30-04-2016

Deliverable D12.3 WebRTC Requirements and R&E Deployment Roadmap

Deliverable D12.3

Contractual Date:	30-04-2016
Actual Date:	30-04-2016
Grant Agreement No.:	691567
Activity:	12/SA8
Task Item:	Task 2 – WebRTC exploration
Nature of Deliverable:	R (Report)
Dissemination Level:	PU (Public)
Lead Partner:	NORDUnet (UNINETT)
Document Code:	GN4-1-16-13B9D3
Authors:	J. Meijer (UNINETT), S. Skrødal (UNINETT)

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 691567 (GN4-1).

Abstract

This document reports on results and findings from an investigation into WebRTC conducted by the Service Activity 8 (SA8), Task 2 team of the GN4-1 project. It provides recommendations for the technology's adoption by the European R&E community.

Table of Contents

Executive Summary 1

1 Introduction 5

1.1 In this Document 5

1.2 Target Audience 5

1.3 Project Background and Introduction 5

1.4 Purpose of Task 6

1.5 Objectives of Task 6

1.6 Scope and Methodology 7

1.7 Task 2 Team Members 8

1.8 Deliverables and Dissemination 9

2 WebRTC Technology 10

2.1 WebRTC Technology Overview 10

2.2 WebRTC Technology: Current Status 14

2.3 WebRTC Market Observations 16

2.4 WebRTC Technology Conclusions 19

3 Task Results and Deliberations 20

3.1 ICE, STUN and TURN: How WebRTC Deals with Firewalls and NAT 23

3.2 RENATER RENdez-Vous: One year of Operational Experience 26

3.3 Distributed RENdez-Vous: An investigation into Scale-out Jitsi 30

3.4 WebTut: A Contextual-Communication Proof of Concept 32

3.5 Service Proposal for a Media API Service 35

3.6 Screencast: Stream and Record Lectures with WebRTC 37

3.7 Unified Communication and WebRTC 40

3.8 WebRTC2SIP Gateway 44

3.9 Janus — General Purpose Gateway 45

3.10 Community Engagement with the TF-WebRTC 46

4 Recommendations and Conclusions 49

4.1 Recommendations for the GÉANT Community 49

4.2 Recommendations for NRENs and R&E Institutions 51

4.3 Closing Words 52

References 54

Glossary 56

Table of Figures

Figure 3.1: Teacher waiting for student in WebTut session 33
Figure 3.2: WebTut — integration management 34
Figure 3.3: Screencast — typical layout composition 38

Table of Tables

Table 1.1: SA8 Task 2 Team Members 8
Table 2.1: WebRTC browser support 17
Table 3.1: Technology scouts overview 22

Executive Summary

This document reports on results and findings from an investigation into WebRTC conducted by the Service Activity 8 (SA8), Task 2 team of the GN4-1 project. It provides for immediate action by the GÉANT community, individual NRENs and R&E institutions to support the technology's adoption by the European R&E community.

Web Real-Time Communications (WebRTC), is a technology (not a turnkey solution) that facilitates audio, video and data communication in a web browser. No extra internal or external plugins are needed, it just works. WebRTC applications are built using the same core technologies that power the rest of the World Wide Web (e.g. JavaScript and HTML5) and may therefore support, leverage and integrate with existing and future web-based services. Of particular interest is the premise of WebRTC to provide contextual communication capabilities, such as form filling or document collaboration, to web-based services pertinent to R&E use by adding real-time communication in any web application.

Key WebRTC advantages include: ease of use, security (end-to-end encryption), speed and reliability (direct between peers), platform/device independence, ease of development and deployment, scalability, interoperability (with legacy systems), open (with free frameworks and components), and high fidelity voice and video codecs. WebRTC provides media streams and data channels that support a wide array of applications, e.g. audio/videoconferencing, text chat, file sharing and autonomous data exchange between devices.

WebRTC standards (defined by W3C and IETF) and frameworks are open [W3C], [IETF]. They are already supported and used by a number of prominent web and unified communications organisations, including Google, Mozilla, Microsoft, Opera, Cisco, Intel, Oracle, IBM, Vidyo, WhatsApp, Ericsson, Facebook and Apple, which have added WebRTC to the list of coming features for Safari's browser engine in April 2016 [WEBKIT].

Even though it already powers many global communication services, WebRTC is a relatively new technology, with the first interoperability between two different browsers achieved in February 2013. Both the standards and their implementation in browsers have yet to be finalised. There is little doubt, however, that WebRTC is "the next big thing" in real time communications. As such, it is imperative that the GÉANT Community positions itself to understand and realise its technological benefits and capabilities within the research and educational context.

Based on a preliminary exploration of the WebRTC landscape, the following areas were selected for further investigation by means of technology scouts (reported in the following technical documentation) [TF-WebRTCWIKI]:

- ICE, STUN and TURN to facilitate NAT/firewall traversal in peer-to-peer WebRTC.
- Native WebRTC-based desktop videoconferencing services, both peer-to-peer (< 10 participants) and star-topology (supporting > 10 participants).
- Contextual communication with WebRTC (Web tutoring service).
- Protocol gateways between WebRTC and other technologies like SIP and H.323.
- Unified Communication and WebRTC.
- Proposal of supporting services that allow for fast and easy development of innovative contextual applications.
- Assess WebRTC technology maturity to add value in other areas of teaching and learning, e.g. lecture streaming and recording.

The SA8 team's investigations (technology scouts) demonstrate how infrastructures required to enable inter-NREN services may be built using open technologies. As a proof of concept, the team has demonstrated what is required to build a contextual communication application based on WebRTC. The task documented a first-hand account of the experiences, challenges and lessons was gained by RENATER from providing a WebRTC-based videoconferencing production service to the entire population of one National Research Educational Network (NREN), as well as the wider R&E community.

To ensure an effective and open dialogue with the wider R&E community, as well as the WebRTC solutions market, the task invested in active participation in the GÉANT taskforce TF-WebRTC which delivered a solid Return of Investments (ROI) case [TF-WebRTC].

WebRTC offers the R&E community:

- A significant improvement of the usability and ease of use of videoconferencing solutions.
- A quantum leap towards contextual communication, giving the ability to embed real-time communication functionality in any web application.

Based on the work performed in SA8 Task 2, the team has put forward two, high-level strategic goals for the European R&E community:

- Use WebRTC to bring easy to use and high-quality desktop videoconferencing functionality at low cost to every European R&E user and do it now.
- Facilitate the adoption of WebRTC for contextual communication, for example in Virtual Learning Environments (VLEs) and research portals.

These strategic goals lead to a number of recommendations for future project work, as well as for all NRENs and R&E institutions.

Key recommendations for collaborative action by the GÉANT community include:

- Make the adoption of useful WebRTC services easier by adding them to the GÉANT Cloud Catalogue, with framework agreements, where appropriate.
- In particular, make one or more easy to use WebRTC desktop videoconferencing service available to all European R&E users through the GÉANT Cloud Catalogue, at the earliest

opportunity. This creates a new lowest common denominator solution for simple online group conversations.

- Recommend requirements specifications for WebRTC services to facilitate procurements and interaction with solution providers.
- Establish a GÉANT STUN/TURN pilot service to support WebRTC technology early adoption.
- Assist the R&E community by providing a technical cookbook for adopting contextual communication.
- Continue to investigate the concept of, and possibilities for, a GÉANT Media API Service to facilitate low-cost, R&E domain-specific, contextual communication.
- Investigate the feasibility of establishing a GÉANT service that may act as a hub for key Unified Communication data, such as presence and buddy lists, from institutions' UC installations. This service may then propagate information to web services supporting contextual communication. This could be combined with the Media API Service proposal.
- Closely cooperate with RENATER to leverage its deployed native WebRTC service RENdez-Vous and its community of users. This has significant benefits for recommended activities on the Media API and the desktop videoconferencing services.
- Consider adding a WebRTC-to-SIP and WebRTC-to-H323 gateway to the GÉANT cloud catalogue.
- Revisit currently immature WebRTC lecture streaming and recording capabilities in early 2017.
- Closely track market developments for the WebRTC data channel as it has potential for interesting R&E use cases.
- Actively promote WebRTC to the NREN community as an emerging technology that will change the face of real time communications in R&E.
- Continue to engage with the wider R&E community and market through the TF-WebRTC.

Recommendations specifically for NRENs and R&E institutions are:

- Ensure users have access to an easy to use WebRTC desktop videoconferencing service.
- Develop NRENs' technical capability for leveraging WebRTC in contextual communication, including skills for procurement, development and support.
- Leverage the TF-WebRTC for building knowledge of, and formulate a position on WebRTC.
- Shift investment in distance-bridging technology for physical rooms from appliances supporting a single communication solution to generic hardware supporting many different (including WebRTC-based) software solutions. Consider publishing a Campus Best Practice paper on this topic.
- Revisit real time communication service portfolio and plan for phasing WebRTC into relevant services and components.
- Engage with streaming and lecture recording solution providers regarding their plans for future support of the emerging WebRTC recording standards.
- Investigate and deploy modern gateway technology to connect unified communications solutions, various traditional videoconferencing systems and WebRTC-based services. This makes it easier for the end user to deal with a heterogeneous, real-time communication environment.

- Consider a joint (open source) effort to further develop a Web Tutoring service.

WebRTC technology allows for a significant improvement in usability of real-time communication solutions. In addition, it represents a quantum leap towards pervasive real-time communication, with its potential for contextual communication in the dominant application deployment framework of today, the World Wide Web. A number of free WebRTC services, such as appear.in, already offer a superior user experience for the simple “some people or rooms want to talk together” use case when compared to previous generation tools, such as WebEx and Adobe Connect [appear.in].

There is no reason to wait with adopting WebRTC.

It's mature enough.

Use it.

1 Introduction

1.1 In this Document

This document reports on results and findings from the WebRTC technology exploration undertaken as part of the GN4-1 project by the WebRTC Task 2 (T2), one of three tasks of the Real Time Communication and Media activity (SA8), which ran from 1 May 2015 to 30 April 2016.

By means of research, technology scouts, demonstrations, pilots and community outreach, Task 2 investigated the WebRTC landscape. The sum of findings and experiences serve as a comprehensive "roadmap", presenting a number of recommendations for immediate action by the GÉANT community as well as individual NRENs and R&E institutions.

The rest of this document contains the following sections:

- Section 2 presents examples of WebRTC technology.
- Section 3 details the Task 2 results and presents a number of findings relevant to WebRTC in a European R&E context and summarises Task Force engagement activity.
- Section 4 closes the document with the team's conclusions and recommendations.

1.2 Target Audience

This document targets technical management and specialists in NRENs and R&E institutions, in particular those working in the fields of real time communications, e-Learning and e-Research.

1.3 Project Background and Introduction

Following TNC2013 (mid-2013) individual members of the NREN community performed preliminary investigations on WebRTC technology. Their experiences made it clear that WebRTC had evolved quickly to a point where a systematic effort of the R&E community to understand WebRTC was prudent.

There is a clear need for the GÉANT Community (the collaboration of NRENs, project partners and users of GÉANT services) to realise the potential benefits of this new capability sooner rather than later, and apply WebRTC technology to a research and educational context. It needs to take an active approach to shape the market, ensuring its relevance for R&E users. This not only requires a good understanding of the technology, the market, service deployment scenarios, business models, infrastructure architectures and integration challenges with existing or legacy infrastructure, but very importantly, also of the institutional and end-user perspective.

To this end, Task 2 of the GN4 Phase 1 Service Activity 8 (SA8), Real-Time Applications and Multimedia Management, explored WebRTC technology under a common and well-coordinated space.

1.4 Purpose of Task

WebRTC offers, to anyone with a web browser, high-quality audio/video communication capabilities, previously available only using proprietary systems and software,. This creates an opportunity for the European R&E community to solve its real time communication challenges in novel ways. WebRTC may finally offer a path towards a large-scale, low-cost and easy-to-use, real-time communication infrastructure for group conversations across institutional boundaries. Whether and how WebRTC may deliver on its promise, needs to be investigated.

The NREN community needs to prepare for this change in real-time communication technology and equip itself to meet the opportunities and challenges with the correct and timely response. A feature-rich web, complemented with real-time communications capability, will offer the opportunity for a more component-based approach to including real-time communication in all kinds of e-Learning and e-Research web applications at a low price point and without locking our community to any particular vendor or solution. The combination of ubiquitous availability of standardized high quality capabilities at the users' endpoint devices, combined with widely accepted open standards, carries significant promise.

Using the web as an application and service delivery model allows for truly large-scale deployments of inexpensive one-to-one, one-to-many and many-to-many (video) communication.

1.5 Objectives of Task

Service Activity 8 undertook to propose a comprehensive and executable roadmap for WebRTC development and deployment in the R&E context with the following larger objectives in mind:

- Investigate whether, and how, WebRTC-based solutions and services can be used to enable large-scale, easy-to-use, easy to integrate and low-cost use of real-time communication across institutional and national boundaries for all researchers, lecturers, administrative staff and students in European R&E.
- To ensure the European NRENs are well positioned to realise the full potential of WebRTC technology for their own user communities as the technology emerges in the years to come.
- Help to establish the NREN community as a recognised representative for the European R&E community in relevant Web-RTC arenas (Section 2.1.1.1).

