논문 19-44-10-20 The Journal of Korean Institute of Communications and Information Sciences '19-10 Vol.44 No.10
https://doi.org/10.7840/kics.2019.44.10.1928

웹 통합인증을 위한 SAML과 OIDC 규약 간 토큰변환 시스템의 개발

조 진 용◆°, 채 영 훈*, 공 정 욱*

Development of SAML-OIDC Token Translation System for Web Single-Sign On

Jinyong Jo◆°, YeongHun Chae*, JongUk Kong*

요 약

연합인증은 다수의 보안도메인에서 웹 통합인증을 가능케 하는 표준화된 사용자 인증 및 인가체계이다. 국제표준 보안규약인 SAML(Security Assertion Markup Language)를 이용하며 호환성 있는 기술 프로파일을 사용함으로써 비밀번호 피로도의 해소, 개인정보 보호의 강화, 연구자원의 공동 활용 가능성 증대와 같은 효과를 기대할 수 있다. 하지만 SAML 인증규약은 기술구현이 어렵고 구동환경의 설정과 메타데이터의 관리가 쉽지 않은 문제점을 가지고 있다. 다수의 국가에서 토큰의 구조가 간단하고 메타정보의 관리가 용이한 OIDC(OpenID Connect)를 차세대 연합인증 규약으로 수용하려는 움직임을 보이고 있다. 본 논문은 웹 응용이 연합인증을 목적으로 SAML 인증규약과 OIDC 인증규약을 모두 사용할 수 있는 토큰변환 시스템을 제안하고 필요성과 세부적인 구현 내용을 소개한다. 또한, 데이터과학 플랫폼인 JupyterHub와 OIDC를 이용해 연동하고 웹 통합인증의 처리과정을 살펴봄으로써 개발한 시스템의 성능과 기능을 정성적으로 분석한다.

Key Words : SAML, OpenID Connect, Token Translation, JupyterHub

ABSTRACT

Federated authentication is a standardized user authentication and authorization scheme that enables web-based SSO (Single-Sign On) in multiple security domains. We can expect the resolution of password fatigue, enhancement of personal information protection, and sharing of research resources by using the SAML (Security Assertion Markup Language), which is an international standard, and using interoperable technology profiles among the domains. However, the SAML-based user authentication is difficult to apply in web applications because it is not easy to implement, configure, and manage. OIDC (OpenID Connect) is showing acceptance in many countries as a next-generation authentication protocol since it has a simple structure of tokens and is easy to manage meta information. This paper proposes a token translation system that makes web applications can use both/either the SAML and/or the OIDC standard for federated authentication. In addition, we qualitatively evaluate the function as well as the performance of the developed system by federating it with the JupyterHub data-science platform acting as an OIDC client.

※ 본 연구는 한국과학기술정보연구원의 지원(K-19-L02-C02)으로 수행되었습니다.
◆° First and Corresponding Author : Korea Institute of Science and Technology Information, jiny92@kisti.re.kr, 정회원
* Korea Institute of Science and Technology Information, proin@kisti.re.kr; kju@kisti.re.kr, 정회원
논문번호 : 201908-147-D-RU, Received August 5, 2019; Revised August 22, 2019; Accepted August 23, 2019

Ⅰ. 서 론

연합인증(Federated authentication)은 다수의 보안 도메인 간에 적용되는 사용자 인증(Authentication) 및 인가(Authorization) 체계이다[1]. 계정연합(Identity federation)은 표준화된 연합인증 정책과 기술 프로파일을 공유하는 연합체로써 식별정보제공자(Identity provider)와 서비스제공자(Service provider) 및 신뢰 프레임워크(Trusted framework) 제공자로 구성된다. 계정연합에 속한 서비스제공자(예, 웹 응용)는 식별정보제공자에게 사용자 인증을 위임한다. 식별정보제공자는 사용자를 인증하고 식별정보(예, 속성정보)를 제공한다. 서비스제공자와 식별정보제공자가 인증정보를 교환하기 위해서 SAML(Security Assertion Markup Language[2]) 인증규약을 이용한다.

연합인증은 웹 응용에서 사용자 계정(식별정보제공자가 관리)을 분리함으로써 규제 적용이나 보안 관리를 간소화시킨다. 연합인증에서 식별정보제공자는 단일 제어점(Single point of access management)으로 동작한다. 검증된 표준규약을 준용함으로써 이질적 정보자원의 상호호환성을 높이며 특정 응용에 종속적인 통합인증 인프라(예, 독점적 통합인증 솔루션)를 지양한다. 또한, 소속기관에서 관리하는 사용자 크리덴셜(ID/비밀번호 등)을 이용해 계정연합에 속한 다수의 서비스제공자에 접속할 수 있기 때문에 비밀번호 피로도(Password fatigue)를 크게 줄이는 장점이 있다.

전 세계적으로 약 70개의 계정연합이 운영 중이다. 연구와 교육(R&E)을 목적으로 제공되는 정보자원에 대한 접근성을 높이거나 공동 활용을 목적으로 연합인증이 활용되고 있다. 미국은 정부기관과 대학 및 국립연구소 등 539개 R&E 기관들이 InCommon 계정연합에 참여하고 있다[3]. InCommon은 R&E 연구자원과 클라우드 서비스를 포함해 경영관리시스템 및 학습관리시스템 등 4,500여 서비스제공자들을 확보하고 있다. 영국의 UKAM 계정연합에는 총 1,175개의 R&E 기관과 서비스제공자가 참여하고 있다[4]. UKAM은 주로 도서관 응용과 전자출판물의 기관외부 접근(Off-site access)을 가능케 하는데 목적이 있다. 현재 40개 이상의 전자저널 출판사에서 연합인증을 지원하고 있다. 우리나라는 2016년도부터 정보자원의 상호호환성 확보와 공동 활용을 목표로 국가과학기술연구망에서 KAFE(Korean Access Federation[5]) 계정연합을 운영 중이다.

SAML 기반의 연합인증이 R&E 분야에서 다양한 목적으로 활용되고 있지만 태생적으로 웹 응용에만 적용할 수 있으며 기술구현이 어렵고 환경설정과 부가정보의 관리가 복잡하다는 문제가 있다. 다수의 계정연합과 서비스제공자들은 SAML 기반의 연합인증이 갖는 문제점을 해결하고자 OIDC(OpenID Connect[6]) 규약을 연합인증에 도입하고 토큰 변환(Token translation)을 통해 SAML과 OIDC를 함께 이용할 수 있는 기술을 개발하고 있다[7-14].

