Cross-domain identity and discovery framework for web calling services

Ibrahim Tariq Javed, Rebecca Copeland, Noel Crespi, Marc Emmelmann, Andreea Ancuta Corici, Ahmed Bouabdallah, Tuo Zhang, Saad El Jaouhari, Felix Beierle, Sebastian Göndör, et al.

Cross-domain identity and discovery framework for web calling services

Ibrahim Tariq Javed[1], Rebecca Copeland[1], Noel Crespi[1], Marc Emmelmann[2], Ancuta Corici[2], Ahmed Bouabdallah[3], Tuo Zhang[3], Saad El Jaouhari[3], Felix Beierle[4], Sebastian Göndör[4], Axel Küpper[4], Kevin Corre[5], Jean-Michel Crom[5], Frank Oberle[6], Ingo Friese[6], Ana Caldeira[7], Gil Dias[7], Nuno Santos[7], Ricardo Chaves[7], Ricardo Lopes Pereira[7]

Abstract Cross-domain identity management remains a major challenge for potential WebRTC adopters. In order to provide a global web-based communication system, it is critical to locate the destination called party, map the identity to the user device, and provide mutual authentication for both caller and called party. In this paper, we present a novel identity management and user discovery framework that enables callers to search and locate users across service domains. The identity management is decoupled from the used calling service, allowing users to manage their profiles and credentials independently of the applications. The framework is designed to preserve privacy and exploit web technology to gain trust and contact list management.

✉ Ibrahim Tariq Javed
ibrahim_tariq.javed@telecom-sudparis.eu

Rebecca Copeland
rebecca.copeland@coreviewpoint.com

Noel Crespi
noel.crispi@telecom-sudparis.eu

Marc Emmelmann
emmelmann@ieee.org

Ancuta Corici
andreea.ancuta.corici@fokus.fraunhofer.de

Ahmed Bouabdallah
ahmed.bouabdallah@imt-atlantique.fr

Tuo Zhang
tuo.zhang@imt-atlantique.fr

Saad El Jaouhari
saad.eljaouhari@imt-atlantique.fr

Felix Beierle
beierle@tu-berlin.de

Sebastian Göndör
sebastian.goendoer@tu-berlin.de

Axel Küpper
axel.kuepper@tu-berlin.de

Kevin Corre
kevin1.corre@orange.com

Jean-Michel Crom
jeanmichel.crom@orange.com

Ingo Friese
ingo.friese@telekom.de

Ana Caldeira
ana.caldeira@tecnico.ulisboa.pt

Gil Dias
gil.dias@tecnico.ulisboa.pt

Nuno Santos
nuno.santos@inesc-id.pt

Ricardo Chaves
ricardo.chaves@inesc-id.pt

Ricardo Lopes Pereira
ricardo.pereira@inesc-id.pt

[1] Institut Mines-Telecom, Telecom Sud-Paris, Evry, Paris, France

[2] NGNI, Fraunhofer FOKUS, Berlin, Germany

[3] Department SRCD, IMT Atlantique, Cesson-Sévigné, France

[4] Technische Universität Berlin, Telekom Innovation Laboratories, Berlin, Germany

[5] Orange Labs Products & Services, Cesson-Sévigné, France

[6] Telekom Innovation Laboratories, Berlin, Germany

[7] INESC-ID, IST, Universidade de Lisboa, Lisbon, Portugal

Keywords WebRTC · Identity management · Trust · Real-time communication · P2P · Directory · Social graph · Registry

1 Introduction

Web real-time communication (WebRTC) is a W3C standard that provides communication capabilities in a peer-to-peer (P2P) fashion to web browsers and applications [1]. WebRTC supports browser-to-browser interoperability, unlike prevalent web communication services (such as Skype and WhatsApp). The advent of browser-based WebRTC calling APIs has made it remarkably easy for any website to offer calling services. For this reason, the potential of WebRTC technology stretches much beyond existing dedicated Voice-Over-IP solutions. WebRTC developers endeavor to provide reliable mechanisms to ensure security and privacy [2]. While current over-the-top web communication services create silos of single-domain users by restricting their subscribers to only communicate within their specified service domains, browser-to-browser WebRTC services provide compatibility across browser users. Today, most users utilize several web calling services, depending on context and preferences. Each service requires separate identifiers and credentials to be maintained. Since they are not shared between services, they cannot be used to discover and connect between users of different services. Hence, cross-domain identity is crucial to fully interoperable web services.

The telecom industry is studying WebRTC with the hope to harness web calling and building new web ecosystems [3]. Some operators consider adopting WebRTC to compete with over-the-top web services, with a much lower cost base of a peer-to-peer service that has no large core backend systems. They plan to enrich the underlying P2P technology by offering enhanced quality of service and cross-domain interoperability. This is the motivation of the telecom partners of the reTHINK project[1] that describes a new communication platform for real-time communication services. The developed framework relies on a module of software logic that is dynamically deployed on end-user devices. This allows session control and media flow management in a P2P fashion [4] at the endpoints. Each communication service provider (CSP) retains knowledge of its logged-in users and allows searching by other domains.

