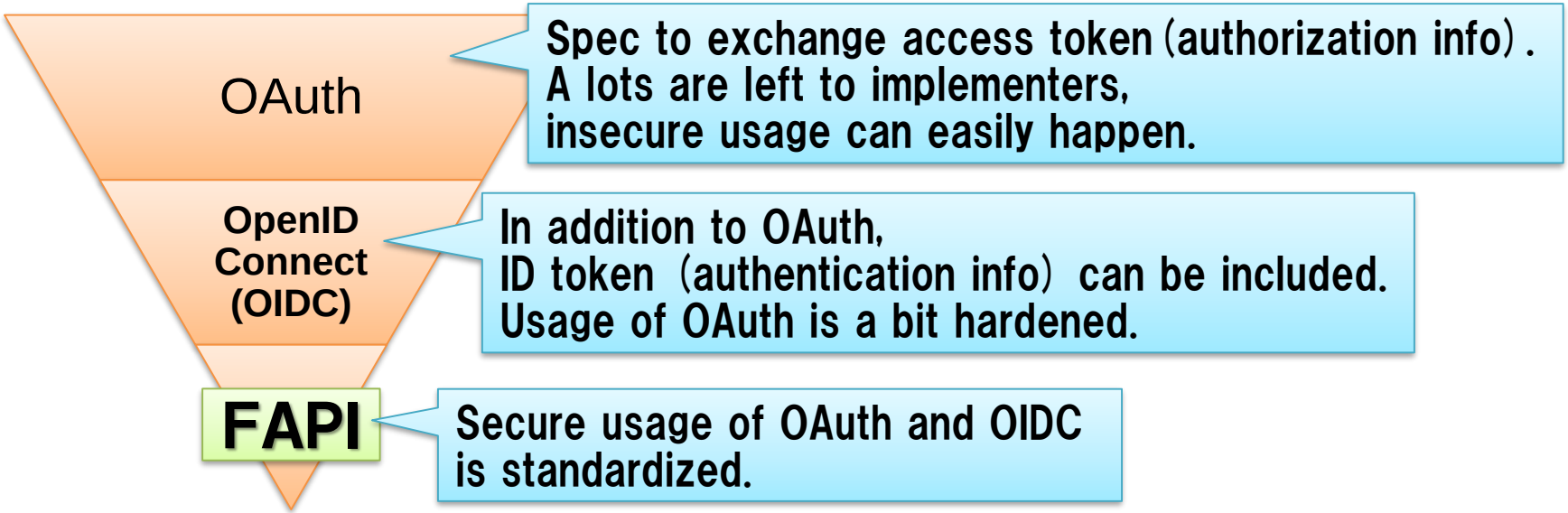



© Hitachi, Ltd. 2019. All rights reserved.

# Compliance to the latest standard: FAPI

HITACHI
Inspire the Next

FAPI (Financial-Grade API) is being standardized in OpenID Foundation.
Part1 (Read Only), Part2 (Read Write), JARM, CIBA




© Hitachi, Ltd. 2019. All rights reserved.

# FAPI in Japan

HITACHI
Inspire the Next

- FAPI is still implementer's draft as of today
- However, being strongly promoted in banking industry

> c With regard to authorization protocols, the OAuth 2.0 authorization framework[25] is recommended. With regard to detailed specifications for applying OAuth 2.0 to APIs in the financial field, standardization efforts are currently being undertaken by the OpenID Foundation Financial API Working Group (FAPI WG) as of June 2017, with a view to ensuring security standards. It is desirable for banks to comply (or give consideration to comply) with the specifications when this organization implements detailed specification for applying OAuth 2.0.[26]

Quoted from "Report of Review Committee on Open APIs: Promoting Open Innovation", Japanese Bankers Association
https://www.zenginkyo.or.jp/fileadmin/res/news/news290713_3.pdf

- We have to prepare for FAPI in advance, because can not implement soon.