- To imagine future WebRTC-based NREN services and infrastructural components that contribute to a smooth evolution of the real-time communications infrastructure.

The roadmap aims to provide the basis for further collaborative WebRTC activities in the GÉANT community and presents an integrated view of the envisioned real-time communication infrastructure for European R&E from 2016-onwards to European NRENs, GÉANT and European R&E institutions.

1.6 Scope and Methodology

This task represents the first major work on WebRTC coordinated by GÉANT. It is, as such, a preliminary study that breaks new ground for future activities. In order to provide a roadmap, knowledge of the terrain is essential. To this end, task members needed to undertake preliminary research activities to better understand the current state of the WebRTC landscape and to identify key areas pertinent to the R&E context. Knowledge from these activities was used to design a scope and methodology suited to a roadmap for future work, with the resources available to the task.

The task members were selected from NRENs with existing national activities and interests in the field of WebRTC. Where possible, the task built on their national work in order to make effective use of our limited task resources as well as to prevent working in isolation.

The following areas were outlined in the task plan:

- WebRTC technology scouting and feature testing
- WebRTC roadmap development
- WebRTC demonstrator
- Liaison with relevant market parties, other GN4 activities and NREN/R&E community.

Task 2's initial wide exploration of the WebRTC technology and landscape highlighted a number of specific areas apt for in-depth exploration through the use of technology scouts, some of whom produced demonstrators. The following areas were addressed:

- ICE, STUN and TURN to facilitate NAT/firewall traversal in peer-to-peer WebRTC.
- Native WebRTC based desktop videoconferencing services.
- Contextual communication with WebRTC.
- Providing gateways to and from other technologies like SIP and H.323.
- Unified Communication and WebRTC.
- Supporting services that allow for fast and easy development of innovative contextual applications.
- Assess WebRTC technology maturity to add value in other areas of teaching and learning, e.g. lecture streaming and recording.

The areas were chosen based on known challenges for the national user communities of NRENs participating in the task, combined with the team's findings of the WebRTC technology exploration.

A number of tech scouts initiatives expanded into collaboration, for example, the Media API Service proposal would not have materialized were it not for tech scouts and work on RENdez-Vous and ICE/STUN/TURN [RENDEZ-VOUS], [RFC5245], [RFC5389], [RFC5766]. Much dialogue and close collaboration with various market players, established and start-ups was essential throughout the project.

Community liaison was addressed by frequent participation in GÉANTs TF-WebRTC, an open forum where the task members could share knowledge with other NRENs, vendors and service providers [TF-WebRTC].

The TF-WebRTC chair and Task 2 team member, Mihály Mészáros, ensured good communication.

Throughout GN4-1, SA8 has contributed to the establishment of a network for the NREN community practitioners working on WebRTC and actively liaised with other networks and (national) forums. Findings of the WebRTC technology scouting were presented and input to the roadmap sought at a number of conferences, for example the GN3plus and GN4-1 Symposium, TNC 2015 and a number of NREN national conferences.

1.7 Task 2 Team Members

Name	NREN	Comments
Allocchio, Claudio	GARR, Italy	
Hansen, Alf	UNINETT (NORDUnet), Norway	Task lead assistance
Loui, Frédéric	RENATER, France	
Meijer, Jan	UNINETT (NORDUnet), Norway	Task lead
Mészáros, Mihály	NIIF Institute, Hungary	
Otto, Stefan	UNINETT (NORDUnet), Norway	
Ribeiro, Rui	FCCN, Portugal	
Rupin, Franck	RENATER, France	Left project 30. November 2015
Skrødal, Simon	UNINETT (NORDUnet), Norway	Entered project 1. January 2016

Table 1.1: SA8 Task 2 Team Members

A total of 22 person-months were allocated for the task.

1.8 Deliverables and Dissemination

1.8.1 Milestone and Deliverable

The formal output of SA8 Task 2 during GN4-1 included:

- A proof of concept (PoC) demonstrator component for WebRTC client signalling, which was delivered by the ICE/STUN/TURN PoC (Milestone 12.2).
- WebRTC requirements and roadmap for deployment in R&E (this report, Deliverable 12.3).

1.8.2 Internal Deliverables

The task produced several internal deliverables documenting the results of the undertaken technology scouts. These documents are not part of this report, but have been published under the umbrella of the GÉANT TF-WebRTC task force [TF-WebRTC-INV], [TF-WebRTC].

1.8.3 Dissemination of Results

Task 2 primarily used the TF-WebRTC and GN4-1 project channels, including the GN4-1 Symposium in Vienna, to disseminate its results. It organised a BoF session at TNC15, and secured two presentation slots at the TNC16 [TNC2015], [TNC2016]. In addition, the team members promoted the task's work through their NREN's national communication channels.

2 WebRTC Technology

At its core, Web Real-Time Communications facilitates synchronous communication of voice, video and data between web browsers, without the need for plugins/extensions of any sort.

WebRTC communication is encrypted and may travel directly between browsers on a peer-to-peer (P2P) network, thus providing both security and privacy. P2P also reduces network latency, i.e. the time it takes for data to travel between peers. If peers are unable to directly connect to each other (e.g. due to firewalls or network address translators (NAT), a special proxy server implementing the TURN protocol may be used to resolve such issues [RFC5766].

Advanced and platform-independent services pertaining to videoconferencing, telecommunications, instant messaging, file sharing and gaming, may all be developed using WebRTC. The end-user only needs access to a web browser to use a WebRTC-based service.

Although browser-centric, WebRTC may also be implemented as native software/apps on mobile and desktop platforms. Well-known web and native apps that already implement WebRTC include: Facebook Messenger, Google Hangouts, WhatsApp, Firefox Hello, appear.in and talky.io.

2.1 WebRTC Technology Overview

While standardisation of WebRTC is work in progress, it already enjoys advanced implementations in a number of popular web browsers: Chrome, Firefox and Opera. In April 2016, Apple announced that it added WebRTC to its list of upcoming Safari features.

This section provides a summary of the WebRTC technology landscape.

2.1.1 WebRTC Specifications and Implementations

WebRTC may refer to a standard specification, as well as several open (source) projects. With a number of organisations involved in progressing the different facets of WebRTC, the terms of reference can get confusing. The following sections present a brief introduction.

2.1.1.1 *WebRTC 1.0 Standard Specification*

The World Wide Web Consortium (W3C) — The W3C WebRTC Working Group is responsible for drafting the WebRTC 1.0 specification [WebRTCSPEC]. This is done in conjunction with the protocol specification developed by the IETF RTCWEB and the API specification to get access to local media devices developed by the W3C Media Capture Task Force [IETFRTCWEB].

The mission of the W3C WebRTC Working Group is to “define client-side APIs to enable real-time communications in Web browsers”. To this end, the group identifies the following client-side technologies in their scope:

- API functions to explore device capabilities, e.g. camera, microphone, speakers.
- API functions to capture media from local devices (e.g. camera and microphone, but also output devices such as a screen).
- API functions for encoding and other processing of those media streams.
- API functions for establishing direct peer-to-peer connections, including firewall/NAT traversal.
- API functions for decoding and processing (including echo cancelling, stream synchronization and a number of other functions) of those streams at the incoming end.
- Delivery to the user of those media streams via local screens and audio output devices (partially covered with HTML5) [W3CWGCHARTER].

The Internet Engineering Task Force (IETF) — The IETF RTCWEB Working Group is responsible for standardising protocols (e.g. used to establish connections between peers) pertinent to WebRTC.

2.1.1.2 *WebRTC Open Projects*

WebRTC

WebRTC — A free and open project supported by a number of organisations, most notably Google, Mozilla and Opera.

The project's mission:

> *To enable rich, high-quality RTC applications to be developed for the browser, mobile platforms, and IoT devices, and allow them all to communicate via a common set of protocols …*
>
> *… It includes the fundamental building blocks for high-quality communications on the web, such as network, audio and video components used in voice and video chat applications.*
>
> *These components, when implemented in a browser, can be accessed through a JavaScript API, enabling developers to easily implement their own RTC web app.* [WebRTC]

The project’s WebRTC architecture (see webrtc.org/architecture) may be summarised as follows [WebRTC_ARCH]:

- *The WebRTC API layer* (according to the World Wide Web Consortium’s (W3C) definition):

    - Used by web app developers to realise WebRTC services.

- The WebRTC Native C++ API layer:
    - Used by browser makers to easily implement the WebRTC API.

- The transport/session layer:
    - Components that implements various network protocols.

- Voice and video frameworks:
    - To access/transfer/receive microphone and camera feeds.

The project provides frameworks for native platforms that support Windows, Mac OS X, Linux, Android and iOS.

OpenWebRTC — Free and open source project by Ericsson; offers an alternative implementation to the WebRTC project.

> *Having independent, interoperable, implementations is important for the health of any standard, and WebRTC is no exception* [OpenWebRTC].

Ericsson's web browser for iOS, Bowser, uses OpenWebRTC. An initiative is also underway (also lead by Ericsson) to add support to the WebKit browser engine using the OpenWebRTC implementation [WEBKIT].

OpenWebRTC provides native platform support on iOS, Android, Mac OS X and Linux (not Windows).

2.1.2 Alternative RTC Implementations

Object Real-Time Communications (ORTC) — The ORTC initiative is a free and open project supported by organisations such as Microsoft, Hookflash and Google [ORTC].

ORTC defines a lower-level API that provides the same components as WebRTC, but also adds greater access to more controls via JavaScript. This to allow web developers more flexibility in the way they build RTC apps and features.

ORTC is supported by the Microsoft Edge web browser and some of the ORTC API elements are being adopted in the WebRTC 1.0 source code.

> ***Q: How does this compare to the work being done in the W3C WEBRTC WG?***
>
> *A: The W3C WEBRTC WG is working toward locking down the functionality in WebRTC 1.0, in order to bring that work to a conclusion. Rather than distracting*

the existing WG from its mission, a separate ORTC Community Group was formed, with the charter of developing a next generation API [ORTC].

A future convergence of ORTC and WebRTC 1.0 is expected.

2.1.3 WebRTC Browser Support

WebRTC support can be identified according to browser engines.

Prominent engines with *WebRTC* support:

- Blink (e.g. Google Chrome, Opera)
- Gecko (e.g. Firefox)

Engines supporting *ORTC*:

- EdgeHTML (powers Microsoft's new browser Edge)

Unsupported engines include:

- WebKit (iOS Safari/WebViews/web clips, Safari, Web)
- Trident (Internet Explorer)

Unsupported browsers, e.g. Safari and Internet Explorer, may achieve WebRTC features by installing third party plugins.

13 April 2016: News broke that WebRTC is finally "in development" for the WebKit engine (which powers Apple's Safari web browser) [WEBKITSPEC].

2.1.4 WebRTC API Core Components

WebRTC consists of three major API components that work together to realise real-time communication and data exchange:

GetUserMedia — Allows a web browser access local multimedia input (e.g. cameras, microphones) on the host system.

- GetUserMedia is defined in the Media Capture and Streams API specification [GetUserMedia].

RTCPeerConnection — Represents a WebRTC connection between the local computer and a remote peer. It is used to handle efficient streaming of data between the two peers.

- RTCPeerConnection is defined in the WebRTC 1.0 specification [RTCPeer].

RTCDataChannel — Provides a bi-directional data channel between two connected peers, allowing data to be shared.

- RTCDataChannel is defined in the Peer-to-peer Data API of the WebRTC 1.0 specification [RTCData].

In addition to these core components, the standard includes a number of supporting features, such as TURN support, multiple streams and recording, to name a few. A more extensive list can be found here [RTCFEATURES].

2.1.5 WebRTC Supporting Services

Firewalls and network address translation (NAT) are important technologies put in place to secure and enhance the usability of networks. However, they also make peer-to-peer (P2P) discovery and connections more complex to achieve. The IETF standards STUN, TURN and ICE, aim to overcome barriers that prevent direct peer-to-peer real-time media communication (essential for WebRTC):

- Session Traversal Utilities for NAT (STUN): a protocol that provides the means for an endpoint to determine the IP address and port allocated by a NAT that corresponds to its private IP address and port (e.g. to detect a client's public IP) [RFC5389].

- Traversal Using Relays around NAT (TURN): a protocol that allows a client behind a NAT to request that a server acts as a relay (e.g. to relay packets between peers if P2P cannot be established) [RFC5766].

- Interactive Connectivity Establishment (ICE): defines a technique for NAT traversal for media streams, established by the offer/answer model, by utilising Session Description Protocol (SDP), STUN, and TURN to discover, verify, establish and keep alive a network path between peers on the Internet (e.g. provide an established connection to the media layer where data is going to be exchanged) [RFC5245].

2.2 WebRTC Technology: Current Status

Specifications — WebRTC is now sufficiently mature to realise a number of quality tools for videoconferencing. However, it is important to be aware that the WebRTC 1.0 standard has not yet been finalised by W3C. The latest specification draft notes, however, states the intention "to publish a Candidate Recommendation soon" [WebRTCSPEC].

