User Controlled Trust and Security Level of Web Real-Time Communications

Kevin CORRE

PhD Defense

May 31st, 2018

Maryline LAURENT, TELECOM SudParis, Reléctrice

Yvon KERMARREC, IMT Atlantique, Relécteur

Walter RUDAMETKIN, Université de Lille, Examinateur

Dominique HAZAEL-MASSIEUX, W3C, Examinateur

Vincent FREY, Orange Labs, Encadrant Industriel

Olivier BARAIS, Université de Rennes 1, Directeur de thèse

Gerson SUNYÉ, Université de Nantes, co-Directeur de thèse

orange™

DiverSE
Diversity-Centric Software Engineering

UMR IRISA

Inria
INVENTEURS DU MONDE NUMÉRIQUE

UNIVERSITÉ DE RENNES 1

Voice Over IP with WebRTC

Context

Voice over IP: the techniques to **communicate** using voice or voice and video over any compatible **IP networks**

WebRTC: W3C **API** and IETF **protocols profiles** for Web based real-time audio, video, and data communication capabilities

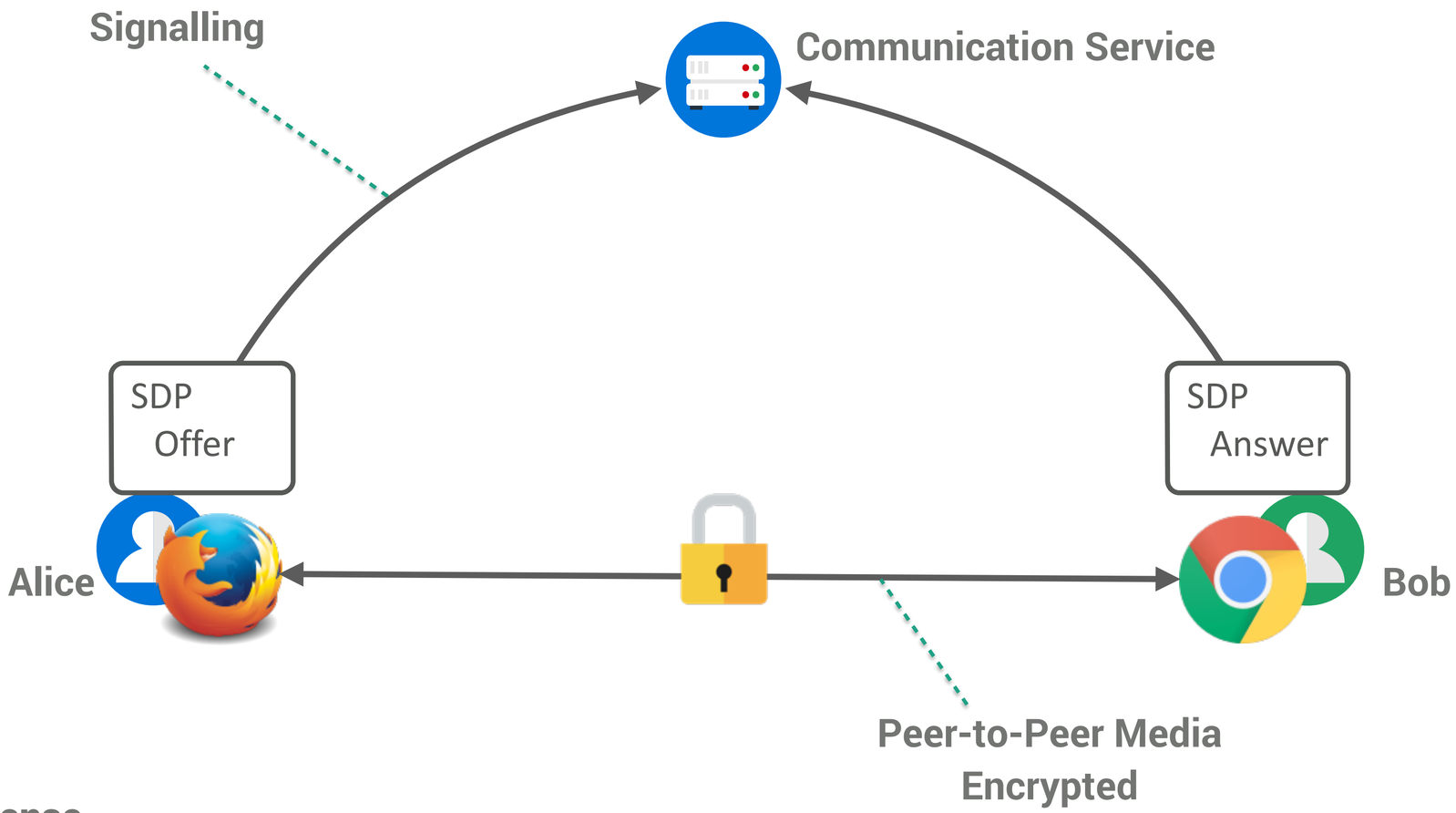
Use cases: VoIP, gaming, streaming, data sharing

Alice and **Bob**: two users of a WebRTC application

Communication Service: the web server providing the WebRTC application

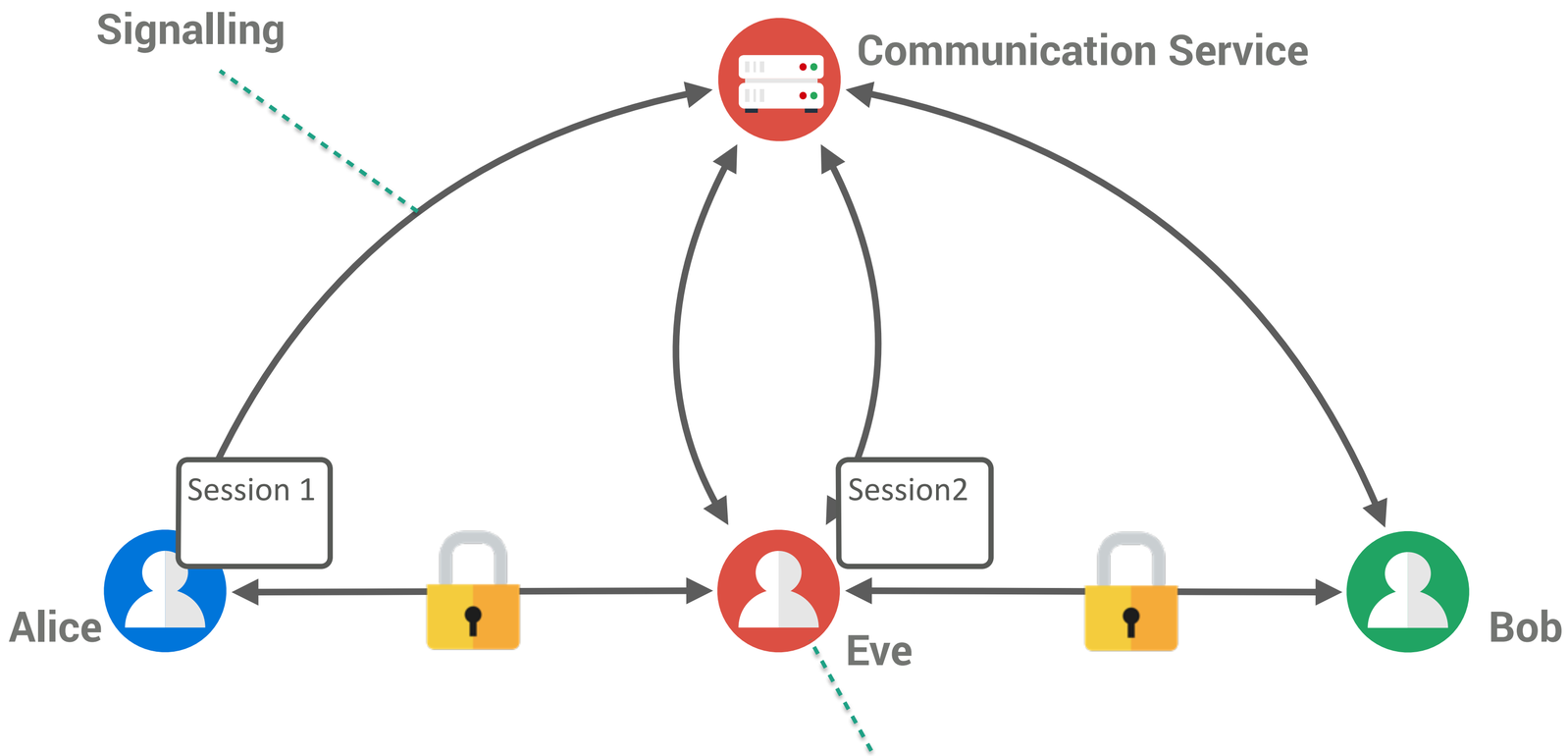
WebRTC Signalling

Establishing a Peer-to-Peer Communication between two browsers



Can Users Trust Any WebRTC Application?

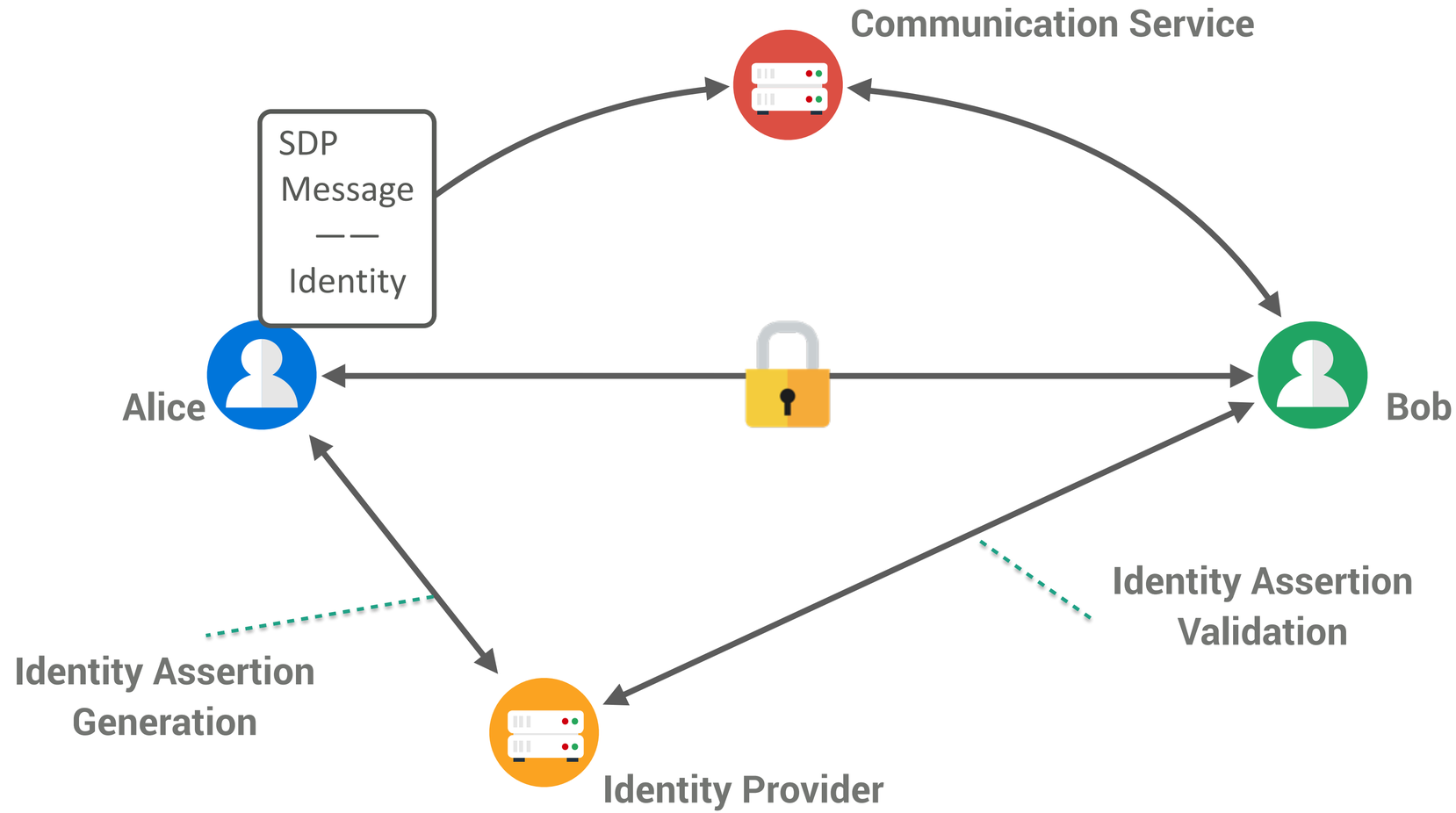
Malicious, or Corrupted Communication Service



The Communication Service can setup a Man-in-the-Middle Attack (MitM)