For global cross-domain communication platforms, the mapping of an identity to an "active" (logged on) end-user device is a critical challenge. The identified IP location can be used to establish a communication session across multiple domains [5]. Unlike mobile networks, web identities are only used currently for authentication purposes and not for discovering the user location and availability to enable routing call. Moreover, existing identity systems are tied to specific administrative domains and are highly dependent on the use of specific identity formats and protocols with static authentication mechanisms [6]. Therefore, a novel identity resolution system is essential to map user identities to the address of the currently used user device, regardless of the service domain. Such identities must be verified and authenticated in an efficient and reliable manner against independent credentials (not only service specific) before a communication session is established.

In this paper, we present a novel identity and discovery framework that allows global discovery of users and a cross-domain identity management system. A new approach is proposed where each collaborating service retains its own directory and user information, but creates an "overlay" of identity management that links their directories to provide a global user discovery. Services that are compliant with the identity management framework can still maintain their internal user identifiers, but relate them to globally unique user identifiers, so that they can be discovered and contacted across multiple domains. It is proposed that user identities are maintained by third party independent identity providers (IdP), who allow communicating participants to verify each other's identity, regardless of the services that they are using. Such identities must be portable, supporting user migration between different domains. The IdP function is to link whatever identifiers the various calling services allocate to the user locally with a globally unique user identifier, Global User Identity (GUID). Furthermore, users' security and privacy are enhanced by computing trust between communicating participants.

The rest of the paper is organized as follows: Section 2 provides the related work. Section 3 details the functional architecture of the cross-domain identity and discovery framework. The details on authentication and trust estimation are provided in section 4. Section 5 describes three of the major directory services involved in identity resolution namely catalogue, registry, and discovery. Section 6 conducts a privacy and security analysis for the framework. Section 7 describes two support services, the policy management and graph connector, whereas section 8 gives the conclusion.

2 Related work

Identity over web is combination of user profile (name, email address, location) and credentials (password and shared secret). Services maintain different levels of knowledge about users but are invariably requiring a service-based identifier. This results in users having to maintain multiple, unrelated identities. To alleviate the burden on users, single sign-on (SSO) systems allow federating identities (when login to one service acts as login to another), or managing linked identities

[1] ReTHINK Project Website: https://rethink-project.eu

via third parties (IdPs) [7]. Here, user authentication is delegated from the service (relying party) to a third party IdP. The IdP allows users to assert their identity using tokens in order to authenticate themselves to the relying party. There are well-developed web protocols that provide generic procedures across services. One such protocol is OAuth 2.0 [8], which provides authorized clients to access protected resources by obtaining access tokens from the IdP. OpenID Connect [9] provides an identity token that contains claims about user authentication. These protocols basically define the interactions between the relying party, user, and IdP. These protocols are traditional user-server authentication procedures, while WebRTC now requires user-user authentication for peer-to-peer communication [10, 11]. However, they may still be used for this purpose under a framework of collaborative procedures.

In telecom networks, the identities are publicly known identifiers (under the international telephone numbering system), which are also used to route the call to the current location of the device. Telecom service providers enforce a standardized set of rules for the identifiers and their "roaming" devices that facilitate both user identification and routing across different domains. Users are authenticated by their "home" server [12], even when they are served by another service provider; hence, this is a centralized identity approach with a federated service approach. By contrast, a web session is established with a current IP address, which is dynamically associated with a URL and a particular device, so finding a called party location requires a special discovery solution. In [13], the mechanism of presence (with subscribe/notify) is used to gain awareness of users' availability and facilitate routing to the currently available IP address.

In online social networks (OSN) such as Facebook, identities are only applicable within their administrative domains and have service-specific formats and authentication procedures. A service-based centralized directory is used to retrieve the user profile associated with its identity. In distributed social networks, user's social profiles can be hosted on any server, which is also responsible for identity management of these users. Servers in the distributed approach can communicate with each other using a federated identity protocol. A major drawback of this approach is that users are bound to trust their server, which can be hosted by anyone, with little or no restrictions. Furthermore, users have no control over their identifiers, which are fully managed by the server. Other approaches use a distributed hash table (DHT)-based P2P network to host signed records for users, while the actual user profiles are stored on servers connected by an open federation protocol, for example SONIC [14]. The Safebook project [15] uses DHT and web trust in a decentralized approach to achieve privacy and identity integrity in social networks.

In summary, cross-domain P2P web communication services face two specific challenges related to identity management: (i) mutual authentication and (ii) discovery. Mutual authentication involves service-independent identification and verification of user identity for both parties. Discovery is required to locate users across different service domains by resolving user identity to the current web location. With these two research problems in mind, we propose a novel identity and discovery management framework for interoperable web communication services.

3 Proposed framework

In this section, we detail the cross-domain identity and discovery framework that enable callers to search, locate, and authenticate users globally. An overview of the functional architecture of the framework is presented in Fig. 1. The proposed framework uses the concept of Hyperty, which is a JavaScript code provided by the CSP and deployed at the user's device [4]. A logged-in communication Hyperty ("live" status) represents a user who is available for connectivity on a specific device. Each CSP retains knowledge of its "live" Hyperties and enables connectivity with Hyperties of other domains. To establish a WebRTC communication session, all the caller requires is the current location (IP address or URL) of the called party's Hyperty that the caller wishes to communicate with.