© Hitachi, Ltd. 2019. All rights reserved.

# Issues toward FAPI in Keycloak

HITACHI
Inspire the Next

Investigated implementation of Keycloak, and reported issues.

We were developing patches with community, major parts were resolved.
Our colleague @tnorimat is mainly working.

| JIRA | Description | Pull Request | Included version |
|---|---|---|---|
| KEYCLOAK-2604 | RFC 7636(PKCE) support | 3831 | 3.1.0 |
| KEYCLOAK-5661 | shall return the list of allowed scopes with the issued access token | 4527 | 3.4.0 |
| KEYCLOAK-5811 | Client authentication client_secret_jwt | 4835 | 4.0.0 |
| KEYCLOAK-6700 | Support of s_hash | 5022 | 4.0.0 |
| KEYCLOAK-6768 | Support of Encrypted ID token | 5779 | Not yet |
| KEYCLOAK-6770 | Signature algorithm (PS256 or ES256) support | 5533 | 4.5.0 |
| KEYCLOAK-8460 | Signature algorithm (PS256 or ES256) support (for request object) | 5603 | 4.7.0 |
| KEYCLOAK-6771 | Support for holder of key mechanism | 5083 | 4.0.0 |

© Hitachi, Ltd. 2019. All rights reserved.

HITACHI
Inspire the Next

# Contents

1. Introduction: background and requirements

2. Usage of OSS to meet requirements
   - Which OSS should be used?
   - Security requirements
   - Availability requirements

© Hitachi, Ltd. 2019. All rights reserved.