최근에 다수의 토큰변환 시스템들이 개발[7-14]되고 있지만 기능블록에 대한 단편적인 소개만 있을 뿐 보안메시지의 처리, 사용자 세션의 관리, 권한이나 자격의 관리방법 등 시스템을 구체적이고 통합적으로 기술한 논문은 없는 것으로 파악된다. 본 논문은 연합인증 환경에서 웹 응용이 SAML이나 OIDC 표준규약을 선택적으로 사용할 수 있도록 토큰변환 시스템을 제안함으로써 궁극적으로 국내 과학기술 응용서비스들이 쉽게 연합인증을 활용케 하는데 목적이 있다. 제안한 토큰변환 시스템을 프로덕션 환경에 적용하고 데이터과학 플랫폼이 JupyterHub[15]와 연동하는 방법을 제시한 점도 본 논문의 기여점이다.

본 논문은 다음과 같이 구성된다. 제2장에서 관련 기술을 소개하고 제3장에서 토큰변환 시스템의 필요성과 기능적 요구사항을 제시한다. 제안한 시스템의 개발 내용과 정성적인 기능 평가는 제4장과 제5장에서 수행한다. 마지막으로 제6장에서 결론을 맺는다.

Ⅱ. 관련 연구

본 장에서는 통합인증 및 연합인증의 표준규약을 살펴보고 제안하는 토큰변환 시스템과 유사한 연구개발 활동을 소개한다. 본 논문에서 토큰은 사용자 정보를 담고 있는 데이터 구조로 정의한다.

2.1 통합인증 표준규약

국제적으로 R&E 분야에서는 SAML을 표준규약으로 채택해 연합인증에 활용하고 있다. SAML은 태생적으로 모바일 환경을 지원하지 않고 메타데이터를 동기화해야 하며 전자서명의 검증과 어설션(Assertion)의 처리가 쉽지 않은 등 인증규약의 적용이 제한적이고 기술구현의 난이도와 복잡도가 높은 문제가 있다. SAML의 문제점을 극복하기 위해서 OIDC Federation[16]에 대한 필요성이 대두되고 있으며 Shibboleth 컨소시엄은 식별정보제공자용 소프트웨어 패키지에 OIDC를 적용할 계획[17]이다. Shibboleth는 SAML 소프트웨어의 일종이다. 장기적으로 연합인증 규약이 SAML에서 OIDC로 전환될 것

으로 예상한다.

SAML은 보안정보를 교환하기 위한 XML 기반의 프레임워크로써 통합인증(SSO, Single-Sign On)과 연합인증(Federated SSO)에 활용되는 표준규약이다. 2003년에 SAML 1.1이 2005년에는 SAML 2.0이 OASIS 표준으로 비준되었다. 인증기능을 제공하는 식별정보제공자와 인가기능을 제공하는 서비스제공자 간에 보안인증 메시지를 교환하기 위해 활용된다. 전송규약은 HTTP나 SOAP(Simple Object Access Protocol)을 이용하며 SAML 어설션에 사용자의 인증정보와 자격정보 및 속성 정보가 포함된다.

OIDC는 OAuth2를 경량화한 통합인증 규약으로서 RESTful API를 통해 보안정보를 교환한다[18]. OpenID Foundation에 의해 2014년에 표준으로 비준되었다. OAuth2 제공자는 사용자의 인증 정보를 반환하지 않지만 OIDC 제공자는 ID 토큰을 이용해 인증정보를 반환한다. 사용자 정보에 대한 접근을 통제하기 위해서 Scope(예, openid, profile 등)를 이용한다. JWT(JSON Web Token)을 이용해 사용자 정보를 전달하기 때문에 SAML 어설션에 비해 구현의 난이도와 복잡도가 현저히 낮아지는 장점이 있다.

2.2 식별정보의 대리 및 중개

본 논문에서 식별정보 대리(Identity proxying)는 웹 통합인증을 처리하는 과정에서 서비스제공자 또는 식별정보제공자를 대리하는 네트워킹 개체(이하, SAML 프록시)로 정의한다. 또한, 식별정보 중개(Identity brokering)는 SAML 개체(식별정보제공자 또는 서비스제공자)와 OIDC 개체(OpenID 제공자 또는 OpenID 클라이언트) 사이에서 토큰변환을 수행하는 네트워킹 개체(이하, SAML 브로커)로 정의한다.

유럽은 식별정보 대리와 중개를 위해 청사진 구조(Blueprint Architecture[19])를 설계하고 국가 간 협업에 필요한 정보자원의 인증 인가 인프라(이하, AAI) 구축에 활용할 수 있도록 권장하고 있다. 청사진 구조는 유럽 AARC(Authentication and Authorisation for Research Collaboration[20]) 프로젝트의 산출물로써 연합인증에 필요한 AAI의 빌딩블록을 정의하고 있다. 총 5개의 구성요소를 제시하고 있다.

본 논문에서 제안하는 토큰변환 시스템은 상호운용성과 호환성을 높이기 위해서 AARC의 청사진 구조를 준용하고 있다. 청사진 구조는 기능블록들 대해서 구체적인 비즈니스 로직을 제시하지 않으며 국가 간의 법령이나 규제도 다르므로 동일한 청사진 구조를 채택한 AAI들도 기술적 차이를 갖게 된다.

표 1. AARC 청사진 구조의 5개 주요 구성요소
Table 1. Five component layers in the AARC blueprint architecture.

Layers	Functional Roles
Identity	Manage digital identities
Attribute	Manage user attributes
Translation	Proxying and brokering
Authorisation	Manage access to resources/services
Services	External end services

CORBEL[8]은 생명과학과 의학연구를 위한 13개 자원인프라로 구성되어 있으며 2017년부터 AAI를 설계하기 시작했다. 총 2개의 SAML 프록시와 1개의 OIDC 브로커로 구성되며 ORCID(Open Researcher and Contributor ID)나 SAML/OIDC 기반의 토큰들은 대리하거나 중개하도록 설계되었다[7]. 개별 구성요소들의 관리운영 주체가 다수이며 X.509 토큰변환을 수행하는 점과 정보자원에 대한 접근제어(권한부여)의 부재 등에서 제안하는 시스템과 차이가 있다.

CTA[9]는 감마선 망원경을 구축하기 위한 범 세계적 프로젝트이다. CTA AAI는 가상조직과 그룹을 관리할 수 있으며 SATOSA[10]를 이용해 토큰변환 서비스를 제공한다[7]. 본 논문에서 제안하는 토큰변환 시스템과 기능적으로는 유사하나 정보자원에 대한 접근제어 기능을 제공하지 않으며 식별정보 대리 기능을 제공하기 위해 1개의 SAML 프록시로 구성한 점 등에서 제안하는 시스템과 구조적으로 차이가 있다.