WebRTC

Identity Architecture : Overview



WebRTC

Identity Architecture : Identity Assertion

```
v=0
o=mozilla...THIS_IS_SDPARTA-54.0.1 5897145307417630851 0 IN IP4 0.0.0.0
[...]
a=fingerprint:sha-256 33:B1:D7:4B:29:29:29:AA:87:01:47:B3:59:41:[...]5D
a=group:BUNDLE sdparta_0 sdparta_1
a=ice-options:trickle
a=identity:eyJhc3NIcnRpb24iOiJleUowZVhBaU9pSktWMVFpTENKaGJHY2lPaUpTVX
pJMU5pSXNJbXAzYXlJNmV5SnJkSGtpT2lKU1UwRWlMQ0p1SWpvaWVHNWxNblpw
[…]
pZHAiOnsiZG9tYWluIjoiZW5lcmd5cS5pZHAucmV0aGluay5vcmFuZ2UtbGFicy5mciIsI
nByb3RvY29sIjoicmV0aGluay1vaWRjIn19
```

identity is a base 64 encoded JSON

```
{
  "assertion": "eyJhc3 …. jIn19" ,
  "idp": {
    "domain": "orange.fr",
    "protocol": "default"
  }
}
```

validateAssertion returns:

```
{
  "identity": "alice@orange.fr" ,
  "content": "fingerprint:sha-256 33:B1[…]5D"
}
```

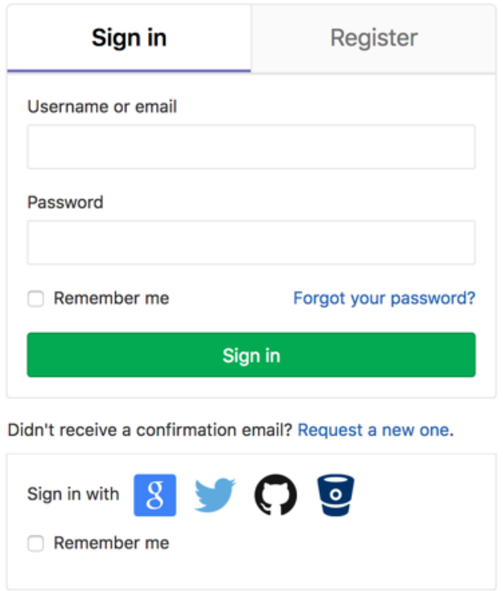
Binds Alice identity to Alice's fingerprint: a sha-256 hash of Alice's public key

Identity Provider

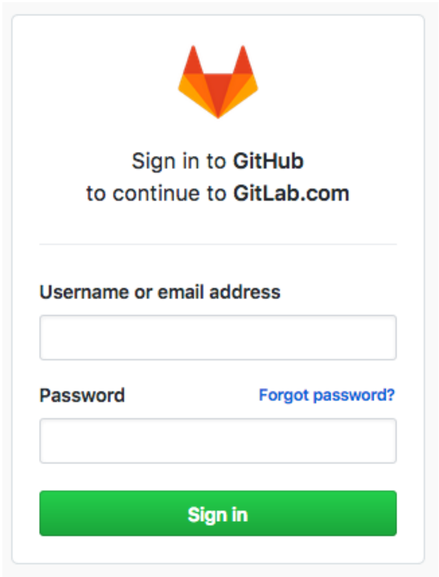
A New Actor in the Communication Setup

Identity Provider: the web server providing an authentication delegation service to other web application

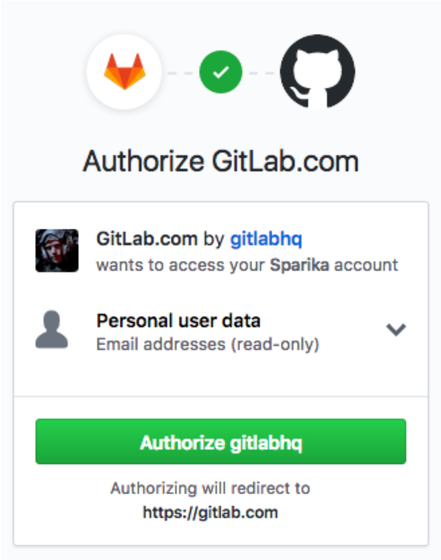
Password fatigue



Authentication

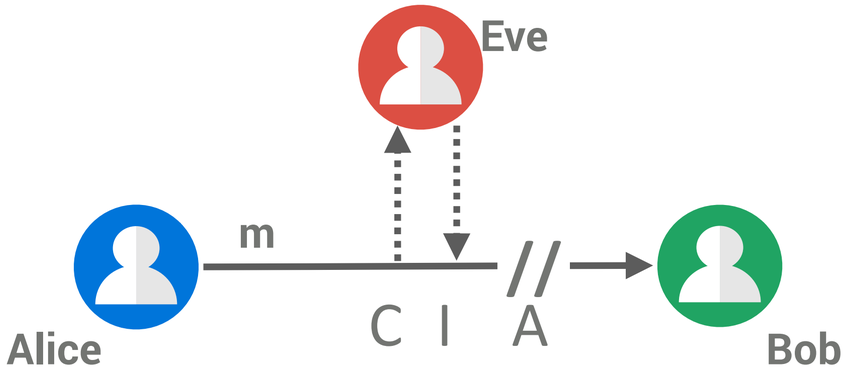


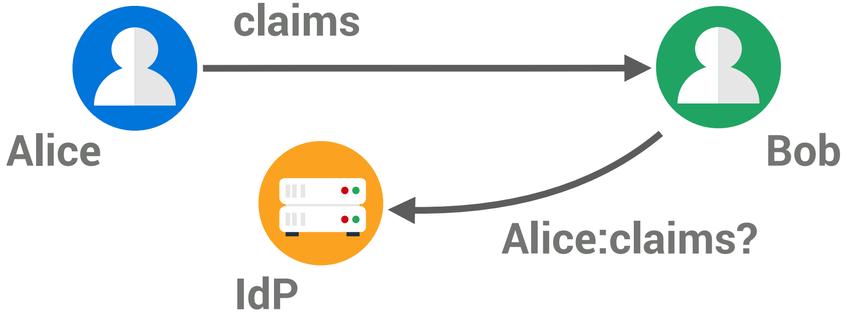
Authorization



Security, Authenticity

Some Definitions





Security is usually defined as **Confidentiality**, **Integrity**, and **Availability**

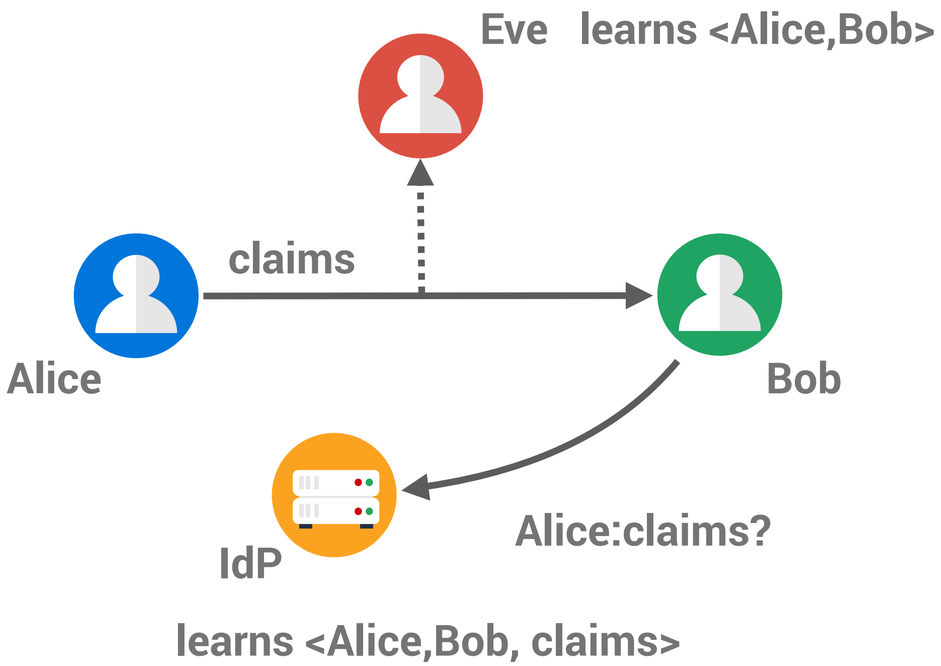
Claims: **identifier**, name, birthdate, access rightname, **fingerprint**, …

Identity: a set of claims

Authentication delegation: letting a trusted **third party assert the validity of identity claims**

Security, Authenticity, and Privacy

Some Definitions



Privacy threats include **surveillance**, correlation, **identification**, **secondary use**, disclosure, exclusion, …

Concern regarding privacy **is drastically increasing** following recent privacy breaches

User-Chosen Independent Identity

Industrial Objective: The Interoperability of TelCo Services for OTT Communications

Ultimately, **users rely on [to be] trusted-actors** to ensure the security, authenticity, and privacy of their communications

We claim that in order to trust their communications, **users must have a choice** regarding the configuration of their communication setup

reTHINK H2020 project, D2.1:

> *« In reTHINK, the aim is to provide **identity** that is **independent of both the front-end applications and the communication providers**, using an independent and unique identifier. This identity should be managed by the user, not the service provider, and is **verified by a user-chosen independent trusted entity**. »*

OUTLINE

The Agenda for Today

1. WebRTC Security and Identity Arch.
Context

2. STATE OF THE ART
on VoIP and WebRTC Security

3. RESEARCH QUESTIONS

4. PRIVACY in WebRTC ID. ARCH.
Contribution 1

5. CONTROLLING WebRTC ID. ARCH.
Contribution 2

6. MODELLING WebRTC ID. ARCH.
Contribution 3

7. CONCLUSION and PERSPECTIVES

Our State of the Art Survey

WebRTC Security 2012-2017

Keromitys published a survey on **« VoIP Security » in 2012**, classifying and reviewing 245 articles

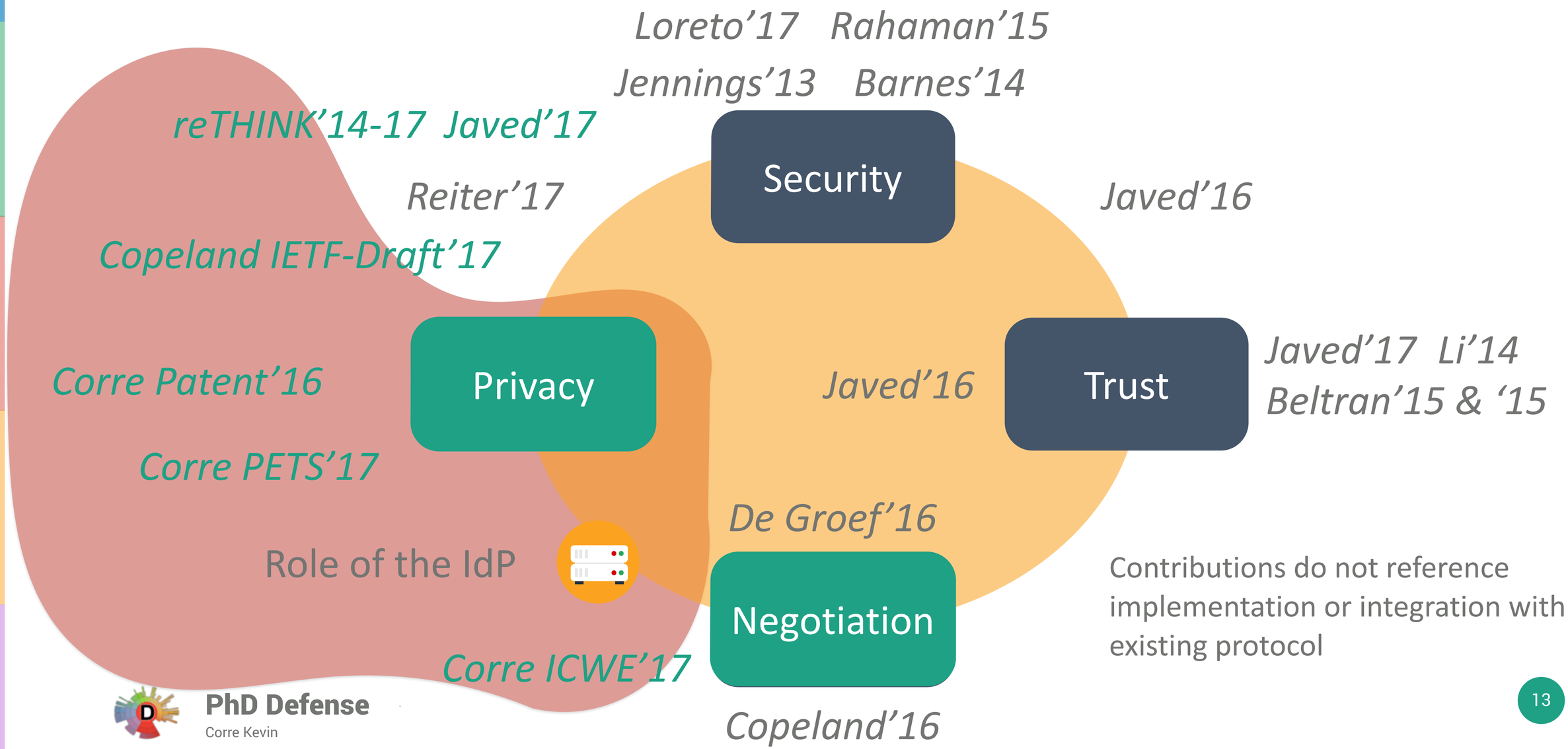
WebRTC W3C Working Group was created in may 2011 and the **security architecture first draft** was published **in January 2012**