# <Recap> Requirements for Availability for banking API

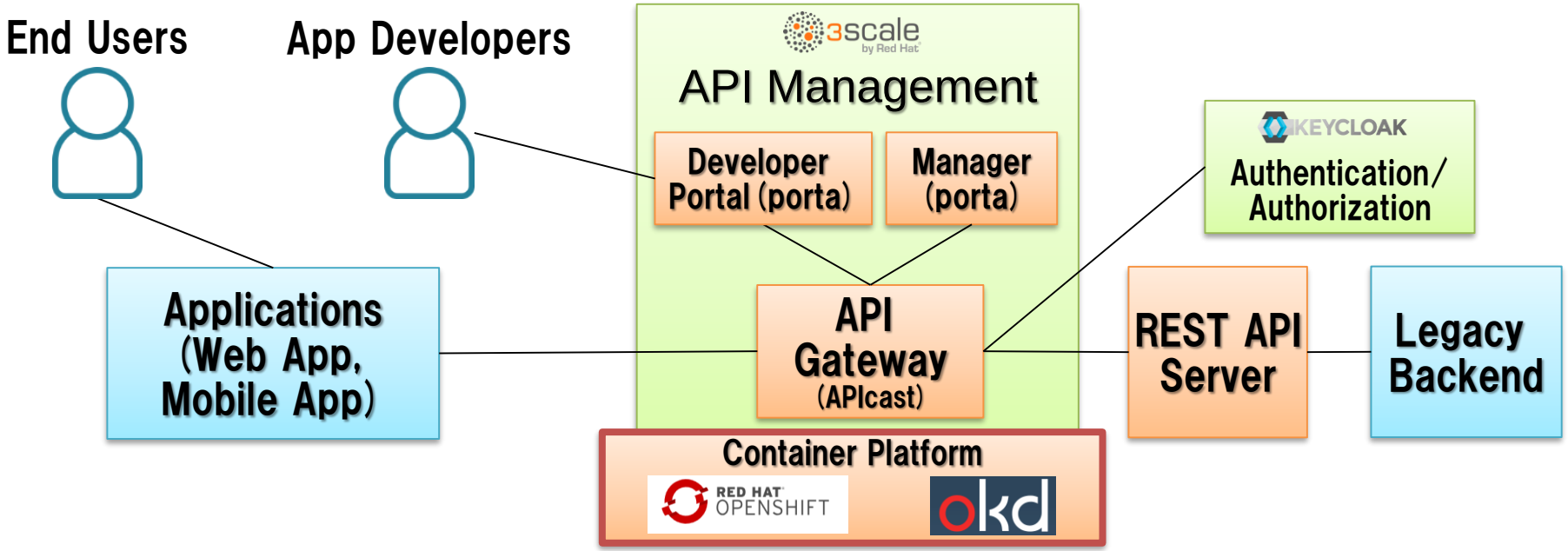
HITACHI
Inspire the Next

| # | Category | Description |
|---|---|---|
| 1 | Fault Tolerance | • HA configuration<br>• Can recover automatically |
| 2 | Fail Soft/<br>Fault Avoidance | • Reduce dependencies of each component |

Implemented these requirements using 3scale, collaborating with OSS community

© Hitachi, Ltd. 2019. All rights reserved.

