electronics

MDPI

Article

Federated-Access Management System and Videoconferencing Applications: Results from a Pilot Service during COVID-19 Pandemic

Jinyong Jo [1,*], Yeonghun Chae [2], Heejin Jang [1] and JongUk Kong [1]

[1] Korea Institute of Science and Technology Information, 245 Daehak-ro, Yuseong-gu, Daejeon 34141, Korea; jhj@kisti.re.kr (H.J.); kju@kisti.re.kr (J.K.)
[2] SEASON Technology, Sejong 30128, Korea; proin@proinlab.com
* Correspondence: jiny92@kisti.re.kr; Tel.: +82-42-869-0585

Abstract: Videoconferencing has become a crucial enabler for sustainable collaboration and learning during the COVID-19 pandemic. However, national regulations often restrict public institutions from introducing commercial videoconferencing services. Open-source software is an attractive option for institutions if it can be protected from potential security threats while ensuring high usability. Unfortunately, to the best of our knowledge, we hardly find available open-source videoconferencing applications in the literature that stress their usability and adopt security-related frameworks. This study presents a federated-access management system called trustHub, which was developed to enable flexible and elaborate access control and protocol-agnostic user authentication. In addition, we introduce two videoconferencing applications that aim to improve the usability of leveraged open-source software. They are prototyped to operate in concert with trustHub to take firm access control and accept various types of identity providers. Consequently, using data collected from trustHub and a prototyped videoconferencing application over a 10-month period, we conduct a comprehensive analysis to understand the usage patterns of federated access and videoconferencing during the pandemic and, thus, verify their feasibility indirectly.

Keywords: identity and access management; videoconferences; authentication and authorization; COVID-19

Citation: Jo, J.; Chae, Y.; Jang, H.; Kong, J. Federated-Access Management System and Videoconferencing Applications: Results from a Pilot Service during COVID-19 Pandemic. *Electronics* **2021**, *10*, 2239. https://doi.org/10.3390/electronics10182239

Academic Editor: George Angelos Papadopoulos

Received: 13 August 2021
Accepted: 10 September 2021
Published: 12 September 2021

1. Introduction

Videoconferencing (VC) has been a critical player for sustainable research collaboration following the emergence of COVID-19 [1–7]. Numerous offline activities have shifted to the online space during the pandemic. Recently, various concerns, such as threat to the security and protection of users' personal information, have been raised. Notably, the interest of users of VC in these threats has seen a significant uptick [8,9]. Identity and access management (IAM) is the first step in improving the security and privacy of users and realizing online collaboration, including the switching of learning activities to online. However, "siloed" identity management (i.e., a stand-alone identity management for an application service) frequently worsens the ease of use and increases management burden on the service providers (SPs). For example, based on analysis of our identity management system, approximately 86% of users did not remember their `userid`, whereas 7% recalled their `userid` but did not recollect their password. Such instances lead to frequent help-desk calls.

There are numerous commercial VC services in the world. However, because of data sovereignty or privacy-related regulations, research institutions and universities in Korea hardly introduce commercial VC services. Open-source VC software [10–13] can provide considerable benefits for them. Unfortunately, at the time of our VC development, we had no available open-source VC software which provides high usability (e.g., room reservation)

while ensuring federated access management and strong VC session management. The importance of those features is growing as cybersecurity and usability become challenges in VC applications [14,15]. As for federated access management, a centrally configurable access management system would be adequate for timely provisioning user access to multiple SPs, including VC applications. In addition, the necessity of a system supporting protocol-agnostic authentication would increase because heterogeneous user authentication standards adopted in SPs can cause interoperability challenges [16]. However, to the best of our knowledge, existing development projects [17,18] overlooked elaborate access control or interoperable authentication among different standards. Details of them are hardly found in the literature.

In this study, two VC applications designed for users in higher education and research institutions in Korea during the COVID-19 pandemic are introduced. Both applications are targeting the improvement of usability. In addition, we detail our federated-access management system (an authentication and authorization (AA) infrastructure), called trustHub, developed for ensuring easy access control to the subjects, user groups, and identity providers. trustHub works in concert with the implemented VC applications for the timely control of user access. Subsequently, a comprehensive analysis to identify the usage patterns of user identities and VC associated with the pandemic is conducted. A detailed analysis on how our identities and VC are used online allows the evolution of identity solutions and VC applications. It should be noted that one of the crucial goals of this study is not to propose new technical concepts but to swiftly advance and apply ideas that have already been explored and validated in similar research projects worldwide [10–13,17,19–21].

We collected data regarding the VC and identity usage of Korean users from the developed VC applications and trustHub for a 10-month period from March to December 2020. Specifically, we examined the usage patterns of VC, the correlation between VC usage and COVID-19 confirmed cases, the usefulness of federated IAM, and the expected capacity of a VC server. We found that the usefulness of institutional identity providers (IdPs) is approximately 87% higher than that of social IdPs. The change in the number of COVID-19 confirmed cases affected the number of VC meetings created more than the number of users who participated in the meetings. In addition, there is a linear relationship of moderate strength between the number of confirmed cases counted for more than a 5-day period and the number of meeting rooms created over the next 5 days. In general, one meeting room holds less than 68.4 participants with a 0.95 probability, and at best 20 of them turn on their webcams. Finally, one VC server is expected to accommodate 22 concurrent meetings, and 58% of meetings are estimated to be lectures. The main contributions of this study are summarized as follows.

- To the best of our knowledge, this is the first study investigating the usage patterns of VC and user identities during the COVID-19 pandemic. We analyze the correlation between the usage patterns and COVID-19 confirmed cases;
- Based on actual usage data collected over a 10-month period, we conduct a comprehensive analysis to derive the findings mentioned above. These findings will provide more authenticity to VC applications and AA infrastructure when creating a design or model;
- Finally, we introduce one AA service and two VC services developed based on operator and user feedback. The feedback reflects the security and usability requirements of higher education and research institutions, which can be used for improving similar services worldwide.

The remainder of this paper is organized as follows. After reviewing related studies in Section 2, we describe the motivation of our approach in Section 3. Next, we detail the trustHub and VC applications in Sections 4 and 5. Subsequently, we present the results of the analysis in Section 6. The limitations of this study and areas of future research are discussed in Section 7. Finally, conclusions are given in Section 8.

2. Related Work

Many research and development projects regarding AA infrastructure have been conducted [17,19–21]. In addition, several types open-source software are being developed for use in VC [10–13]. This study focuses solely on the AA infrastructure and software that are mostly similar to our research. The aim of this study is not to establish the superiority of our research but to introduce the developed infrastructure and services and present an analysis of their usage patterns during the COVID-19 pandemic.

First, for the development of trustHub we were motivated by CILogon [17]. One of our goals was to provide a feature-rich central access management system to domestic SPs, whose staff members generally have poor knowledge on maintaining or adopting federated IAM. CILogon built the open-source COmanage [22] and Shibboleth software, provides X.509 certification authority, an OpenID Connect (OIDC) provider, and security assertion markup language (SAML) attribute authority. COmanage is a software platform for managing a collaborative organization (CO) or virtual organization (VO [23]). We believe CILogon and trustHub have similar architectures because both comply with the authentication and authorization for research and collaboration (AARC) blueprint architecture [24,25]. The detailed architecture in this case is hidden.

As the main difference between trustHub and CILogon, the former does not act as an X.509 certificate authority. We rarely use X.509 certificates in science and education sectors in Korea. In addition, trustHub provides fine-grained access controls to subjects, groups, and IdPs. Instead of using COmanage, we developed a new group manager that is simpler to apply than COmanage and yet supports the widely used functions of COmanage (e.g., identity linking, CO management, and the lightweight directory access protocol (LDAP)).

From an access-control perspective, trustHub has a similar design principle as the SURF Science Collaboration Zone (which recently evolved into the SURF Research Access Management (SRAM [18])) in that it allows users to easily manage the access to their research resources. SRAM consists of an identity proxy, an LDAP interface for non-web applications, and a membership management system. It also implements the AARC blueprint architecture. However, to the best of our knowledge, details in the literature are scarce, and thus we omit the comparison with trustHub.

The pandemic entailed the increased use of VC for education or collaboration purposes. Although there are numerous commercial VC software [1], in this section, we only introduce two open-source applications selected based on three criteria: free and open-source, the use of web real-time communication (WebRTC), and the applicability of open-standard authentication protocols or specifications. We must address the fact that both Webmeet [26] and Webinar [27], which we developed, rely heavily on open-source VC software, which will be subsequently introduced. Webmeet and Webinar are front-end interfaces for VC software (herein, noted as VC server unless otherwise mentioned).