CILogon[11]은 특정 응용에 종속되지 않고 범용으로 활용할 수 있다. EGI Check-in[12]는 컴퓨팅 자원을 연합해 활용하기 위한 AAI로 개발되었으나 범용성을 확보하고 있다. WLCG[13] 및 EISCAT_3D[14]는 특정 응용(예, 과학 분야)을 지원하기 위한 AAI로써 AARC 청사진 구조에 따라 설계되었다. 하지만 소개한 AAI들은 이용된 공개 소프트웨어, OIDC의 지원여부, 권한관리 기능의 제공여부, 사용자세션의 유지여부, 개발기술 산출물의 응용서비스 적용여부, 식별정보의 변환여부 등에서 제안한 토큰변환 시스템과 차이가 있다.

Ⅲ. 요구사항 및 필요성

본 장에서는 토큰변환 시스템의 기능적 요구사항과 필요성에 대해서 살펴본다. 연합인증 기술을 적용하기 위해 웹 응용 개발자들과 협업하는 과정에서 다음과 같은 필요성과 요구사항이 제기되었다.

· **쉬운 인증규약**, 국내에서는 SAML 인증규약이 행정기관을 중심으로 활용[21]되었고 민간으로는 크게 확산되지 못했기 때문에 관련된 산업생태계가 열악한 상황이다. 웹 응용에 SAML을 적용하고 구동환경을 설정하는 과정이 국내 개발자들에게 생소한 실정이다. OIDC 규약은 메타데이터의 관리가 쉽고 RESTful API 수준에서 통합인증을 쉽게 구현할 수 있는 장점이 있다. JupyterHub나 Openstack과 같이 근래에 개발되고 있는 다수의 R&E 플랫폼들은 OIDC를 기본적인 인증규약으로 수용하고 있다. 따라서 R&E 응용이 SAML과 함께 OIDC 규약도 선택적으로 이용할 수 있어야 한다.

· **자원 및 사용자권한 관리**, 특정 사용자의 자격과 권한을 통제하기 위해서 정보자원에 대한 접근제어가 필요하다. 하지만 상용 웹 응용은 접근관리 기능이 제품에 종속적이므로 관리의 유연성이 떨어질 수 있다. 예를 들어, 사용자 수에 비례해 과금하는 일부 상용 웹 응용들은 수익증대를 목적으로 접근관리 기능을 제공하지 않는 경우도 있다. 상업적인 이유로 인해 웹 응용이 메타데이터를 동기화하지 못하거나 하나의 식별정보제공자만 가질 수 있도록 제한되기도 한다. 정보자원의 접근관리를 중앙화해 개방함으로써 종단 웹 응용에 대한 제어가능성을 높이고 관리비용을 줄일 수 있어야 한다.

· **소셜 로그인**, 낮은 보증레벨(Level of Assurance)을 인내할 수 있거나 제공하는 자원의 중요도가 낮은 웹 응용(예, 화상회의 등)들은 이용편의성을 제공하기 위해 소셜 로그인을 채택하고 있다. 최후의 인증수단(Last resort identity provider)으로도 소셜 로그인을 이용하기도 한다. 소셜 로그인은 OIDC나 OAuth2 인증규약을 이용하므로 연합인증에서 사용하는 SAML과는 함께 사용될 수 없다. 웹 응용이 수용하는 인증규약의 수가 늘어날수록 개발비용이 증가하므로 SAML의 문맥에서 웹 응용이 다수의 소셜 로그인을 통합할 수 있어야 한다.

· **사용자 동의 및 접근로그 관리**, 개인정보 보호법 제17조의 '개인정보의 제공'과 정보통신망 이용촉진 및 정보보호 등에 관한 법률의 제24조의 '개인정보의 제공 동의 등'에 따르면 개인정보를 제3자에게 제공하기 위해서는 정보주체에게 알리고 동의를 받아야 한다. 중앙화된 정보시스템(식별정보 처리목적)은 법령이나 규정 및 지침을 준수를 위해 정보주체로부터 전자적 동의를 받고 동의 결과를 관리해야 한다. 또한 식별정보의 오용을 예방할 수 있도록 인증된 사용자의 로그정보를 유지해야 한다.

Ⅳ. 토큰변환 시스템의 구현

본 장에서는 제안한 토큰변환 시스템의 내부 구성요소를 살펴본다. 제안한 시스템은 호환성과 확장성을 높이기 위해 AARC 청사진 구조에 따라 설계되었다.

그림 1은 제안한 시스템의 기능블록과 연동관계를 간략히 보여준다. 주요 구성요소는 토큰변환을 수행하는 SAML 프록시(Proxy)와 OIDC 브로커(Broker), 프록시와 브로커를 제어하거나 메타정보를 유지하는 정책관리시스템(Policy Management) 및 부가정보를 제공하는 속성관리시스템(Attribute Authority)를 포함한다. 추가적으로 SAML 프록시는 사용자동의(User Consent), 탐색서비스(Discovery Service) 및 OIDC 인터페이스(Interface) 등 다수의 마이크로서비스를 포함한다.

그림 1의 종단 서비스(End Services) 블록은 사용자가 이용할 수 있는 정보자원이고 식별정보(Identity) 블록은 사용자를 인증하기 위한 개체이다. 사용자는 식별정보 블록에서 로그인할 수 있다. 변환(Translation) 블록은 토큰변환을 통해 OIDC와 SAML 플로우 간에 호환성을 제공한다. 추가적으로 법령준수나 편의성 제공을 위한 기능을 포함하고 있다. 인가(Authorisation) 블록은 변환 블록에서 처리되는 OIDC 또는 SAML 플로우를 제어하고 플로우에 포함된 식별정보를 수집해 관리한다. 속성(Attribute) 블록은 사용자로부터 부가속성을 등록받거나 부가속

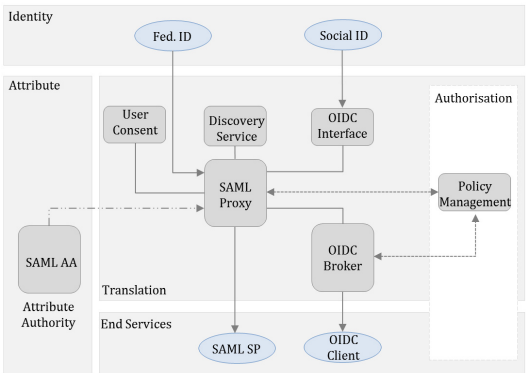

그림 1. 제안한 토큰변환 시스템의 개념도
Fig. 1. High-level overview of the token translation system.

성을 변환 블록에 제공한다.