We survey **VoIP security research since 2012 to 2017**

We **collect and classify 208** articles based on title and abstract and then review the **25 articles dealing with WebRTC security**

Security of the WebRTC Identity Architecture

State of the Art on WebRTC Security for 2012-2017



Contributions do not reference implementation or integration with existing protocol

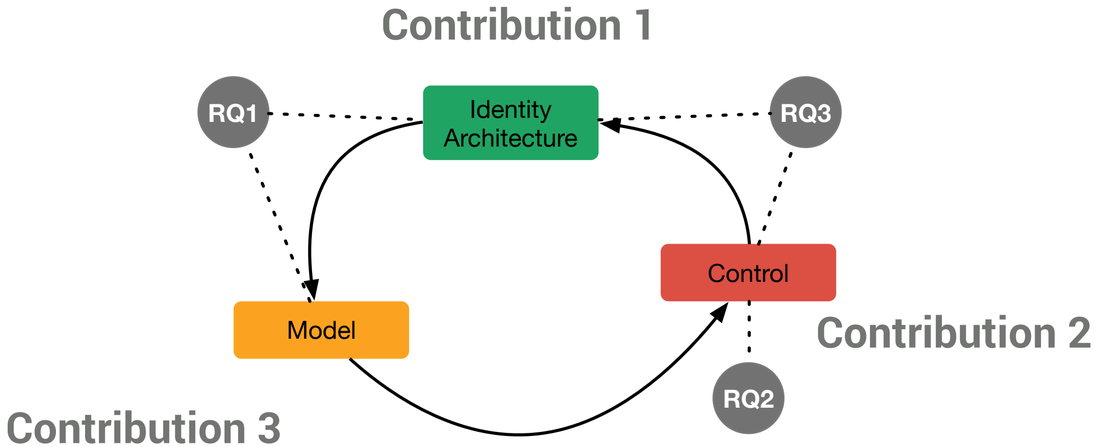
Research Questions

The Problem We Are Addressing

RQ1: What are the risks for the user of a WebRTC session and **which abstractions can we use to show these risks to the user?**

RQ2: Can we act on a WebRTC session to **raise the trust and security level?**

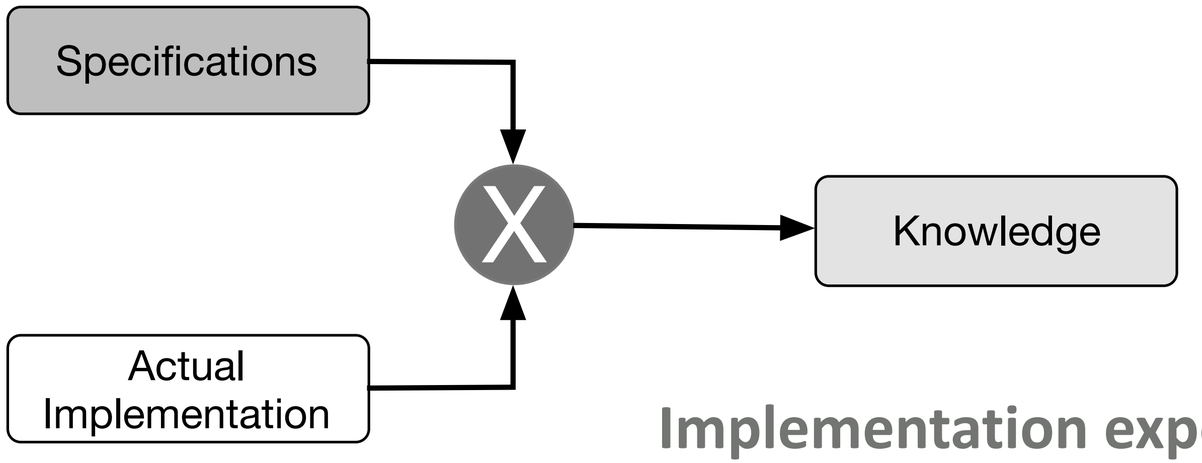
RQ3: Can we let **users chose actors they trust** to participate in the communication setup?



Privacy / Role of the IdP

Methodology

An Empirical Methodology for Studying WebRTC Security



Implementation experiments:

- Reveal implementation and integration issues

Deployment surveys:

- Demonstrate if and how a feature is used
- Measure a feature's interest in the community

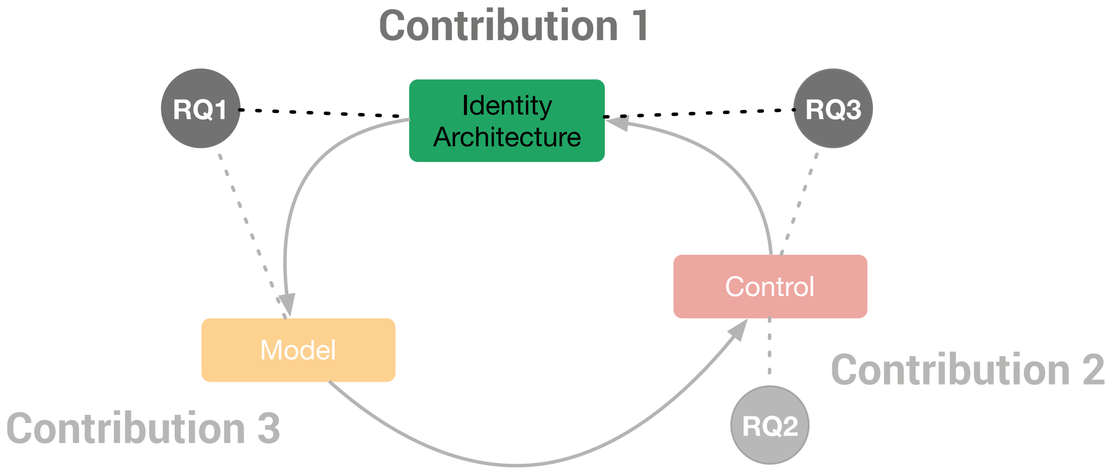
Privacy in the WebRTC Identity Architecture

Contribution 1

RQ1: What are the risks for the user of a WebRTC session and **which abstractions can we use to show these risks to the user?**

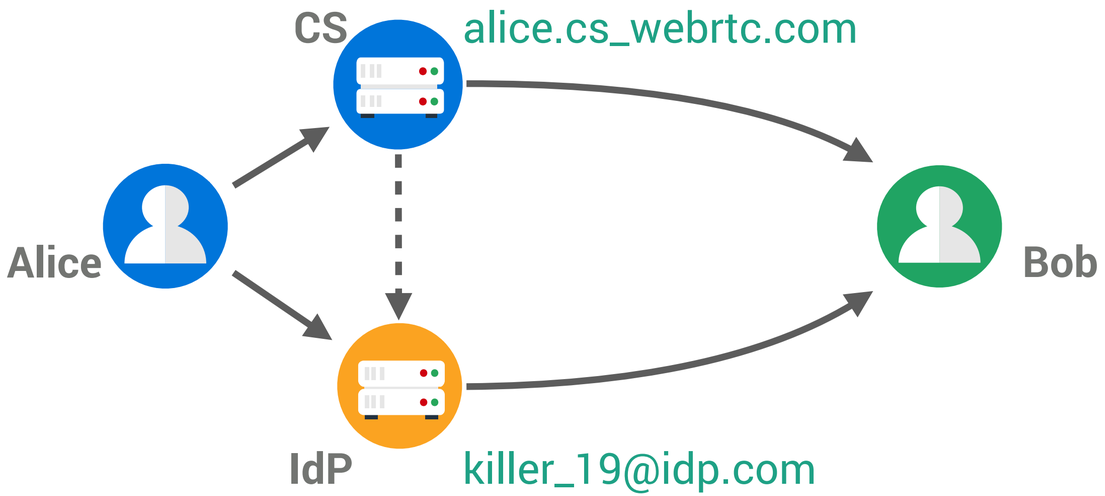
RQ2: Can we act on a WebRTC session to **raise the trust and security level?**

RQ3: Can we let **users chose actors they trust** to participate in the communication setup?



The Identity Continuity Principle

A Potential Privacy Issue



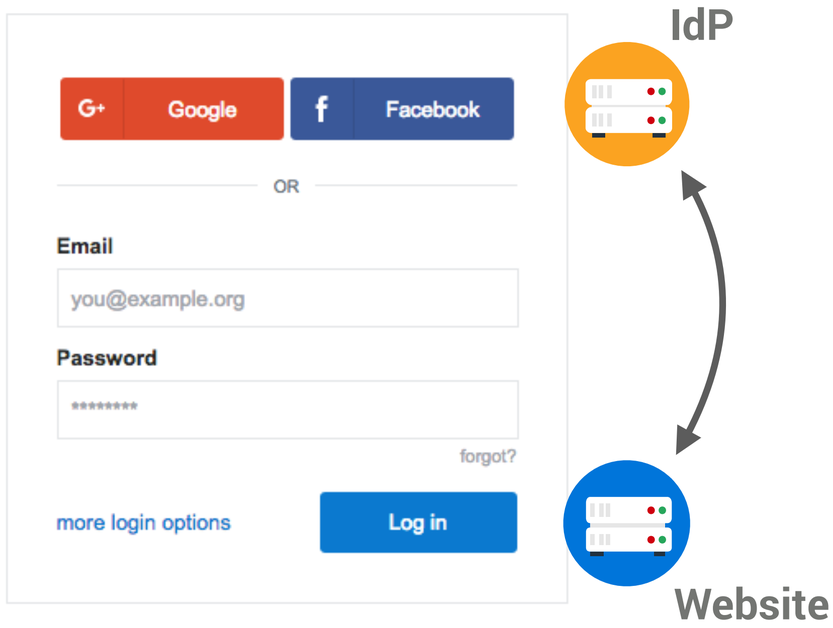
1. Alice's CS responsible for configuring the IdP

2. Alice's identities in CS context and in WebRTC context must be coherent to Bob

The **choice of an IdP is limited** in the same way **for WebRTC as it is on the** general **Web**

Users Cannot Choose their Identity on the Web

Which limits trust on the Web.



On the Web, users are presented with a very limited choice of IdP

Vapen'*15* reports that **47% of 77 websites offers only one IdP** and only 19% offers 4 or more IdPs

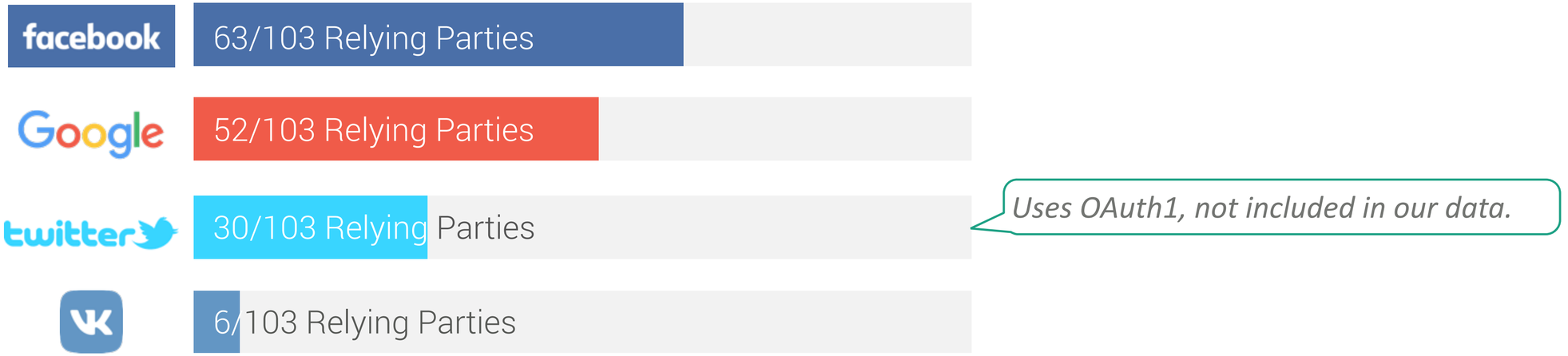
Authentication delegation is a practical architecture for security on the Web, however **the domination of a few IdPs is a privacy issues for end-users**

OAuth2 and OpenID Connect Data Collection

Overview

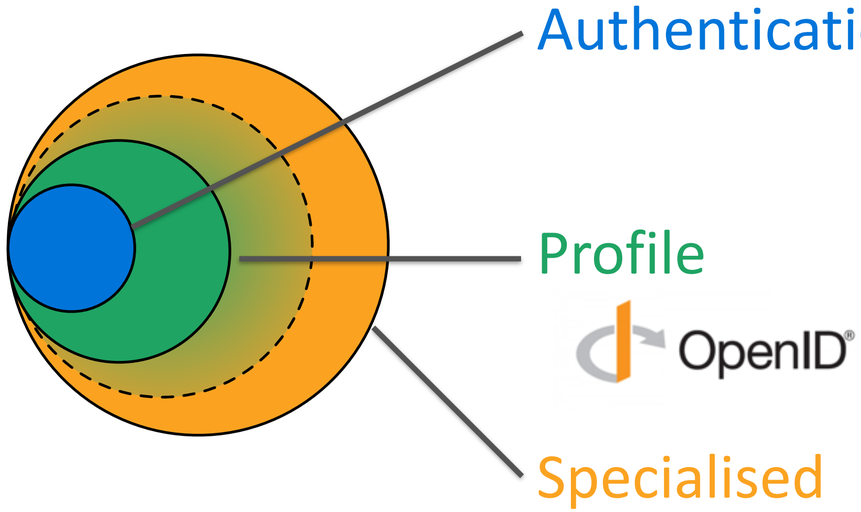
Why can’t users choose their IdP? **Could users choose any IdP?**

We collect OAuth2/OIDC usages on websites from alexa.com top-500
Out of 500 websites, we collected **103 OAuth2/OIDC Relying Parties**, using **23 OAuth2/OIDC providers**



Websites Require Different Kind of Authorization

Authorization Classes



https://github.com/login/oauth/authorize?
scope=user:email

https://www.facebook.com/dialog/oauth?
scope=email,user_birthday,user_likes,
user_friends,publish_actions

(MIN, MAX) classification

https://accounts.google.com/o/oauth2/auth?
scope=/auth/plus.login,/auth/
userinfo.email

RQ3.1 Do Websites Require Specialised API?