BigBlueButton [12] mainly consists of Freeswitch [28] for dealing with audio, Kurento [29] for streaming multimedia, and HTML5 clients and servers to provide a graphical user interface. It supports selective forwarding unit (SFU)-based WebRTC streaming, which receives all audio and video streams and selectively forwards them to the HTML5 clients. Its HTML5 user interface is more suitable for 1:N lecture-type VC than N:N meeting-type VC. Webinar exploits BigBlueButton as a back-end VC server. Greenlight is the front-end interface of BigBlueButton, enabling open-standard authentication, e.g., the OAuth2/OIDC protocol. There are some differences between Greenlight and Webinar. The latter underlines the proactive roles of service managers, such as room-stat monitoring, the banning of VC rooms, and user-event traces. Webinar can protect meeting rooms with solid passwords and alert users if the recording of a VC session is initiated. As another difference, Webinar supports the SAML specifications, whereas Greenlight supports OIDC.

Jitsi Meet [11] is another open-source VC software that is used as the back-end VC server of Webmeet. It consists of Videobridge to route video streams, conference focus (jicofo) to manage media sessions, and a Prosody extensible messaging and presence protocol (XMPP) server [30] for signaling. Similar to BigBlueButton, it forwards media packets

based on an SFU webRTC. Videobridge in Jitsi Meet is a better lightweight forwarding solution than Kurento in BigBlueButton, e.g., in our experience, the former utilizes much fewer CPU resources. However, Jitsi Meet provides an extremely simple front-end interface allowing only meeting creation. Webmeet has additional features compared with Webinar, such as meeting reservation.

Analyses of the required VC features allowing continuous education under the pandemic and case studies of the VC services employed in the eHealth or telemedicine sectors was recently conducted [1,4–7]. However, such analyses have focused on reviewing the education activities shifted to online mode, whereas this study examines the usage patterns of user identities and VC, as well as their correlation with COVID-19 confirmed cases.

3. History and Data Collection

A trustHub has been developed under a 3-year plan since 2018. The COVID-19 pandemic started in early 2020, when the third year of the development phase began. The demand for VC services increased significantly immediately after the outbreak. At that time, we were operating a VC service based on BigBlueButton; however, the service was about to be stopped because of the expected imminent blocking of flash players on Chrome. Old versions of BigBlueButton had relied on a flash player. Therefore, we upgraded BigBlueButton and implemented Webinar. Subsequently, we applied trustHub to Webinar to enable social networking authentication, which allows more users to access the VC service. Shortly after launching the service as a pilot, security and privacy threats of Zoom [9] emerged. In addition, user feedback related to the usability of Webinar was provided. The threats motivated us to reinforce trustHub to ensure elaborate access control to application services connected to trustHub. We enhanced trustHub and applied it to another VC application called Webmeet, which was developed based on user feedback.

Data collection is essential for analyzing the usage patterns of user identities and VC. First, we collected COVID-19 infection data [31] provided by the Ministry of Health and Welfare, Korea. After refining the data (e.g., removing duplicates or correcting errors), we obtained the number of daily COVID-19 confirmed cases. One of our aims was to investigate which types of IdPs (e.g., social or institutional IdPs) are preferred more by research and education sectors experiencing the pandemic. Because trustHub and VC applications record all login and user-consent events for tracking users, we can acquire login-associated data (e.g., IdPs or SPs that accessed by users, user identities, and user-consent status) from such records.

We leveraged VC usage data (e.g., room id, creation time, and the maximum number of users) stacked in Webinar. However, the data available were extremely limited because Webinar is designed to save only essential metrics for service provisioning and management. For example, from the stored data only, it was impossible to explore some usage statistics, such as the number of webcams turned on per meeting or the average duration of the meetings. When there were no related data in Webinar, we estimated the stats based on 1-year event-history records (e.g., bandwidth consumption) in our network monitoring system. We also complementarily utilized the log files created by BigBlueButton. Finally, an analysis was conducted using Python programming packages, such as scikit-learn, pandas, numpy, and scipy.

4. Federated-Access Management System

4.1. Functional Overview of trustHub

trustHub is built on two open-standard AA frameworks, OIDC and SAML. Figure 1 shows a high-level overview of trustHub, which is composed of four functional subsystems: group manager, SAML proxy, hub manager, and two brokers. In this study, we refer to a broker as a system that conducts a token translation between different AA frameworks (e.g., converting an SAML assertion into an OIDC token or vice versa). A proxy is a system that relays AA messages that request user authentication or assert the user's identity. The proxy complies with the SAML specifications.

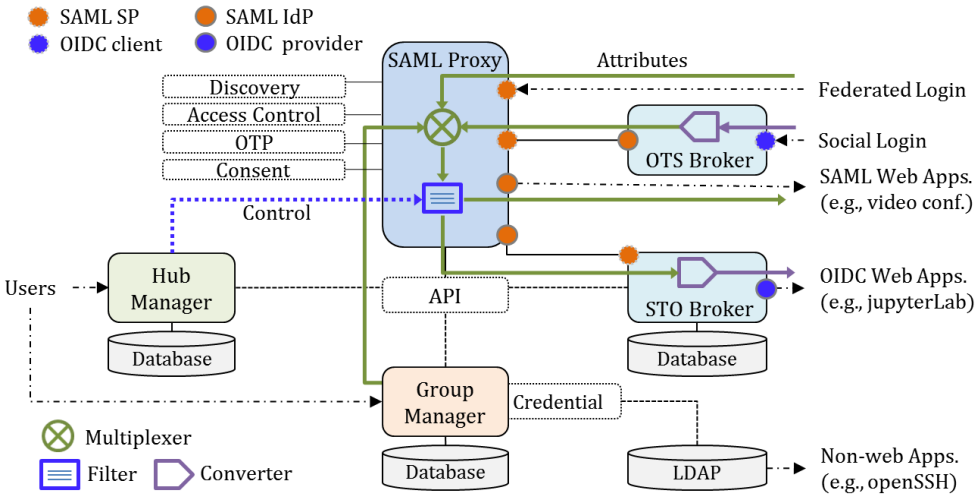


Figure 1. High-level overview of the developed federated-assess management system.

The roles and necessities of each subsystem are as follows:

- **Group Manager**: A role of the group manager is to manage VOs, including user groups. If the members of a research community share a unique group identifier, application services, such as NextCloud [32] or OpenStack [33], can easily manage their storage or computing resources on a group basis. The group identifier can represent the set of users allowed or denied to access a particular resource, such as a file.
  The second role of the group manager is to administer third-party (TP) attributes registered by the users or superusers of the group manager. Occasionally, SPs require user or system attributes that IdPs do not provide. We call such attributes TP attributes because TP attribute authorities provide them. The group identifier is a TP attribute. Note that the group manager also keeps a subset of the home organization (HO) attributes of each user authenticated from his or her home (or institutional) IdP.
  Finally, the group manager provides service interfaces for the SAML proxy to allow the proxy to take the TP attributes of the user attempting to be authorized by a relying party (i.e., an SAML SP or an OIDC client). In addition, the service interface is used for user management in LDAP to enable openSSH authentication;
- **SAML Proxy**: This is a crucial component of trustHub, and its first role is to enable an SP to communicate with multiple IdPs. There are many types of SAML SP softwares that only communicate with a single IdP. Without the SAML proxy, an SP implemented with such a software cannot accept users enrolled in different IdPs.
  The second role is to serve as a central policy enforcement point. Access control of the message flows becomes straightforward if all authentication request and response messages go through a central point, i.e., the SAML proxy. Each microservice (e.g., IdP discovery, access control, two-factor authentication, or user consent) enforces its AA policy based on the HO/TP attributes and rule sets collected from an IdP and other subsystems (i.e., hub manager, group manager, and broker). Thus, the collection of attributes and rule sets is another important role of the SAML proxy.
  Finally, the SAML proxy enforces an attribute filtering based on a rule set defined in the hub manager, i.e., it only posts a predefined set of HO and TP attributes to an associated relying party;
- **Hub Manager**: This is the policy decision point of trustHub. Its primary role is to keep some of the rule sets used by microservices in the SAML proxy. For example, in the hub manager, service owners can define access-control rule sets for their relying parties. The SAML proxy then retrieves the associated rule sets from the hub manager and enforces access control as it receives authentication messages to the service. It also keeps the attribute-filtering rule set of each relying party.