Web browsers — The browser support scorecard provides an apt illustration of the various APIs pertinent to WebRTC and how the major browsers are positioned in their implementations of these [SCORECARD]. Note that individual browsers still operate with vendor-specific prefixes in their implementations, and these are subject to change as the implementations are finalised.

2.2.1 Technology Stability

W3C specification status, browser implementation maturity, as well as first-hand experience, suggests that the technology is still in flux. Browser vendors in particular may implement new — and alter existing — functionalities that *can break an application*.

For example, the web is still riddled with demonstration apps that are broken in the Chrome web browser, following Google's decision in late 2015 to block WebRTC that is not run from a secure origin (e.g. HTTPS or localhost). Vendor prefixing will also change in the future, thus using adapters and shims to insulate apps from spec changes and prefix differences is still a recommended practice. For now, developers must be extra stringent and continuously monitor and test their WebRTC applications to ensure they operate as intended.

Whilst finalizing this report, news broke that WebRTC is "in development" for WebKit, the engine for Apple's Safari web browser.

2.2.2 What Works

While it is prudent to be wary of technology stability of a work in progress, the fact is that core functionality (e.g. audio/video/data exchange) of WebRTC works incredibly well in supported browsers.

One of the most actively used services for frequent communications between team members in Europe and Australia, appear.in proved superior to readily available non-WebRTC alternatives, such as Skype (for Business) and Adobe Connect, for a number of reasons (e.g. better audio, better video, less latency/lag, quick to get started, no accounts required, simple and intuitive). The same benefits were found when testing other free WebRTC services, such as rabb.it, room.co and Jitsi-based products such as talky.io and RENdez-Vous. While the UIs and extra features may vary, the audio/video quality remains identical. It should be noted that, in the Task 2 team's mode of operation, we have not required extra features pertinent to e.g. online education, such as multi-party (>8 participants), presence, whiteboards, break-out-rooms, etc.

In our experience, the only service matching audio/video quality was FaceTime, but it only supports two participants and requires Apple hardware (and an account).

For audio and video, WebRTC Data Channels provide peer-to-peer and fully encrypted transfers [WebRTCDATA]. It is the natural choice for enabling instant messaging in WebRTC videoconferencing services. However, the Data Channel can be (and is) used for so much more. A number of services, such as RTCCopy.com, sharedrop.io, pipe.com and webtorrent.io, demonstrate the Data Channel's capacity to allow files of any size to be securely shared directly between two web browsers. Games making use of the Data Channel are also starting to emerge, one novel example turns your mobile phone into a Lightsaber [STARWARS].

The Data Channel is also a strong candidate as a standard to power information exchange for Internet of Things applications (Section 2.3.1), for example, remote doctor-patient, or personal trainer, consultations combine the use of video, voice and data transfer (WebRTC for video conversation, Data Channel for automated and secure P2P transfer of sensor/instrument data).

2.2.3 What Does Not Work

As discussed, implementations using the "core" functionalities of WebRTC are fully functional and may even provide user experiences superior to other, more established, services.

Some extending APIs, however, do not yet enjoy the same stability and implementations. Examples include APIs for recording and screen sharing (discussed in more detail in Section 3.6) and statistics.

The WebRTC Statistics API provides clients with parameters such as CPU performance, bandwidth, jitter and latency in real time [WebRTCSTATSAPI]. This information allows clients to dynamically adapt to changes in the environment, which is essential in e.g. multi-part conferences, where participants typically have varying device and network resources available.

As noted by Sam Agnew (Twillio) in a recent blog post:

> *The WebRTC statistics spec details an API that gives developers access to a ton of statistical information about a WebRTC peer connection. It is currently evolving and is partially implemented in Chrome and Firefox. Neither browser has their stats API implementation up to the full spec yet and they both vary in execution. Code you write for one browser will almost certainly not work in the other browser* [TWILLIOBLOG].

Thus, although the API functionality is specified and being implemented, it does not yet permit larger scale use in a heterogeneous environment. For situations where communication end points have a significant difference in CPU and bandwidth performance, some end points may receive larger media streams than they can cope with. In this case the resulting user experience will be sub-optimal until the Statistics API is widely supported and facilitates dynamic scaling of media streams without reloading the WebRTC application.

For an up-to-date overview of browser support for the various WebRTC APIs, please check [BROWSER].

2.2.4 Codecs

The standard mandates support for the VP8 and H.264 video codecs in order for a web browser to be WebRTC compliant. At present, VP8 enjoys broader support, but H.264 is set to catch up as vendor implementation of codecs progresses.

Support for audio codecs is more streamlined, both Opus and G.711 must be supported by compliant WebRTC end points. Opus is the codec of choice to achieve high-quality audio. G.711 enables interoperability with other systems, most importantly with PSTN.

2.3 WebRTC Market Observations

WebRTC technology uptake is tightly coupled with browser vendors' attention and support of the standard. At present, a majority of the modern (and relevant) web browsers do support WebRTC — with one major exception, Apple's Safari (pending).

Browser Engine	Major Browsers	WebRTC Capabilities?
Blink (WebKit fork)	Chrome, Opera	Yes
WebKit	Safari	No[1] (but on the roadmap)

Browser Engine	Major Browsers	WebRTC Capabilities?
Gecko	Firefox	Yes
EdgeHTML	Edge	Yes (ORTC)
Trident	Internet Explorer	No[2]

[1] *Late announcement April 2016 stated development on WebRTC for the WebKit browser engine.*

[2] *Internet Explorer is still enjoying a considerable market share, but is being replaced by Edge. IE version 11 is effectively the last release of this browser.*

Table 2.1: WebRTC browser support

Adobe Flash, which used to be *the* major enabler of real-time communication in the web browser, is a dying technology. Already in July 2015, Adobe, which is a major player in the field with its Adobe Connect web conferencing system, noted:

> *As WebRTC matures, Adobe Connect plans to be ready to support HTML5 on an even broader scale. We are already developing an Adobe Connect web client fully on HTML5 – so we are planning to be ready once HTML5 can support large scale collaboration across browsers* [ADOBE].

Recent news on WebRTC development for WebKit is big, and also underpins the pertinence and importance of GÉANTs initiative in this area. Native WebRTC support in Safari (thus the entire Apple ecosystem of devices), effectively closes the circle of web browser support, making the technology *even more* attractive to existing and coming providers of real-time communication services, such as Adobe.

Adobe believes that HTML5/WebRTC will solve three important challenges:

- Content compatibility: The promise of HTML5 is that content (e-Learning or otherwise) will work well on desktop and across mobile devices.
- Content portability: HTML5 will allow users to content create once and deploy it across platforms.
- Cross-browser real-time collaboration: WebRTC aims to provide cross-browser real-time collaboration.

Microsoft, which is already active in the standardization process, also have clear intentions of a plugin-less future for its Skype portfolio:

> *We also want to ensure web-based Skype experiences connect smoothly with the hundreds of millions of other Skype and Skype for Business clients running on desktops and various mobile platforms around the world. As a result, we're making significant investments in standards-based, protocol-level support for ORTC and WebRTC interoperability across our platforms* [SKYPE].

Another major provider of communication solutions, Vidyo, announced its support for WebRTC mid-April 2016 [VIDYO].

The standardisation of browser APIs that offer high-quality RTC functionality on every modern device significantly lowers the barrier of market entry. It implies that vendors run a significant risk of being overtaken by new players offering solutions that better serve users' needs than current mature products, as pointed out by Clayton Christensen in his seminal work "Disruptive Innovation" [CHRISTENSEN].

With all browsers supporting WebRTC, over 1 billion people have WebRTC technology in their hands, just waiting to be unlocked.

2.3.1 WebRTC and the Internet of Things

The Internet of Things (IoT) implies networked objects other than the devices we have already grown accustomed to, e.g. computers, mobile and wearable devices. In the world of IoT, just about anything can be connected and exchange information. A number of futuristic scenarios have already become reality in our increasingly connected world, with broadband Internet, Wi-Fi, smart phones and sensors becoming cheaper and mainstream. Soon, modern cars will likely be able to automatically pay for fuel/parking (a "thing" in the car communicates with a "thing" on the pump/meter), be controllable over the network by other "things" (e.g. an IoT thermometer may trigger the car's heater), and so on.

Gartner's forecast from late 2015 suggested that:

> *6.4 billion connected things will be in use worldwide in 2016, up 30 percent from 2015, and will reach 20.8 billion by 2020. In 2016, 5.5 million new things will get connected every day* [GARTNER].

Cisco, Juniper Research and IDC predict even higher numbers. As expected with technology in an early stage of development, a range of predictions have been made pertaining to the possible increase of connected devices (up to 200 billion). Regardless of which of these proves the most accurate, it is clear that WebRTC will play a central role in the future of IoT, as it aims to:

> *...enable rich, high-quality RTC applications to be developed for the browser, mobile platforms, **and IoT devices**, and allow them all to communicate via a common set of protocols* [WebRTC].

A key strength of WebRTC, and one that makes it particularly pertinent to IoT, is what the WebRTC project refers to as 'a common set of protocols'. The Bluetooth wireless technology standard, an important enabling technology for IoT, enables *short-range* communication between billions of fixed and mobile devices. WebRTC, on the other hand, is designed to exchange data over the Internet using open and freely implementable core technologies, such as HTTP and TCP/IP.

To what extent WebRTC may be utilised in the context of IoT, and how IoT may be used to benefit R&E, is up for speculation. It is nonetheless an area that is gaining traction (e.g. IBMs multi-billion dollar IoT-related launch [IBMIoT], and one that could further spur, and be spurred by, implementation of WebRTC standards.

2.4 WebRTC Technology Conclusions

This section described the core features of WebRTC (what they are and what they do) and the major parties involved in its standardization, support and implementation. It provided a snapshot of the technology's current capabilities, as well as a brief look at market trends and likely evolution. There are a number of key points:

- WebRTC is defined by the organisations that articulate web and Internet standards.
- The majority of browser vendors work together to ensure WebRTC interoperability between browsers and platforms.
- Several key players work together to provide tools and resources (e.g. APIs and frameworks) that facilitate development of rich and high-quality RTC applications.
- WebRTC (standard and project resources) is open and free to use in any context.
- Although the standard has not been not finalised, WebRTC has tremendous momentum and is ready for use, by developers and users alike.
- WebRTC improves current applications, in particular, videoconferencing, with superior ease of use as well as high quality audio and video.
- WebRTC enables contextual communication on an unprecedented scale: putting real time communication functionality in *any* web application with low effort and cost.

The most apparent use cases for WebRTC are audio/videoconferencing and applications for exchanging arbitrary data (e.g. chat and file transfer). A number of free and open production services that exhibit these capabilities may already be accessed. So too is the availability of open source frameworks and components to assist developers in creating new WebRTC-powered applications. In addition to the core functionalities that enable these service implementations, the standard also includes a number of supporting features that achieve extended functionalities, although not all of these are, as of yet, sufficiently supported by web browsers to be used in production services.

While WebRTC services may already be enjoyed on just about any device, the advent of WebRTC for WebKit (thus the Safari web browser) ensures native support across all major web browsers, platforms and devices. The recent news about this will surely resonate through the industry, as it further solidifies WebRTCs position. With giants such as Google, Microsoft and Apple on board, it is increasingly difficult to imagine WebRTC not becoming an integral part of their future RTC portfolios. This is bound to impact the overall market, especially when considering that a number of powerful players, such as Mozilla, Opera, Cisco, Ericsson, Adobe, Intel, Oracle, IBM, Vidyo, Facebook and Blackboard, are already involved. WebRTC may be an enabling standard for other emerging technologies, such as those pertaining to IoT.

SA8 Task 2 is composed of a multi-disciplinary team in scattered geographical locations. As a result, our work has heavily depended on RTC. Team members use different platforms and hardware, and our systems operate on Windows, OSX, Linux, Android and iOS. In our experience, although subjective and not easily quantified, WebRTC already provides superior or at least equal) web conferencing services to available alternatives (including a number of which we, as NRENs, provide as production services to our institutions). WebRTC works out of the box (or, should we say, web browser), on any device, provides high audio and video quality, a positive user experience and is stable.

3 Task Results and Deliberations

This SA8 Task 2 deliverable has been primarily informed by a number of technology scouts. These were designed and undertaken in order to provide first-hand experience and understanding of key facets of WebRTC pertinent to the European R&E context.

Connectivity is, of course, a basic requirement for real-time communication. Various network implementations, however, add layers of complexity that can interrupt or even prevent WebRTC connections altogether. These complexities must be addressed in order to reduce failure rates that would otherwise render any WebRTC-based service unusable for many. A recent report from the WebRTC monitoring cloud service *callstats.*io also supports this [WebRTCSTATS] and shows that 1 in every 8 sessions are never set up due to NAT/firewall traversal issues. To this end, we undertook a technical exploration to learn more about ICE, STUN and TURN; three widely accepted open standard-based protocols that address and solve complex problems with NAT/firewall traversal and IPv6 transitioning. The technology scout provides rationales for each protocol, applicability to our community, a proof-of-concept implementation of the technologies, as well as proposing a service recommendation.