# HA configuration / Automatic recovery



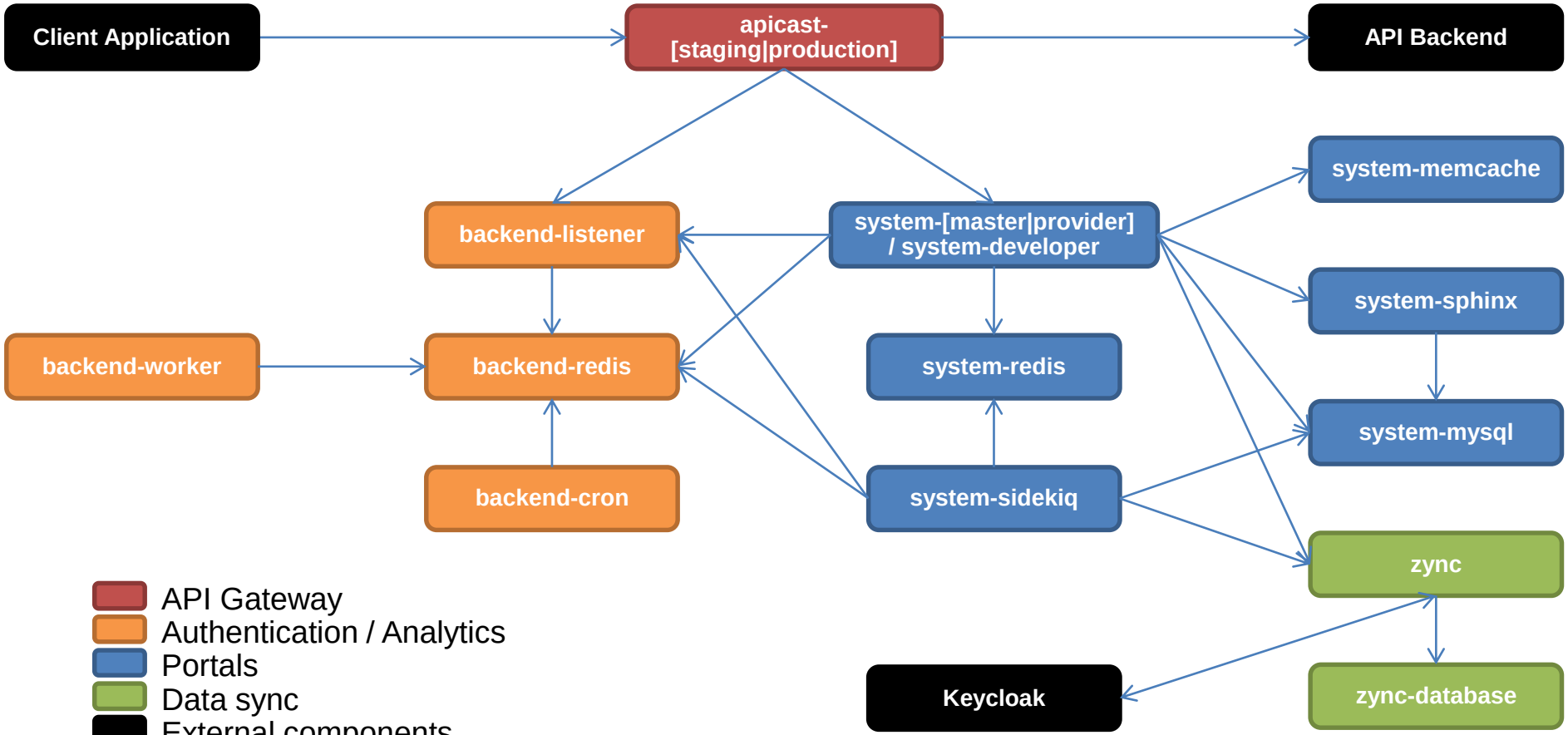

OpenShift provides:

- Automatic recovery/Automatic rerouting -> **Automatic recovery**
- Flexible scaling -> **HA configuration**

© Hitachi, Ltd. 2019. All rights reserved.