Another vital role of the hub manager is to enable service owners to register their OIDC clients. Registered client information is used by the STO broker. The STO broker, in part, is an OIDC provider, and it communicates only with registered OIDC clients. Thus, the registration is mandatory for processing authentication messages for OIDC clients.
In addition, the hub manager governs the intra- and inter-federation (i.e., eduGAIN [34]) metadata of SAML entities (i.e., SAML IdPs or SPs), and maintains the history of user events, such as consent records;

- **OTS/STO (OIDC to SAML/SAML to OIDC) Broker**: Many science applications, such as Jupyter [35] and Globus [36], are beginning to support the OIDC protocol. In addition, some of them accept social IdPs (e.g., Google) as the last IdP option. Social IdPs adopt the OIDC protocol and are frequently used for less security-sensitive applications. Unfortunately, current federated IAM for research institutions and universities generally leverage the SAML specification, which leads to a concern regarding the interoperability between SAML entities and OIDC entities (i.e., OIDC providers or clients).
  The brokers ensure interoperability between the two standard frameworks. More specifically, it enables SAML-based federated IAM to work with the OIDC entities. Each broker has one SAML entity and one OIDC entity, and one entity acts as a database providing user attributes or claims against the other entity. For example, the front-end entity of the OTS broker (i.e., an SAML IdP) leverages the user claims obtained from its back-end entity (i.e., an OIDC client) to build an authentication response message. The STO broker operates in a similar way as the OTS broker, except that the front-end is an OIDC provider and the back-end is an SAML SP.

4.2. Subsystems for Authentication and Attribute Management

First, service owners have to register their relying parties to the hub manager to control the authentication flows of their users. The owners govern their relying parties by themselves under supervision of the trustHub administrators. We disabled trustHub from accepting dynamic client registration owing to security concerns. Thus, all OIDC clients have to be registered to the hub manager. The OIDC client information required for registration includes the client name, callback or redirect URLs, and the OIDC scopes including claims. The claims are equivalent to SAML attributes, and the OIDC scope is a string that represents a set of claims. When a service owner registers an OIDC client, the hub manager returns `client_id` and `client_secret` for the client. Each client authenticates itself to trustHub with its `client_id` and `client_secret`. Using a RESTful application programming interface (API), the hub manager saves the client information into the database of the STO broker, which is an OIDC provider communicating with the clients.

The service owner has to create a VO and a service interface for the relying party in the group manager if the party requires TP attributes. Basically, the group manager is composed of three functional blocks, namely, user-profile management (M_u), VO management (M_v), and service-interface management (M_s).

The M_u block enables users to register their identity attributes and credentials (e.g., password and SSH key). The credentials are used for authenticating the users accessing application services registered in the M_s block. We designed the group manager such that the superuser can flexibly define the types and values of the identity attributes that the M_u block collects. Because users must log into the group manager using their institutional IdP, their user identifier in the hub manager becomes synchronized with that in the institutional IdP. If users have multiple user identifiers (e.g., authenticated from multiple IdPs), they can select one user identifier as a primary because the M_u block provides an identity linking service.

Users can create VOs at the M_v block if a superuser of the group manager approves. Each VO can have multiple groups; however, we limited the depth of each group to one to

reduce the complexity of the implementation. Each VO manager designates the leaders of his or her groups for autonomous management. After a VO is created, the VO manager or the group leaders have to enroll users into their VO or groups. The group manager provides three enrollment flows: an email invitation, an assignment by each manager or leader, and manager or leader approvals after user applications. Each VO or group has a unique identifier representing itself and uses a set of user attributes for the enrolled users. For example, every user in a group can have a common group identifier (e.g., `isMemberOf`). The set of attributes used by the VO or group is fully configurable.

As noted, a service interface (I_s) has to be created if an application service or a relying party needs TP attributes. The M_s block provides I_s for relying parties or openSSH servers. We have to address the fact that the TP attributes consist of the user attributes and system attributes. In addition, I_s maintains the system attributes (e.g., `gidNumber` and `uidNumber` for openSSH), and M_u and M_v maintain the user attributes. Because I_s links more than one VO, it knows the attributes of each user enrolled in the linked VOs. Consequently, the group manager collects TP attributes (A_m) as $A_m = A_m^u \cup A_m^v \cup A_m^s$, where A_m^u is the user attributes from M_u, A_m^v is the user attributes from M_v, and A_m^s is the system attributes from M_s or I_s.

Each I_s has its unique identifier, which is equivalent to that of the associated relying party (e.g., SAML `entityID` or OIDC `client_id`). Thus, all subsystems of trustHub can query the TP attributes of a user to I_s as long as they know the identifier of the relying party. The key string for identifying a user is `eduPersonPrincipalName`, which is an SAML user attribute.

4.3. Subsystems for Authentication Processing

The two subsystems mentioned above do not directly engage in controlling the authentication flows. Authentication request and response messages are delivered only to the SAML proxy, and to the two brokers if OIDC entities are involved. For example, if a user wants to access the jupyterLab in Figure 1 with a social login, the two brokers have to engage in the user authentication. Note that it is also possible for the user to be authenticated with the federated login if the user is enrolled in an institutional IdP. The federated login in the figure signifies the user authentication with a guest or an institutional IdP. The guest IdP can be seen as an IdP holding users from multiple organizations, whereas the institutional IdP keeps users who are affiliated with a single institution.

We exemplify the jupyterLab user illustrated above to explain how the brokers are implemented. An authentication request from a user agent (e.g., a web browser) is sequentially delivered to the STO broker, SAML proxy, and OTS broker and is finally handled by a social IdP. An associated response is delivered in reverse order. The back-end of the STO broker is an SAML SP, and it can then communicate with the front-end of the SAML proxy (i.e., SAML IdP). Similarly, the back-end of the SAML proxy is an SAML SP, and it works with the front-end of the OTS broker, which is an SAML IdP. The back-end of the OTS broker is an OIDC client. We implemented the OTS broker using simpleSAMLphp and the STO broker using SATOSA [37]. We configured the STO broker to support only an authorization code flow [38] out of three OIDC authentication flows.

If a user successfully logs into an IdP, the user's his or her HO attributes are posted to the SAML proxy. The proxy is a policy enforcement point utilizing the HO/TP attributes and the rule sets collected from other subsystems. To allow the proxy to obtain the rule sets and other auxiliary information, we implemented tiny RESTful API services to each subsystem. In addition to the RESTful API, the proxy can query the TP attributes of each user through a simple object access protocol (SOAP). The SAML proxy was implemented using simpleSAMLphp [39].

The policy enforcement is conducted by several microservices modularized in the proxy. These include IdP discovery, access control, second-factor authentication (2FA), and user consent. We implemented these microservices to enhance the ease of use, security, and legal compliance. First, IdP discovery enables users to choose their IdPs out of the allowed

IdPs. In the hub manager, service owners can limit the lists of IdPs (i.e., the allowed IdPs) that are accessible to their relying parties. Next, we leveraged Google authenticator [40] for the 2FA that works on a relying-party basis. The service owner can optionally activate the 2FA for the relying party. Third, the user-consent microservice takes online consent for the provisioning of personal information and records who is consenting what. We cover the access control service in the following subsection.

Whenever the SAML proxy receives an authentication response for a user, it takes the TP attributes of the user from the group manager and merges them with the HO attributes of the user. Subsequently, the merged attributes are passed to an associated relying party.

Upon receiving the SAML attributes, the STO broker converts them into the OIDC claims. The naming convention of the user attributes (e.g., mail) in the SAML context is different from that in the OIDC protocol. We only partially followed the naming convention recommended by the Research and Education Federations (REFEDS) group [41] because the naming convention that was used before the recommendation had to be maintained. Table 1 shows some examples of the mapping convention applied. Detailed descriptions of the attributes and claims are omitted for the sake of brevity.

Table 1. Examples of mapping SAML attributes to OIDC claims.

OIDC Scope	OIDC Claim	SAML Attribute
sub	sub	eduPersonTargetedID
profile	name	displayName
email	email	mail
userinfo	ismemberof	isMemberOf
	eppn	eduPersonPrincipalName

In addition, the STO broker converts the name of the authentication policies (e.g., authentication context class reference (`acr`)). The OIDC `acr` claim (called `AuthnContextClassRef` in an SAML context) is used to determine the level of assurance of an authentication request [42]. For example, an IdP can assert the use of a multi-factor authentication (MFA) by inserting the `AuthnContextClassRef` element into SAML messages. Relying parties receiving the messages can assess whether the IdP authenticating the user applied an MFA.

We implemented the SAML proxy so that it adds the `AuthnContextClassRef` element to the SAML response messages if the 2FA provided by the proxy is activated for associated relying parties. The response messages are then passed to an SP or the STO broker. Subsequently, if the broker takes the messages, its token endpoint inserts the `acr` claim into the ID token as it issues the token. The REFEDS MFA profile [43] is adopted for the `acr` or the `AuthnContextClassRef`.