Hyperties are executed in web runtime environment on endpoints which can interwork with a web browser or native app. Hyperty fundamentally consists of a static and a dynamic part. The former is defined when the Hyperty is provisioned and remains unchanged until the Hyperty is removed. The dynamic part concerns a Hyperty instance created when a Hyperty is deployed. The Hyperty life cycle is used to determine the perimeter of the data model, which pertains to Hyperties as well as the structure of the associated

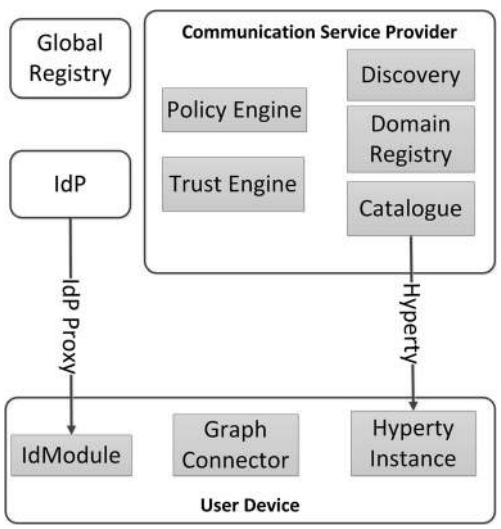

Fig. 1 Cross-domain identity and discovery functional architecture

information, which has to be locally maintained by the involved entities during the life cycle of the Hyperty. For details of the global structure and description of the data model, we refer readers to our technical report [16].

An active communication Hyperty represents an endpoint that is associated with a particular identity. To achieve interoperability and openness, the identities are managed by a third party IdP, who can support multiple calling services. This allows communicating parties to authenticate and validate each other independently of their chosen services. The device-based Identity Module (IdModule) component is a CSP client responsible for user registration, identity provisioning, and storing identity assertions (IA). For service-independent authentication, the IdP-Proxy is downloaded from the IdP's URL. The IdP-Proxy provides an interface between IdP and IdModule for user authentication. The IdModule receives and stores the IA to authenticate a user to its CSP and communicating participants. The function of the trust engine is to estimate trustworthiness of the communicating participants in order to minimize the risk involved in establishing a connection with an unknown party. For user discovery, the framework includes three types of directories: a registry for information about available Hyperty instances for communication, a catalogue for a list of available service functions provided by Hyperties of various CSPs, and a discovery service for finding users across various domains.

To achieve global reachability and discovery, the framework uses two unique identifiers for users, namely, Global User Identity (GUID) and User Identity (UserID). The GUID is a unique and domain-independent identifier that remains the same, irrespective of the CSP. This GUID can be used to contact any available communication endpoint of the user without the need to know where he is subscribed. The UserID is the identifier which is used to get the actual location of user device by discovering its Hyperty instance within the CSP domain it is registered to. Every CSP maintains a list of currently available Hyperties of a user. As soon as a user downloads a Hyperty to an endpoint device, the CSP registers the IP address for this instance, thus storing the user availability status and the routing network address where the user is currently using a particular service that is compatible with the framework. As each Hyperty belongs to a specific calling service, it is related to the CSP-given UserID, which is registered in the CSP local directory. Users are allowed to maintain the linking of UserIDs to a globally unique identifier (GUID) that they can manage independently of any service. To manage the linking of several CSP's UserIDs with the user-controlled GUID, an independent identity provider (IdP) service is required, which is acceptable to all participating CSPs. This GUID is essential for service mobility as it allows users to retain their identity while switching between CSPs.

Figure 2 illustrates linking the service-based UserID and the service-independent GUID. It shows how Alice uses the framework directory services to discover and authenticate Bob before establishing a connection. To initiate a call request, Alice needs to know Bob's well-known identifier for the service that Bob is using. Alice uses the discovery service of her own CSP to discover Bob's GUID that Bob is currently using. The global registry, which is independent of the CSP, finds Bob's GUID and can link it to one or more service-based UserIDs. If Alice has previously contacted Bob then the GUID can be accessed from the local address book "graph connector". After discovering Bob's GUID, Alice must establish whether Bob is currently contactable on any on his services with their associated different identifiers. This is facilitated by using the UserIDs found on the global registries to access the local domain-based registries, where the currently active Hyperties status is registered dynamically. Hence, Alice may find at least one UserID for Bob that has a current IP (or none if Bob is not contactable at this moment). The mechanism of registering running Hyperties by each compatible calling service is therefore the means of setting up connectivity between Alice's own Hyperty and Bob's Hyperty. Before establishing a communication session, Alice and Bob authenticate each other using third party identity providers. Furthermore, the trustworthiness of the communicating participant can be checked using the trust engine service.