# Reduce dependencies of each component

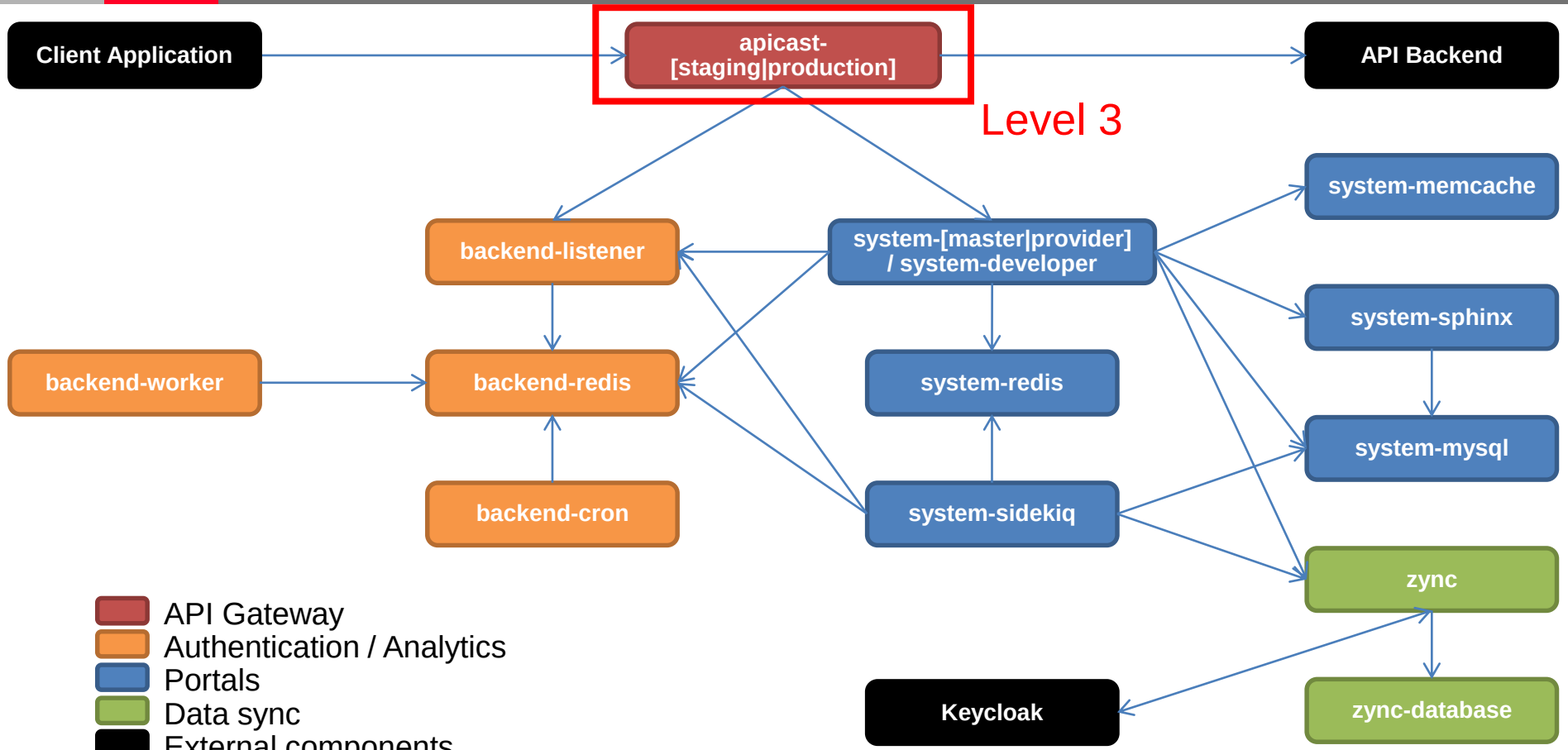
HITACHI
Inspire the Next




© Hitachi, Ltd. 2019. All rights reserved.