Our results

Min/Max Classes	Observed
Authentication/-	10% (10)
Authentication/Auth	1% (1)
Authentication/Profile	9% (9)
Authentication/Special	6% (6)
Profile/-	13% (13)
Profile/Profile	2% (2)
Profile/Special	17% (18)
Specialised/-	26% (27)
Specialised/Special	5% (5)
No Scope	11% (11)
Total	100% (102)

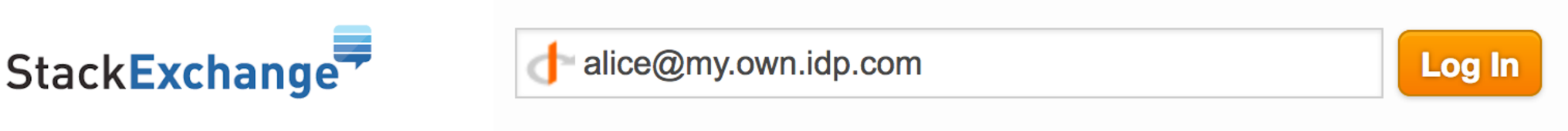
1. **58% do not require specialised API** and data, and **could accept any IdP** from an authorization point of view

2. **OIDC standardises user profile** scopes and data, but it is **scarcely implemented**

RQ3.2 Do IdPs Offer Dynamic Discovery ?

Our results

Dynamic discovery let users select their own IdP



OpenID Connect discovery JSON metadata should be available at

/.well-known/openid-configuration

No implementation of OIDC discovery metadata on IdPs

No implementation of OIDC discovery form on websites

Privacy in the WebRTC Identity Architecture

Contribution 1

RQ1: What are the risks for the user of a WebRTC session and which abstractions can we use to show these risks to the user?

The IdP can gather critical call informations

Users do not have much choice regarding their IdP in WebRTC

RQ3: Can we let users chose actors they trust to participate in the communication setup?

58% of websites only require authentication or profile authorization

OIDC -standard profile, discovery- is scarcely implemented

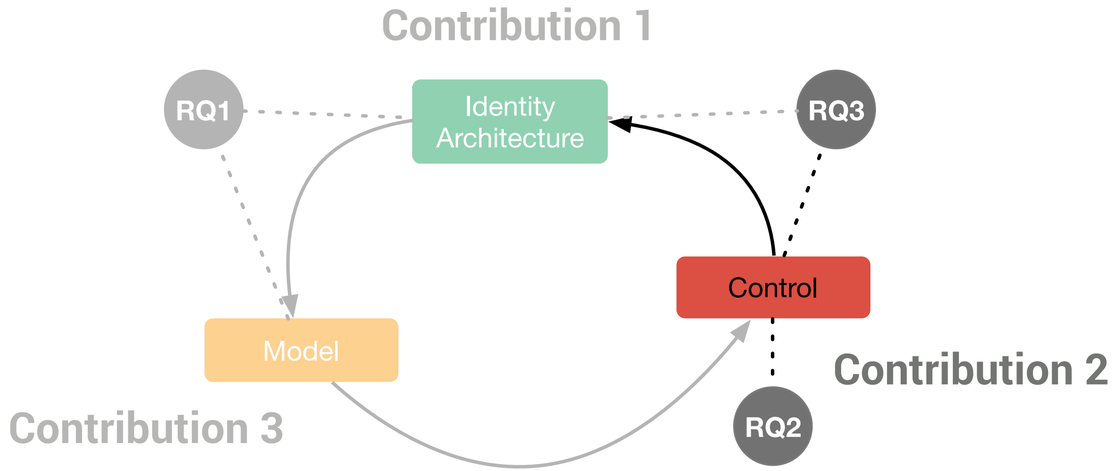
Controlling WebRTC Identity Parameters

Contribution 2

RQ1: What are the risks for the user of a WebRTC session and **which abstractions can we use to show these risks to the user?**

RQ2: Can we act on a WebRTC session to **raise the trust and security level?**

RQ3: Can we let **users chose actors they trust** to participate in the communication setup?

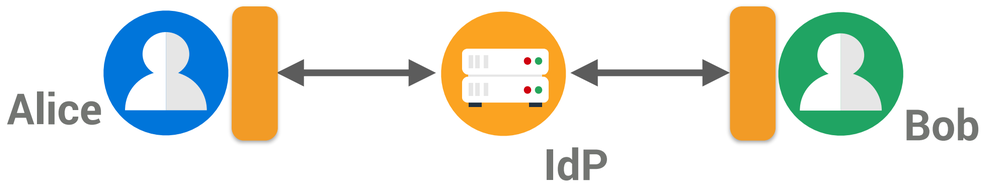


Giving More Control to WebRTC Users

Contribution 2

Claim: **a trust decision (to trust) implies that a choice is possible**:

- Alice should be able to choose her IdP,
- Bob may want to have some control too



RQ2.1: How to let users negotiate the other peer's identity parameters?

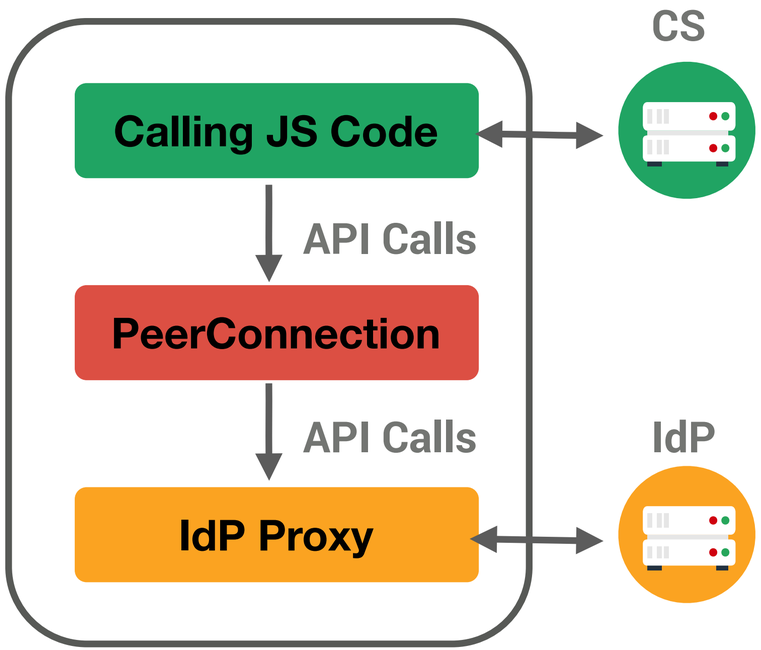
RQ3.4: Can we leverage the WebRTC identity architecture to let users chose their IdP for user-to-server authentication?

WebRTC IdP Proxy

WebRTC Identity Discovery

IdP Proxy standard location is ***DOMAIN/.well-known/idp-proxy/PROTOCOL***

*This **standard location** acts as **a discovery mechanism***

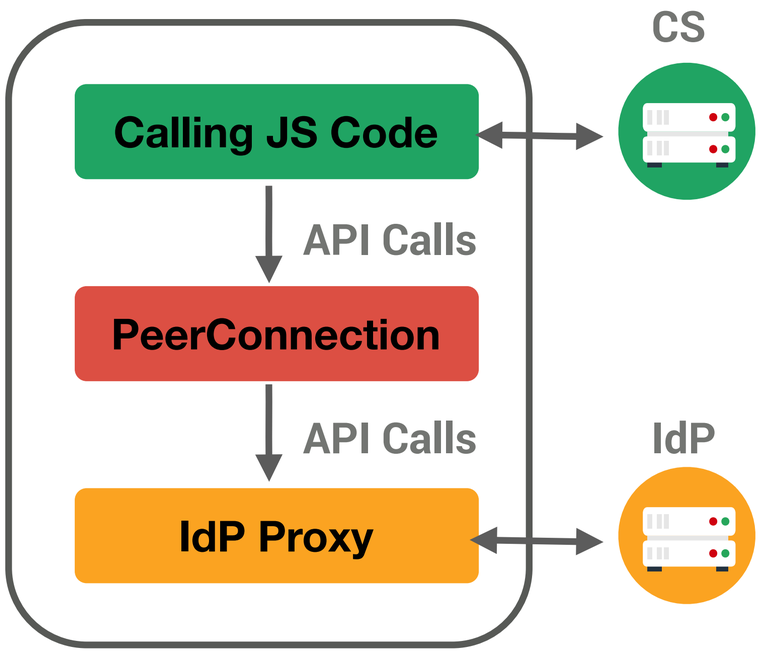



```
{
  "assertion": "eyJhc3 …. jIn19" ,
  "idp": {
    "domain": "orange.fr",
    "protocol": "default"
  }
}
```

https://orange.fr/.well-known/idp-proxy/default

WebRTC IdP Proxy

The CORE Component of the WebRTC Identity and Security Architecture



The IdP Proxy serves as an **authentication protocol abstraction layer**

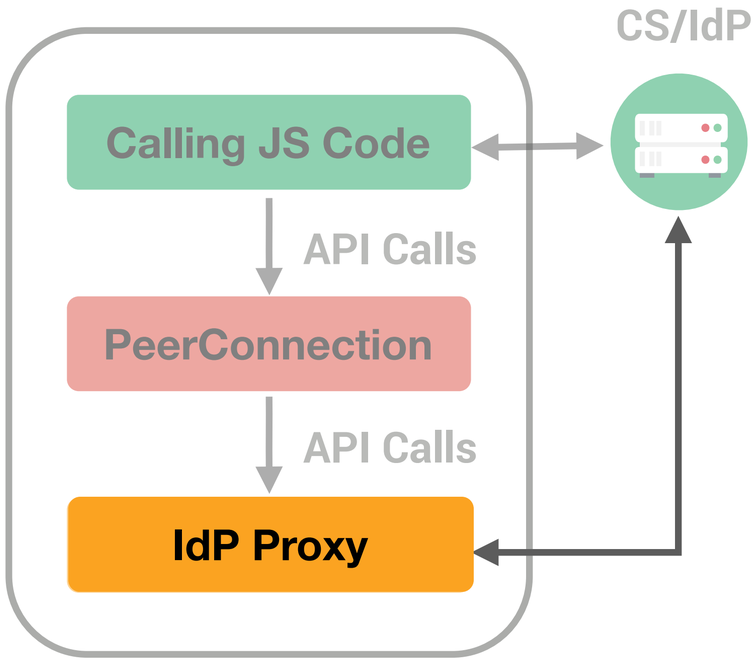
```
generateAssertion(fingerprint) -> identityAssertion

validateAssertion(identityAssertion) -> identity, fingerprint
```