통합인증 플로우를 간략히 설명하면 다음과 같다. 사용자는 SAML 서비스제공자(SP)나 OIDC 클라이언트에 접근해 로그인을 요청한다. OIDC 플로우는 SAML 플로우로 변환되어 SAML 프록시에 전달된다. SAML 프록시는 이용이 가능한 식별정보제공자의 목록을 사용자에게 제시(즉, 탐색서비스를 제공)한다. 정책관리시스템은 종단 서비스별로 이용 가능한 식별정보제공자의 목록을 관리한다.

선택한 식별정보제공자에서 사용자가 로그인에 성공하면 인증정보와 속성정보가 명기된 어설션을 발급해 SAML 프록시에게 전달한다. 소셜 ID 제공자가 전달한 사용자정보는 OIDC 인터페이스를 통해 SAML 어설션으로 변환되어 전달된다. SAML 프록시는 법령 준수를 위해 사용자 동의를 받고 어설션을 재구성한 후에 종단 서비스에게 전달한다. 속성관리시스템으로부터 해당 사용자의 부가정보(예, 그룹정보 등)를 획득하고 SAML scope(인증주체)가 프록시로 변경되므로 어설션을 재구성한다. OIDC 클라이언트로 전달해야 할 경우에는 OIDC 브로커가 SAML 어설션을 토큰으로 변환해 제공한다.

정책관리시스템은 토큰변환 시스템의 중앙 제어기로 동작한다. SAML 또는 OIDC 플로우에 포함된 사용자의 속성정보를 이용해 정보자원에 대한 접근을 통제한다. 또한, 개별 종단 서비스에 접근할 수 있는 식별정보제공자를 제어함으로써 웹 응용의 보안성을 높이고 연합인증 활용의 유연성을 제공한다. 법령의 준수와 보안사고 시 빠른 대응을 위해서 사용자의 동의기록과 서비스 이용기록을 실시간으로 수집해 관리한다. 마지막으로 승인된 종단 서비스만 토큰변환 시스템을 이용할 수 있도록 서비스제공자 및 OIDC 클라이언트에 대한 관리등록(Managed registration) 기능을 제공한다.

4.1 OIDC 인터페이스의 구현 및 토큰 변환

제안한 시스템의 빠른 구현과 안전성을 확보하기 위해 SAML 소프트웨어인 simpleSAMLphp[22]와 OIDC 라이브러리[23]을 이용해 OIDC 인터페이스를 구현했다. simpleSAMLphp는 환경설정이 쉽고 모듈화를 통해 비즈니즈 로직을 효과적으로 구현할 수 있는 장점이 있다.

OIDC 인터페이스는 OIDC 토큰을 SAML 어설션으로 변환하므로 OAuth2/OIDC를 백앤드(Backend)로 갖는 SAML 식별정보제공자로 여길 수 있다. OIDC 인터페이스를 통해 Google과 Naver를 SAML 프록시에 연동했으며 OIDC Scope(정보제공 범위)으로 openid, profile, email을 이용했다. OIDC 인터페이스는 OIDC 클레임(Claims)을 SAML 속성(Attributes)로 변환해야 한다. 하지만 SAML 속성명에 대해서 OIDC 제공자마다 사용하는 OIDC 클레임의 이름이 다르므로 표 2와 같은 변환표를 이용해 표준화했다.

표 2의 'sub'와 'id'는 사용자 ID를 나타내는 OIDC 클레임이다. 평문형태의 'sub'나 'id'가 노출되지 않도록 SAML 속성으로 변환하는 과정에서 해당 클레임을 단방향 암호화 했다. ePPN(eduPersonPrincipalName)은 사용자의 고유식별자이며 myid@univ.ac.kr와 같은 형태를 갖는다.

표 2. SAML 속성과 OIDC 클레임 간의 변환표
Table 2. Conversion map between SAML attributes and OIDC claims.

SAML Attributes	Google Claims	Naver Claims
sn	family_name	sn
givenName	given_name	gn
cn	name	name
displayName	name	nickname
mail	email	email
ePPN	sub@issuer	id@issuer

4.2 SAML 프록시 및 OIDC 브로커의 개발

그림 2는 개발된 토큰변환 시스템의 내부구조를 보여준다. OIDC 브로커와 SAML 프록시는 각각 SATOSA와 simpleSAMLphp를 이용해 구현했다. 사용자 동의(Consent), 자격권한(Entitlement), 속성확보(Attribute), 탐색서비스(Discovery), 접근관리(Access) 및 등록관리(Registration)는 simpleSAMLphp와

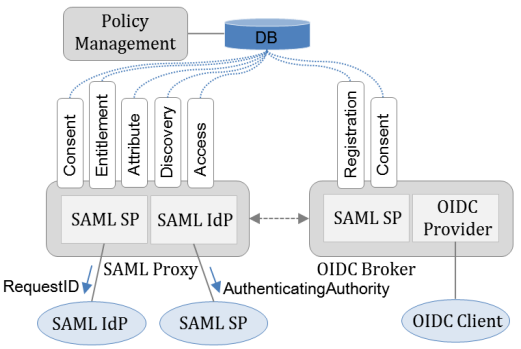

그림 2. 토큰변환 시스템의 기능적 구성요소
Fig. 2. Functional components of the token translation system.

SATOSA의 소프트웨어 모듈로 개발했다. 개발된 모듈이 제공하는 기능을 본 논문에서는 '마이크로서비스'로 정의한다.

그림 3은 제안한 토큰변환 시스템을 이용해 웹 브라우저(Agent)가 OIDC 클라이언트로부터 사용자정보(UserInfo)를 획득하는 전체 과정을 보여준다. 로그인 과정과 동의 과정은 그림에서 생략되어 있다. OIDC는 인가 코드 플로우(Authorization Code Flow)를 이용하는 것으로 가정한다.

웹 브라우저가 OIDC 브로커에게 인가코드를 요청(1)하면 OIDC 브로커의 서비스제공자(O_s)는 SAML 프록시에 포함된 식별정보제공자(S_i)에게 사용자 인증을 요청(2)한다. SAML에서 인증 요청은 HTTP Redirect를 이용한다. SAML 프록시의 서비스제공자(S_s)는 종단 식별정보제공자(E_i)에게 사용자 인증을 요청(3)한다. E_i에서 로그인에 성공하면 사용자가 인증된다.

사용자의 인증정보와 속성정보는 S_s에게 전달(4)되고 S_i에 의해 O_s의 ACS(Assertion Consumer Service)에게 전달(4)된다. SAML에서 인증 응답은 일반적으로 HTTP POST를 이용한다. OIDC 클라이언트(O_c)가 인가코드를 획득(5)하면 OIDC 브로커에게 ID/Access 토큰을 요청한다. OIDC 브로커에 포함된 OIDC 제공자(O_p)는 해당 토큰을 반환하고 추가적으로 O_c가 사용자정보(UserInfo)를 획득함으로써 사용자 인증과정이 완료된다.