WebRTC is first and foremost a technology standard that lends itself to services for audio/videoconferencing, with a number of such services, both free and commercial, readily available. There are, however, a number of reasons why it is important to investigate the feasibility of developing and hosting our own WebRTC service — e.g. experience, technology/market maturity, scalability, usability, diversity, features, interoperability, federations, flexibility, cost, predictability, and so on. The French NREN, RENATER, has offered a WebRTC-powered videoconferencing solution, RENdez-Vous, as a production service since 2015. The service informs two of our technology scouts, so as to better address and understand the many facets involved in setting up and rolling out a WebRTC conferencing service to an R&E community on a global scale:

- The document *RENdez-Vous: 1 year of operation experience*, provides an insight into the history behind the service, motivation, experiences, benefits, challenges and future direction.
- RENdez-Vous presents a brilliant opportunity to further investigate the technical and practical feasibility of various deployment configurations to realize a theoretically unlimited scalable multi-party web conference service. The document "Distributed RENdez-Vous" provides the full story.

The diverse nature of our R&E community demands a variety of services to support, enhance and add value to the teaching and learning process. These services are increasingly based on web technology, as is WebRTC. This opens a plethora of new opportunities for integration and interoperability between different tools and services. WebRTC, in particular, holds great potential with regard to contextual communication, and the many ways it can transform processes of teaching and learning. The "WebRTC Media API Service proposal" describes the concept of an easy to implement API that makes it possible to create, integrate (in-context) and control WebRTC applications inside other services such as Virtual Learning Environments [TF-WebRTC-WIKI]. The "WebTut" technology scout provided a prototype that demonstrated contextual communication in a realistic application for teaching and learning.

As mentioned, real-time communication in the form of audio/videoconferencing is the most common application for WebRTC. However, it also provides building blocks to enable a number of other services (e.g. P2P file sharing, gaming and data exchange for the Internet of Things), many of which may provide useful applications for R&E. Services for lecture streaming and recording are already commonplace in universities worldwide, and WebRTC may also add value in this area. To learn more about WebRTCs readiness to power applications in this domain, the technology scout "Stream and record lectures with WebRTC" deconstructs and investigates the availability, background, features, maturity and implementations of APIs required to build a screencast web application (see Table 3.1).

WebRTC is not deployed in a vacuum. Most R&E institutions have or are planning deployment of a Unified Communication (UC) solution. In the UC technology scout the impact of WebRTC on UC, and vice versa, was explored.

The section concludes with a brief summary of community outreach and market dialogue work through the TF-WebRTC.

Each technology scout includes an introduction to the topic, a description the work undertaken by SA8T2, provide the reason for the particular tech scout and summarize the results for that particular tech scout. We then discuss these results, put them in context with other technology scouts, our knowledge of WebRTC as well as our market and community observations and describe our conclusions and recommendations. An overview of all technology scout reports is provided in Table 3.1.

All technology scout reports, as well as any pertaining resources (e.g. source code), are available via the TF-WebRTC wiki [TF-WebRTC].

Section ref	Technology scout	Topic	Rationale	Result
Section 3.1	ICE, STUN and TURN: How WebRTC deals with firewalls and NAT	Addressing NAT and firewall traversal with peer-to-peer WebRTC communication	Investigation into important infrastructure component that could be offered by NRENs	Report, demonstrator
Section 3.2	RENdez-Vous: One year of operational experience	Describes experiences of the French nationally deployed WebRTC desktop videoconferencing service	Harvest experience from first national deployment of a native WebRTC desktop videoconferencing service	Report

Section ref	Technology scout	Topic	Rationale	Result
Section 3.3	Distributed RENdez-Vous: An investigation into scale-out Jitsi	Assess possibilities for Jitsi software to scale out to a distributed deployment supporting all of EU R&E	RENdez-Vous would be a possible candidate for a shared EU R&E desktop videoconference system, if the choice is build-your-own.	Report
Section 3.4	WebTut: A contextual-communication PoC	A proof of concept implementation of an online web tutoring service	Assess difficulty and cost of building an in-context application/VLE	Report, Demonstrator
Section 3.5	Service proposal for a Media API Service	Proposes a service offering high-level, real-time communication building blocks for contextual communication	Possible lower integration cost for in-context communication, possible infrastructure component in NREN infrastructure	Report
Section 3.6	Screencast: Stream and record lectures with WebRTC	Investigate recording with WebRTC	Opportunity for simplification of lecture/screencast recording and streaming, new services possible	Report
Section 3.7	Unified Communication and WebRTC	Assessment of impact of WebRTC on Unified Communication in R&E	R&E trend is towards UC deployments at every R&E institution	Report
Section 3.8	WebRTC2SIP gateway	Explore opening the legacy world of SIP for browser-based communication	Address interaction of new technology with installed base	Report
Section 3.9	JANUS — General purpose gateway	Explore JANUS' capabilities through a PoC	Circumvent WebRTC's peer-to-peer limitations in regard to number of concurrent users in a room	Report, PoC

Table 3.1: Technology scouts overview

3.1 ICE, STUN and TURN: How WebRTC Deals with Firewalls and NAT

3.1.1 Introduction

In this section we sum up the ICE, STUN and TURN technology scout.

As described in Section 2.1.5, the Interactive Connectivity Establishment (ICE), Session Traversal Utilities for NAT (STUN) and Traversal Using Relays around NAT (TURN) protocols are mandatory to implement for WebRTC-compliant endpoints. STUN/TURN parameters are automatically negotiated by WebRTC clients when needed.

A major cloud service provider for WebRTC statistics, callstats.io, stated that *more than 10% of WebRTC talks would fail* without these protocols because an audio and/or video stream could not be set up [WebRTCSTATS]. For the end user, this typically manifests itself by being unable to hear/see other participants, or be seen/heard themselves, which leads to "it doesn't work" frustration. ICE, STUN and TURN prevent this unfortunate user experience from happening.

While capable web developers at R&E institutions can easily include peer-to-peer real time communication support in any web application, with no central signal multiplexing infrastructure required, a significant number of connections will fail without support for NAT and firewall traversal.

3.1.2 Rationale

Most R&E users encounter firewalls and NATs on campus, at home and on the road. These middle-boxes cause problems for real time communication data streams (audio, video, data). The ICE, STUN and TURN protocols are the widely accepted open standards to address and solve complex problems with NAT/firewall traversal and IPv6 transitioning. Any WebRTC deployment in R&E will therefore need to address how it deals with ICE, STUN and TURN (STUN/TURN). This technology scout provides the technology background for a recommendation on a STUN/TURN infrastructure recommendation for the European R&E community.

3.1.3 Summary

The technology scout started with desk research into STUN/TURN technology standards, available products and services and how this relates to WebRTC. After this phase it was clear that anyone wanting to deploy WebRTC services, applications (especially peer-to-peer based) or gateway boxes would quickly find the need for a STUN/TURN service. Such a service would have to be procured or built.

To facilitate a build or buy decision, a STUN/TURN proof of concept service to support both WebRTC applications and legacy SIP devices was built. The goal with the PoC was threefold:

- Explore the use of federated authentication to allow WebRTC applications to authenticate to a STUN/TURN service.

- Explore the feasibility of building a distributed STUN/TURN service in the GÉANT community with a high degree of deployment automation.
- Gain practical experience with STUN/TURN server technology for use in the requirements specification process of a possible procurement or building a community service.

For the PoC, the most popular and mature open source software product for providing a STUN/TURN service, coTURN, was used. coTURN is also used in, for example, Google Hangouts. The PoC implemented a STUN/TURN service and an associated web portal with federated authentication.

- **A STUN/TURN service** with a central control node capable of automatically deploying STUN/TURN nodes on IaaS-clouds using Ansible automation scripts. This allows STUN/TURN nodes to be located as close as possible to the end user requiring the functionality, which reduces latency in the real time communication traffic and improves user experience.

  The PoC had modest system requirements and we estimate an equally modest effort is required to operate a service based on the PoC. The underlying assumption is access to one or more STUN/TURN experts for subject matter expert (Tier-3) support.

- **A web portal with federated authentication** through eduGAIN, to obtain both a never-expiring, long-term credential and a time-limited, long-term credential.

  The *long-term credential* is typically a username/password used by newer multi-protocol video bridges and legacy SIP/H323 appliances to authenticate to the STUN/TURN service [eduGAIN].

  The *time-limited long-term credential* is used by WebRTC applications to transparently authenticate the end user's WebRTC client application to the STUN/TURN service, From the end user's perspective "it just works", even a login is not needed as the web application owner needs the eduGAIN authentication to seed the application with the credential. The coTURN software needed to be modified to support the time-limited long-term credential mechanism.

  A *third authentication mechanism based on OAuth* is currently being defined by the IETF and is on the roadmaps of the big browser vendors. As implementations are currently lacking, it was not possible to investigate this further.

The PoC was tested with the WebTut contextual communication technology scout (see Section 3.4) and was a success. STUN/TURN nodes were deployed to NIIF, UNINETT and FCCN.

3.1.4 Conclusions

- Every peer-to-peer WebRTC application needs access to a STUN/TURN service to address NAT and firewall traversal. If unaddressed, about 10% of users will experience audio, video and data streams not being established. Gateway boxes like Pexip's multi-protocol videoconference bridge have a similar requirement.
- There are two main security considerations:
  - Without authentication the TURN server, in particular, can be abused for anonymously relaying arbitrary traffic.

  - Calls are encrypted and protected against eavesdropping on the STUN/TURN service. It is, however, possible to gather meta-information about which endpoint communicates with which endpoint and when.
- A STUN/TURN service should support a credential mechanism to authenticate WebRTC web applications in a secure and user friendly way. The service should support eduGAIN to let the web application owner initialize this credential.
- A STUN/TURN server should be close to the end user in need of NAT traversal. This reduces latency in the RTC streams resulting in a better real time user experience.
- There are currently two relevant STUN/TURN providers in the market. They typically charge for bandwidth used by TURN.
- Using the coTURN open source software we demonstrated the feasibility of building a highly automated distributed STUN/TURN service that could be deployed in the GÉANT service area. The manpower needed for operations is estimated to be around than 0.25 FTE under the assumption one has access to a STUN/TURN expert for Level 3 support.
- The Proof of Concept is accessible via the TF-WebRTC pages [TF-WebRTC].

3.1.5 Discussion and Recommendations

WebRTC has been primed over the past few years and the technology and implementations are now mature enough to be useful. Institutions are already trying out the technology, e.g. by developing simple contextual communication trials, testing simple WebRTC services and procuring gateway boxes.

It is essential that the European R&E community have access to a STUN/TURN service in order to stimulate adoption of peer-to-peer WebRTC applications and modern multi-protocol video bridges. Without it, *at least 10% of sessions will fail!* In the eyes of the end user, such rates will render WebRTC a broken technology rather than the easy to use real time communication solution it can and should be. The lack of a STUN/TURN service must not be an excuse for shying away from WebRTC.

We are not, however, in a position to make build or buy recommendations. At the moment, there is no quantifiable demand for STUN/TURN in European R&E. Although some market parties offer STUN/TURN services, we do not know if they can integrate with existing R&E infrastructures; notably the eduGAIN authentication federation and the R&E network. We also lack practical experience with using these commercial services.

The GN4 SA7 activity on cloud services informs us that the R&E service providers in different European countries have differing opinions on cloud services. While we do expect these opinions to converge in the next three years, they currently obscure our target of a shared STUN/TURN service to stimulate WebRTC adoption.

A built service is best achieved as a shared European infrastructure via GÉANT, as it is the only way to achieve a distributed, and ultimately global, footprint against low cost. National-only deployment will lead to sub-optimal user experiences for users on the road, as well as increase the total cost of service.

The commercial service path is currently unclear and will take time to resolve, but may prove to offer better value for money over time than what an in-house service can deliver. The commercial path vs.

in-house provisioning of a STUN/TURN service should be properly investigated to enable an informed sourcing choice to be made in due time.

The STUN/TURN PoC created by Task 2 is a scalable working solution that can be productized to a pilot service that can be put to use immediately. In the next couple of years, both the commercial and community service strands should be pursued.

The Task 2 team recommends the GÉANT community to:

- Move the STUN/TURN PoC to a 2-year pilot service operated by GÉANT as part of a collaborative project. After the 2-year period a decision can be made whether to continue the service as part of the GÉANT service portfolio or not. This gives the European R&E community a time window in which to explore using STUN/TURN services against easy conditions, e.g. no invoicing, eduGAIN authentication. It avoids stalling WebRTC adoption with discussions regarding desirability of using cloud services for STUN/TURN
- Document the technical requirements for STUN/TURN services for R&E and make a recommendation based on these on possible inclusion of STUN/TURN cloud services in the GÉANT cloud catalogue. This requires practical tests of at least some of the services.