Implementing the Missing Part

Local Authentication Scenario

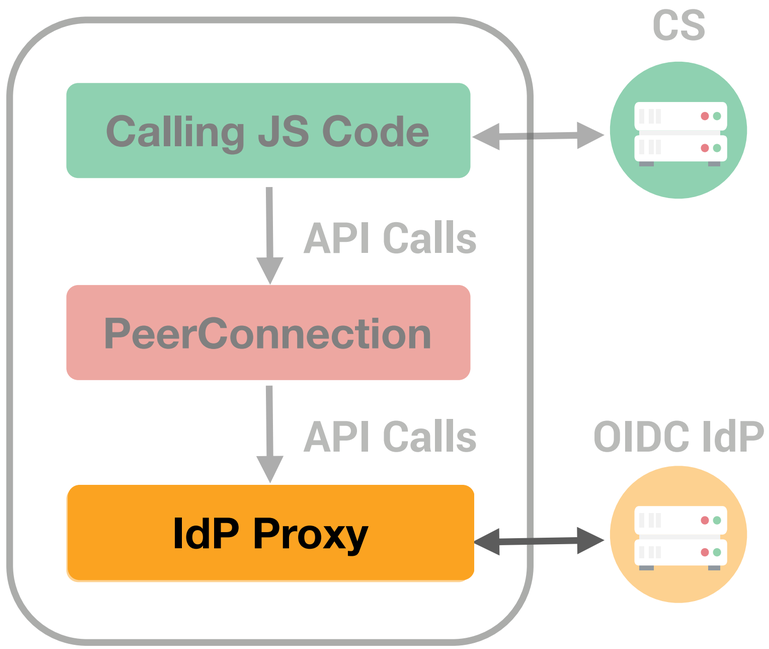
Local Authentication Scenario

- The **CS plays the IdP's role**
- Useful in multi CS architecture



OIDC Sketched in WebRTC annex

- **Require modification of OIDC**



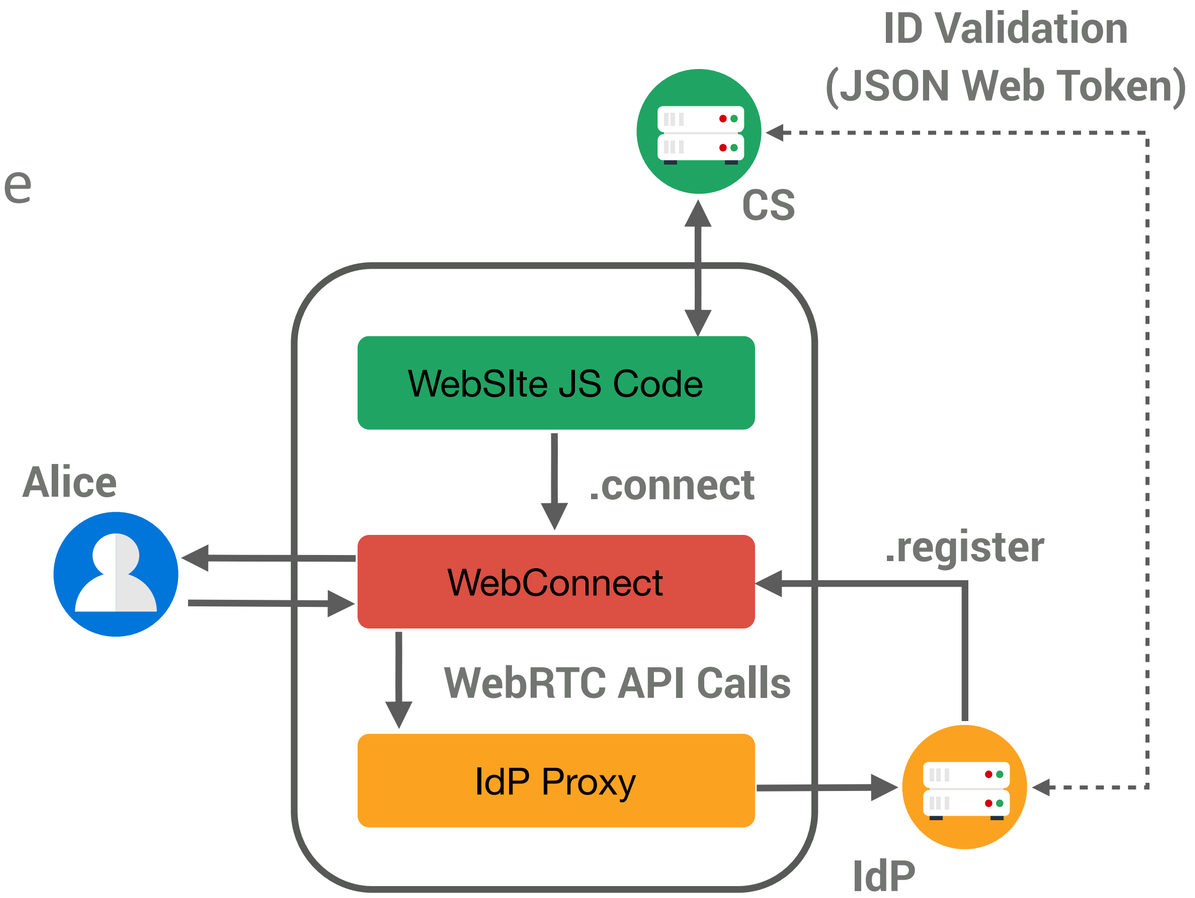
WebConnect Web API

Leveraging WebRTC Identity Architecture for User-to-Server Authentication

WebConnect provides an API and a graphical user interface, on top of the WebRTC Identity Architecture

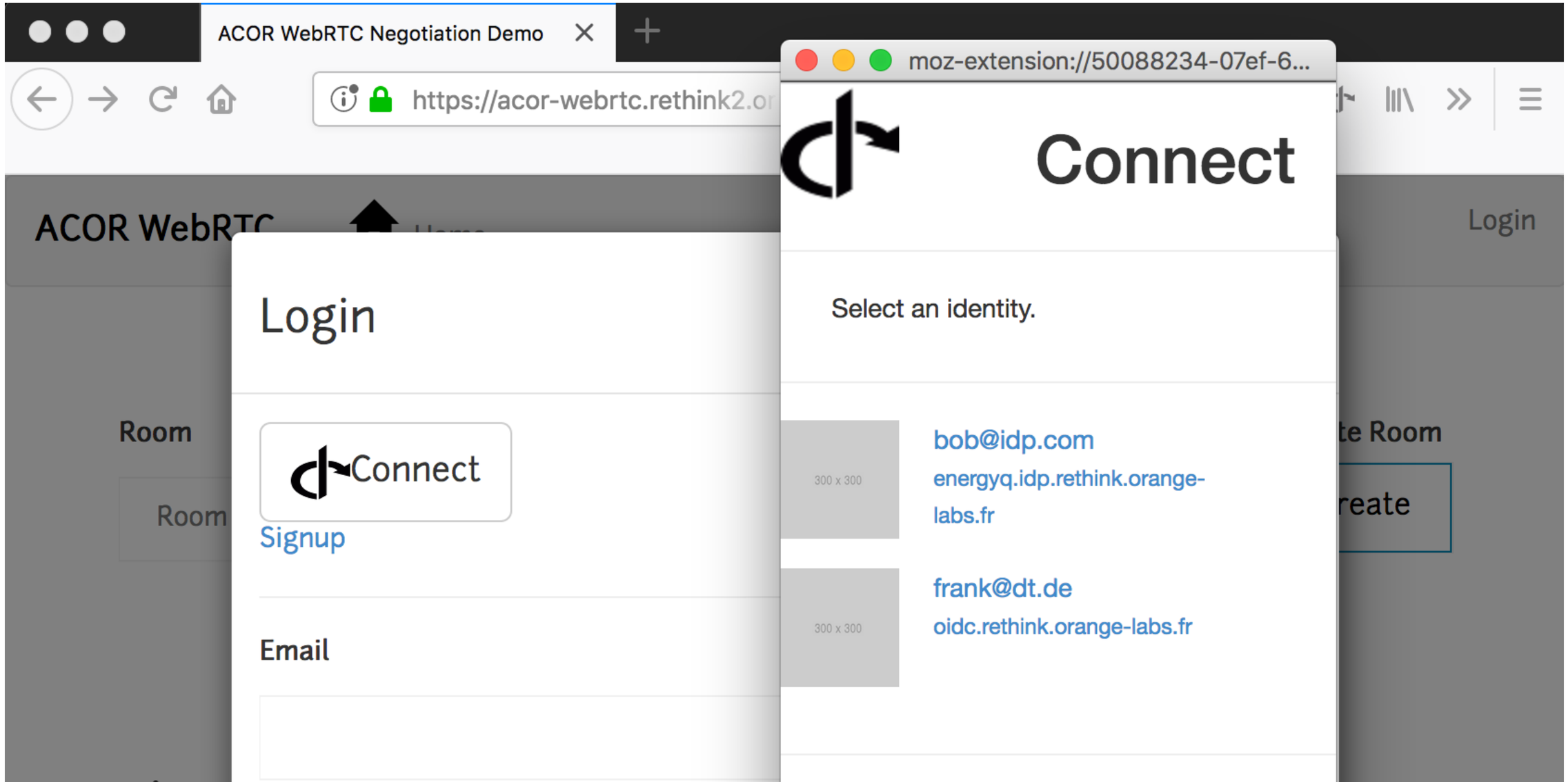
A web browser extension simulate the following web API:

```
void register (String iss,
         String proxy,
         String sub,
         String name,
         String picture);

Promise<JWT> connect(Object request);
```



WebConnect: a Browser-based Identity Metasystem

The User Experience

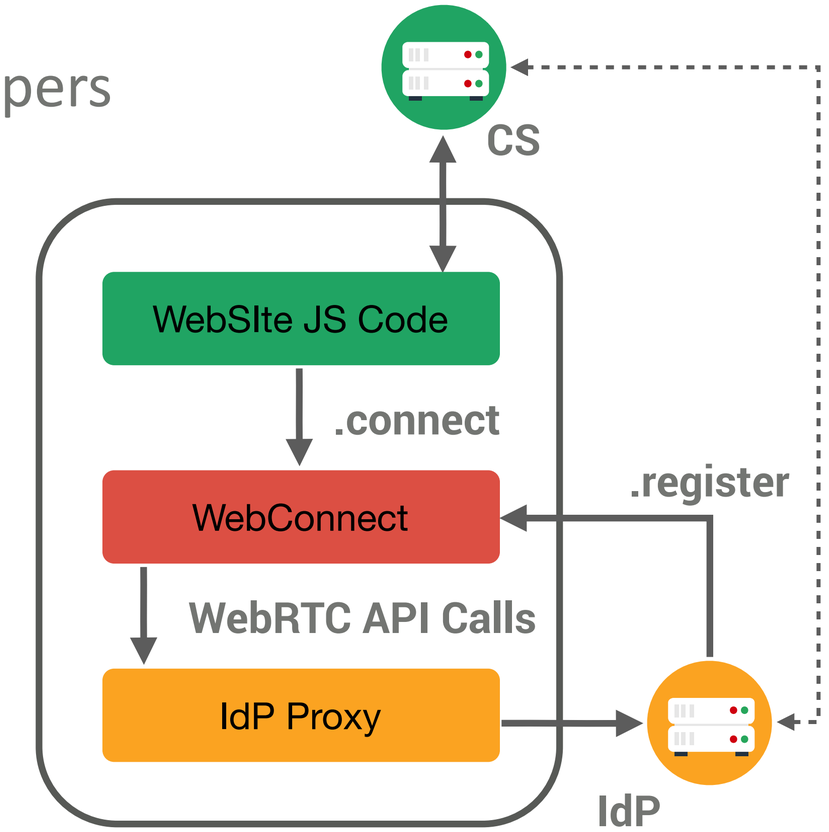


WebConnect Developer Usability

Evaluation

Implementation work is a constraint for web developers

Module	Total code lines	New code lines
Firefox Addon	0	417
IdP Proxy	0	197
Client site (.js)	693	66
Client site (.conf)	457	70
Passport JWS	1242	60
Total	2392	810



WebConnect integration to an existing website is simple: **under 200 lines of code** using a library such as Passport

WebConnect Privacy

Evaluation

This solution allows users to select **any** compatible and **trusted IdP to authenticate** on a website offering WebConnect

For instance, a user could choose an IdP implementing **privacy preserving solutions**, **privacy preserving policies**, or **self-host its own IdP**

Privacy by Design Foundation

Cozy Cloud

WebID-TLS

WebConnect Security

Evaluation

WebConnect do not follow OIDC standard

Website can authenticate the IdP without prior registration

The IdP cannot authenticate the website and cannot control authorization delegation

Same limit as for WebRTC Identity Architecture

Security relies on secure implementation and control of the API by the browser

Controlling WebRTC Identity Parameters

Contribution 2

RQ2: Can we act on a WebRTC session to raise the trust and security level?

- SDP attribute extension to negotiate identity parameters
- The WebRTC API does not handle authentication assurance level

RQ3: Can we let users chose actors they trust to participate in the communication setup?