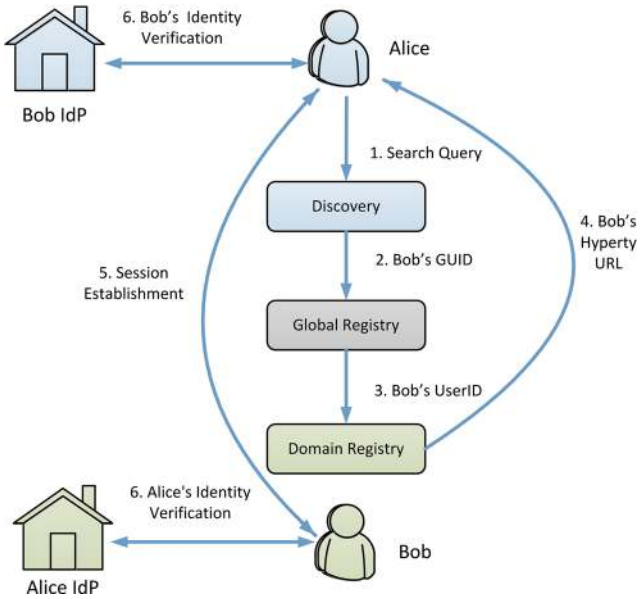

Fig. 2 Cross-domain user discovery

4 Authentication and trust estimation

The framework supports peer-to-peer authentication that allows not only user-to-service authentication, but also user-to-user by the verifications of identity assertions (IAs). In order for the mutual authentication to be successful, all messages are required to have an IA, which is a digital certificate.

Therefore, to authenticate a message, the sender's IA that is obtained from the sender's own IdPs is attached to the message, containing the user's public key. To confirm that the public key actually corresponds to the claimed identity, the receiving user (i.e., the "called party") contacts the sender's IdP to validate the content of the sender's IA. When the receiver validates the sender's digital signature (the confirmation of the assertion), he can encrypt the response to the sender's challenge with his own identity assertion. Then, to conclude the procedure, the mutual authentication inverts the roles, so that the receiving party becomes the one who must prove his or her identity assertion, using the same procedure.

The IdP role is to issue such IAs and to confirm the ownership of the identity in response to enquiries by other parties. In WebRTC, it is proposed that IAs are generated and verified through the IdP-Proxy mechanism [10]. Similarly, in our framework, an IA (e.g., implemented by a JSON Web Token [17], as in OpenID Connect) is attached to the call offer as well as the answer, so the calling parties exchange identity assertions that are then confirmed by their respective IdPs. As the IA contains the IdP's URL, the other party can contact the issuing IdP. Hence, the WebRTC authentication process is supported by the framework that identifies the called party's IdP even if the caller's service that initiated the connection request is not aware of it beforehand.

While these flexible features allow greater inter-service cooperation, they also raise some security and integrity concerns, because the originating service is not able to set some defense mechanisms, such as limiting the intended audience that has access to the full identity information. Difficulties or even security breaches can be caused if the IA is available to any party, not only those that the IA was intended for, when implementing IdP-Proxies for standard protocols such as OpenID Connect. To resolve this, the IdModule component at the user device includes the GUID in the assertion, which limits the scope of intended audience, and refreshes the user's assertion token frequently. Every time a user starts a communication session with another user, the process of mutual authentication commences using the particular requirements of the calling services (formats, protocols), and a dialogue based on the TLS handshake is initiated, setting the required parameters accordingly. This process of authentication results in the exchange of the symmetric keys to be used in secure communication, so even if one user seeks anonymity, the other user is still authenticated, in order to establish a secure channel.

The sequence diagram of the registration of a new identity in the IdModule is presented in Fig. 3. To initialize the registration procedure, the IdP's URL is provided by the local runtime in the user's device to the IdModule, which is a generic endpoint application that supports all the different Hyperties from multiple CSPs. The IdP's URL allows the IdModule to retrieve an IdP-Proxy and instantiates it temporarily on the user's device. The "runtime" generic client must authorize the IdP-Proxy to serve as an IdP delegate in order for the IdP-Proxy to connect to the issuing IdP and retrieve a digital signature. The authenticated GUID identity is associated with the downloaded Hyperty that requested authentication, thus linking the UserID and the GUID. Further details including sequence diagrams of identity association with Hyperty instance, identity assertion generation, and verification are found in our report [18].