NREN recommendations:

- Ensure institutions understand the purpose of a STUN/TURN service with WebRTC technology. The goal here is to prevent dismissing the technology as immature because connections are blocked by NATs and firewalls. This is an awareness raising activity that not only needs to target technologists developing and deploying (web)applications but also support teams for e-Learning and e-Research.

3.2 RENATER RENdez-Vous: One year of Operational Experience

3.2.1 Introduction

The French NREN, *RENATER,* was the first NREN to provide a native WebRTC-based online desktop audio/videoconferencing service to its community. The RENdez-Vous service was set up as a pilot in the fall of 2014 and announced as a production service in March 2015. eduGAIN authentication was added as part of RENATER's national WebRTC effort during 2015 to allow harvesting experiences from a wider user community [RENDEZ-VOUS], [eduGAIN]. The service builds on the open source software product Jitsi, and is currently the only known example of a Selective Forwarding Unit (SFU) system.

As part of Task 2, RENATER wrote down its experiences with operating and delivering the service. This paragraph summarises those experiences.

3.2.2 Rationale

To systemise RENATER's hands-on experience with operating, delivering and supporting a native WebRTC desktop videoconferencing solution. This information is particularly apt with regard to deliberations to facilitate possible build-or-buy decisions in European R&E.

3.2.3 Summary

RENdez-Vous is the first widely deployed WebRTC use case within the French R&E community. It provides an easy to use desktop videoconferencing solution that, at the same time, is so affordable and scalable that it can be used to pave the way to a digital collaborative platform.

The service has been well received by the user community as an easy to use and good enough alternative to Skype (a service that users in the French public sector are actively discouraged from using due to security considerations). RENdez-Vous offers a good "one click and use" experience, similar to previously mentioned services such as Talky.io and appear.in. Despite little active promotion the service uptake caught RENATER completely by surprise, with approximately 10 000 unique users each month using the service for 2500 conferences.

RENdez-Vous has required significant less support than traditional H.323 videoconferencing systems. Relative to usage volume, RENATER’s support system recorded 1/5 less tickets pertaining to RENdez-Vous than those for H.323 service. The nature of support requests was also different, those for the WebRTC service were typically software-based and less complex to solve than those for the H.323 service. Problems with H.323 often pertain to network, firewall and configuration issues. Support issues for RENdez-Vous were typically related to incompatibilities between browser and the server software after a browser update, users not being used to the audio and web cam controls of their operating system and non-WebRTC issues pertaining to the server software.

RENdez-Vous is built with Jitsi, an open source video bridge using a Selective Forwarding Unit (SFU) to distribute media streams. Where a Multipoint Conferencing Unit (MCU) mixes different media streams to a single one, a SFU is capable of deciding which media streams should be forwarded to specific participants. Compared to the full-mesh with peer-to-peer WebRTC conversations, a SFU means a participant receives streams from all participants but it sends only one stream. This uses less CPU and bandwidth on the user's desktop. The difference in functionality between a SFU and a MCU means a SFU-based system requires less resources compared to an MCU, making it cheaper to procure, operate and scale. This is confirmed by the RENdez-Vous experience, modest processing resources are required to provide the service: a handful of not too high-end virtual machines and the staff to operate those and the application.

The service was developed and operated in a close partnership with Blue Jimp, a French university-originated start up and the lead developer of the Jitsi software. Blue Jimp was recently bought by Atlassian, specialists in monetising open source enterprise collaboration software, such as the Confluence wiki used by GÉANT.

This partnership brought about new features such as SAML 2.0 integration. Recording functionality and a media API is currently being worked on.

Future work is tightly coupled with feature requests from the user base:

- Improve user experience, features and audio/video quality.
- Add features indicating the qualitative user experience in a meaningful way (jitter, latency etc.). A user needs to get user friendly feedback indicating one of the other participants no longer hears them and how this may be resolved.
- Improve mechanisms for user feedback to the service provider.
- Add a presence management component to support ad-hoc meetings.
- Integrate RENdez-Vous with existing SIP solutions to provide audio, as a minimum.
- Recording and other typical webinar behaviour, such as that used in Adobe Connect (highly requested).
- Device and application compatibility: ensure RENdez-Vous works in popular browsers while the WebRTC standards are still being worked on, and consider support for mobile devices.

3.2.4 Conclusions

The first year of operation experience with RENdez-Vous has been very exciting and rewarding. Judging by community feedback, building a new service on top of innovative WebRTC technology has been both beneficial and successful. While the service is young and has room for improvement, both users and CIOs of French R&E institutions have nonetheless embraced it.

RENATER recognises that the user experience currently offered by RENdez-Vous has room for improvement. While the one-click start capability is brilliant in its simplicity, the service may still be adjusted to allow for an even smoother experience, particularly to assist those new to videoconferencing.

3.2.5 Discussion and Recommendation

In an environment where the institutional offerings of RTC services were lacking, and the use of free services (such as Skype) were strongly advised against, RENdez-Vous addressed a very real demand in the French R&E community. This is evident from the overwhelming uptake, support and feedback of the service, directly from the end-users, as well as CIOs, from several institutions.

RENATER has effectively provided the rest of the European R&E community with a large-scale field test of an end-user facing native WebRTC conferencing service. The RENdez-Vous story shows a WebRTC production service may be achieved with moderate levels of technical and human resources. It also clearly demonstrates the WebRTC technology is mature enough to be used by large user groups.

Where most commercially available WebRTC conferencing services use peer-to-peer for sending and receiving media streams, the Jitsi software underpinning the RENdez-Vous software has a SFU. Both approaches have pros and cons. Peer-to-peer lets media traffic take the shortest possible path between meeting participants leading to a good user experience. A SFU-based approach opens for larger participant groups, indicating a group size of 15 should be possible to support. This is important in R&E with its teaching scenarios and plenary discussions.

NRENs have expressed a need for a service like this, as not all R&E online meeting scenarios can live with the limitation to the number of participants a peer-to-peer service by its nature has.

The results of the RENdez-Vous service study combined the Task 2 team's own experiences with native WebRTC conferencing tools and our observations from the market all point to the same direction: WebRTC is mature enough to support the use case of simple videoconferencing calls for both small and larger user groups.

Many free native WebRTC tools offer user experiences and high-fidelity audio superior to most commercially available, non-WebRTC, solutions.

As far as its own observations are concerned, the Task 2 team concludes the mandatory OPUS audio codec performs well, also over long latency connections. For example, in a recent meeting between UNINETT (Norway) and DeIC (Denmark) using a WebRTC tool, one participant from DeIC noted that using a telephone for communication was unpractical as he was located in Singapore. Until this point, participants in Norway were oblivious to his location; nothing in the audio/video quality or latency gave it away.

Regarding ease of use, two observations are key:

- Finally, a videoconference may be initiated in any (WebRTC-compliant) web browser simply by sending participants a meeting room URL — without any further set-up/installations involved. It will just work, out-of-the-box.
- WebRTC is a native web technology, meaning that it can be sliced, diced, wrapped, packaged and presented in web applications in as many different ways as there are web designers. As recognised by WebRTC start-ups, abstracting the difficult part of real-time communication technology foundation away through simple native web-APIs means service developers can truly focus on the user experience.

The promise to simply click and talk is finally delivered.

Based on the Task 2 team's experience and the RENATER field test with RENdez-Vous, there is great value in having a simple desktop videoconferencing service for R&E users. We need to ensure all European R&E users have access to such functionality.

It is proposed that all European R&E users have access to a native WebRTC-based, easy to use, no-nonsense desktop videoconferencing service. This service would replace the telephone as the new lowest common denominator for real-time communication for collaboration involving European R&E users.

Key features of such a service would include:

- Zero-install — the service MUST run in a standard WebRTC compliant browser with no need for extensions.
- No clutter — the service needs to support multi-part audio and videoconferences with chat in advance of adding other features.

- Authentication with eduGAIN — a user should be able to start using the service without any initialisation steps.

This would allow collaborations across organisational boundaries without searching for a usable real time communication tool first. While a number of services that support few participants are readily available, other use cases should be supported by services made available in the GÉANT cloud service portfolio.

Recommendations for the GÉANT community:

- Make the adoption of useful WebRTC services easier by adding them to the GÉANT cloud catalogue, with framework agreements, where opportune/appropriate.

- In particular, make one or more easy to use WebRTC desktop videoconferencing services available to all European R&E users through the GÉANT Clouds Service catalogue, as soon as possible. This creates a new lowest common denominator solution for simple online group conversations.

- Some NRENs prefer a commercial cloud service. Others prefer to build their own or to use a community service. The discussions about using cloud services are at different stages in different countries. Our tech scouts clearly show both options are viable for bringing desktop videoconferencing to the masses. We recommend investigating whether the RENdez-Vous service can be added to the GÉANT cloud service catalogue.

- RENATER will continue its RENdez-Vous effort. It would be very beneficial for the wider R&E community to be able to learn from the experiences of this large-scale field-test of native WebRTC (and SFU) technology. We recommend enabling RENATER to provide at least an annual update of their experiences.

Recommendations to NRENs and R&E institutions:

- Ensure your users have access to an easy to use zero-install native WebRTC desktop videoconferencing service, compliant with local policies.

3.3 Distributed RENdez-Vous: An investigation into Scale-out Jitsi

3.3.1 Introduction

In the previous section, we learned about RENATER's experiences with their RENdez-Vous service. Based on the technologies underpinning RENdez-Vous, this technology scout sought to investigate their potential in realising a larger-scale distributed service to facilitate more users in the wider GÉANT community.

3.3.2 Rationale

The national RENdez-Vous service pilot utilises the Jitsi open source software, using a Selective Forwarding Unit (SFU) as a central server component rather than routing media traffic peer-to-peer. Experiences with R&E web conferencing services show online meetings of eLearners and researchers often reach a group size of 10–15 people, which is too challenging for the peer-to-peer technology used by most commercial native WebRTC solutions.

It made sense to investigate the technical potential for a scalable GÉANT desktop videoconference service based on Jitsi: a distributed RENdez-Vous.

3.3.3 Summary

The explored technical Jitsi-setup involved a central control node and distributed SFU nodes. Users enter via the central control node where authentication and authorisation is handled. The media traffic is handled on a specific SFU node, which could be the meeting host's national SFU node. A SFU node would automatically be instantiated on a local (national) IaaS service. This concept would mean a very scalable desktop videoconference service with a cost-by-use cost scaling model. No heavy investment in large-scale online infrastructure is needed.

Using the global NREN network footprint, as well as the GÉANT community's ability to set up collaborations with overseas NRENs, this could then be scaled out to a global footprint.

The primary concept was validated with 2 SFU nodes in 2 countries, the distributed SFU model worked. Unfortunately, the project staff member undertaking the work left for a new job and there was not enough time available to replace his unique expertise in the project. As such, the automation of the process could not be validated, or tested on an even larger scale.

3.3.4 Conclusions

The technical concept was validated: distributed SFUs worked across national borders. Departure of a key project member resulted in a lack of progress on process automation for SFU instantiation.

3.3.5 Discussion and Recommendations

The Jitsi software, and thus the RENdez-Vous service, provides something that seems difficult to find in the current commercial WebRTC market: the ability to support larger user groups with a native WebRTC service. We recognise this from earlier ventures in the web conferencing solution market, where few vendors support a requirement to support many concurrent talking heads.

Both technology scouts pertaining to RENdez-Vous provided valuable input that led the task to explore the idea of a Media API Service, for which there also is little support in the market. The Jitsi software is nevertheless on its way to implement some basic features of the concept.

One member of the European NREN community (RENATER) now has a service that caters to users with a needed technology, the SFU. The service enables us to further explore ideas for a Media API, using the French user community as a testbed. The technology used by RENATER is open source, and through RENATER, the GÉANT community has a well-functioning collaboration with the lead developers of the software. An investment in Jitsi may therefore be leveraged by the wider community and offers a golden opportunity to further explore the concepts of SFU, Unified Communication presence propagation, integration etc., on a live testbed.

Recommendations for the GÉANT community:

- Use RENATER's RENdez-Vous effort to further explore the concepts of scalable SFU-based native WebRTC desktop videoconferencing services and further develop the concepts of a Media API Service, especially integrating UC presence propagation.
- Closely track the market for solutions similar to a distributed SFU and strongly consider vendor collaborations for development of a Media API Service concept.

3.4 WebTut: A Contextual-Communication Proof of Concept

3.4.1 Introduction

One of the key benefits of WebRTC is how it allows for inclusion of real-time communication in any web application with the same ease as adding any other standard web feature. The concept is commonly known as contextual communication, which makes for a very smooth user experience. The web developer may choose which real time communication features are used where and when; audio conference, videoconference and data sharing can appear exactly where they are needed inside the same window, without cluttering other parts of the application. This bodes well for e-Learning and e-Research applications — e.g. including RTC at the right place in a learning plan for a MOOC or VLE was never as straightforward as with WebRTC. In addition, with lower development costs expected as a result of this new approach, one can also imagine smaller e-Learning and e-Research applications dedicated for specific processes, e.g. for tutoring or research proposal writing.