- **The IdP Proxy provides discovery and authentication protocol abstraction**
- **We use the IdP Proxy in a user-server authentication**

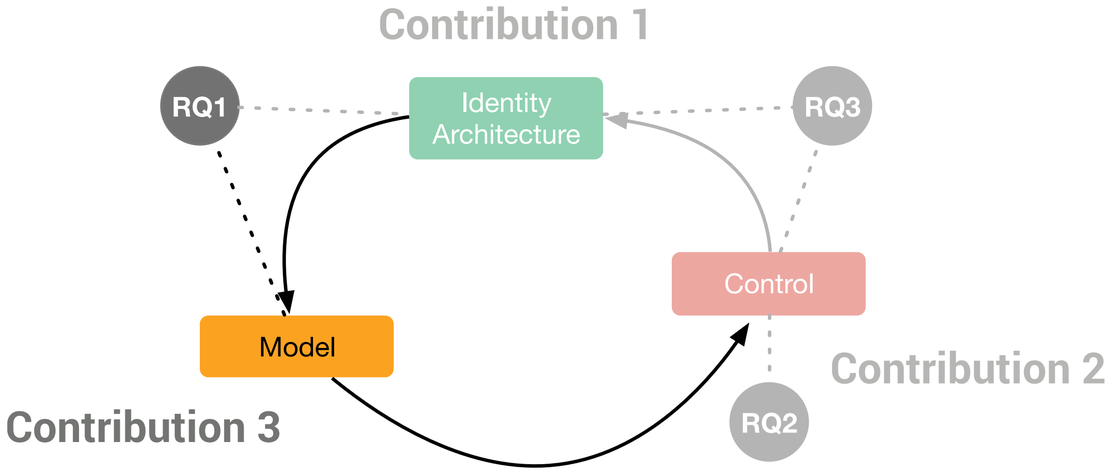
Modelling Trust and Security of a WebRTC Session

Contribution 3

RQ1: What are the risks for the user of a WebRTC session and **which abstractions can we use to show these risks to the user?**

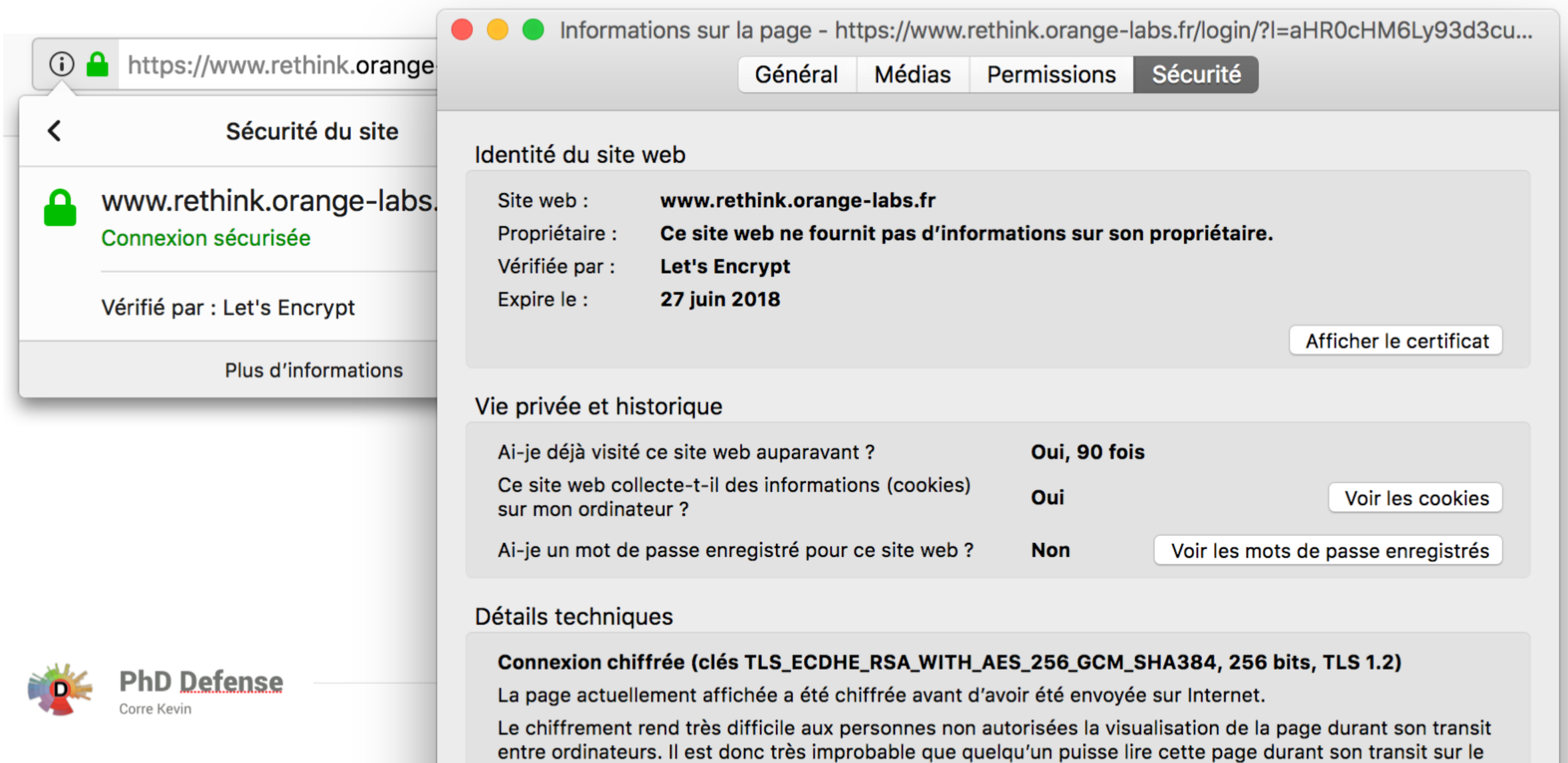
RQ2: Can we act on a WebRTC session to **raise the trust and security level?**

RQ3: Can we let **users chose actors they trust** to participate in the communication setup?



Security on the Web vs WebRTC

Advertising Security Configurations to Users

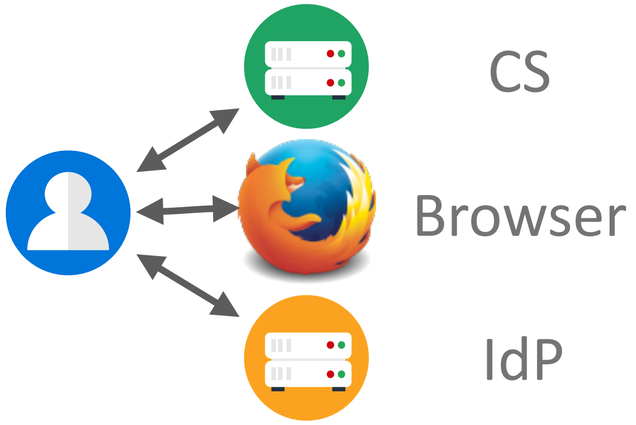


Preliminary Survey on Advanced Users

Validating our Approach

"the model [...] is interesting for people who do not have much knowledge in the field but still are interested in knowing how it roughly works"

Which actor <**browser**, **CS**, **IdP**> should play the role of the **trusted recommendation source?**



Conclusion

Our results

Contribution 1: **Demonstrated the privacy risks related to the role of IdP** in the WebRTC communication setup and some of the technical reasons for this situation

Contribution 2: **Proposed** two **solutions** for allowing users to choose which actors are allowed to participate in the peer authentication

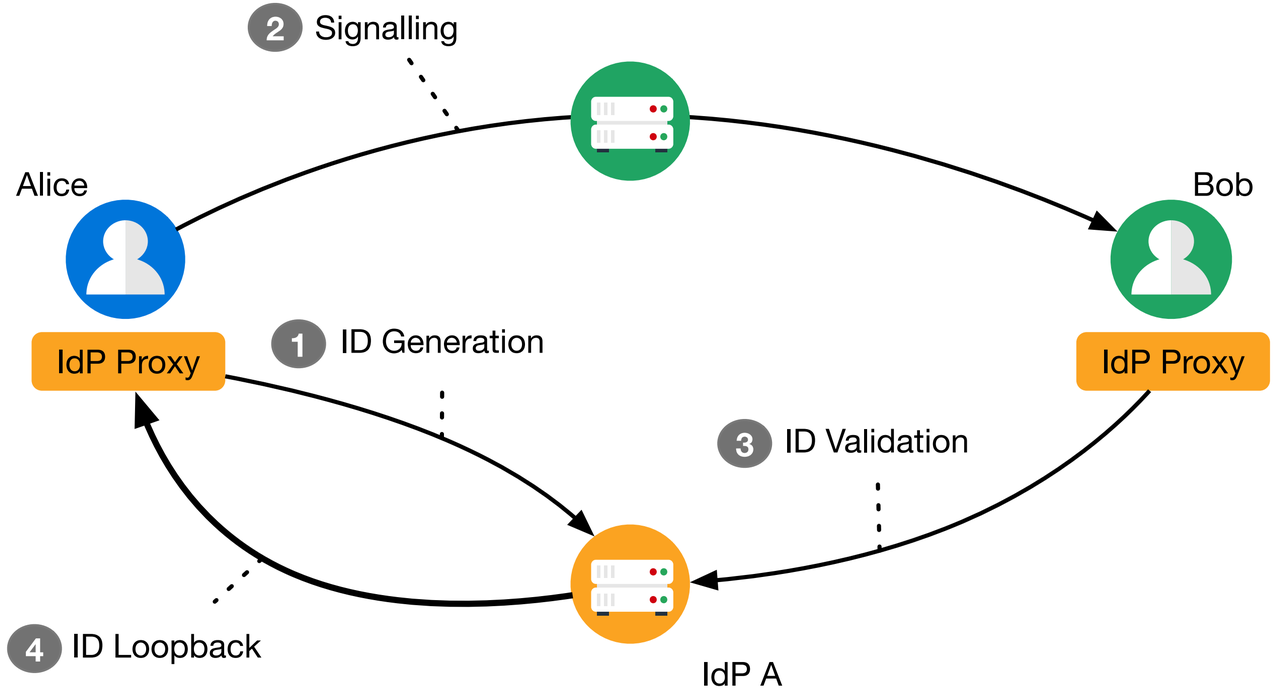
Contribution 3: **Proposed** a **model** of a **WebRTC session security** targeted at [advanced] **users** to facilitate the comprehension of their security configuration to help them decide/choose

We validated our contributions through implementation or deployment survey

WebRTC Identity Proxy Interface

Short-Term Perspective

In the current architecture **Alice does not get feedback** from her IdP <**the trusted actor**>
i.e. Alice does not know to whom she is authenticating



We propose that IdPA loopback on Alice when the identity assertion is validated

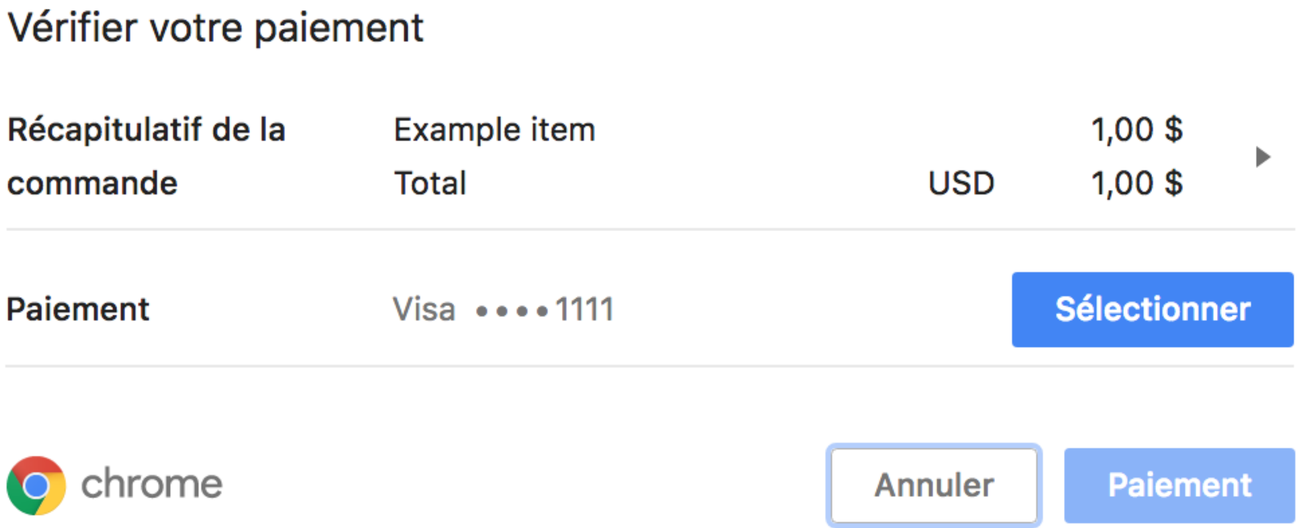
How can IdP A authenticate Bob?

Authenticating with the WebPayment API

Mid and Long Term Perspectives

W3C **Web Payment Working Group, 3 may 2018**:

*« Right now there seem to be no major obstacles to resolving our list of issues for exiting Candidate Recommendation and **advancing Payment Request API to Recommendation by Q4 of this year.** »*



What if we request payment for 0€?

Is there an API to abstract payment, authentication, and authorization?