플로우의 처리과정에서 정책관리시스템과 마이크로서비스들은 SQL 질의 또는 RESTful API를 이용해 정보를 공유한다. 정책관리시스템은 독립형(Standalone) 시스템인 프록시와 브로커를 중앙에서 조화롭게 제어하므로 제안한 시스템의 이용편의성을

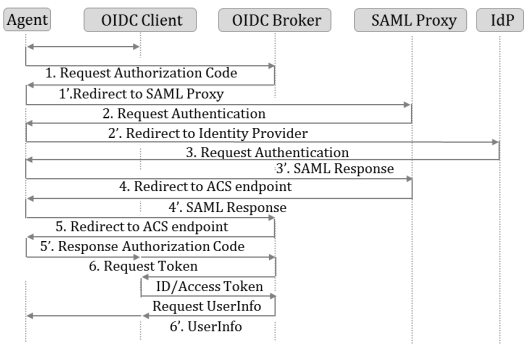

그림 3. 제안한 시스템의 웹 통합인증 플로우
Fig. 3. Web Single-Sign On flow of the proposed system.

표 3. 개발된 마이크로서비스의 역할
Table 3. Roles of developed microservices.

Microservice	Functional Roles
Consent	Display and get User Consent
Entitlement	Check eligibility for a service/client
Attribute	Get user attributes from AA
Discovery	Access control of Identity providers
Access	Track user consent and access
Registration	Set/get Client Registration

높이고 관리비용을 줄이는 효과가 있다. 정책관리시스템이 관리하는 각 마이크로서비스의 역할은 표 3과 같다.

앞서 기술했듯이 SAML 프록시는 simpleSAMLphp를 이용해 구현되었다. simpleSAMLphp는 기본적으로 식별정보제공자 또는 서비스제공자로 동작하지만 구동환경을 설정함으로써 프록시 기능을 활성화시킬 수 있다. 종단의 서비스제공자(E_s, 그림 2에서 타원으로 표시된 SAML SP)나 O_s는 S_i와 1:1로 연동된다. SAML 프록시(S)의 S_s가 다수의 외부 E_i와 연동되므로 하나의 식별정보제공자만 허용하는 상용 웹 응용도 프록시를 통해 다수의 식별정보제공자를 이용할 수 있게 된다.

식별정보제공자는 웹 응용의 통합인증을 위해서 로그인한 사용자의 세션정보를 유지한다. E_s에 접근한 사용자가 SAML 프록시를 통해 E_i에 로그인하면 S_i와 E_i가 해당 사용자의 세션정보를 각각 유지해야 한다. 이 상태에서 사용자가 통합 로그아웃(Single Logout)을 요청하면 명시적인 로그아웃(예, 웹 브라우저에서 로그아웃 버튼을 클릭)을 E_s, S_i 및 E_i에서 각각 수행해야 하므로 사용자 경험(User experience)이 크게 훼손된다. 통합 로그인(Single Login)의 경우에도 사용자가 E_i에 로그인하면 S_i도 해당 사용자의 세션정보를 유지하므로 브라우저 캐시가 지워지지 않는 한 다른 식별정보제공자에 로그인할 수 없는 문제가 있다.

통합 로그인과 통합 로그아웃 과정에서 발생하는 사용자 경험의 훼손을 방지하기 위해서 프록시의 세션관리 기능을 재 구현했다. 프록시 S는 E_s에 접속한 사용자의 세션정보를 E_i로부터 SAML 어설션을 전달받아 E_s에 반환한 시점(그림 3의 4번 구간)까지만 유지한다. 즉, 로그인에 성공한 사용자의 세션정보를 저장하지 않으므로 프록시 사용으로 인한 통합로그인과

통합 로그아웃의 문제를 해결할 수 있다.

SAML 어설션은 cross-site HTTP POST 방식(즉, 서비스제공자가 도메인이 다른 식별정보제공자의 세션정보를 전달)으로 전달되는데 Chrome 웹 브라우저는 cross-site 간 정보공유를 명시적으로 금지할 예정이다[24]. SAML 프록시에서 세션정보를 일시적으로 유지하기 때문에 cross-site로 인해 식별정보제공자의 세션정보가 서비스제공자에게 전달되지 않는 문제가 발생할 수 있다.

하지만 사용자 동의나 탐색서비스 등 마이크로서비스의 제공을 위해서 세션정보가 반드시 필요하다. 예를 들어, E_i나 E_s는 S_s나 S_i가 아닌 E_s나 E_i가 종단 개체라는 사실을 각각 SAML 메시지의 RequestID나 AuthenticatingAuthority 구분자를 통해 확인할 수 있다. 프록시는 세션정보를 활용해 해당 구분자를 설정한다. Chrome 웹 브라우저에서 발생하는 cross-site 문제를 회피하기 위해서 SAML 프록시의 웹 구동환경을 SameSite=None으로 설정함으로써 웹 브라우저가 cross-site 검사를 수행하지 않도록 했다.

OIDC 브로커(O)는 SATOSA를 이용해 구현했다. O_p의 입장에서 O_s는 백앤드 데이터베이스처럼 동작한다. SATOSA는 OIDC와 SAML 규약 간에 토큰변환이 가능한 시스템으로써 파이썬으로 구현되어 있다. SAML 프록시를 이용하지 않아도 다수의 식별정보제공자들을 연동할 수 있으며 비즈니스 로직을 효과적으로 구현할 수 있도록 모듈화되어 있다.

제안한 토큰변환 시스템이 SAML 프록시를 별도로 분리한 이유는 1) SAML 서비스제공자를 수용할 수 있어야 하고 2) 서비스 지속성을 높이는 측면에서 타 OIDC 브로커(예, KeyCloak[25])와 쉽게 연동할 수 있는 구조를 우선 시 했기 때문이다. 또한, 3) SAML 프록시에서 마이크로서비스를 구현하거나 검증하기 쉬우므로 시스템의 빠른 프로토타이핑이 가능했기 때문이다.

OIDC 제공자는 등록된 클라이언트에게만 토큰을 제공한다. 제안한 시스템은 사용자와 클라이언트에 대한 보안 추적과 감사가 용이하도록 클라이언트의 관리등록 또는 수동등록만 허용한다. 관리자가 정책관리 시스템을 이용해 OIDC 클라이언트를 등록할 수 있도록 구현했다. 또한, 등록된 클라이언트의 정보를 RESTful API를 통해 정책관리시스템이 획득할 수 있도록 OIDC 브로커에 해당 API 모듈을 구현했다. OIDC 브로커는 O_p를 위해 클라이언트의 정보를 관리해야 한다. 등록해야 하는 클라이언트 정보는 표 4와 같다.