3.4.2 Rationale

Given the potential benefits of contextual real-time communication, it was important to investigate the actual effort involved. How consuming is it to build a PoC demonstrating a useful contextual communication application for R&E? How mature are the browser implementations of required WebRTC APIs and software libraries using them? These were some of the questions this technology scout area sought to answer.

3.4.3 Summary

As a proof of concept, a small web application was developed for a web tutoring service, which enables an expert (the teacher) to be available for a tutoring session for students. The students queue and are let in one after another. Authentication was implemented using eduGAIN. Peer-to-peer WebRTC was used to allow the teacher and student to talk together. Features that indicated to a user whether audio and video were working before engaging in a session were implemented. The web tutoring, WebTut, PoC was tested with the STUN/TURN Proof of Concept service for NAT and firewall traversal.

The application was demonstrated at the GN4-1 Symposium in March 2016. A demonstration screencast is available on the TF-WebRTC project result page [TF-WebRTC].

Figure 3.1: Teacher waiting for student in WebTut session

3.4.4 Conclusions

- Adding WebRTC to a web application is now easy. In fact, developing the queuing mechanism took more time than adding video calling functionality.
- The expert-layman model can easily be transposed to other domains, for example doctor – patient applications. Although the user interface requirements and context would differ, the principle is the same and demonstrates the value of low-cost contextual communication.
- From discussions during the development stage, it was clear how the WebRTC real-time data channel may be useful in a doctor–patient context: transferring medical sensory output from a device worn by the patient to the doctor while having an online consult. One would assume similar uses could be found, especially in research contexts where sensors or instruments provide data and researchers communicate about that data.
- Feedback from the Portuguese R&E community indicated an interest for a web tutoring service such as the WebTut Proof of Concept.

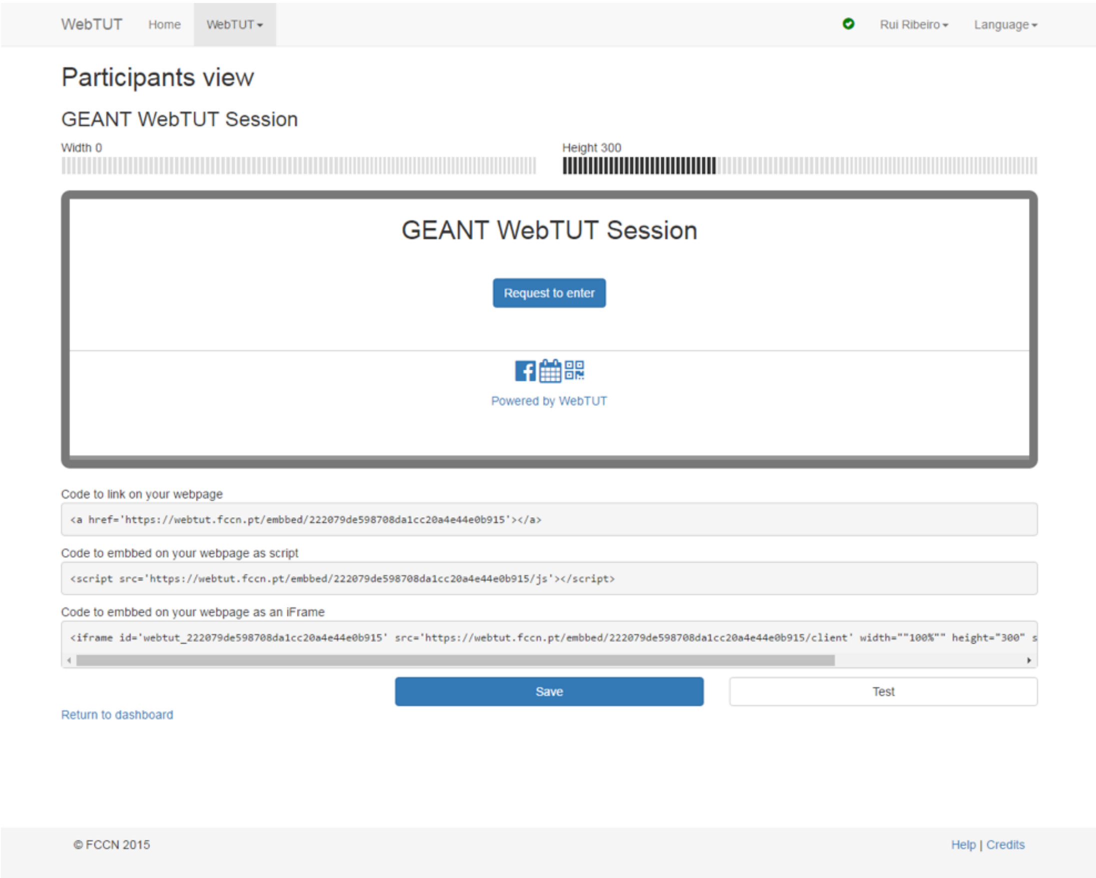

Figure 3.2: WebTut — integration management

3.4.5 Discussion and Recommendations

The WebTut web tutoring PoC, as well as observations in the field, signify the potential of contextual communication using WebRTC. It allows for bespoke integration of real-time communication support in many different e-Learning and e-Research services, both nimble and complicated, at low cost.

Any R&E institution wanting to create in-context WebRTC applications will need to consider the following:

- Management, support functions and users must understand enough of the concept and functionality it enables in order to be able to formulate their requirements.
- The organisation must develop its technical capability. Regardless of whether development is typically done in-house or outsourced, the ability to specify requirements must be developed. If in-house development is desired, the developers need to (be) train(ed) on the use of WebRTC.
- Access to a STUN/TURN service must be available (see Section 3.1).

We recommend the GÉANT community to:

- Develops an online resource with recipes and code examples (cookbook), making it easy for institutions to find their way in the various software components when implementing contextual communication using WebRTC.
- Help NRENs to actively promote the contextual communication concept to R&E institutions and stimulate their members in capability building.

We recommend for NRENs and R&E institutions to:

- Develop technical capability for leveraging WebRTC in contextual communication, including skills for procurement, development and support.
- The Web Tutoring proof of concept (WebTut) looks like an idea with potential. Consider a joint (open source) effort to further develop the Web Tutoring service idea.

3.5 Service Proposal for a Media API Service

3.5.1 Introduction

An application programming interface (API) provides the developer with building blocks to make service development faster and easier. Faster, because the API typically provides well documented and well defined routines to achieve frequently requested functionalities. Easier, because the API abstracts away many complexities of the underlying systems. Since APIs define a standard way of operations (e.g. routes, protocols, data structures) they implicitly promote reuse.

SA8 Task 2 proposes the development of a Media API Service, a centralised web service that presents a set of tools (as a JavaScript library) as-a-service that makes it trivial to add high-level WebRTC components to any web application. The API service can be integrated with other central, national or institutional infrastructure components, e.g. recording services, authentication and authorisation, logging, captioning, presence (see Section 3.7), legacy infrastructures (SIP), etc.

The proposed service will allow for the creation of a standard set of well-functioning virtual lecture rooms that are easy to embed in any virtual learning environment. It will also facilitate standard components for multi-party videoconferencing to be embedded in the context of any other web based tool. The service will lower the cost for contextual communication as well as to increase the quality of such integrations.

By allowing a deconstruction of virtual room functionality the service can, in a simple and affordable manner, be used to select and add feature and UI combinations/subsets inside any web application. The implied use-cases are limitless and the benefits potentially priceless.

The service proposal is a natural extension of SA8 T2s focus on WebRTC's enabling potential for contextual communication (see e.g. Section 3.4 on WebTut).

3.5.2 Rationale

WebRTC may be used to enable in-context communication (i.e. embedding RTC capabilities inside a non-RTC web application) at low cost. The proposed Media API Service facilitates domain-specific in-context communication to virtually any web-based service.

Requirements for contextual communication in R&E are normally quite similar, thus offering the potential for reuse. The Media API Service proposition promotes high-level components to assist web developers in European R&E and spare them from "re-inventing the wheel".

In addition, a Media API Service would allow the R&E community to integrate key value-adding services at a central point. Note that this does not necessitate the services themselves to be centralised.

A GÉANT Media API Service will reduce integration time and cost for developing contextual communication in web applications, as well as increase the quality of the individual components.

3.5.3 Summary

An existing API for Jitsi gives an idea of the type of functionalities that would be expected from a Media API Service [JitsiAPI]. Jitsi's API allows meetings with a custom GUI to be created.

The Media API Service would support functions like:

- Room operations: create, delete, modify behaviour.
- Participant operations: invite, add, kick, etc.
- Participant media operations: mute/unmute, disable/enable video.
- Reporting on session QoS parameters allowing dynamic changes.

- The Media API would allow use cases such as :Add a meeting room to an LMS. The components of the meeting room and their layout should be flexibly defined by the teachers involved in each course.

- Add multipart-audio conference, chat, screen sharing and document upload capability to a NOC ticketing system to be available in the context of a trouble ticket.

- Create a simulator to train aspiring doctors in doctor-patient interactions. Create two main video windows, show monitoring teacher, show comments by teacher to group for learning purposes. Record session and link to involved student(s) portfolio. Use eduGAIN for authentication.

Twillio currently offers a similar product with some of the aforementioned functions. As the WebRTC market is in rapid development, more platforms like these will probably appear [TWILLIO].

3.5.4 Conclusions

The proposed Media API Service would include an interface that will not only simplifies the integration of WebRTC with web applications, it would also provide the R&E community with a single point of integration for value-adding services, an attribute especially important in today's world of cloud services. The API would enable the R&E community to create best practice implementations of domain-specific solutions that can then easily be integrated in any web application.

It is a logical optimisation step for supporting contextual communication.

3.5.5 Discussion and Recommendations

The Media API Service idea is not yet sufficiently developed to procure or build. Based on market observations, combined with the RENdez-Vous and the Unified Communication technology scouts, there are some potentially very interesting developments that need to be investigated further.

We recommend the GÉANT community to:

- Further develop the concept of the Media API Service by developing a Proof of Concept and build a pilot service based on this. A build-or-buy decision would have to be made after the PoC phase.

We recommend the R&E community to:

- Establish a collaborative project defining domain-specific, high-level building blocks useful for contextual communication scenarios.

3.6 Screencast: Stream and Record Lectures with WebRTC

3.6.1 Introduction

Much attention is given to WebRTC and its potential for online education in the context of videoconferencing. Less known is the WebRTC standard’s ambition to facilitate desktop capture (acquire and stream what happens on the screen) as well as recording of media streams.

“Screencast”, a type of lecture capture, is particularly suitable for presenting content that requires excellent representation of information otherwise difficult to capture with a video camera. Hence, it is a popular solution for capturing lectures centred around digital content, such as those delivered by overhead projectors (e.g. “slide-based”) and interactive whiteboards.

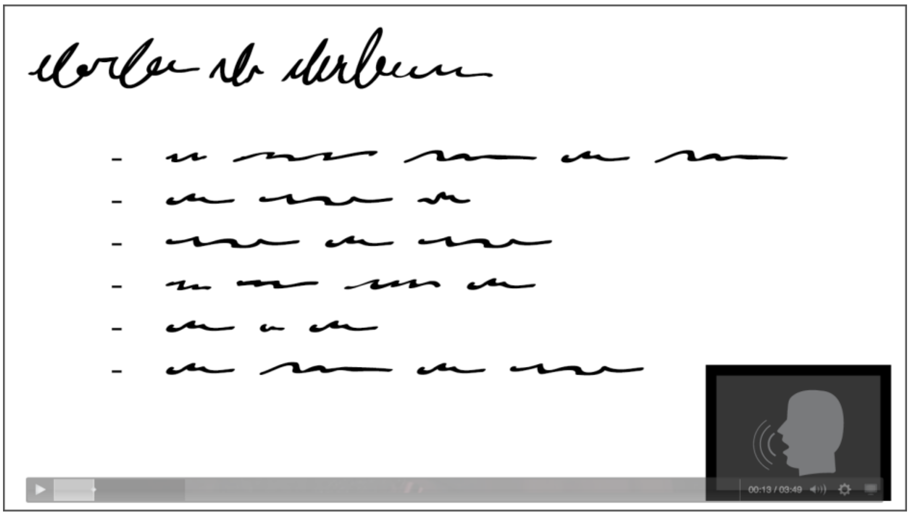

Figure 3.3: Screencast — typical layout composition

The screencast format lends itself particularly well to the APIs provided by WebRTC. Screencast multimedia compositions are typically comprised of a feed from the computer screen (Screen Capture API) accompanied by audio narration and a "talking head" camera feed (Media Stream API), all recorded to a flat video file (Media Recorder API).

3.6.2 Rationale

WebRTC-based services share many common benefits already established across all the scouts (e.g. lower cost, zero-install, better user experience, open standards, federated identities and cross-platform support, to name a few). This technology scout, which was performed in the final stages of the SA8 Task 2 working period of GN4-1, was added to complement and underline benefits, findings and recommendations from other technology scouts and proposals. In other words; to highlight that proposed WebRTC infrastructures and services (e.g. applications and APIs) may be leveraged and/or repurposed to realize new services for R&E.