Publications

Our results

[1] Kevin Corre, Simon Bécot, Olivier Barais, and Gerson Sunyé. **"A WebRTC Exten- sion to Allow Identity Negotiation at Runtime"**. *Web Engineering - 17th International Conference, ICWE 2017, Rome, Italy, June 5-8, 2017*

[2] Kevin Corre, Olivier Barais, Gerson Sunyé, Vincent Frey, and Jean-Michel Crom. **"Why can't users choose their identity providers on the web?"** *PoPETs 2017.3 (2017), pp. 72–86*

[3] Rebecca Copeland, Kevin Corre, Ingo Friese, and Saad El Jaouhari. **Requirements for Trust and Privacy in WebRTC Peer-to-peer Authentication**. *Internet-Draft draft-copeland-rtcweb-p2p-idp-auth-00. IETF Secretariat, Sept. 2016*

[4] Kevin Corre and Vincent Frey. **"Method of managing the authentication of a client in a computing system"**. *WO2017006013 A1 Patent App. PCT/FR2016/051,601. 2016*

[9] Ibrahim Tariq Javed, et al. **"Cross-domain identity and discovery framework for web calling services"**. *Annales des Télécommunications 72.7-8 (2017), pp. 459–468*

WebConnect
https://github.com/Sparika/WebConnect
https://github.com/Sparika/passport-jwt

ACOR SDP
https://github.com/Sparika/ACOR_SDP

OIDC/WebRTC Integration
https://github.com/reTHINK-project/dev-IdPServer
https://github.com/reTHINK-project/dev-IdPServer-phpOIDC

Trust Viz
https://github.com/Sparika/trustModelSurvey

Acknowledgment

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

CONVENTION CIFRE N° 2014 / 1185

Additional Slides

WebRTC Identity Proxy Interface

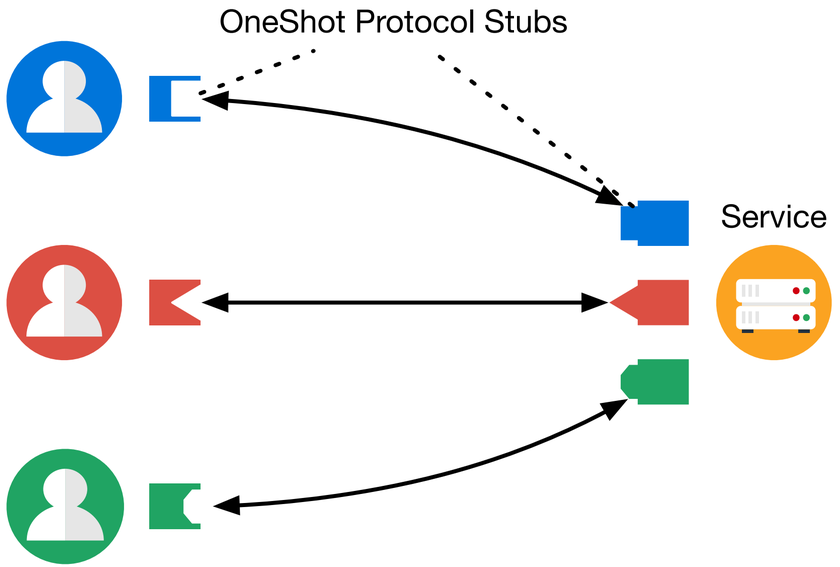
Perspectives

We focused on user control and manual configuration, ultimately we want to implement automatic reconfiguration of the WebRTC session

A protocol composition language may facilitate recomposition at runtime

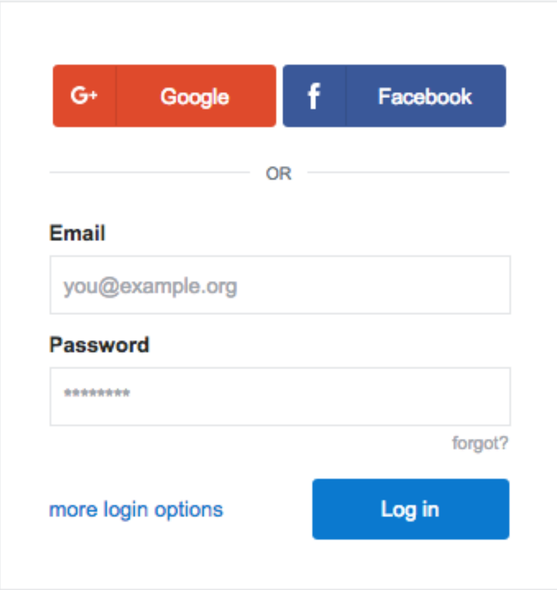
We also want to add trust context (i.e. types) to the WebRTC security and trust model

Such model could be used to diversify security configuration and protocols



Authorization and Authentication Delegation

OAuth2 and OIDC Authorization URLs



https://accounts.google.com/o/oauth2/auth?

client_id=74[...].googleusercontent.com&

redirect_uri=http://www.dailymail.co.uk/registration/signin/google.html&

scope=email+https://www.googleapis.com/auth/plus.login&

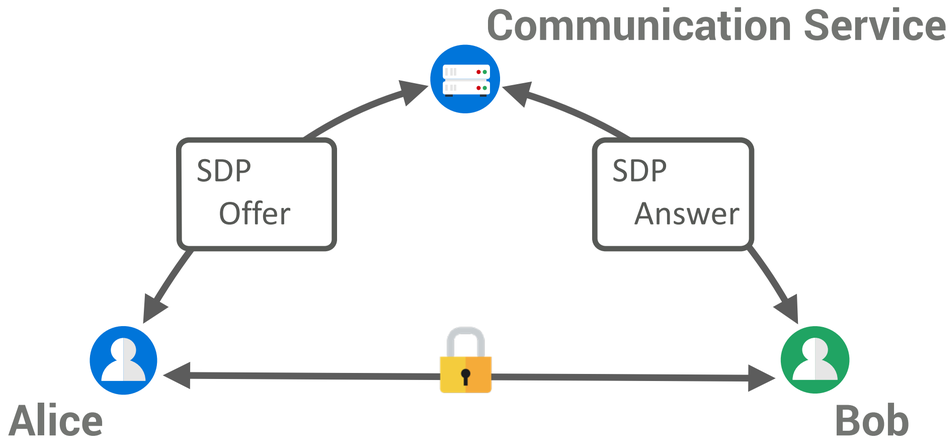
[...]

Negotiating the Other Peer's Authentication

Integrating into the SDP Session Negotiation Protocol

Can we extend the SDP identity-attribute?

```
identity-attribute  = "identity:" identity-assertion
                      [ SP identity-extension
                       *(";" [ SP ] identity-extension) ]
```



The identity-attribute grammar does not allow to use extensions while being anonymous (no identity-assertion provided).

Negotiating the Other Peer's Authentication

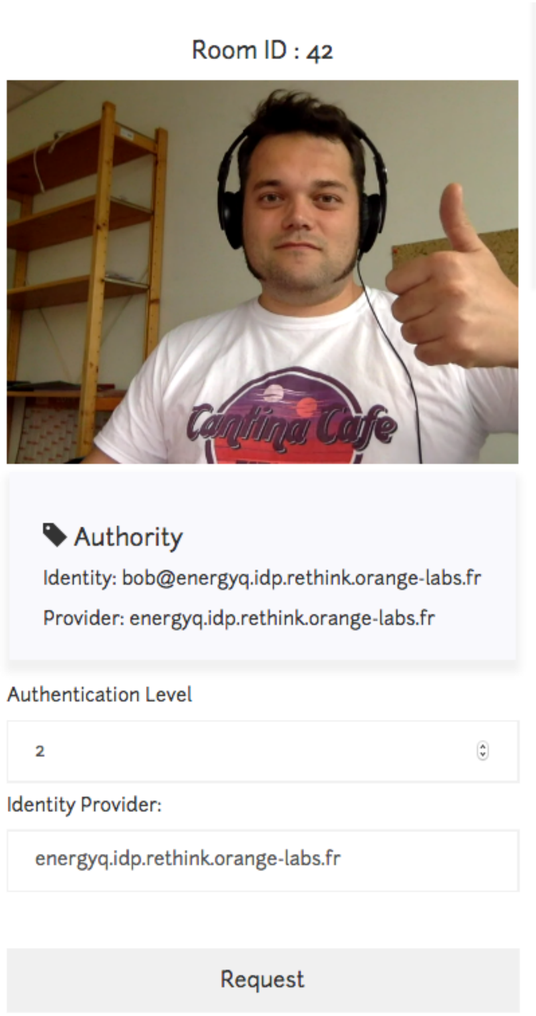
Integrating into the SDP Session Negotiation Protocol

Instead, we define a new session level attribute for Authentication Class and Origin Request (ACOR)

acor = "acor:" List<Authentication Class Values>
";" List<Identity Provider Origin>

We implement it in a simple communication service written in NodeJS.

https://github.com/Sparika/ACOR_SDP/



Negotiating the Other Peer's Authentication

Results

	Can be set	Can be verified
Origin Request	✓	already available
Authentication Class Request	✓	✗

The data format for identity assertion verification does not allow extensions. This may be a problem even outside negotiation scenario.

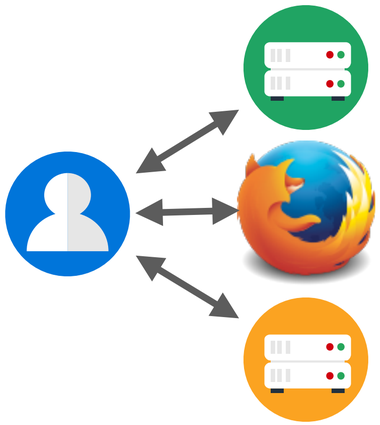
We can initiate an anonymous session and then request authentication, albeit without being able to verify the authentication strength.

It is also impossible to provide multiple identities or change identity after an initial authentication.

Negotiating the Other Peer's Authentication

Recommendation Sources

In order to negotiate its peer's authentication, the user must rely on a trusted recommendation source

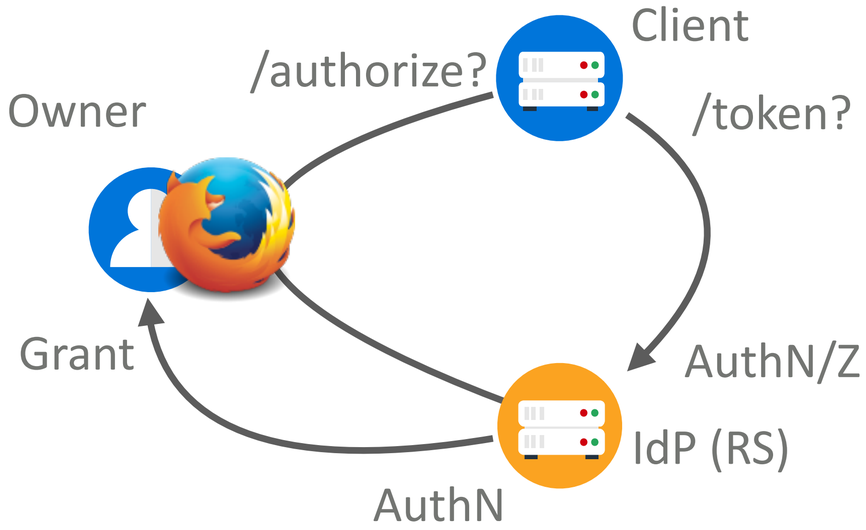


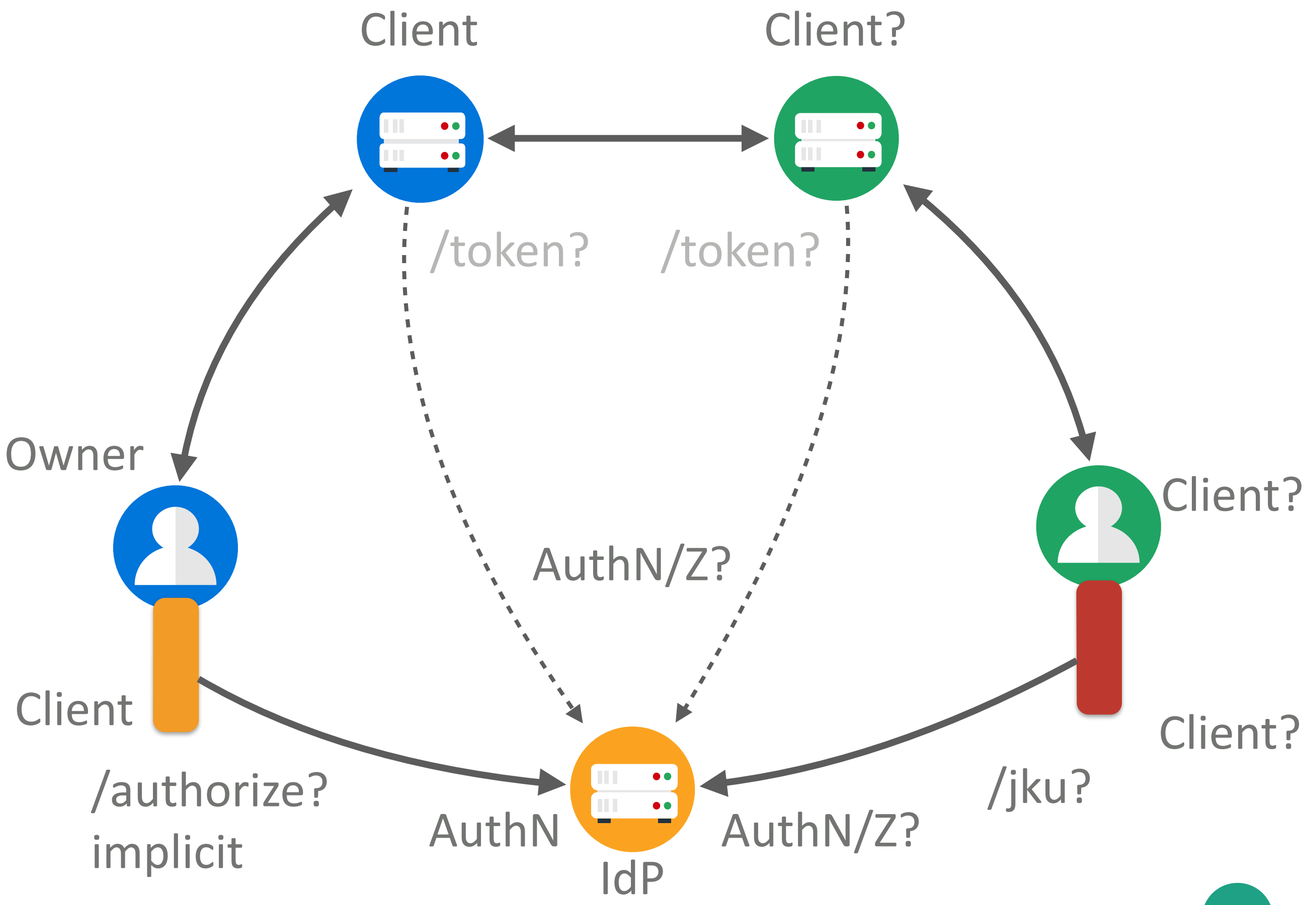
	Trusted	Authentication aware	Call context aware
CS	++	++	++
Browser	++	++	++
IdP	++	++	++