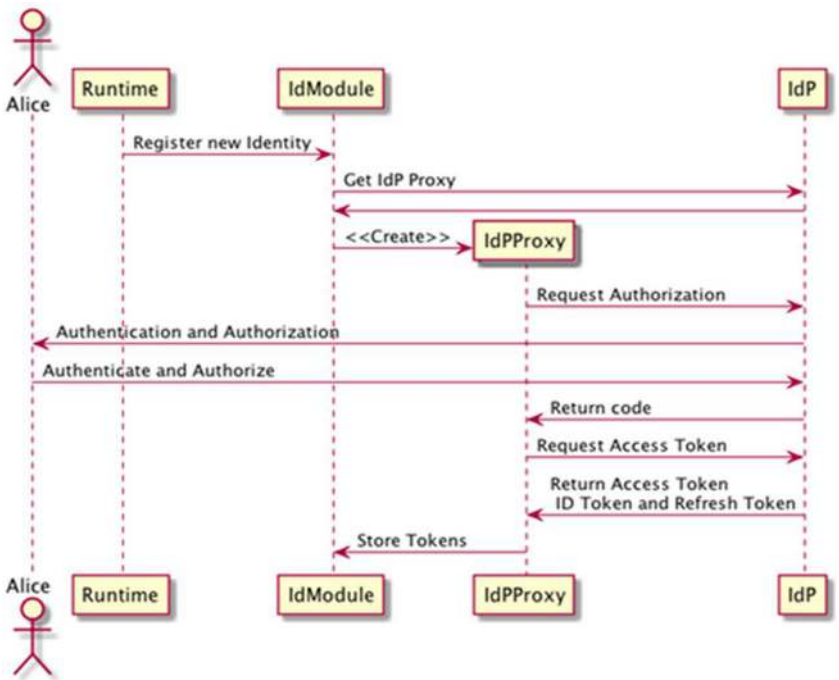

Fig. 3 Identity registration sequence diagram

While the framework facilitates mutual authentication for given identifiers, this does not ensure that the users are trusted to act in an acceptable, responsible, and legitimate manner. Users may be involved in spam calls, sending malicious content, phishing, identity misrepresentation, etc. We define trust between communicating participants as the belief that they will act in an acceptable and legitimate manner over the established communication session. Establishing trust between communicating parties will reduce uncertainty and risk involved while establishing a communication session. The evaluated trust enhances users' security and privacy by minimizing unwanted call activities. Various parameters have been previously considered for the computation of trust, such as identification, experience, and recommendation [19]. We propose a reputation-based trust model that uses recommendations and user behavior to evaluate trust. Recommendations are based on user experiences whereas call characteristics (such as incoming/outgoing and talk time) are used to predict the user's popularity and acceptability in the network. Further detail on the evaluation of trust and implementation of trust engine can be found in [20].

5 Directory services

The endpoint discovery and reachability are designed in a modular way, using three directory services: registry, catalogue, and discovery [21]. The catalogue stores descriptors of Hyperties that the users can utilize (i.e., services and domains that the user can log into); the registry stores information on how to reach a Hyperty instance that the user has activated (i.e., login status for a particular calling service); whereas, discovery services provide ways for users to find other users to initiate communication for a discovered identifier and domain.

5.1 Registry

In order to initiate a connection to a specific user Hyperty instance, it is required to know its current network address. Our framework allows frequent changes of locations (IP addresses and devices) and of domains (CSPs calling services) by the user. All information required to initiate a connection to a Hyperty is published in a registry service upon the initiation of a Hyperty instance and is removed from the registry when the instance is terminated. If the network address of the device running the Hyperty instance changes, the information is updated automatically to provide a seamless way to connect to the Hyperty instance. Hence, the registry provides a directory of users who are available to receive communication requests.

Our framework allows seamless migration of users between different CSPs. A globally unique identifier (GUID) is assigned to each user. These GUIDs are domain agnostic and can be kept even after changing the association to a service provider. A GUID is derived from a user's public ECDSA key and a cryptographic salt, using the key derivation function PBKDF#2, where the GUID, the public key, the salt, and other relevant information are published as a digitally signed JSON Web Token [17] in a distributed directory service, the global registry. The GUID can be used for identification purposes regardless of the CSP's domain; hence, it allows mobility between CSPs. Each user is also identified within the CSP domain by the UserID, which is the identifier that unlocks access to the particular service. As users may want to use services of multiple CSPs, each user may have more than one UserID.

Registry services in the cross-domain identity framework comprise of two main components: the global registry and the domain registry. The global registry is built on P2P technology using a Kademlia-based DHT, similar to the global social lookup service [22]. Following this approach, a single point of failure is avoided, resulting in a distributed and domain-independent directory service. While the global registry is able to provide fast response times for read operations, write operations are much slower [23]. Hence, data that has to be updated frequently is stored in domain registry services, which follow a traditional client-server approach to facilitate fast response times not only for read operations but also for frequent write operations, as a result of updated client information. The global registry resolves a user's GUID to the CSP-specific UserID, whereas the domain registry translates UserIDs to the information about the Hyperty instances of this user, i.e., the IP location or URL of a reachable endpoint for this user. This allows other users and network devices to initiate a connection to the actual Hyperty location of the user. Figure 4 outlines the relationships between Hyperty, global registry, and domain registry. When initiating a connection to a Hyperty instance, the global registry is queried using GUID, to obtain the target user's UserID and its current domain registry server. In the following step, the domain registry uses the user's UserID to obtain the current Hyperty IP address.

5.2 Catalogue

The catalogue service conceptually acts as a software repository that contains information and the executable code for Hyperties. The catalogue is the initial entry point providing components to be executed at end-user devices. Access to the catalogue has to be provided via standard protocols, widely used. The resource-based view of catalogue entries allows representing them according to OMA-TS-Lightweight M2M [24]. Create, read, update, and delete (CRUD)-based access can be directly mapped to http or lwm2m/coap-based operations. The architecture follows for its catalogue RFC 6690 [25], specifying URLs to descriptors as entries of a "well-