4.4. Access Control and Attribute Filter

We designed trustHub to federate various types of IdPs (e.g., social, guest, and institutional providers), potentially leading to security threats or a misuse of the application services. The low assurance level related to user identities makes it difficult to track users if security incidents occur. Furthermore, if one user has multiple user accounts, they can abuse the system resources of the application services. Thus, trustHub needs to support fine-grained access control to federated relying parties. As noted, in the hub manager, service owners can define the access-control rules of their relying parties. The SAML proxy fetches the stored rules and conducts access control for the authentication flows traversing it.

The access control applied in trustHub is described in the following. Service owners can enforce access control to their relying parties based on the subjects, groups, or IdPs. For example, a service owner may want to only allow students in one university among all users in an identity federation to access the owner's application services. There are two

approaches that trustHub applies to control user access to the relying parties: manipulating the IdP list shown in the discovery service or providing access control based on the user attributes.

Access control based on the discovery service operates as follows. A service owner configures an access-control rule for the relying party, such that $F_I^i : I \rightarrow I_i$, $(I_i \subset I)$, where F_I^i is a function for filtering the IdPs, I denotes all IdPs in a single federation, and I_i denotes the IdPs allowed to access the relying party i. We should note again that the configured rule in the hub manager is enforced by the SAML proxy.

For attribute-based control, the owner sets a rule (R) consisting of n conditions (C), such that R_D or $R_A : \{C_1 \oplus C_2 \oplus ... \oplus C_{n-1} \oplus C_n\}$, where the operator $\oplus$ includes $\cap$ and $\cup$. A deny rule is R_D, and an allow rule is R_A. For example, for a relying party i, a rule for allowing access can be set as $R_A^i : \{(a_1 = *a.univ) \cap (a_2 = stud*)\}$, which implies that the users are allowed if the values of their attributes a_1 and a_2 contain strings ending with *a.univ* and starting with *stud*, respectively. The owner can use group identifiers as attribute values to achieve a group-based access control.

Attributes used for such access controls are collected from an IdP and the group manager. Note that the group manager also functions as a standalone attribute authority that is not subjected to trustHub. The TP attributes of a user have to be filtered, allowing an associated relying party to take only the required attributes. In the service interface, an attribute filter can be defined as $F_G^i : A_m \rightarrow A_i^m$, $(A_i^m \subset A_m)$ for a relying party i, where A_m denotes all TP attributes of the user (i.e., $A_m = A_m^u \cup A_m^v \cup A_m^s$) and A_i^m is filtered the TP attributes for the relying party. Thus, the relying party takes only A_i^m. The SAML proxy is also a relying party.

Because the SAML proxy intervenes an authentication flow, it collects the $A_p^u \cup A_i^m$ attributes of the user, where A_p^u is a set of user attributes posted by an IdP. The SAML proxy needs another attribute filter because A_p^u contains more attributes than the relying party of an application service (e.g., jupyterLab) requires. IdPs connected to trustHub do not know what user attributes are required by each application service, and are only aware of the user attributes required by the proxy. The attribute filter is defined as $F_s^i : (A_p^u \cup A_i^m) \rightarrow A_i^s$, $(A_i^s \subset (A_p^u \cup A_i^m))$. Relying party i consumes the filtered attributes (A_i^s). If the STO broker is the relying party, it uses the filtered attributes to compose the OIDC scopes and claims for the OIDC client that initiated the authentication flow.

4.5. Example Authentication Flow

To summarize this section, we exemplified how trustHub and other entities handle an authentication flow of a user. Figure 2 shows an authentication message flow engaged by trustHub. The flow does not include any interactions with the hub manager or the group manager. We assume that a relying party is an OIDC client, and an SAML IdP authenticates the user. In addition, we presume that the relying party already obtained its `client_id` and `client_secret` from the hub manager. Note again that the STO broker supports only the authorization code flow [38].

The authentication flow begins as a user agent (e.g., a web browser) requests an OIDC authorization code to the authorization endpoint of the STO broker. The OIDC client needs the authorization code to obtain an ID token and an access token from the STO broker (see steps 6 and 6′ in Figure 2). Because the user is not authenticated yet, the STO broker has to redirect an authentication request to the SAML proxy. Similarly, the SAML proxy redirects the request to the IdP that the user selected in the discovery service because it also has no authentication session (i.e., a session created for an authenticated user) for the user. We designed the SAML proxy to not keep the authentication sessions of the users because a single logout problem (e.g., an explicit logout is required from both the proxy and the IdP) occurs if the proxy maintains the sessions.

If the IdP successfully authenticates the user, an SAML response message is returned to the SAML proxy and then to the STO broker. If the proxy blocks the user based on an access-control rule, it stops the authentication flow. Otherwise, the STO broker receives the

SAML message and obtains the authentication information, including the user attributes, from the message. Subsequently, the STO broker sends an authorization code to the agent.

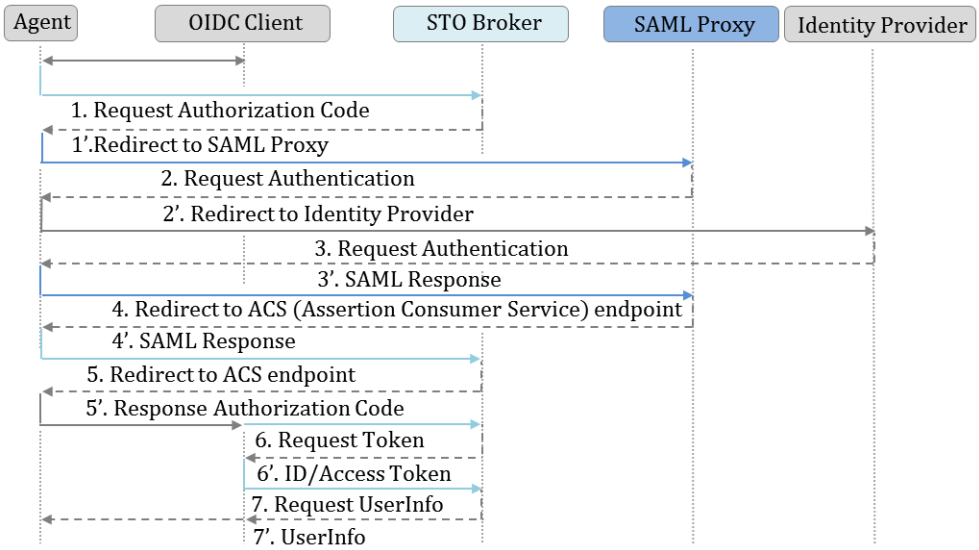


Figure 2. Authentication message flows.

The OIDC client uses the authorization code to obtain an ID token and an access token from the STO broker. The ID token contains a piece of authentication information (e.g., an issuer and a subject identifier, the authentication time, and `acr`). The access token is required to access the `userinfo` endpoint of the STO broker, which provides the claims of the user. The subsequent steps are the same as the authentication code flow of the OIDC protocol [38]. Consequently, the OIDC client exploits the ID token and the `userinfo` to authenticate and authorize the user.

Since 2020, the developed trustHub has been providing federated AA services for a total of ten applications, including two VC applications and two OIDC clients. We had rapidly prototyped two VC applications and provided them to Korean users to prevent the spread of COVID-19. The following section details the architecture and functional components of the VC applications.

5. Videoconferencing Applications

The first confirmed case of COVID-19 in South Korea was reported on 20 January, 2020. From mid-February we began developing a VC application called Webinar using open-source software (BigBlueButton) and started a pilot service. Initially, all functional blocks were implemented on a single physical server. Over time, the usage increased, and a need began for reducing the server load. We separated the functions of the VC application, and implemented each function to run on a separate server. Figure 3 shows the high-level system architecture of the Webinar VC application.

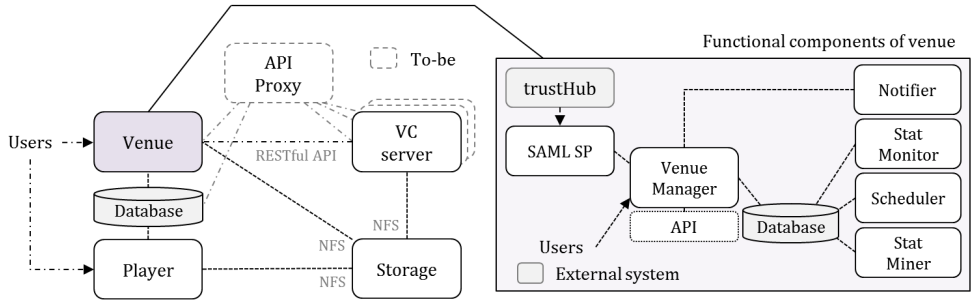


Figure 3. System architecture and software components.