표 4. OIDC 클라이언트의 수동 등록
Table 4. Managed Registration of an OIDC Client.

Attributes	Description
Client	URI of OIDC client
Name	Name of the client
Country	Providing country
Scope	Required OIDC Scopes
Privacy	URL of the client's privacy policy
Redirection	Redirect endpoint

표 4에 나열된 정보들은 마이크로서비스(예, 사용자 동의)가 사용자에게 공지해야 하는 항목을 가시화하기 위해서도 활용된다. OIDC 브로커에서 openid, email, profile, kafe.userinfo를 Scope로 이용할 수 있도록 개발했다. kafe.userinfo는 OIDC 클레임에서 지원하지 않는 SAML 속성(예, 그룹정보나 자격정보 등)을 OIDC 클라이언트에서 활용하기 위해 정의했다. 반환주소(Redirect endpoint)는 사용자 인증에 성공하면 리디렉트시킬 URL 주소이다. 클라이언트가 성공적으로 등록되면 정책관리시스템은 클라이언트 ID(client_id)와 비밀번호(client_secret)를 발급한다.

SAML 프록시 S가 O_c에서 시작된 플로우에 대해 마이크로서비스를 제공하기 위해서는 O_c를 식별하고 O_c에 대한 정보를 얻을 수 있어야 한다. S_i는 O_s를 종단 서비스로 인식하기 때문에 일반적으로 O_c를 식별할 수 없다. O_s가 S_i에게 SAML 인증요청 메시지를 전달할 때, O_c의 임시식별자를 생성해 RelayState에 포함하고 S_i가 임시식별자를 킷값으로 RESTful API를 호출해 O_c에 대한 정보를 얻을 수 있도록 개발했다.

SAML RelayState는 HTTP 매개변수로써 통합인증 플로우의 처리과정에서 서비스제공자의 상태정보를 확인하기 위해 활용한다. 특정 서비스제공자가 인증 요청에 명시한 RelayState 값은 식별정보제공자의 응답 메시지에 포함되어 해당 서비스제공자에게 반환된다.

V. 평 가

본 장에서는 쉬운 인증규약인 OIDC를 활용해 데이터과학 플랫폼인 JupyterHub와 토큰변환 시스템을 연동하고 웹 통합인증 플로우를 살펴봄으로써 개발한 시스템의 기능과 성능을 정성적으로 검증한다.

5.1 JupyterHub와 토큰변환 시스템의 OIDC 연동

JupyterHub와 토큰변환 시스템이 OIDC 인증규약을 이용해 연동될 수 있도록 클라이언트 측 OIDC 인증모듈인 OAuthenticator[26](KafeOAuthenticator)를 개발하고 JupyterHub에 추가했다.

OIDC 클라이언트인 JupyterHub가 토큰변환 시스템을 이용하기 위해서는 정책관리시스템에 JupyterHub를 등록하고 클라이언트 ID와 비밀번호를 확보해야 한다. JupyterHub가 이용할 OIDC Scope로 그림 4와 같이 openid, email, kafe.userinfo를 선택하고 정책관리시스템에 등록했다. OIDC Scope은 임의로 선택할 수 있다. 마지막으로 반환주소(callback_url)를 설정하고 client_id와 client_secret를 발급 받았다.

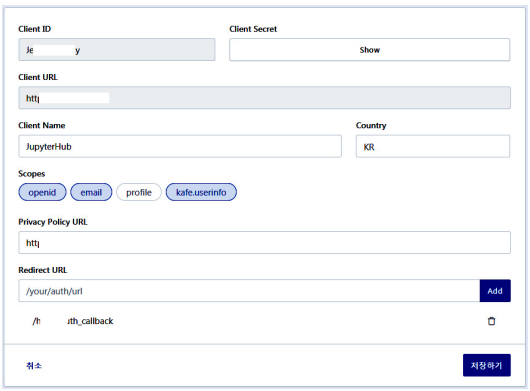

그림 4. 정책관리시스템에서 클라이언트의 수동등록
Fig. 4. Managed Client Registration on the Policy Management.

표 5. JupyterHub의 구동환경 설정(config.py)
Table 5. Configuration of JupyterHub(config.py)

```
c.KafeOAuthenticator.oauth_callback_url = '[url]'
c.KafeOAuthenticator.client_id = '[client id]'
c.KafeOAuthenticator.client_secret = '[client secret]'

from oauthenticator.kafe import KafeOAuthenticator
from jupyterhub.auth import LocalAuthenticator
from oauthenticator.kafe import *
c.KafeOAuthenticator.username_claim = 'email'

class LocalKafeOAuthenticator(LocalAuthenticator,
KafeOAuthenticator):
    def normalize_username(self, username):
        username = username.replace('@', '').lower()
        return username.replace('.','')
...,
```

마지막으로 정책관리시스템에서 획득한 정보를 이용해 표 5와 같이 JupyterHub의 구동환경을 설정했다. 검증을 위해 개발한 KafeOAuthenticator 클래스를 인증모듈로 활용하고 사용자를 구분할 고유식별자(username_claim)로 email 클레임을 이용했다. email 클레임에 포함된 특수문자는 모두 제거된 후에 고유식별자로 이용된다.

5.2 JupyterHub에 대한 사용자 권한의 관리

그림 5는 정책관리시스템에서 JupyterHub의 이용자격을 설정하는 화면이다. 사용자의 속성정보를 기반으로 이용자격을 설정한다. SAML 프록시는 식별정보제공자와 속성관리시스템으로부터 로그인에 성공한 사용자의 속성정보를 획득한다. JupyterHub에 대한 모든 사용자의 접근권한을 Allow로 설정했다. 다수의 속성정보를 조합해 접근권한을 통제할 수 있도록 개발함으로써 정책적용의 유연성을 높였다.

마지막으로 각각 1개씩의 SAML 식별정보제공자와 소셜 ID 제공자만 JupyterHub를 이용할 수 있도록 정책관리시스템에서 탐색서비스를 설정했다. 소셜 로그인과 식별정보제공자를 SAML의 문맥에 함께 수용하므로 정보자원에 대한 이용권한의 부여방식을 다양화하고 이용편의성을 높일 수 있다. 지면 관계상 탐색서비스를 설정하는 그림은 생략한다.

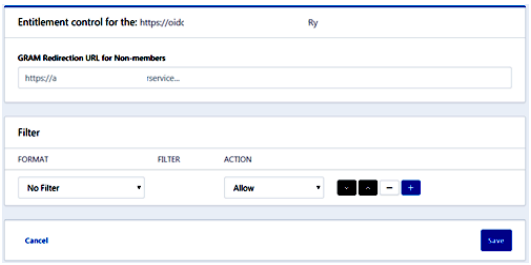

그림 5. 자격증명의 관리
Fig. 5. Entitlement management.

