living draft-00
last edited on July 19, 2021

Identity-Based Network

Igor Zboran
izboran@gmail.com
July 19, 2021

Abstract

The Internet is a complex interconnection of people and services with the ability to exchange data. The networking industry has to agree on how to assemble a trusted, secure infrastructure for connecting these entities. In this paper the OpenID Connect (OIDC) protocol in combination with the User-Managed Access (UMA) protocol is proposed to overcome the identity management, access control and decentralized data storage issues of the current Internet.

Introduction

In the early days the Internet was a protocol driven – FTP, SMTP, HTTP, XMPP – decentralized ecosystem. Today's Internet has turned into a centralized group of cloud services. There is no easy way to run a widely accessible service without being tied to a specific cloud provider. The real cloud lock-in is an Identity and Access Management (IAM) with a Single-Sign On (SSO) authentication service.

Problem

IAM services were built for internal use behind the firewall. They were not designed to manage identity and access control across corporate boundaries. In the new cloud, social and mobile era, many enterprises now find themselves with more identity silos than ever.

Current Situation

.

Current Flaws

.

Proposed Solution

.

Motivation

.

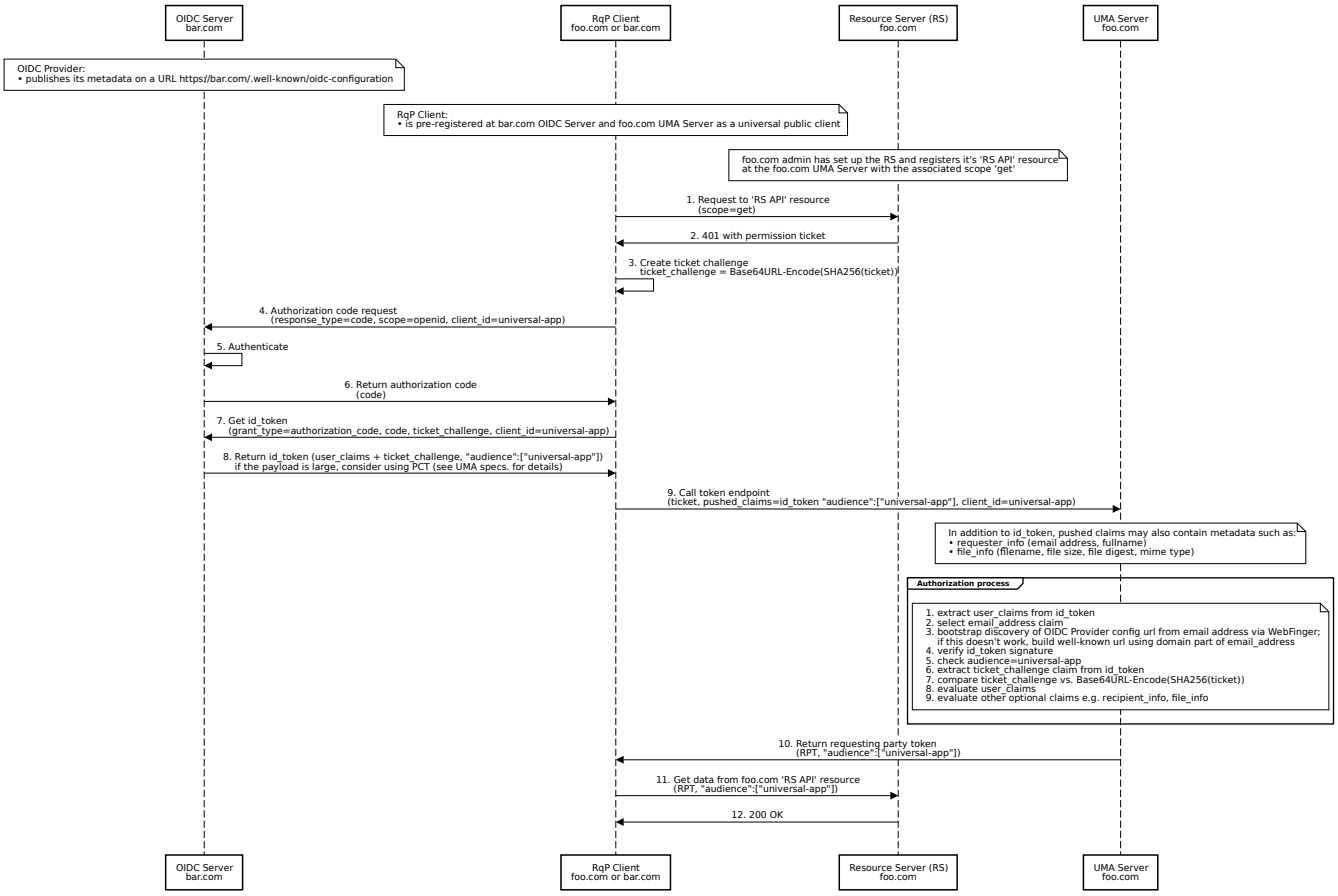
Main Concept

There are five principal issues that are going to have to be resolved.

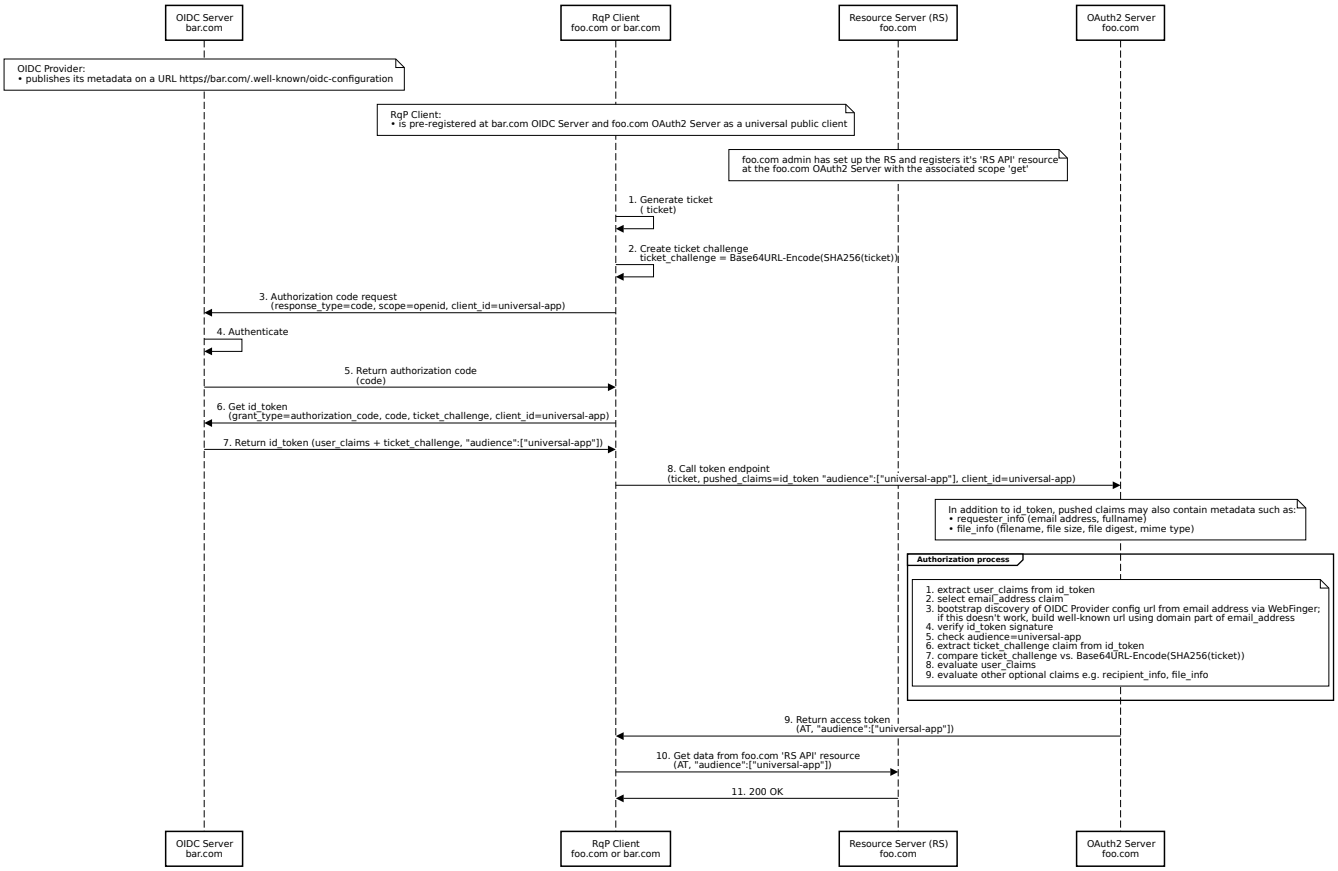
- Connectivity
- Identity
- Manageability
- Data Storage
- Interoperability