In the case of a screencast application:

- The ICE/STUN/TURN infrastructure would also be essential to allow the streamed lecture reach its audiences.
- Concepts, expertise and even implementations from a Media API for in-context applications may also be of value for lecture capture solutions (and vice versa).
- Experiences and lessons learnt from operating a WebRTC videoconferencing service provides important, and likely 1:1, input to the operation of a WebRTC video recording service.
- RENdez-Vous' user interface involved research and implementation of some WebRTC APIs exactly the same as those required for a screencast service.

- Similarly, the WebTut PoC provides knowledge and hands-on experience from building a WebRTC-powered web application from scratch.

In addition, it was important to provide NREN and R&E institutions with up-to-date information regarding WebRTC-based recording for use in their dialogue with vendors and service providers of screencast and lecture recording solutions.

3.6.3 Summary

The primary objective of the technology scout was to investigate WebRTCs maturity regarding the various APIs required to implement a zero-install, browser-based, screencast (web)application for streaming and recording combined feeds from microphone, web camera and screen — utilising only WebRTC and HTML5 standards.

The scout also discusses a number of benefits that WebRTC may offer in this domain, e.g. different approaches for content management and how a web-based service is primed to leverage existing storage solutions.

3.6.4 Conclusions

The Task 2 team's findings suggest that browser implementations of WebRTC standards, specifically pertaining to recording and screen capture, are insufficient at present time. Implementations are, however, progressing at a very rapid pace.

While WebRTC falls just short of supporting a screencast application as per our specification, the technology scout nevertheless established that all required building blocks are available, at the very least as specification drafts. If compromises were made on the no-extension requirement for screen capture, an early implementation of a screencast service could commence already today.

Knowledge from this technology scout, in particular, the fact that the standard is so close to supporting our suggested functionalities, provides food for thought and further actions on two fronts:

- Make plans for preliminary developments in-house in the near future.
- Approach vendors of solutions that we employ today to enquire/encourage more focus on this area.

The benefits offered by WebRTC throughout the entire environment, ranging from the user on one end to the decision makers on the other, most certainly warrants further testing.

3.6.5 Discussion and Recommendations

Attention to open standards, which is central to WebRTC's specifications, facilitates ease of implementation, significant opportunities for integration and interoperability, future proofing, user-friendliness, broad platform support and so on. Many of these benefits are amplified when other

WebRTC-related efforts in the European R&E are taken into consideration, e.g. most of the work pertaining to the SA8T2 deliverables.

Lecture capture was investigated, in part, to highlight WebRTC's potential to enable a wider array of services than those directly belonging to the web conferencing domain. Indeed, there are still a number of other technologies and services that could be realised with WebRTC that we have not detailed in the Task 2 team's work, e.g. P2P file sharing/transfer, gaming, IoT, many of which may also be pertinent to the area of R&E.

Recommendation to the GÉANT community — Give the WebRTC recording standards and implementations time to mature and revisit this area in early 2017.

Recommendation to NRENs and R&E institutions — Engage with streaming and lecture recording solution providers regarding their plans for future support of the emerging WebRTC recording standards.

3.7 Unified Communication and WebRTC

3.7.1 Introduction

Unified Communication (UC) is sufficiently mature to replace telephone systems at R&E institutions. Solutions from Microsoft and Cisco, in particular, are offered at very favourable conditions to the R&E community, complemented with cloud delivery models that lower deployment costs significantly.

Different national communities are at different points in this transition. While the Norwegian NREN (UNINETT) seems to be at one end of the spectrum, with a shared community service for Microsoft Skype for Business, institutions in other countries are just now starting to discuss the transition to UC. The benefits of real-time communication, tightly integrated with features such as email and calendaring, are too valuable to the core business processes used in R&E to ignore them.

Extrapolating the trends in this field leads to the conclusion that in a few years, perhaps five, most European R&E institutions will have deployed a UC solution to all of their end users. These UC solutions will likely absorb current solutions for videoconferencing.

3.7.2 Rationale

If we assume that UC solutions will be present at most institutions in European R&E some years from now, they will represent the legacy technology WebRTC solutions are most likely to meet. At the moment, both UC and WebRTC solutions are being developed and deployed. This makes it very timely to investigate how the two play together and whether there are smart moves that R&E institutions should make now to support a smooth and effective RTC infrastructure in the near future.

3.7.3 Summary

Task group discussions and desk research was combined with the practical experience from UNINETTs real time communication team to do an analysis of the implications of WebRTC for UC deployments and vice versa.

An obvious, but nonetheless important, improvement WebRTC brings to UC, is improved guest access and easier deployment to licensed users. It enables UC vendors to create a web-based client that offers the same quality audio and video as native clients. This will also help address one of UC's key weaknesses: the difficulty of letting clients from different UC solutions talk smoothly together. R&E is a global undertaking with people (staff, researchers, students) from many independent organisations working together across organisational borders in collaboration patterns that cannot be pre-determined or predicted; you cannot predict which UC solution your user's collaboration partners have. Guest clients with WebRTC support will make this problem easier to address.

Less obvious is WebRTCs capability to de-unify Unified Communications. When people work together, communication is contextual. For example, they discuss the document they work on, the research data they produced, the MOOC/VLE learning plan under development, etc. Solutions that support communication should therefore also support its context. Instead of moving context into communication tools, however, WebRTC can easily and seamlessly embed RTC into any web application. This approach erodes the business case for UC solutions. Much communication will simply happen in the tools people use to get their jobs done. What is lost in such a scenario is the unified aspect of UC. There is no longer a single concept of user presence; a user is present in dozens of applications, so which is the best tool to address the user? This is called "Dis-Unified Communications".

The prospect of creating contextual-communication islands forces a change of focus for UC solutions. You cannot force users to go for the less productive solution of moving their job context into a general tool if they have the alternative of not doing so. It is possible, however, to bring presence and improved integration to the contextual applications.

WebRTC does not solve signalling, buddy lists or presence. An UC installation could act:

- As a presence engine for both dedicated and contextual communication (across organisational boundaries).
  - Serve up buddy lists.
  - Propagate a single sense of presence to multiple applications and services.
  - Re-unify communications by including contextual communications! If you want to get hold of someone, you don't want to know where they happen to be, you simply want to say "chat to" or "call" and for that to work the user needs to see the communication attempt in the context they're in.
- As a central integration hub for an institution, allowing contextual communication implementations access to recording solutions for example, or access to a gateway for traditional videoconferencing units.

The last aspect explored is room-based videoconferencing. When an institution deploys UC to its end users, it makes a lot of sense to let their meeting room systems expose their users to the same UC experience. Anecdotal evidence from Portugal and Norway supports this case with UiT, the Arctic

University of Norway, providing a clear example. When faced with how to equip *all* rooms (meeting rooms, education rooms) in a new building with distance-bridging technology, they went for a solution based on general-purpose hardware combined with smart camera and microphone choices, running a UC client. The drivers for this decision were cost and flexibility.

Collaboration mandates greater flexibility for users than found in existing tools. Not every user external to an organisation is willing or capable to adapt to its specific toolset. This is already a problem, and WebRTC will make it worse. The number of different distance-bridging solutions a physical room needs to support explodes. Not only will there be an increase in the number of dedicated (web-based) communication solutions people use, contextual communication means that any web application suddenly can have a solid business need to effectively become an audio/videoconferencing solution that needs to be supported by physical rooms, both meeting and lecture rooms. In turn, this means that equipment installed in a physical meeting room needs to be able to support a number of different solutions.

Luckily, all these solutions are likely to be WebRTC-based in not too many years from now. However, physical meeting room equipment typically has a long investment cycle, with rooms waiting up to five years for a total equipment overhaul. To ensure ongoing support for modern distance-bridging services some years from now, the investment stream for distance-bridging technology in rooms needs to be shifted from appliances supporting one solution to generic solutions supporting any (web based) solution. It is important that this shift must happen to equip organisations for users' future needs.

3.7.4 Conclusions

If you look at the world through UC glasses, WebRTC is just another useful feature. It will enable UC vendors to create native web based clients with full functionality. It is easy to use, easy to deploy and any users may be invited by just sending a link.

Looking at the bigger picture, WebRTC has the potential to disrupt the UC market. Users move to the communication tools that best get their jobs done; smooth contextual communication in (collaborative) services trumps talking through a dedicated UC solution. This is an undesirable situation, as you lose the unified aspect of communication. Users no longer have control over where and when they can be reached. One would expect UC vendors to act on this.

Opportunities for UC are in the propagation of presence information to the contextual communication implementations, providing a "single point of entry" for a user's presence information and for being the single integration point with any other type of communication infrastructure or third party value-adding solution, e.g. recording, buddy lists, etc.

UC solutions are often silos, an approach that creates suboptimal results when researchers and educators want to communicate across organisational boundaries. This situation is made worse when considering contextual communication.

To address this, a shared Media API solution for European R&E can be developed, as a meeting point for presence and catalogue (buddy list) information from various UC solutions deployed in the R&E

community, services with contextual communication features and other infrastructures like the R&E authentication infrastructure via eduGAIN.

To ensure the upcoming change in distance-bridging services and solutions also can be used in physical rooms, the investment budget for distance-bridging technology in physical meeting rooms needs to be shifted from appliances dedicated to one solution to generic equipment supporting any solution based on WebRTC. Generic hardware that supports a web browser is a good place to start.

3.7.5 Discussion and Recommendations

Current trends suggest that every institution will move to some form of UC solution for its communications needs. The contextual communication feature unlocked by WebRTC will eat market share from UC. UC solutions typically lack interoperability, leading to problems for collaboration across organisational boundaries, which is typical in R&E.

The R&E community will be unable to standardise on a single UC solution for standardisation is not an option. With WebRTC, however, it is possible to create an umbrella for specific attributes of different UC solutions. Propagation of presence, buddy lists and the ability to interoperate audio and video communication, can all be integrated in, and serviced up, by a Media API. It is also possible to acquire the necessary globally unique userID required to make this work using the eduGAIN infrastructure.

The recommendation for the GÉANT community is:

- Investigate the feasibility of establishing a GÉANT service that may act as a hub for key Unified Communication data, such as presence and buddy lists, from institutions' UC installations. This service may then propagate this information to web services supporting contextual communication. This could be combined with the Media API Service proposal. (see Section 3.5).

The recommendations for R&E institutions are:

- Engage with UC vendors regarding support for contextual communication and WebRTC in general.
- Shift_investment in distance-bridging technology for physical rooms, from appliances supporting a single communication solution to generic hardware supporting many different (including WebRTC-based) software solutions.

A particular recommendation to NRENs is:

- Write a recommendation regarding the shift of investment in distance-bridging technology for physical rooms. Consider publishing a Campus Best Practice on the topic.

3.8 WebRTC2SIP Gateway

3.8.1 Introduction

The Session Initiation Protocol (SIP) is a system of rules that governs the signalling and controlling of multimedia communication sessions. This technology scout investigated how the WebRTC ecosystem may integrate and support legacy SIP equipment and the type of value-adding use cases this could address.

3.8.2 Rationale

There is a large installed base of SIP infrastructure in the R&E community. Effectively every Unified Communication installation adds to this. That makes it very interesting to investigate how WebRTC can be used to add value to this installed base, and how the installed base can be used to facilitate WebRTC adoption.

3.8.3 Summary

The concept and construction of a WebRTC2SIP gateway was explored. Several challenges were identified:

- Transfer protocol
- Codecs
- NAT traversal
- Encryption of traffic

A Proof of Concept was constructed using open source components. It uses Kamailio as the SIP proxy and Sipwise rtpengine as the RTP-proxy. The solution worked and supported the following use cases:

- Easy to integrate call buttons where web page visitors can ring an organisation's internal telephone network by pressing a button in their browser.
- Browser participation window within a classic audio-only conference.
- Build Point-of-Presence systems that can interact both with classical telephony and legacy video systems wherever it is important to connect a browser to telephony or legacy SIP video systems.

3.8.4 Conclusions

A WebRTC to SIP gateway opens the SIP world to browser-based functionality, creating an opportunity for adding value to existing SIP infrastructure services, making them easier to use and facilitate adoption of WebRTC in a more evolutionary way. The proof of concept based on open source components worked well *and is now going to be taken into production in the UNINETT real time communication infrastructure as part of the flat-rate audio conference service*.

3.8.5 Discussion and Recommendations

This technology scout has enabled UNINETT to effectively turn their telephone conference service into a browser based audio conference service that has telephone ring-in capability. This helps especially users participating from countries with a suboptimal technical telephone infrastructure or cost structure. It also makes it trivial to add WebRTC audio call functionality on a web page.

Based on the work done in this technology scout we estimate it should be fully possible to buy this functionality from vendors. It might be an overkill solution at the moment from a price/value point of view, we would expect cheaper gateway appliances to appear in the market.

Although the tech scout focussed on WebRTC-to-SIP we know there to be a significant deployment of H323 equipment in the R&E community which may or may use SIP for signalling. A protocol gateway deployment should also support communication from WebRTC to H323.