Webinar consists of a venue, VC server, player, and storage server. The significant role of the VC server is to disseminate video and audio streams to the connected clients. The server adopts an SFU-based WebRTC for dissemination. Compared with a multipoint control unit (MCU) model, an SFU server can accommodate more concurrent clients because it is less computationally intensive [44]. However, clients with an SFU model require a wider network bandwidth than those with an MCU model because they are likely to receive multiple media streams.

The storage saves the recordings of the VC sessions (e.g., video footage) and the loggings (e.g., log files) of the VC server. It also provides network file system (NFS) access for other systems (i.e., venue, player, and VC server). Users take their recordings from the storage and play them using the player. We enabled the users to protect their recordings through password-based authentication. The player shares a database with the venue to retrieve the password for playing the recorded media. The venue is a web application allowing the users to create a meeting room, join a meeting, search an active session, and manage recordings. The venue exploits RESTful APIs to interact with the VC server (e.g., create a meeting and take the meeting status information). We will scale up the VC servers when the capacity of one server reaches the threshold (as discussed in Section 6). The API proxy will ensure load balancing among the multiple VC servers.

Following 2 months of operation of the VC application, we received feedback for functional improvements. The feedback ranged from demand for media quality to an easier use of the application. We focused on improving the ease of use instead of enhancing the media quality because the former is necessary for users untrained with VC applications. We classified the feedback into usability- and identity-related problems. The users wanted to be able to schedule or reserve meetings in advance and share meeting addresses through a URL. They also requested consistent meeting names, e.g., holding lectures during a semester with the same meeting name. As an identity-related problem, many users were unable to remember their `userid` or password, which increased help-desk calls and increased operational burdens. In addition, privacy and security concerns were raised owing to the corresponding threats, e.g., online classroom hijacking [45].

In Figure 3, we can see the functional components of the venue, based on the user feedback. This figure shows another VC application called Webmeet, which leverages another open-source software (Jitsi Meet [11]). We recently implemented Webmeet instead of upgrading Webinar based on user opinions, indicating that when the sharing of presentation materials is not required, the HTML5 layout of Webinar (when showing a default presentation file) is visually unappealing. Consequently, we motivated Korean users to use Webinar for 1:N lecture-style meetings and Webmeet for N:N group meetings.

The SAML SP in the figure connects federated IdPs through trustHub to authenticate users. The SP is implemented using simpleSAMLphp. Users create, join, and manage their meeting rooms, harnessing the IFrame API of Jitsi Meet. Stat Miner collects the status (e.g., number of participants) of active meeting rooms and provides the status to the venue manager. The venue manager uses the status information to present the list of all active meeting rooms, including the number of participants in each room. We managed to obtain the status information from a log file of jicofo because no API was available to retrieve information.

We separated the name and identifier of a meeting to allow the users to have a consistent meeting name. Each room is assigned a label presenting a room number for users to distinguish rooms having the same names. Once a user enters the venue to create a room, the venue sets the URL and label for the room. Next, we implemented a scheduler and a notifier to let users book meetings and share the URLs of the meetings using the most widely used instant messenger service (KakaoTalk) in Korea [46]. A user creating a meeting room can set the start time of the meeting and the meeting-initiating password. The scheduler periodically examines the start time of the reserved meetings and changes the status to active if the start time has been reached. Anyone who knows the

initiation password can start the meeting. The notifier is implemented using the RESTful API provided by the messenger service.

Since early March 2020, we have allowed users to log in to Webinar using social IdPs. We set the filtering function F_I^i in trustHub to enable access from all types of IdPs (i.e., social, guest, and institutional IdPs). Institutional IdPs are IdPs operated by research institutions and universities. In early April 2020, severe security and privacy concerns were raised against VC services in Korea because of the use of Zoom. Consequently, we had to strengthen our AA policies and apply them to Webmeet. We set trustHub to disable social IdPs and only allow guest users with institutional email addresses or users from institutional IdPs. We enforced the access control by configuring F_I^i and R_D^i. A guest user is defined as a user registered in a guest IdP, i.e., an IdP granting user self-registration.

6. Results

In this section, we discuss the usefulness of institutional IdPs and the correlation between VC usage and the number of daily confirmed COVID-19 cases. In addition, we present the analysis of the usage patterns of a VC application and estimate the capacity of the VC server. We conducted the analysis using the event-history (e.g., log in, create, or leave a meeting) stored in the databases depicted in Figures 1 and 3, as well as using the log files of the VC server. The VC server has 2 CPUs operating at 2.6 GHz, 14 physical cores per CPU, 128 GB of memory, and a 10-Gbps network interface card.

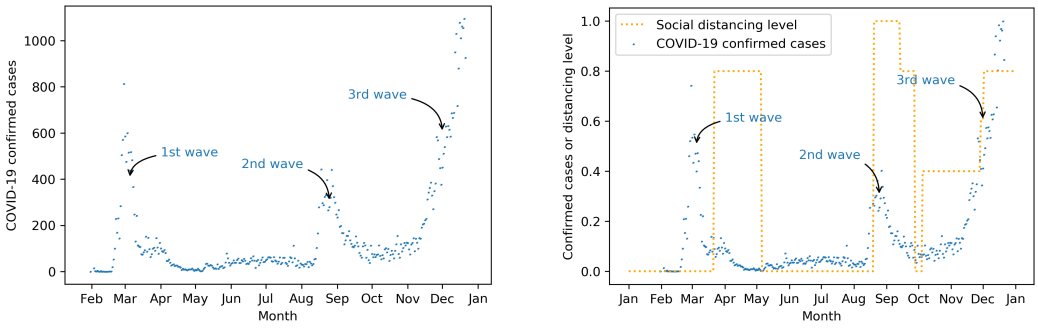
All statistics presented in section are derived from Webinar, which has more data for analysis than Webmeet. Moreover, Webinar is appropriate for estimating the usefulness of institutional IdPs over guest and social IdPs: Webmeet only accepts institutional IdPs, whereas Webinar accepts all types of IdPs. Data regarding VC were collected from 5 March to 20 December 2020. Webinar held a total of 7651 meetings during this period, and 77,891 users participated. User authentications were carried out by one guest IdP, two social IdPs, and 17 institutional IdPs.

Figure 4 shows the number of COVID-19 confirmed cases during 2020 and the social distancing level recorded by the Korean government. Observance of social distancing was compulsory. The level of social distancing was changed from a 3-level to a 5-level system from the beginning of November 2020. The maximum level of distancing has remained at Level 3. The meaning of each level before and after November 2020 is different. However, their details are omitted here.

First, we evaluate the usefulness of the institutional IdPs. Figure 5 and Table 2 summarize the VC usage statistics of each type of IdP. We enabled two social logins (Google and Naver) for the VC service and assumed that the users who join a VC room use the same type of IdP as the room creator. The users did not need to log in when they participated in a previously created VC room.

$$v_i = \frac{\psi_i/\rho_i}{\sum_j \psi_j/\rho_j}. \tag{1}$$

The usefulness of each type (v_i), listed in Table 2, is calculated using Equation (1). For instance, the usefulness of social IdPs (v_1) is 0.09, which is lower than those of the other two types. Although VC users create many rooms through social IdPs, not many people actually attend such rooms (average is only 1.24 users per room). Therefore, v_1 is low. Although the users tend not to prefer the guest IdP, its usefulness is 22% higher than that of the social IdPs. The institutional IdPs are tens of magnitude more useful than the guest or social IdPs.

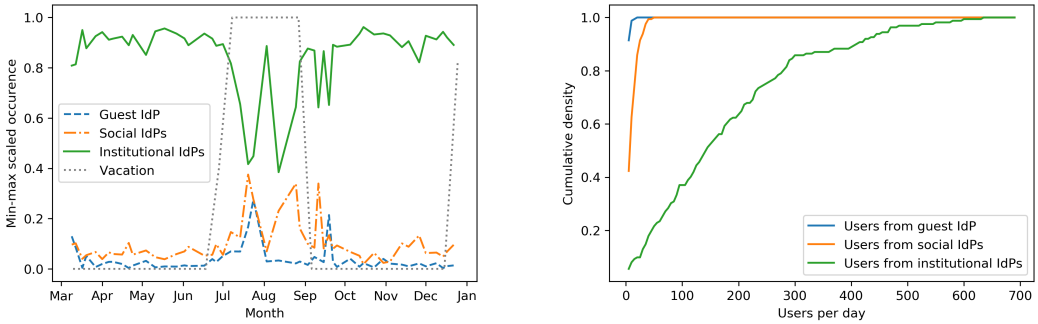


(**a**) (**b**)

Figure 4. COVID-19 confirmed cases and social distancing level in Korea (2020). (**a**) COVID-19 confirmed cases; (**b**) Social distancing level (min-max scaled).