# Reduce dependencies of each component

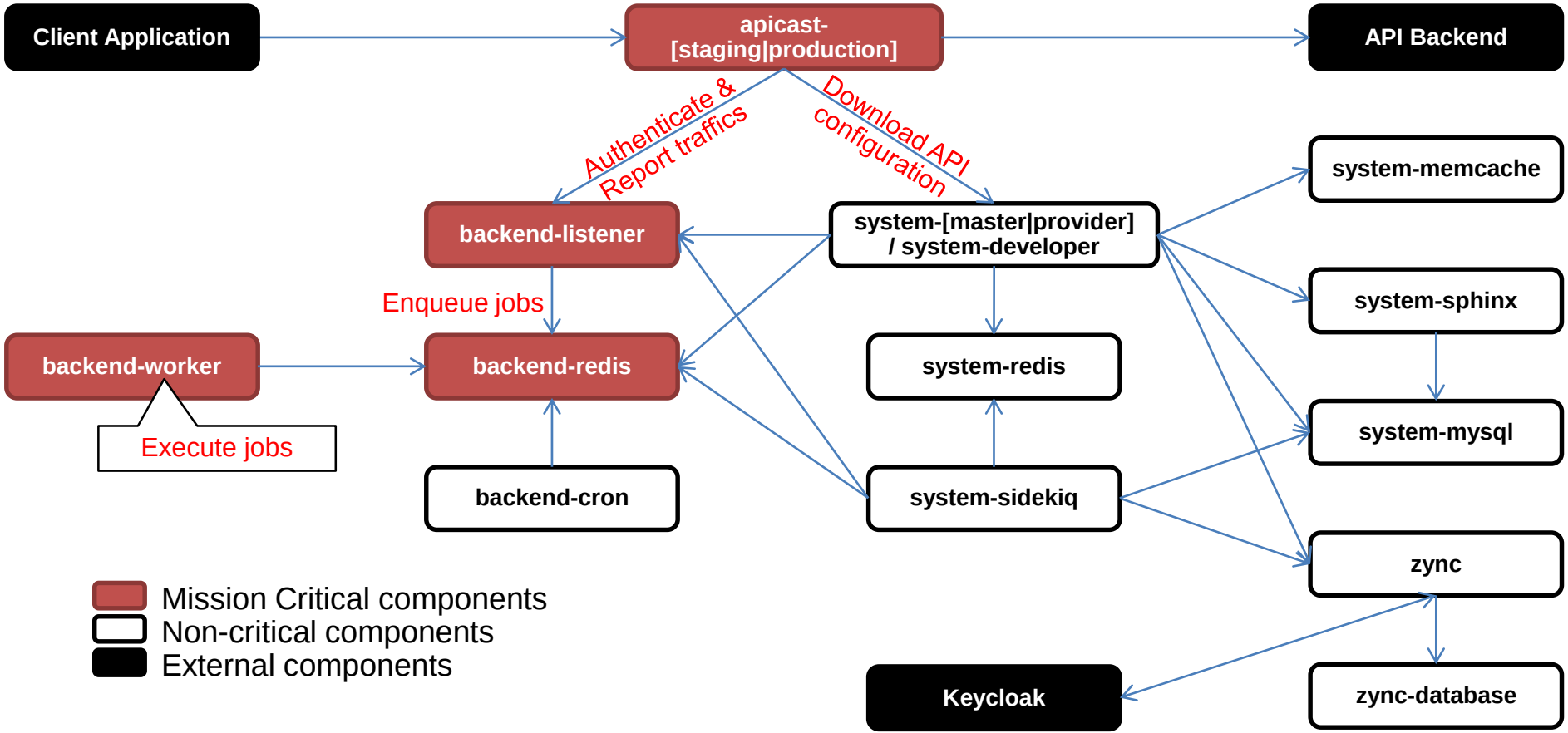
HITACHI
Inspire the Next




© Hitachi, Ltd. 2019. All rights reserved.

# Reduce dependencies of each component

HITACHI
Inspire the Next




© Hitachi, Ltd. 2019. All rights reserved.

# Reduce dependency (APIcast to Backend)


We have to consider:

1. how to authenticate API requests
2. how to report traffics

} when backend-listener is down

© Hitachi, Ltd. 2019. All rights reserved.

# Reduce dependency (APIcast to Backend)

HITACHI
Inspire the Next


We have to consider:

1. how to authenticate API requests
   ☞ **cache the result of authentication and authenticate using cache**

   *Caching policy*

   -> cannot authenticate newcomers and results to opportunity loss
   ☞ **allow newcomers without cache authentication and with alternative authentications**