None of these three actors appears best placed to serve as the trusted recommender

OpenID Connect vs WebRTC

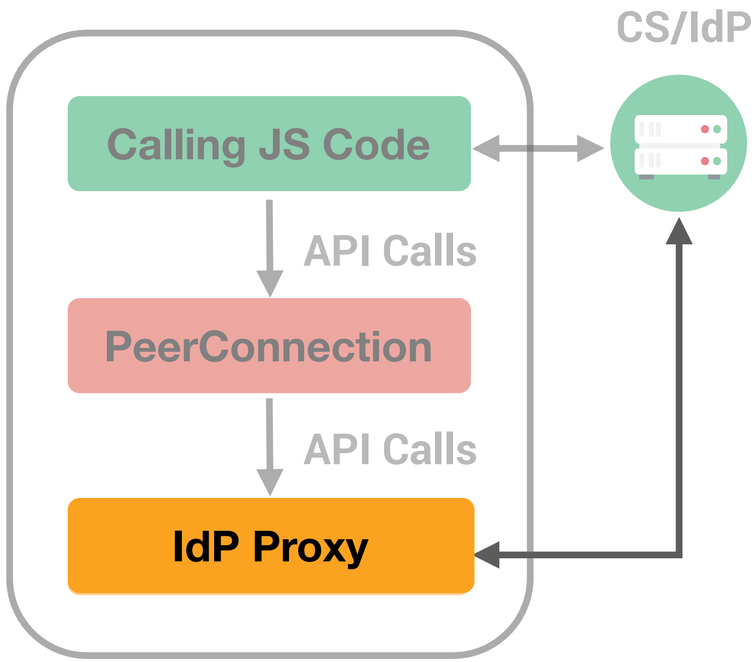
Comparison of Role





Implementing the Missing Part

Local Authentication Scenario

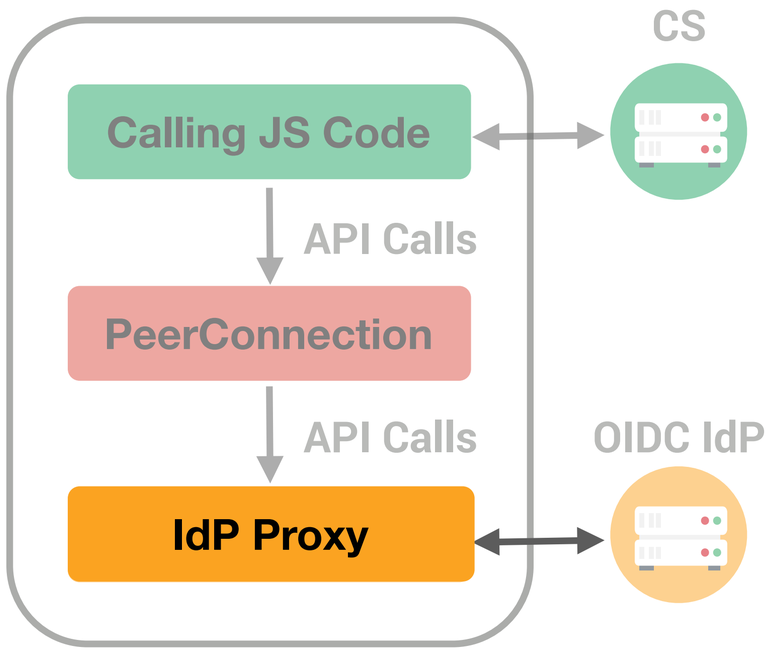


Local Authentication Scenario

- The **CS plays the IdP's role**
- Implemented a **Map(Assertion->Identity)** with a simple **REST** interface
- Could be useful in multi CS architecture

Implementing the Missing Part

OpenID Connect Integration



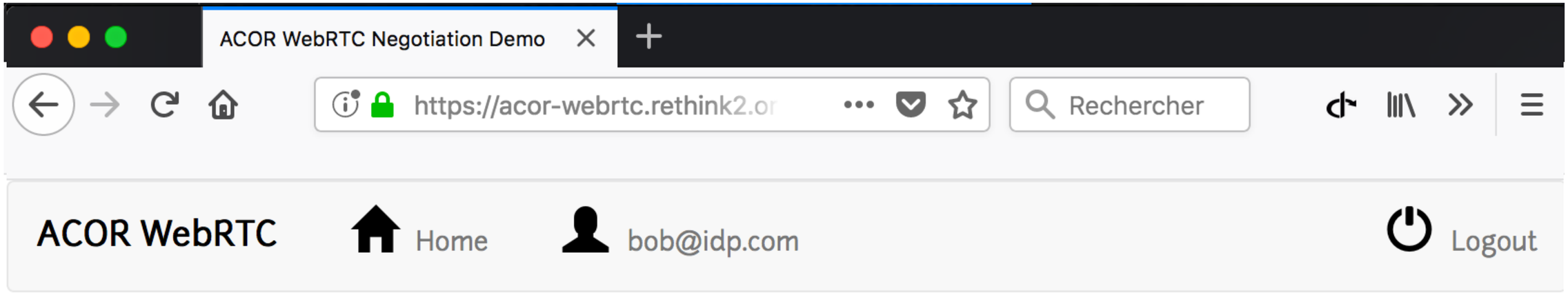
Initially sketched in WebRTC annex, **can we integrate OIDC with WebRTC?**

Integration requires modifications to the protocol:

- We reuse OIDC Identity Assertion in WebRTC
- **New OIDC parameters** for the session fingerprint
- **New OIDC response mode**

WebConnect: a Browser-based Identity Metasystem

The User Experience



WebConnect Security

Evaluation

WebRTC IdP Proxy	WebConnect
IdP Proxy served from an **HTTPS well-known** origin and whose domain matches the identity ending in @domain	**HTTPS JSON Key URL (JKU)** on a **well-known** origin and whose **domain matches the JWT:ISS domain claim**

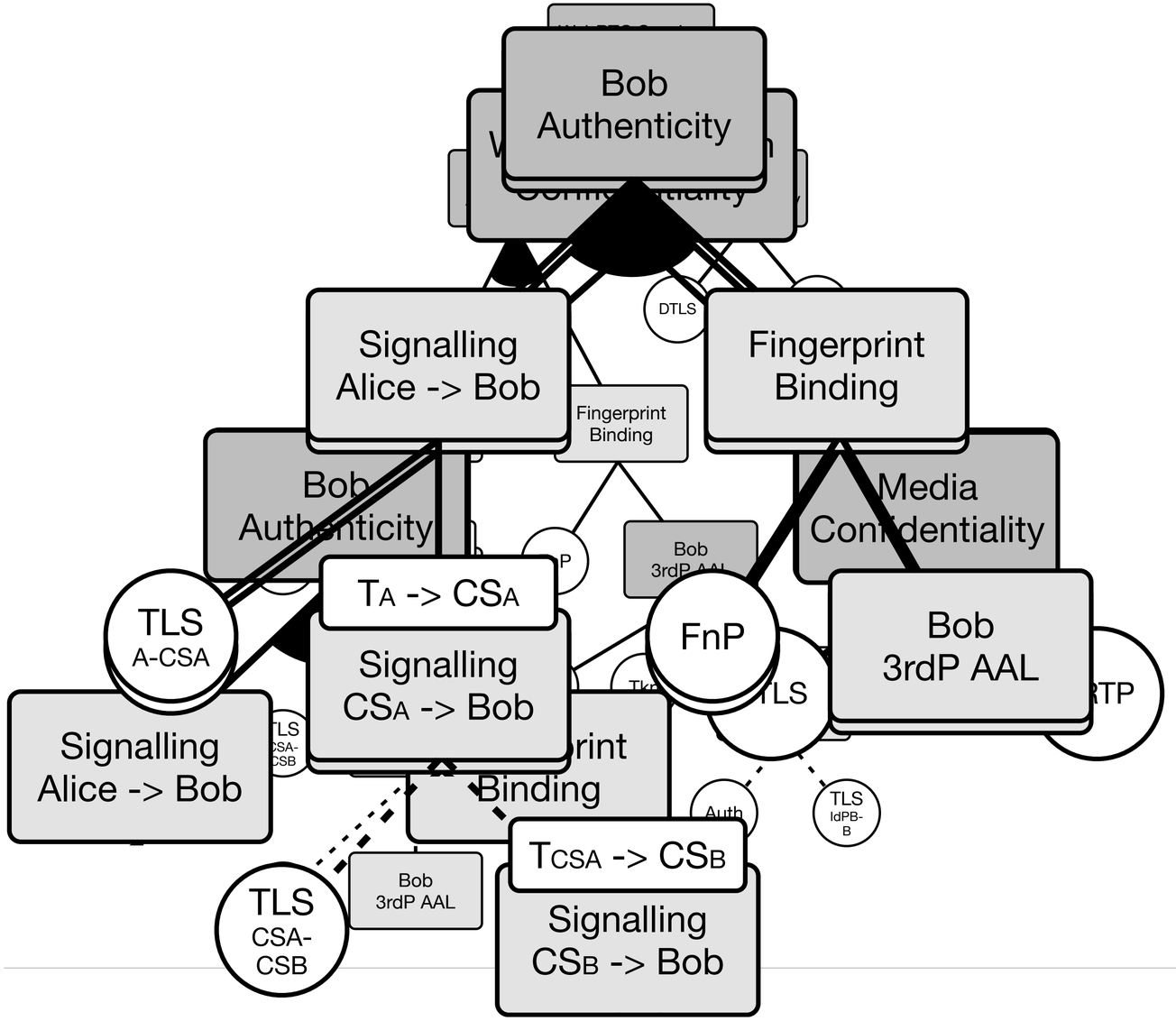
Our implementation modify OIDC standard to allow **the website to authenticate the IdP without prior registration**

However, **the IdP** is unable to authenticate the website and as such **cannot control authorization delegation**

Security relies on secure implementation and control of the API by the browser

A WebRTC Security and Trust Model

Transitive Defense Tree



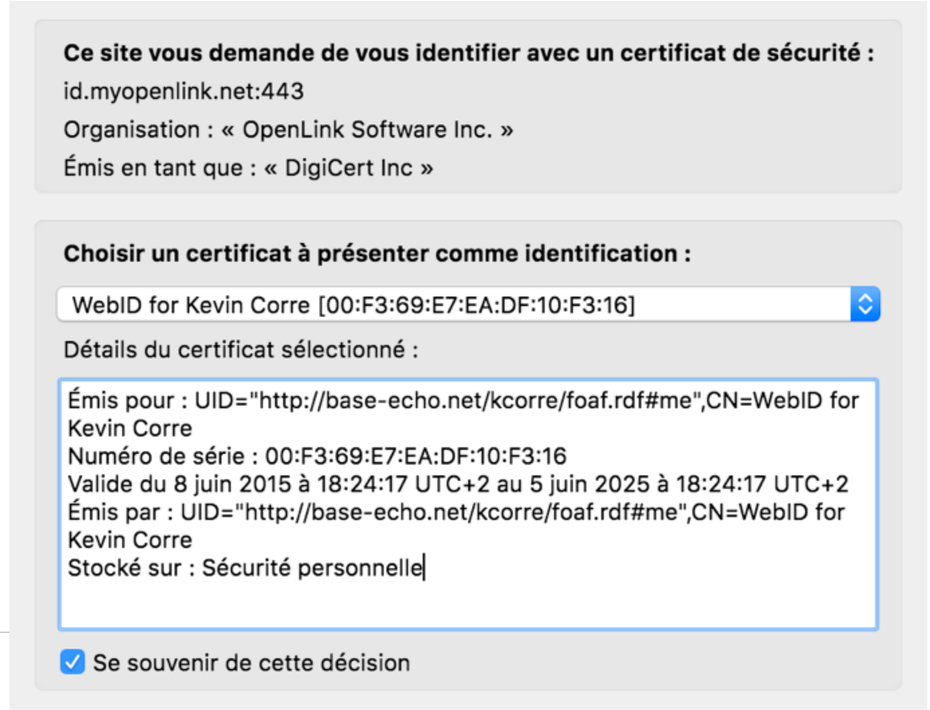
WebRTC Identity Proxy for WebID-TLS

Short-Term Perspective

WebID-TLS is a W3C protocol for self-hosted identity and TLS client-authentication

Is **WebID-TLS compatible with WebRTC IdP Proxy** (sandboxed JS)?

Can it be updated to a JSON Web Token based authentication signed using client-side Javascript?



WebPayment, WebID-TLS, WebConnect

Perspectives

We questioned wether WebRTC is a protocol abstraction capable of handling OIDC

Similar user experience and interface

Browser acts as the Trusted Computing Base

Similar request protocol

WebPayment	WebConnect	WebID-TLS	OAuth2/OIDC
Payment	Authentication	Authentication	Authorization

Is there a protocol abstraction interface capable of handling payment, authentication, and authorization?