Table 2. Usefulness of each type of identity providers.

Index (i)	Type of IdP	Avg. Rooms/Day (ρ)	Avg. Users/Day (ψ)	Usefulness (v)
1	Social	11.33	14.08	0.09
2	Guest	2.76	4.28	0.11
3	Institutional	16.91	185.15	0.79

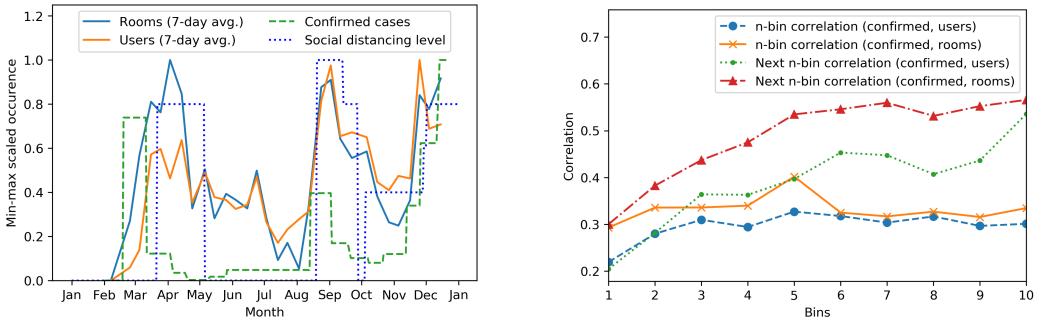


(**a**) (**b**)

Figure 5. Usage statistics by identity management types. (**a**) Users per day (time series) ; (**b**) Users per day (density).

Based on Figure 5 and Table 2, we can infer the usage patterns of the three IdPs. First, the users with the guest or social IdPs are more likely to create small-sized VC rooms because the number of daily users per room is less than 1.55. Second, users seem to favor social IdPs only to see how the VC service operates because a room created by a social IdP typically has only 1.24 users. Finally, users do not prefer to use the guest IdP if social or institutional logins are enabled, and it shows minor usage out of the three types available.

We will now discuss the correlation between COVID-19 confirmed cases and VC usage. Figure 6a shows that the number of VC sessions created per day increases as the distancing level increases. Similarly, as the number of confirmed cases or distancing level decreases, the number of VC uses declines. Interestingly, the gradual decline in VC usage continued until the arrival of a new wave (e.g., examining the section from mid-March to mid-August 2020 in Figure 6a, the number of VC uses decreased up to the second wave). We can easily infer a correlation between the number of confirmed cases and the amount of VC usage.

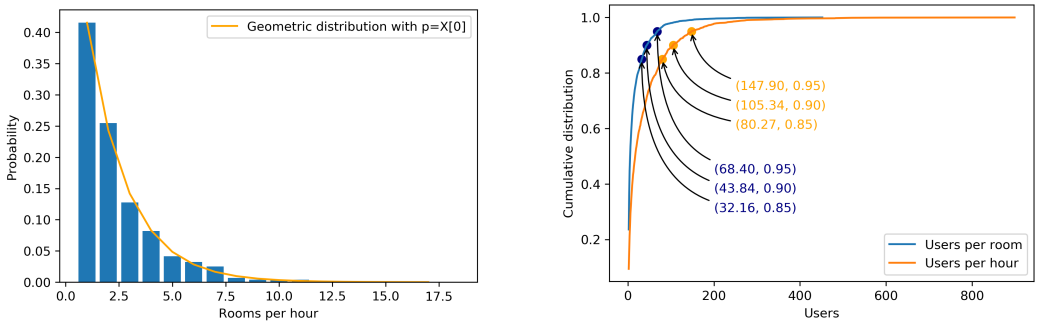


(**a**) (**b**)

Figure 6. Statistics regarding number of rooms, users, and confirmed cases. (**a**) Rooms, users, and confirmed cases of COVID-19; (**b**) Correlation among rooms, users, and confirmed cases of COVID-19.

Figure 6b illustrates such correlation. The n-bin in the figure denotes n working days, and the next n-bin correlation signifies the correlation between the data in n days and additional data in the next n days. For example, the next 5-bin correlation compares the data from this week with additional data from the next week. The number of confirmed cases during n days and the number of created VC rooms during the next n days have a linear relationship with a moderate strength if $n > 5$ ($0.5 < R$, correlation coefficient < 0.7). Contrary to our expectations, the total number of users has only a weak relationship with the number of confirmed cases. Although when the number of confirmed cases increases in a single week, the number of VC rooms to be created next week also increases; however, it is difficult to infer whether the number of users participating in the VC rooms also increases.

Figure 7 shows probability distributions of users and created rooms. Given that one or more rooms were created over a 1 h period, the distribution of the number of created VC rooms fits the geometric distribution with $X[0] = 0.42$ (Figure 7a). The probability regarding the creation of six or fewer rooms within a 1 h period is 0.95. Under the same conditions as in Figure 7a, the probability that the total number of users in one room will be less than 68.4 is 0.95. We can also determine that 85% of the VC rooms have 32.16 or fewer users, which signifies small-sized meetings are common. Finally, the probability that the total number of participants in all VC sessions during a 1 h period will be less than 147.9 is 0.95.



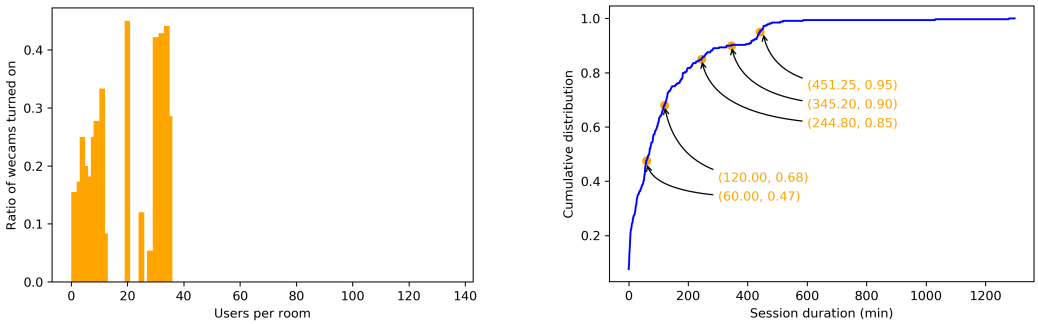
(**a**) (**b**)

Figure 7. Distribution of users and rooms per hour. (**a**) Number of created rooms per hour; (**b**) Number of users per room or hour.

Again, our goal of developing the VC applications was not to outperform other VC solutions but to enable the timely use by higher education and research institutions in Korea as a COVID-19 response measure. Therefore, we concentrated on analyzing the usage pattern of the VC application rather than comparing the performance (e.g., scalability) with other VC solutions.

How do we estimate the capacity of one VC server based on the number of VC rooms or users? We can determine the capacity if the number of concurrently running VC sessions and shared webcams (i.e., webcams turned on) in each session is known. BigBlueButton leverages SFU. Thus, the CPU load is proportional to the amount of outbound network traffic of the VC server, which is primarily affected by the number of shared webcams. The load determines the capacity of the server. However, regarding to shared webcams, we only had of 1 month worth of data.

We collected log files (e.g., *bbb_webrtc_sfu.log*) for a 1 month from the end of December 2020 to January 2021 and mined the start and end times of each VC session, including the numbers of participants and shared webcams. Figure 8 shows the results. The maximum average number of shared webcams was 14 when the room size was 34, which is lower than our estimation. A maximum of 20 webcams were turned on in a meeting with 33 participants. No more than 1 webcam was shared when the room size was over 35 users, which suggests that the users had used the VC service for lecture-type meetings. It is estimated that approximately 53% of the meetings are lectures if we assume that a meeting that lasted longer than 60 min is a lecture (Figure 8b).