5.3 웹 통합인증 플로우의 처리 과정 검증

본 절에서는 JupyterHub에 접근이 허용된 식별정보제공자와 소셜 ID 제공자(Google™) 중에 식별정보제공자를 통해 웹 통합인증을 처리하는 과정을 살펴본다. 식별정보제공자를 선택해 검증함으로써 OIDC 클라이언트를 SAML의 문맥에서 처리할 수 있음을 확인한다.

그림 6은 사용자가 SAML 식별정보제공자를 이용해 OIDC 클라이언트인 JupyterHub에 로그인하는 과정을 보여준다. 사용자가 1) JupyterHub에 접근해 로

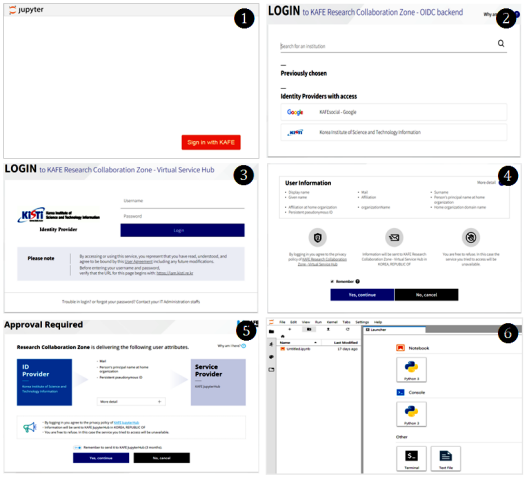

그림 6. 사용자 관점의 웹 통합인증 플로우
Fig. 6. Flow of user-perspective Web Single-Sign On.

그인을 시도하면 2) 토큰변환 시스템의 탐색서비스는 정책관리시스템에서 설정한 식별정보제공자와 소셜 ID 제공자의 목록을 보여준다. SAML 프록시는 OIDC 브로커의 요청으로 탐색서비스를 제공한다. 3) 사용자가 식별정보제공자를 선택하고 로그인에 성공하면 4) 식별정보제공자는 사용자 정보의 제3자 제공에 대한 동의를 받은 후에 속성정보를 토큰변환 시스템에게 전달한다. 토큰변환 시스템이 제3자에 해당된다. 5) 토큰변환 시스템의 SAML 프록시는 JupyterHub에게 속성정보를 제공하기 위해 사용자 동의를 추가적으로 받는다. 마지막으로 JupyterHub가 토큰을 확보하면 6) 사용자에게 정보자원의 이용권한을 부여한다.

5.4 사용자 동의 및 접근로그의 관리

법령 준수와 보안사고의 대응을 위해 정책관리시스템은 그림 7과 같이 웹 통합인증 플로우에 포함된 모든 사용자의 접근기록을 관리한다. 접근기록은 동의의 형태, 속성정보, 접근한 IP 주소와 사용자를 인증한 개체 및 날짜를 포함한다. 관리 편의성을 확보하기 위해

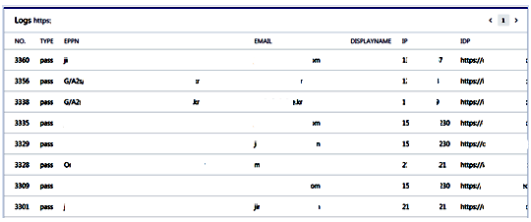

그림 7. 사용자 동의 로그의 관리
Fig. 7. Maintenance of Consent log.

서 종단 서비스에 대한 사용자의 동의기록이 초기화될 수 있도록 개발했다. 정책관리시스템은 종단 서비스에 대한 사용자의 동의 의사를 3개월 간 선택적으로 유지함으로써 재 동의로 인한 사용자 불편을 최소화시킨다.

5.5 처리 성능의 검증

마지막으로 그림 3의 각 단계(예, 그림 8의 축 '6'은 그림 3의 6-6' 단계)에서 소요된 시간을 측정해 그림 8과 같이 도시화했다. 측정된 시간은 프록시와 브로커 및 식별정보제공자에서 요청을 처리하는데 걸린 시간과 네트워크 지연을 포함한다. 총 10회 측정한 후 평균을 취했으며 사용자와 상호작용해야 하는 시간(예, 사용자 동의 등)은 제외했다. 웹 브라우저에서 각 서버에게 HTTP 요청을 보내고 응답을 받는데 까지 걸린 시간은 평균 10ms 이하였다.

로그인을 완료하는 데까지 걸린 시간(그림 8의 1부터 6까지 단계를 모두 완료)은 최소 0.761초이고 최대 2.108초였다. 평균은 약 1.395초가 걸렸다. 측정된 전체 시간은 다수의 웹 응답시간을 합한 결과이다. 예를 들어, 웹 통합인증의 처리과정은 전자적 동의와 같은 사용자 인터랙션(Interaction)을 포함하기 때문에 사용자가 실제로 느끼는 웹 응답시간은 측정된 값보다 훨씬 작아진다. OIDC 브로커(그림 8의 1, 5, 6 단계)에서 걸린 시간이 SAML 프록시에 비해서 상대적으로 높은 것을 확인할 수 있었다. 사용자가 인내할 수 있는 웹 응답시간(하나의 HTTP 요청에 대한 응답시간)이 최대 4초[27]이므로 토큰변환 시스템에서 발생한 시간지연이 사용자 경험에 미치는 영향이 크지 않을 것으로 판단된다.

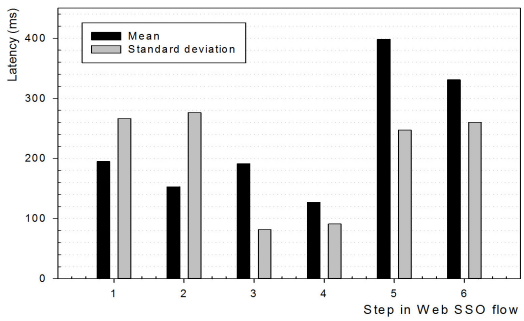

그림 8. 웹 통합인증의 각 과정에서 측정된 평균 지연
Fig. 8. Average latency taken in each step of SSO flow.

Ⅵ. 결 론

본 논문은 SAML 인증규약과 OIDC 인증규약을 선택적으로 이용해 웹 통합인증을 처리할 수 있는 토큰변환 시스템을 제안하고 데이터과학 플랫폼인 JupyterHub를 연동함으로써 개발한 시스템의 성능을 정성적으로 분석했다. 향후 OIDC 토큰의 생명주기 관리와 지원하는 통합인증 플로우의 추가 및 개념설계 중인 속성관리시스템의 프로덕션화 개발을 수행할 예정이다. 또한, 속성관리시스템과 정보자원의 연동 기술을 추가적으로 개발할 계획이다.