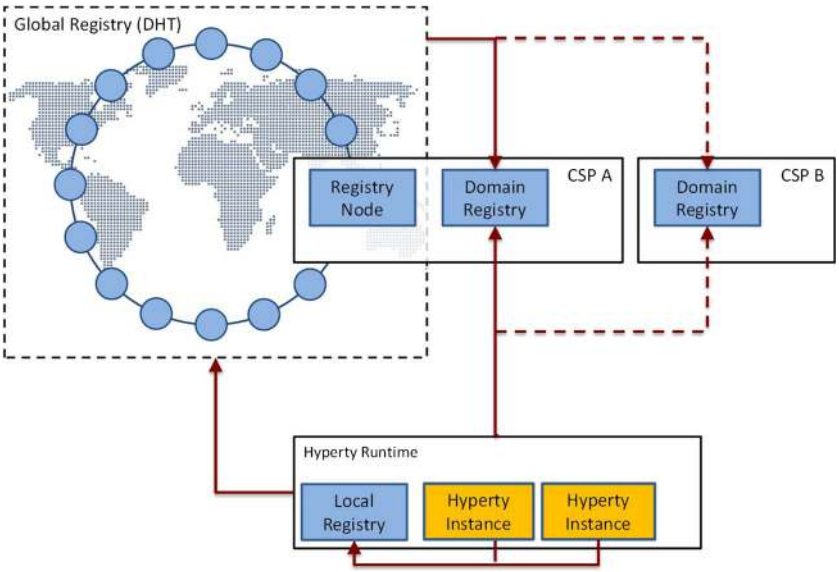

Fig. 4 Interplay of global registry and domain registry

known" core, which allows standard-compliant discovery of all stored resources via an http-get operation. The catalogue is implemented in two components: the catalogue broker and the catalogue database. The broker acts as an aggregation point for all databases storing catalogue objects; hence, the broker is contacted by the framework components to request information on a catalogue object stored in any of the attached databases. The advantage of this approach is that databases from anywhere in the world may be attached and detached, to allow deploying newly developed Hyperties on the fly.

5.3 Discovery

The discovery component allows searching for conversational partners in a similar way to Internet search engines. The discovery service assists users who are looking to connect to people for whom they do not have contact details or an address book entry. For better usability, active endpoints for a user must be found even without knowing the CSP, the UserID, or the GUID, by searching on users' characteristics and attributes. The discovery service may find more than one profile matching the search query, since users could be using several devices and services simultaneously.

A RESTful API allows passing search requests to the semantic interpretation component within the discovery service, which returns matching data records, including the respective registry keys. The registry keys are used for a lookup of the communication endpoints in the global registry. The discovery service implementation is a combination of a search engine and a directory service. Users are able to create accounts and store directory profiles, so that the service can discover users by various attributes, such as email addresses, links to social network profiles, or phone numbers. These profiles are only visible per user-defined privacy policies, meaning that a user can configure who can see what parts of the information and under what conditions. The core of the discovery in our implementation is based on the Apache Solr[2] search engine. Solr can be run as a single instance or in a distributed manner. User logins, profiles, and the related policies can be stored in distributed databases. Once a profile is searched, it can be loaded from the related database, so profiles can be maintained by several players.

6 Privacy and security analysis

The migration of the authentication task to the endpoint allows the framework to provide a consistent authentication method to any compliant service. However, this entails further measures of security and privacy assurance. For example, the IdModule should be setup to refresh the security tokens for users regularly and determine the target audience who may receive the identity assertion, with its user information. Each party performs authentication at the respective endpoint, regardless of the CSP, but more parties are now involved, when including independent IdPs and different CSPs.

The two identifiers—the GUID and the UserID—are used by the framework as static and correlated levels of identification. The universal level is defined by the GUID, which is accessible through a discovery engine. It ensures uniqueness and accessibility of the associated user. The UserID is specific to an administrative domain associated with a CSP. The UserID is also the user subscription ID at the CSP, which allows access to the hyperties that are provided by the CSP. The linking of multiple identities with the GUID and to a profile with user attributes adds resilience against identity

[2] Apache Lucene Project Website http://lucene.apache.org

theft, in the same manner as multi-factor authentication procedure. Users can use an independent IdP to manage several CSP-bound identifiers and link them to a single profile. The user can define under which strategy and which IdP the different subsets of CSP-bound identifiers may be allocated. As shown in Fig. 5, each user has a unique GUID and several CSP-bound identifiers maintained by the IdP that can be presented for authentication.

The relationship between the user, IdPs, and CSPs is determined by the manner of choosing them and the details of the subscription contract. IdPs and CSP have access to the "user digital life" that is marketable for advertising, and may be seen as infringement of privacy. A CSP has complete knowledge of all the communication activities of the UserID that it has allocated to the user. An IdP has knowledge of all the authentication requests that involve any of the identifiers that it manages, i.e., multiple CSPs' UserIDs for the same GUID. However, if the user subscribes to more than one IdP, no one IdP can have a full view of all the user's communication. Hence, a user strategy of using IdPs, but distributing CSP identifiers between several IdPs, can prevent one party acquiring full knowledge of all the user's activities.

The requirements for authentication should support different levels of user privacy and anonymity, such as untraceable identity, pseudonymous identity, and unlinkability [26]. Privacy features for disclosure or surveillance [27] are a well-discussed topic for Internet services, but little has been implemented. Our distributed identity framework considers such issues from the design phase onwards, to avoid current retrofitting issues, as seen on the web. In addition, the architecture enables user choice of IdPs, putting the user in full control of information that is shown to others, provided the IdP enforces agreed policies. The implemented discovery process allows configuring visibilities, ranging from being visible only to selected few users to being universally visible to all.