(**a**) (**b**)

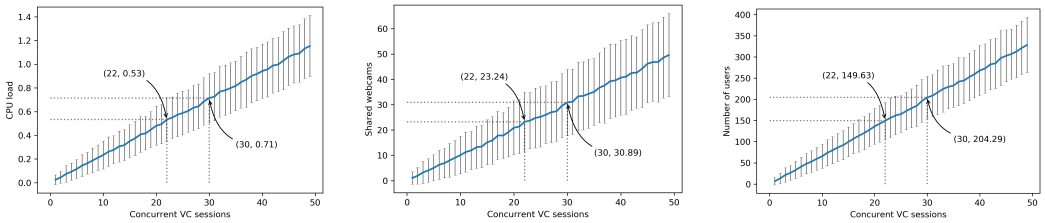
Figure 8. Measured usage statistics. (**a**) Number of shared webcams; (**b**) Duration of VC sessions.

Next, we estimated the correlation between the outbound network traffic and CPU load of the VC server. The outbound network traffic of the server (in Gbps) has a high correlation with the CPU load of 0.86. The derived regression line has a coefficient (α) of 2.05 and an intercept (β) of 0.018. We obtained the amount of outbound traffic of the server and the CPU load from the Zabbix monitoring system [47]. The granularity of the measured time interval was 1 h.

$$l_c = \alpha(n_w \times (n_u - 1) \times b_v) + \beta. \quad (2)$$

The CPU load (l_c) in one VC room can be calculated using Equation (2). The amount of outbound traffic of the VC server per room is $n_w \times (n_u - 1) \times b_v$, where n_w is the number of shared webcams, n_u is the number of users in a VC room, and b_v is the required bandwidth in Gbps for a single user. We assumed that all webcams had a video resolution of 320 × 240, generating approximately 0.5 Mbps of network traffic (i.e., b_v = 0.0005). As one of the worst-case scenarios, two concurrent VC sessions with 34 users each would generate approximately 500 Mbps of outgoing network traffic and consume 100% of CPU resources if 14 users per session turned on their webcams.

Finally, we conducted a simple simulation to evaluate the number of concurrent meetings or users a single VC server can endure. The number of users in a room was modeled with the distribution shown in Figure 7b, and the number of shared webcams in the room was determined using the results in Figure 8a. We calculated the CPU load using Equation (2). The simulation was conducted 10^3 times for every number of concurrent VC sessions. Figure 9 shows the simulation results. The error bars in each figure represent $\pm 1\sigma$ (68% confidence interval).



(**a**) (**b**) (**c**)

Figure 9. Simulated server capacity for concurrent VC sessions. (**a**) CPU loads; (**b**) Shared webcams; (**c**) Users.

As a rule of thumb, it is known that the audio quality starts degrading as the CPU load reaches 0.8 [12]. However, we frequently experienced audio glitches even at an average CPU load of approximately 0.5. Thus, we believe that VC system managers must consider the increase in server capacity if the CPU load continuously exceeds 0.5. On average, if there are 22 meetings in progress, they will have a total of 149.62 users and 23.24 shared webcams and produce a CPU load of 0.53, as shown in Figure 9a–c. Considering the error bar in each figure, the CPU load can reach 0.51 even with 16 concurrent VC sessions, holding a total of 142.52 participants and 25.32 shared webcams.

7. Discussion and Future Work

We believe that the results of this study are a good representation of the usage patterns of user identities and the VC of higher education and research institutions in Korea during the COVID-19 pandemic. However, there are some limitations to fully generalizing these results. First, there may be distortions in our data because we found many VC meetings having only a single participant. A single-user-meeting can be held for testing or lecture-recording purposes. Second, our results were analyzed using insufficient number of cases, e.g., the total numbers of participants and meetings created were less than 100,000 and 10,000, respectively. In particular, we utilized data for only a 1-month period to estimate the number of shared webcams in a meeting and the meeting duration (Figure 8); therefore, more data need to be collected and further analyzed. Finally, most of the users were affiliated with the research and education sectors in Korea, and thus, our results do not signify the usage patterns outside these sectors.

From Figure 6, we can estimate that the frequency of online and offline activities is inversely proportional during the COVID-19 pandemic. A negative slope (e.g., number of rooms created) between the first and second waves, or the second and third waves, implies increasing offline activities, resulting in the next wave. Enforcement of social distancing by the government is apparently more effective than the voluntary distancing of individuals, e.g., the number of COVID-19 confirmed cases during the second wave is smaller than that during the first wave. However, the enforcement of more stringent social distancing leads to a greater use of our VC application. In addition, the duration of the effect is gradually shortening as the high distancing level is being enforced again. We believe that the psychological fatigue regarding the distancing induced such shortening. In future research, we will comprehensively study how the frequent enforcement of the distancing affects the shortening based on additional data collected.

From a technical perspective, we collected a wide range of user feedback regarding security and privacy concerns about VC applications. A group of users believed that the VC security measures were insufficient, having only transport-layer data encryption, such as datagram transport layer security (DTLS), and a secure real-time protocol (SRTP) [48,49]. A straightforward technical approach to mitigating such security concerns would be to strengthen the AA policies. We can also consider applying technical strategies to automatically de-identify personal data embedded in the shared files or text chats of the VC applications. We plan to conduct future research on detecting and de-identifying personal or confidential information in VC content through a machine learning approach.

A virtual private network (VPN, e.g., IPsec [50]) can ensure more secure connections between a VC server and end-user VC agents compared with transport-layer encryption alone. A performance penalty and high cost are the major hurdles in deploying a VPN for VC services, requiring extensive network bandwidth in general. Users have to log in to both the VPN client and the VC services, which is another usability challenge. We will conduct further research on applying lightweight VPN protocols, such as WireGuard [51] to our VC services and resolve the challenges regarding the usability.

8. Conclusions

This study presented a federated-access management system called trustHub that provides flexible and elaborate access control for two VC applications. We implemented trustHub to provide a protocol-agnostic authentication and attribute-based access control for any relying parties supporting standard authentication frameworks. We expect relying parties to apply federated IAM without changing their authentication frameworks and easily delegate access-control functions to trusHub, which enhances flexibility and reduces implementation costs. Next, we implemented two VC applications using open-source software as a COVID-19 response for higher education and research institutions in Korea. We focused on improving the usability of the open-source software, such as adopting SAML specification, enabling users to reserve meeting rooms, and strengthening the application managers' authorization control. Our VC applications can be utilized as a low-cost VC solution for the institutions that hardly introduce commercial VC services.

We also analyzed the usage patterns of a VC application using the data collected from a pilot service carried during the COVID-19 pandemic. The fact that we successfully collected the data implies that trustHub functioned well to interconnect IdPs and relying parties based on defined rule sets, such as enabling social IdPs. Our finding is that users prefer to use their institutional credentials rather than those registered in social IdPs or guest IdPs: this indicates that institutional IdPs could be commercially attractive to service providers. Then, we found that the number of created VC rooms correlated with the number of COVID-19 confirmed cases over the past 5 days. In addition, the number of concurrent users which a single server can accommodate was tied with the number of shared webcams. We believe the quality degradation of VC due to the explosive growth of concurrent users can be indirectly mitigated by timely provisioning user access using trustHub.

Author Contributions: Conceptualization, J.J.; formal analysis, J.J. and J.K.; investigation, J.J., H.J., and Y.C.; software, J.J. and Y.C.; writing—original draft, J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Korea Institute of Science and Technology Information, K-21-L02-C02-S01.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sidpra, J.; Gaier, C.; Reddy, N.; Kumar, N.; Mirsky, D.; Mankad, K. Sustaining education in the age of COVID-19: A survey of synchronous web-based platforms. *Quant. Imaging Med. Surg.* **2020**, *10*, 1422. [CrossRef] [PubMed]
2. Chidambaram, S.M.D.N. Success of Online Teaching and Learning in Higher Education-COVID-19 Pandemic: A Case Study Valley View University, Ghana. *J. Appl. Eng. Res.* **2020**, *15*, 735–738.
3. Skulmowski, A.; Rey, G.D. COVID-19 as an accelerator for digitalization at a German university: Establishing hybrid campuses in times of crisis. *Hum. Behav. Emerg. Technol.* **2020**, *2*, 212–216. [CrossRef] [PubMed]
4. Byrnes, K.G.; Kiely, P.A.; Dunne, C.P.; McDermott, K.W.; Coffey, J.C. Communication, collaboration and contagion: Virtualisation of anatomy during COVID-19. *Clin. Anat.* **2021**, *34*, 82–89. [CrossRef] [PubMed]
5. Gonzales-Zamora, J.A.; Alave, J.; De Lima-Corvino, D.F.; Fernandez, A. Video conferences of Infectious Diseases: An educational tool that transcends borders. A useful tool also for the current COVID-19 pandemic. *Infez. Med.* **2020**, *28*, 135–138. [PubMed]