   *Keycloak Role Check policy, Edge Limiting policy, Token Introspection policy*

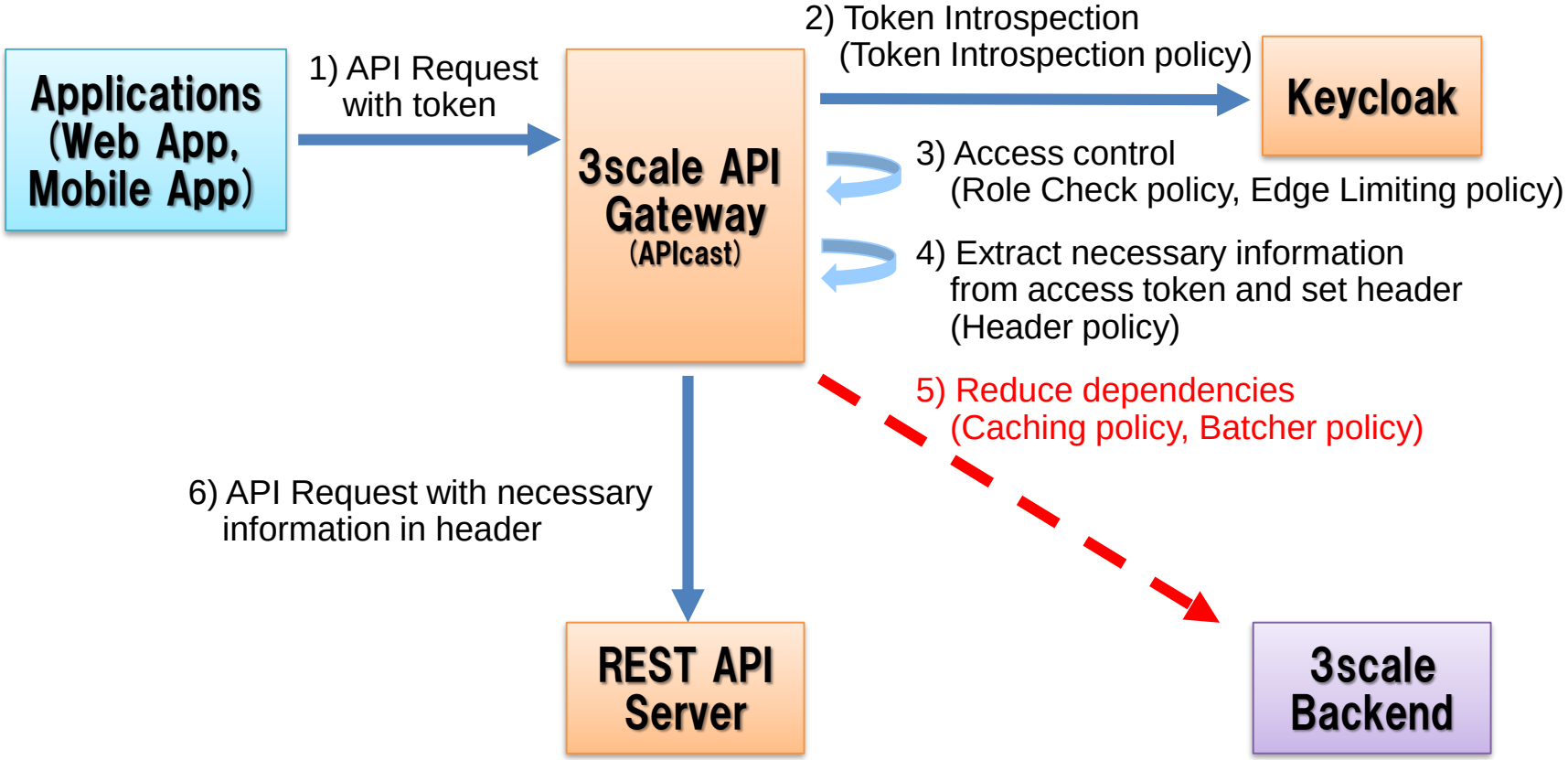
2. how to report traffics
   ☞ **cache traffics and report them all together when backend-listener comes back**

   *Batcher policy*

© Hitachi, Ltd. 2019. All rights reserved.

# How API is called in 3scale 2.3

HITACHI
Inspire the Next




© Hitachi, Ltd. 2019. All rights reserved.

# Summary

HITACHI
Inspire the Next

- OAuth is recognized as a key technology for banking API systems
- Requirements to be considered around OAuth
  - Authentication, Access control, Token management,
    Latest standard (OIDC, FAPI)
- Requirements to be considered around Availability
  - HA configuration, Dependencies
- Applied OSS (3scale + Keycloak) to achieve them
  - Improved with OSS community
    - 3scale: enhanced rate limit, access control
    - Keycloak: Features required for FAPI

    -> Improvements are included in the latest version
- Let's work with OSS community ! 3scale and Keycloak are great community.

© Hitachi, Ltd. 2019. All rights reserved.

# Trademarks

HITACHI
Inspire the Next

- Red Hat is a trademark or registered trademark of Red Hat, Inc. in the United States and other countries.
- OpenShift is a trademark or registered trademark of Red Hat, Inc. in the United States and other countries.
- WSO2 is a trademark or registered trademark of WSO2 in the United States and other countries.
- OpenID is a trademark or registered trademark of OpenID Foundation in the United States and other countries.
- GitHub is a trademark or registered trademark of GitHub, Inc. in the United States and other countries.
- Twitter is a trademark or registered trademark of Twitter, Inc. in the United States and other countries.
- Facebook is a trademark or registered trademark of Facebook, Inc. in the United States and other countries.
- Other brand names and product names used in this material are trademarks, registered trademarks, or trade names of their respective holders.

© Hitachi, Ltd. 2019. All rights reserved.

HITACHI

Inspire the Next