The screening of communication partners is likely to be even more attractive as threats and nuisance calls become more prevalent and more pernicious. In particular, cybercrimes based on misrepresentation to obtain sensitive information are fast growing. Furthermore, web calling services are used to distribute malicious content, viruses, and spywares. Therefore, in order to enhance security, methods of estimating user trustworthiness should be introduced in to web calling services.

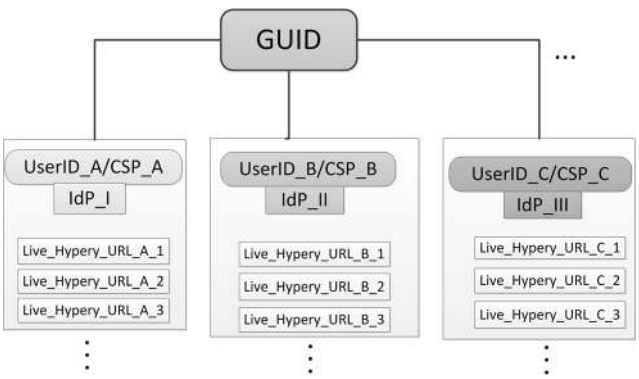

Fig. 5 Levels of user identification

7 Support services

In our framework, the Hyperties are created by and received from remote CSPs. To ensure the correct governance of the Hyperty at the endpoint, the downloaded Hyperties need to obey the policies defined by their respective CSPs, hence multiple policy rules need to be considered. Therefore, the framework must provide policy management component to manage rules and policies, obeying several CSPs, but coordinating between them and the users. Another supporting service is exploiting techniques borrowed from online social networks, such as the social graph, to link contact lists and enquire on unknown callers' reputation, using the framework graph connector service.

7.1 Policy management

CSPs provide several supporting services that maintain potentially sensitive information, the access to which must be controlled. The domain registry, for example, should be able to restrict user discovery according to preferences expressed by the user when subscribing to the service. In the same way, the information about the current live Hyperties maintained in the domain registry should be accessible in a controlled way. The correct governance within the framework is enforced using policies that make authorization decisions, such as who has access to what, at which time, and under which conditions. The delivery of information is controlled using a classical policy decision point (PDP) and policy enforcement point (PEP) structure [28]. The policy decision-making and enforcement are carried out independently by PDP and PEP components, which are driven by the policy engine. The policy engine acts as an access control point in the system: all messages originating from or delivered to Hyperties in the user device runtime environment are subject to interception and authorization by the policy engine. When a message is intercepted, the policies specified by the user are loaded and validated against that message. The reasonable mutualization of the PDPs is not suitable for the distributed nature of the CSP domain. To cope with the presence of multiple dedicated PDPs, distributed operation is used. We introduce a policy orchestrator that maintains global consistency between the different points of policy evaluation and enforcement. Different CSPs may need dynamic and time-critical negotiated policies to be applied to inter-domain sessions. Such negotiations can be carried out by a policy broker in the policy orchestrator. The description of the framework policies are

recorded by XACML, using the XML specification language, or Ponder [29]. Following [30], JSON can be a valuable and "fat-free" alternative to XML.

7.2 Graph connector

The framework includes a module to learn about previous connections and provides a method of contacting previous calling parties without having to search for them. By managing a list of known communication endpoints, users can stay connected to other users independent of their location or context. The graph connector acts as a local address book or contact list stored in a distributed manner. The distributed graph indicates not only friendship relations but also relations like similar taste in music, similarity in location traces, etc., thus forming additional tiers in the social graph based on common context and location [31]. The idea is to have different applications build on different edges of the graph.

This distributed graph information may also be used to estimate the trust level between users that have not previously interacted with each other. Receiving an incoming call from an unknown GUID, by checking the user's contacts, the framework runtime service at the endpoint can determine if there are mutual contacts with the caller, indicating a trustworthy relationship. In order to respect the user's privacy, hashing algorithms can be employed in order to mask identifiers or profile data. Existing research utilizing bloom filters when comparing user profiles while preserving privacy seems the most promising [32]. Using a bloom filter minimal data structure enables calculations to be made on smartphones with bandwidth and battery constraints. Users store the GUIDs of their direct contacts in such bloom filters. Users can look up a specific GUID (a caller, for example) to determine if this GUID has mutual direct contacts. Bloom filters allow probabilistic checks for set membership but do not allow direct lookup of data belonging to other users; hence, they provide the required information (e.g., matched mutual friends) while protecting privacy (e.g., not disclosing the whole list). It is also possible to set a privacy flag that that will prevent a particular GUID from being hashed into the bloom filter, for added selective security.

8 Conclusion

In this paper, we present a novel cross-domain identity and discovery framework that allow users to be discovered, identified, and authenticated across different service domains. The proposed solution is based on registering active users in their own service domain, but allowing the availability status to be searched by all participating CSPs, thus facilitating discovery of contactable users. The framework identity management is underpinned by correlating the service login identifiers that are allocated by the CSPs to a user-selectable, globally universal, service-independent identifier (GUID) that can be searched globally. The authentication procedure is P2P-based, conducted by the calling parties respective IdPs, and is decoupled from the service logic. The discovery, authentication, and contact list services are designed for controlled privacy. Hence, this framework is a new method of enabling cross-service domain communication that empowers user choice and supports privacy.

Acknowledgements This work has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 645342, project reTHINK.