6. Wlodarczyk, J.R.; Wolfswinkel, E.M.; Carey, J.N. Coronavirus 2019 video conferencing: Preserving resident education with online meeting platforms. *Plast. Reconstr. Surg.* **2020**, *146*, 110e–111e. [CrossRef] [PubMed]
7. Fatani, T.H. Student satisfaction with videoconferencing teaching quality during the COVID-19 pandemic. *BMC Med. Educ.* **2020**, *20*, 1–8. [CrossRef] [PubMed]
8. Okereafor, K.; Manny, P. Understanding Cybersecurity Challenges of Telecommuting and Video Conferencing Applications in the COVID-19 Pandemic. *IJMR* **2020**, *8*, 13–23.
9. Kagan, D.; Alpert, G.F.; Fire, M. Zooming Into Video Conferencing Privacy and Security Threats. *arXiv* **2020**, arXiv:2007.01059.
10. Wenzel, M.; Meinel, C. Full-body WebRTC video conferencing in a web-based real-time collaboration system. In Proceedings of the 2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Nanchang, China, 4–6 May 2016.
11. Grozev, B.; Marinov, L.; Singh, V.; Ivov, E. Last N: Relevance-based selectivity for forwarding video in multimedia conferences. In Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, Portland, Oregon, USA, 18–20 March 2015.
12. BigBlueButton. Available online: https://docs.bigbluebutton.org/support/faq.html (accessed on 10 February 2021).
13. eduMEET Web-Based Videoconferencing Platform. Available online: https://edumeet.org (accessed on 10 March 2021).
14. Hasan, R; Ragib, H. Towards a Threat Model and Security Analysis of Video Conferencing Systems. In Proceedings of the IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2021.
15. Correia, A.; Chenxi, L.; Fan, X. Evaluating videoconferencing systems for the quality of the educational experience. *Distance Educ.* **2020**, *41*, 429–452. [CrossRef]
16. Selvanathan, N.;Dileepa, J.;Violeta, D. Federated identity management and interoperability for heterogeneous cloud platform ecosystems. In Proceedings of the 14th International Conference on Availability, Reliability and Security, Canterbury, UK, 26–29 August 2019.
17. Basney, J.; Flanagan, H.; Fleury, T.; Gaynor, J.; Koranda, S.; Oshrin, B. CILogon: Enabling federated identity and access management for scientific collaborations. In Proceedings of the International Symposium on Grids & Clouds, Taipei, Taiwan, 31 March–5 April 2019.
18. SURF Research Access Management: Ease Collaboration with Fellow Researchers. Available online: https://sbs.sram.surf.nl/landing (accessed on 10 March 2021).
19. Klaas, W.; Leif, J.; Christos, K.; David, G.; Davide, V.; Nicolas, L. *EOSC Authentication and Authorization Infrastructure*; EU Publications: Luxembourg, 2021.
20. Procházka, M.; Licehammer, S.; Matyska, L. Perun—Modern approach for user and service management. In Proceedings of the IST-Africa Conference, Pointe aux Piments, Mauritius, 7–9 May 2014.
21. AARC Projects. Available online: https://wiki.geant.org/display/AARC/Pilot+res\ults+and+demos (accessed on 10 March 2021).
22. Internet2 COmanage. Available online: https://incommon.org/software/comanage (accessed on 10 February 2021).
23. Nishimura, T.; Sakane, E.; Yamaji, K.; Nakamura, M.; Aida, K.; Klingenstein, N. Virtual organization platform interoperability provides the long tail an eScience environment. *J. Inf. Process.* **2016**, *24*, 609–619. [CrossRef]
24. Biancini, A.; Florio, L.; Haase, M.; Hardt, M.; Jankowski, M.; Jensen, J.; Kanellopoulos, C.; Liampotis, N.; Licehammer, S.; Memon, S.; et al. AARC: first draft of the blueprint architecture for authentication and authorization infrastructures. *arXiv* **2016**, arXiv:1611.07832.
25. Introduction to the AARC Blueprint Architecture (AARC-BPA-2017). Available online: https://aarc-project.eu/wp-content/uploads/2019/03/Introductory-module-on-BPA.pdf (accessed on 10 March 2021).
26. Webmeet. Available online: https://webmeet.kafe.or.kr (accessed on 19 March 2021).
27. Webinar. Available online: https://webinar.kafe.or.kr (accessed on 19 March 2021).
28. Minessale, I.A.; Maruzzelli, G. *Mastering FreeSWITCH*; Packt Publishing Ltd.: Birmingham, UK, 2016.
29. Garcia, B.; Lopez-Fernandez, L.; Gallego, M.; Gortazar, F. Kurento: The Swiss army knife of WebRTC media servers. *IEEE Commun. Stand. Mag.* **2017**, *1*, 44–51. [CrossRef]
30. Prosody, I.M. Available online: https://prosody.im (accessed on 10 March 2021).
31. Corona 19 Infection Status. Available online: https://www.data.go.kr/data/15043376/open-api.do (accessed on 15 March 2021).
32. NextCloud. Available online: https://nextcloud.com (accessed on 19 February 2021).
33. Sefraoui, O.; Mohammed, A.; Mohsine, E. OpenStack: Toward an open-source solution for cloud computing. *J. Comput. Appl.* **2012**, *55*, 38–42. [CrossRef]
34. eduGAIN. Available online: https://edugain.org (accessed on 10 February 2021).
35. Milligan, M.B. Jupyter as common technology platform for interactive HPC services. In Proceedings of the Practice and Experience on Advanced Research Computing, Pittsburgh, PA, USA, 22–26 July 2018.
36. Chard, K.; Foster, I.; Tuecke, S. Globus: Research data management as service and platform. In Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact, New Orleans, LA, USA, 9–13 July 2017.
37. SATOSA. Available online: https://github.com/IdentityPython/SATOSA (accessed on 10 February 2021).
38. Sakimura, N.; Bradley, J.; Jones, M.; De Medeiros, B.; Mortimore, C. Openid Connect Core 1.0 Incorporating Errata Set 1. 2014. Available online: https://openid.net/specs/openid-connect-core-1_0.html (accessed on 2 September 2021)

39. Andronache, I.; Nisipasiu, C. Web single sign-on implementation using the simpleSAMLphp application. *J. Mobile Embed. Distrib. Syst.* **2011**, *3*, 21–29.
40. Two-Factor Authentication Module for Simple SAMLphp Using Google Authenticator. Available online: https://github.com/NIIF/simplesamlphp-module-authtfaga (accessed on 2 June 2021).
41. Niels, V.D. *White Paper for Implementation of Mappings between SAML 2.0 and OpenID Connect in Research and Education*; REFEDS: 2018.
42. Kemp, J.; Cantor, S.; Mishra, P.; Philpott, R.; Maler, E.; Cahill, C.P.; Hughes, J.; Lockhart, H.; Beach, M.; Metz, R.; et al. Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS. 2005. Available online: https://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf (accessed on 11 September 2021).
43. REFEDS MFA Profile. Available online: https://refeds.org/wp-content/uploads/2017/06/REFEDS-MFA-Profilev1.0.pdf (accessed on 10 February 2021).
44. Petrangeli, S.; Pauwels, D.; van der Hooft, J.; Wauters, T.; De Turck, F.; Slowack, J. Improving quality and scalability of WebRTC video collaboration applications. In Proceedings of the 9th ACM Multimedia Systems Conference, Amsterdam, The Netherlands, 12–15 June 2018.
45. Lowenthal, P.; Borup, J.; West, R.; Archambault, L. Thinking Beyond Zoom: Using Asynchronous Video to Maintain Connection and Engagement During the COVID-19 Pandemic. *J. Technol. Teach. Educ.* **2020**, *28*, 383–391.
46. Lee, S.Y.; Lee, S.W. Social media use and job performance in the workplace: The effects of facebook and kakaotalk use on job performance in South Korea. *Sustainability* **2020**, *12*, 4052. [CrossRef]
47. Olups, R. *Zabbix Network Monitoring*; Packt Publishing Ltd.: Birmingham, UK, 2016.
48. McGrew, D.; Rescorla, E. *RFC 5764: Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-Time Transport Protocol (SRTP)*; IETF: Fremont, CA, USA, 2010.
49. Gallenmüller, S.; Schöffmann, D.; Scholz, D.; Geyer, F.; Carle, G. DTLS Performance-How Expensive is Security? *arXiv* **2019**, arXiv:1904.11423.
50. Kotuliak, I.; Rybár, P.; Trúchly, P. Performance comparison of IPsec and TLS based VPN technologies. In Proceedings of the 9th International Conference on ICETA, Stara Lesna, Slovakia, 27–28 October 2011.
51. Jason, A.D. WireGuard: Next Generation Kernel Network Tunnel. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 26 February–1 March 2017.