References

[1] J. Jo, H. Jang, K. Kong, and Y. Chae, "Federated IAM service of KAFE identity federation," *J. KICS*, vol. 43, no. 12, pp. 2200-2214, Dec. 2018.

[2] OASIS Security Services Technical Committee, "*Security Assertion Markup Language (SAML) Version 2.0 Specification Set*," OASIS Standard, Retrieved Jul., 30, 2019 from http://docs.oasisopen.org/security/saml/v2.0/saml-2.0-os.zip.

[3] *InCommon*, Retrieved Aug. 1, 2019 from https://www.incommon.org/.

[4] *UK Access Management Federation*, Retrieved Aug. 1, 2019 from https://www.ukfederation.org.uk/.

[5] *Korean Access Federation*, Retrieved Aug. 1, 2019 from https://www.kafe.or.kr/.

[6] N. Sakimura, J. Bradley, M. jones, B. de Medeiros, and C. Mortimore, "OpenID connect core 1.0 incorporating errata set 1," *The OpenID Foundation, specification*, 2014.

[7] EU Research & Innovation, "*Deliverable DSA1.1: Results on Pilots with Communities part 1*," May 2018.

[8] *CORBEL: Shared services for life-science*, Retrieved Jul., 30, 2019 from https://www.corbel-project.eu/.

[9] G. Pareschi, et al., "Status of the technologies for the production of the Cherenkov Telescope Array (CTA) mirrors," *Optics for EUV, X-Ray, and Gamma-Ray Astronomy VI*, International Society for Optics and Photonics, vol. 8861, p. 886103, Sep. 2013.

[10] *SATOSA*, Retrieved Jul. 30, 2019 from https://github.com/IdentityPython/SATOSA/wiki.

[11] J. Basney, T. Fleury, and J. Gaynor, "CILogon: A federated X.509 certification authority for cyberinfrastructure logon," *Concurrency and Computation: Practice and Experience*, vol. 26, no. 13, pp. 2225-2239, 2014.

[12] *Check-in BETA: Login with your own credentials*, Retrieved Jul. 30, 2019 from https://www.egi.eu/services/check-in/.

[13] *WLCG: Worldwide LHC Computing Grid*, Retrieved Aug., 1, 2019 from http://wlcg-public.web.cern.ch/.

[14] EISCAT3D project website, Retrieved Aug., 1, 2019 from https://www.eiscat.se/eiscat3d/.

[15] F. Perez and B. E. Granger, *Project jupyter: Computational narratives as the engine of collaborative data science*, Tech. Rep., 2015. Retrieved Aug., 1, 2019 from https://blog.jupyter.org/project-jupyter-computational-narratives-as-the-engine-of-collaborative-data-science-2b5fb94c3c58.

[16] R. Hedberg, M. Jones, S. Solberg, S. Gulliksson, and J. Bradley, "OpenID Connect federation 1.0 – draft 08," Retrieved Jul., 30, 2019 from https://openid.net/specs/openid-connect-federation-1_0.html.

[17] *Shibboleth*, Retrieved Jul. 30, 2019 from https://wiki.shibboleth.net.

[18] N. Naik and P. Jenkins, "An analysis of open standard identity protocols in cloud computing security paradigm," in *Proc. IEEE Int. Conf. on Dependable, Autonomic and Secure Comput., on Pervasive Intell. and Comput., on Big Data Intell. and Comput. and Cyber Sci. and Technol. Congr.*, pp. 428-431, 2016.

[19] A. Biancini, L. Florio, M. Haase, M. Hardt, M. Jankowski, J. Jensen, C. Kanellopoulos, N. Liampotis, S. Licehammer, S. Memon, N. van Dijk, S. Paetow, M. Prochazka, M. Salle, P. Solagna, U. Stevanovic, and D. Vaghetti,

"AARC: First draft of the blueprint architecture for authentication and authorisation infrastructures," *CoRR*, vol. abs/1611.07832, 2016.

[20] *AARC: Authentication and Authorisation for Research Collaborations*, Retrieved Jul. 30, 2019 from https://aarc-project.eu/.

[21] Ministry of Public Administration and Security, *Technical specification of SSO (Single-Sign On) gateway*, 2017.

[22] A. Lonut and C. Nisipasiu, "Web single sign-on implementation using the simpleSAMLphp application," *J. Mob., Embedded and Distrib. Syst.*, vol. 3, no. 1, pp. 21-29, 2011.

[23] *OpenID Connect (OAuth2) Client Library,* Retrieved Aug. 1, 2019 from https://github.com/ivan-novakov/php-openid-connect-client.

[24] *Chromium Blog: Improving privacy and security on the web,* Retrieved Aug. 1, 2019 from https://blog.chromium.org/2019/05/improving-privacy-and-security-on-web.html.

[25] M. A. Christie, A. Bhandar, S. Nakandala, S. Marru, E. Abeysinghe, S. Pamidighantam, and M. E. Pierce, "Using keycloak for gateway authentication and authorization," Presented at *Gateway 2017*, University of Michigan, Ann Arbor, MI, Oct. 2017.

[26] *KAFE OAuthenticator for JupyterHub,* Retrieved Aug. 1, 2019 from https://git.kreonet.net/kafe-private/oauthenticator.git.

[27] C. Lorentzen, M. Fiedler, H. Johnson, J. Shaikh, and J. Ivar, "On user perception of web login - a study on QoE in the context of security," in *Proc. Australian Telecommun. Netw. and Appl. Conf.*, Auckland, New Zealand, pp. 84-89, 2010.

조 진 용 (Jinyong Jo)

2003년 : 광주과학기술원 정보통신공학과 석사
2013년 : 광주과학기술원 정보통신공학과 박사
2003년~현재 : 한국과학기술정보연구원
2016년~현재 : eduGAIN 운영그룹 위원

<관심분야> 연합인증

[ORCID:0000-0001-6830-3604]

채 영 훈 (YeongHun Chae)

2015년 : 고려대학교 전자 및 정보공학과 학사
2017년 : 과학기술연합대학원대학교 빅데이터과학 석사
2017년~현재 : 한국과학기술정보연구원

<관심분야> 딥러닝, 연합인증

[ORCID:0000-0002-6860-7533]

공 정 욱 (JongUk Kong)

1998년 : 포항공과대학교 석사
2015년 : 충남대학교 정보통신공학과 박사
2002년~현재 : 한국과학기술정보연구원

<관심분야> 네트워크 자원제어, 사용자정의 네트워킹

[ORCID:0000-0002-8703-2798]