References

1. A Bergkvist, Burnett DC, Jennings C (2015) WebRTC 1.0: real-time communication between browsers, W3C Working Draft, 10 February
2. Barnes R, Thomson M (2014) Browser-to-browser security assurances for WebRTC. IEEE Internet Comput 18(6):11–17
3. E Bertin, S Cubaud, S Tuffin, N Crespi, V Beltran (2013) WebRTC, the day after: what's next for conversational services? International Conference on Intelligence in Next Generation Networks (ICIN 2013)
4. I Javed et al. (2016) Global identity and reachability framework for interoperable P2P communication services, 19th Conference on Innovations in Clouds, Internet and Networks (ICIN 2016)
5. S Becot, E Bertin, J Crom, V Frey, S Tuffin (2015) Communication services in the Web era: how can Telco join the OTT hangout?, International Conference on Intelligence in Next Generation Networks (ICIN 2015)
6. Lampropoulos K, Sanchez D, Almenares F, Weik P, Denazis S (2010) Introducing a cross federation identity solution for converged network environments, principles, systems and applications of IP Telecommunications (IPTComm '10). ACM, New York, pp 1–11
7. Beltran V (2016) Characterization of web single sign-on protocols. IEEE Commun Mag 54(7):24–30
8. M Jones and D Hardt (2012) The OAuth 2.0 authorization framework: bearer token usage, IETF RFC6750
9. N Sakimura, J Bradley, M Jones, B Medeiros, C Mortimore (2014) OpenID connect Core 1.0, The OpenID Foundation
10. E Rescorla (2016) WebRTC security architecture, IETF internet draft, standards track
11. Beltran V, Bertin E, Crespi N (2014) User identity for WebRTC services: a matter of trust. IEEE Internet Comput 18(6):18–25
12. R Copeland (2009) Converging NGN wireline and mobile 3G networks with IMS. CRC Press, Talor & Francis Group, Boca Raton
13. L Li, W Chou, T Cai, Z Wang, Z Qiu Mirror presence: secure web identity resolution and call control for WebRTC, Proceedings of International Conference on Information Integration and Web-based Applications & Services (IIWAS 2013) ACM, New York, pp 523–532
14. S Göndör, H Hebbo (2014) SONIC: Towards seamless interaction in heterogeneous distributed OSN ecosystems, IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Larnaca
15. Cutillo L, Molva R, Strufe T (2009) Safebook: a privacy-preserving online social network leveraging on real-life trust. IEEE Commun Mag 47(12):94–101

16. A Bouabdallah Data models and interface specification of the framework, reTHINK project Deliverable. http://dx.doi.org/10.18153/RTH-645342-D2_2.
17. M Jones, J Bradley, N Sakimura (2015) JSON Web Token (JWT), IETF Standard
18. J-M Crom (2015) Management and security features specifications, reTHINK project Deliverable http://dx.doi.org/10.18153/RTH-645342-D4_1
19. I Javed, K Toumi, N Crespi, A Mohammadinejad Br2Br: a vector-based trust framework for WebRTC calling services. IEEE IEEE International Conference on High Performance Computing and Communications (HPCC 2016), 12–14 December, Sydney, Australia
20. J-M Crom Implementation of governance and identity management components, reTHINK Project Deliverable
21. I Friese, R Copeland, S Göndör, F Beierle, A Küpper, R Pereir and J-M Crom (2017) Cross-domain discovery of communication peers. Identity mapping and discovery services (IMaDS), IEEE European Conference on Networks and Communications (EuCNC)
22. S Göndör, F Beierle, S Sharhan, A Küpper (2016) Distributed and domain-independent identity management for user profiles in the SONIC Online Social Network Federation, International Conference on Computational Social Networks, Springer
23. S Göndör, F Beierle, E Küçükbayraktar, H Hebbo, S Sherhan, A Küpper (2015) Towards migration of user profiles in the SONIC Online Social Network Federation, International Multi-Conference on Computing in the Global Information Technology (ICCGI)
24. Alliance, Open Mobile (2013) Lightweight machine to machine technical specification. Technical Specification OMA-TS-LightweightM2M-V1
25. Z Shelby (2012) Constrained RESTful environments (CoRE) link format, IETF standard
26. R Copeland, K Corre, I Friese, S El Jaouhari (2016) Requirements for trust and privacy in WebRTC peer-to-peer Authentication IETF internet draft
27. A Cooper et al. (2013) Privacy considerations for Internet protocols, IETF RFC 6973
28. R Yavatkar, D Pendarakis, R Guerin (1999) A framework for policy-based admission control, IETF RFC 2753
29. Damianou N, Dulay N, Lupu E, Sloman M (2001) The ponder policy specification language, policies for distributed systems and networks. Springer, Berlin, pp 18–38
30. D Crockford (2006) JSON: the fat-free alternative to XML, XML 2006 Conference, Boston
31. F Beierle, S Göndör, A Küpper (2015) Towards a three-tiered social graph in decentralized online social networks, Workshop on Hot Topics in Planet-scale mobile computing and online Social networking (HotPOST ‘15), ACM
32. Alaggan M, Gambs S, Kermarrec A (2012) BLIP: non-interactive differentially-private similarity computation on bloom filters, symposium on self-stabilizing systems. Springer, Berlin