RESTful Flows

UMA/GET



OAuth2/GET


OIDC Server
bar.com
RqP Client
foo.com or bar.com
Resource Server (RS)
foo.com
OAuth2 Server
foo.com
OIDC Provider:
• publishes its metadata on a URL https//bar.com/.well-known/oidc-configuration
RqP Client:
• is pre-registered at bar.com OIDC Server and foo.com OAuth2 Server as a universal public client
foo.com admin has set up the RS and registers it's 'RS API' resource
at the foo.com OAuth2 Server with the associated scope 'get'
1. Generate ticket
(ticket)
2. Create ticket challenge
ticket_challenge = Base64URL-Encode(SHA256(ticket))
3. Authorization code request
(response_type=code, scope=openid, client_id=universal-app)
4. Authenticate
5. Return authorization code
(code)
6. Get id_token
(grant_type=authorization_code, code, ticket_challenge, client_id=universal-app)
7. Return id_token (user_claims + ticket_challenge, "audience":["universal-app"])
8. Call token endpoint
(ticket, pushed_claims=id_token "audience":["universal-app"], client_id=universal-app)
In addition to id_token, pushed claims may also contain metadata such as:
• requester_info (email address, fullname)
• file_info (filename, file size, file digest, mime type)
Authorization process
1. extract user_claims from id_token
2. select email_address claim
3. bootstrap discovery of OIDC Provider config url from email address via WebFinger;
if this doesn't work, build well-known url using domain part of email_address
4. verify id_token signature
5. check audience=universal-app
6. extract ticket_challenge claim from id_token
7. compare ticket_challenge vs. Base64URL-Encode(SHA256(ticket))
8. evaluate user_claims
9. evaluate other optional claims e.g. recipient_info, file_info
9. Return access token
(AT, "audience":["universal-app"])
10. Get data from foo.com 'RS API' resource
(AT, "audience":["universal-app"])
11. 200 OK
OIDC Server
bar.com
RqP Client
foo.com or bar.com
Resource Server (RS)
foo.com
OAuth2 Server
foo.com

About the Author

Igor Zboran is a mechanical engineer by education with professional experience as a software engineer and solutions architect. He'd like to transform his knowledge into a useful system or service that people would love to use.

Igor received Ing. degree in Mechanical Engineering from the University of Žilina, Slovakia in 1988. After graduating, he worked in several small private companies as a software developer. From 2008 to 2009, he provided expert advice to Prague City Hall IT department, Czech Republic as an external consultant. He invented a new decentralized Identity-Based Privacy (IBP) trusted model built around OAuth2 and OpenID Connect standards. Igor is a strong proponent of open source software and open standards.

References

[1] OpenID Connect (OIDC) https://openid.net/connect/
[2] User-Managed Access (UMA) https://en.wikipedia.org/wiki/User-Managed_Access
[3] WG - User Managed Access https://kantarainitiative.org/confluence/display/uma/Home