We recommend the GÉANT community to:

- Consider adding a WebRTC-to-SIP and WebRTC-to-H323 gateway service to the GÉANT Cloud Catalogue.

We recommend to NRENs and R&E institutions to:

- Gain access to functionality similar to a WebRTC2SIP gateway to link real-time audio communication via the web browser to your existing SIP/telephony solution. This will help you with flexible and cheap audio conferences, click-to-call buttons, etc.

3.9 Janus — General Purpose Gateway

3.9.1 Introduction

WebRTC's peer-to-peer functionality provides many use cases and advantages, but scalability in "one to many" or "many to many" applications is limited and will fail when used by a large number of concurrent users.

Janus, an open source library, presents itself as a "general purpose WebRTC Gateway". It was conceived to be general and not focused on any specific functionality. As such, it provides basic functionality to set up WebRTC connections with its clients/peers and to relay RTP/RTCP messages. Other features and applications are provided by server-side plugins that implement the logic and actions on the streams and extend the API exposed to browser applications.

This technology scout set out to investigate the features and ease of use of the Janus software, particularly in regard to support multi-point videoconferencing. It also studied its support for other applications, such as video streaming, recording and distribution as a video-on-demand service.

3.9.2 Rationale

WebRTC provides peer-to-peer connections and media flow to facilitate real-time communication. However, some R&E applications require “one to many” and “many to many” use cases on a scale larger than what WebRTC is able to deliver out of the box using its direct peer to peer connections.

In order to satisfy these cases, a WebRTC multi-point capable server must be added to the application architecture. This element will allow the application architecture to scale and provide centralised value-adding services.

3.9.3 Summary

A demonstration web application was implemented, for which the Janus server had to be set up and configured. The Janus server, configured and supplied with a streaming plugin, acted as the broadcast server for WebRTC media streams. A standard LAMP stack was put in place to deliver the demonstration application, which was set up to provide a multi-view broadcast for testing.

3.9.4 Conclusion

Janus provides an immense opportunity for competent developers to deploy extra services within an application, as it facilitates filter-based streaming from any source to any destination. Although not stress tested, own observations and the documentation suggest that each WebRTC connection will consume ~500kb of RAM, thus allowing a 2GB machine to connect around 3000 clients. Based on the fact that CPU usage is also rather low (only for process SSL flows), making the server’s network bandwidth the main limiting factor.

Janus provides an efficient and low impact WebRTC multipoint server, suitable for medium-high scale scenarios. It is flexible and may be adapted to suit specific use cases, at the cost of increased complexity.

3.10 Community Engagement with the TF-WebRTC

3.10.1 Introduction

The GÉANT task force on Web-RTC (TF-WebRTC) was established on 1 October 2014, and received a two-year mandate. The task force was designed to be open for participation from NRENs, universities, research institutions, commercial partners, and even knowledgeable individuals contributing to the following objectives:

- To investigate whether and how WebRTC based solutions and services can be used to enable large scale, easy-to-use, easy to integrate and low-cost use of real-time communication across institutional boundaries for all researchers, lecturers, administrative staff and students in European R&E.

- To ensure the European NRENs are well positioned to realise the full potential of WebRTC technology for their community as the technology emerges in the years to come.
- To foster the establishment of an NREN knowledge community as a recognised representative for the European R&E in relevant Web-RTC arenas.
- To build competence, track national developments, collect use cases as well as demonstrate possibilities and identify possible challenges.
- To imagine future WebRTC-based NREN services and infrastructural components that contribute to a smooth evolution of the real time communication infrastructure.
- To liaise with the GN4-1 SA8 WebRTC Task as well as with commercial partners and industry led standardization activities related to WebRTC by providing an open, public forum for gathering and exchanging wider aspects, knowledge and expertise.

The first TF-WebRTC meeting (hosted by RENATER in France on 15 December 2014), was attended by 35 participants and invited speakers representing 13 NRENs and 12 other organisations, and included individuals active in IETF standardisation and WebRTC development, discussed the latest developments in WebRTC and agreed the scope for the new task force.

The task force Chair, Mihály Mészáros (NIIF Institute) explained the rationales behind the task force as follows:

- To organise the community and to prepare for forthcoming project work in order to maximise the benefit of the funding made available.

- To be the public face and open community for discussing development directions of integrated Web-based real-time communication services and application for research and education.

3.10.2 Rationale

As a project this task is a small effort where only a very limited subset of the community could participate. In addition, the task had limited resources to engage in a wider useful dialogue with market and industry. Through the TF-WebRTC we could address both factors.

3.10.3 Summary

To address the challenges ahead, TF-WebRTC worked closely together with the GN4-1 SA8 WebRTC Task and engaged with the largest group of stakeholders possible.

TF-WebRTC organised three meetings (Paris, Budapest and Stockholm) and two other meetings (Berlin and Helsinki) are lined up before end-2016. The first three meetings were attended by 73 people in person, and another 58 people followed the discussion track remotely. All presentations are available online [TF-WebRTC-WIKI].

The task force has been particularity successful in the engagement with commercials and industry. The past meetings featured presentations from the big industry players such as Adobe, Cisco, Ericsson Research, Intel, Mozilla and Oracle and also from the start-ups and SMEs such as Callstats.io, Jitsi/Blue Jimp, Kurento, MashMeTV, Mconf Tecnologia, Pexip and Vidyo.

The extensive discussions helped to forge a better understanding of how the products and the market itself evolve and the active task force participants built technical competence and knowledge on the user preferences and development tracks experienced by the NRENs in different countries.

TF-WebRTC is also meant as a platform for dissemination of and further collaboration on Task 2 results. Therefore, the results will be presented in-depth at the TF-WebRTC meeting on 2-3 May in Berlin.

3.10.4 Conclusions

In conclusion, TF-WebRTC efficiently and effectively brought together the key stakeholders of the community and engaged with both the NREN community and their customers, and the major industry and standardisation players.

The task force helped this task to achieve its goals of open dialogue with market and industry, as well as facilitating an efficient dialogue with the NRENs (and other interested parties) not part of the WebRTC task in the GN4-1 project. In addition, co-location of meetings lowered travel expenses, both in hard currency and by cutting down on time spent travelling.

Task 2 would also like to thank the Hungarian NREN NIIF for enabling and funding Mihály Mészáros to chair the TF-WebRTC as a non-partial community effort.

3.10.5 Discussion and Recommendations

The recommendation to the GÉANT community is:

- To continue with active participation in the TF-WebRTC.
- To continue to co-locate task physical meetings with the TF-WebRTC.

The recommendation to R&E institutions and NRENs is:

- Leverage the TF-WebRTC for building knowledge of, and formulate a position on WebRTC.

4 Recommendations and Conclusions

WebRTC is a technology for native browser-based, real-time communication, but it does not provide a turn-key solution. Much in the same way as TCP/IP allows the building networks without specifying network blueprints, so too does WebRTC allow the building of communication solutions.

WebRTC provides the R&E community:

- A significant improvement of the usability and ease of use of videoconferencing solutions.
- A quantum leap towards contextual communication, with the ability to embed real time communication functionality in any web application.

Based on the work performed in this task, the Task 2 team can formulate two, high-level strategic goals for the European R&E community:

- Adopt WebRTC now to bring easy to use and high-quality desktop videoconferencing functionality at a low cost to every European R&E user.
- Facilitate the adoption of WebRTC for contextual communication.

These strategic goals lead to a number of specific recommendations for collaborative action by the GÉANT community, as well as recommendations for individual NRENs and R&E institutions. These recommendations are detailed below with the paragraph number and title indicating the Task 2 area from which they stem.

4.1 Recommendations for the GÉANT Community

From Section 3.1 ICE, STUN and TURN: How WebRTC Deals with Firewalls and NAT***:***

- Move the STUN/TURN PoC to a two-year pilot service operated by GÉANT. After a two-year period, a decision can be made whether to continue the service as part of the GÉANT service portfolio or not. This gives the European R&E community a timeframe in which to explore using STUN/TURN services against easy conditions, e.g. no invoicing, eduGAIN authentication. It avoids stalling WebRTC adoption with discussions regarding desirability of using cloud services for STUN/TURN.
- Document technical requirements for STUN/TURN services in R&E and make a recommendation based on these on possible inclusion of STUN/TURN cloud services in the GÉANT cloud catalogue. This requires practical tests of at least some of the services.

From Section 3.2 RENATER RENdez-Vous: One year of Operational Experience***:***

- Make the adoption of useful WebRTC services easier by adding them to the GÉANT Cloud Catalogue, with framework agreements where opportune/appropriate.
- In particular, make one or more, easy to use, WebRTC desktop videoconferencing services available to all European R&E users through the GÉANT Clouds Service catalogue, as soon as possible. This creates a new lowest common denominator solution for simple online group conversations.
- Some NRENs prefer a commercial cloud service. Others prefer to build their own or to use a community service. The discussions about take-up and use of cloud services are at different stages in different countries. The Task 2 tech scouts clearly show both options are viable for bringing desktop videoconferencing to the masses. Task 2 recommends to investigate whether the RENdez-Vous service can be added to the GÉANT Cloud Service catalogue.
- RENATER will continue its RENdez-Vous effort. It would be very beneficial for the wider R&E community to learn from the experiences of this large-scale field-test of native WebRTC (and SFU) technology. RENATER should be enabled to provide at least an annual update of their experiences. Cooperate closely with RENATER to leverage their deployed native WebRTC service RENdez-Vous and its community of users.

From Section 3.3 Distributed RENdez-Vous: An investigation into Scale-out Jitsi***:***

- Use RENATER's RENdez-Vous effort to further explore the concepts of scalable SFU-based native WebRTC desktop videoconferencing services and further develop the concepts of a Media API Service, especially integrating UC presence propagation.
- Closely track the market for solutions similar to a distributed SFU solution and strongly consider vendor collaborations for development of a Media API Service concept.

From Section 3.4 WebTut: A Contextual-Communication Proof of Concept***:***

- Develop an online resource with recipes and code examples (cookbook), making it easy for institutions to find their way in the various software components when implementing contextual communication using WebRTC.
- Help NRENs to actively promote the contextual communication concept to R&E institutions and stimulate their members in capability building.

From Section 3.5: Service Proposal for a Media API Service***:***

- Further develop the concept of the Media API Service by developing a Proof of Concept and build a pilot service based on this. A build-or-buy decision would have to be made after the PoC phase.

From Section 3.6 Screencast: Stream and Record Lectures with WebRTC***:***

- Revisit currently immature WebRTC lecture streaming and recording capabilities early 2017.

From Section 3.7: Unified Communication and WebRTC***:***

- Investigate the feasibility of establishing a GÉANT service that may act as a hub for key Unified Communication data, such as presence and buddy lists, from institutions' UC installations. This is one way that this service may then propagate this information to web services supporting contextual communication. This could be combined with the Media API Service proposal. (Section 3.5).

From Section 3.8 WebRTC2SIP Gateway

- Consider adding a WebRTC-to-SIP and WebRTC-to-H323 gateway service to the GÉANT cloud catalogue.

From Section 3.10 Community Engagement with the TF-WebRTC***:***

- Continue with active participation in the TF-WebRTC and continue to co-locate task physical meetings with the TF-WebRTC.

From Section 2 WebRTC Technology*: the data channel:*

- WebRTC supports a data channel. The market is still in the process of out what it can best be used for, and this was one area that the Task 2 team was unable to further investigate. There are a number of interesting use cases indicating potential for R&E use cases, especially with sensors and instruments. Market developments for the WebRTC data channel should be closely tracked in future.

4.2 Recommendations for NRENs and R&E Institutions

From Section 3.1 ICE, STUN and TURN: How WebRTC Deals with Firewalls and NAT***:***

- Ensure institutions understand the purpose of a STUN/TURN service with WebRTC technology. The goal here is to prevent dismissing the technology as immature because connections are blocked by NATs and firewalls. This is an awareness raising activity that not only needs to target technologists developing and deploying (web)applications but also support teams for e-Learning and e-Research.

From Section 3.2 RENATER RENdez-Vous: One year of Operational Experience***:***

- Ensure end-users have access to an easy to use zero-install native WebRTC desktop videoconferencing service, compliant with local policies. Ensure / follow-up this point, regardless of future work priorities.

From Section 3.4 WebTut: A Contextual-Communication Proof of Concept

- Develop technical capability for leveraging WebRTC in contextual communication, including skills for procurement, development and support.
- The Web Tutoring proof of concept looks like an idea with potential. Consider a joint (open source) effort to further develop the Web Tutoring service idea.

From Section 3.5: Service Proposal for a Media API Service**:**

- Establish a collaborative project defining domain-specific high-level building blocks that could be used in contextual communication scenarios.

From Section 3.6 Screencast: Stream and Record Lectures with WebRTC**:**

- Engage with streaming and lecture recording solution providers regarding their plans for future support of the emerging WebRTC recording standards.

From Section 3.7: Unified Communication and WebRTC**:**

- Engage with UC vendors regarding support for contextual communication and WebRTC in general.
- Shift your investment in distance-bridging technology for physical rooms, from appliances supporting a single communication solution to generic hardware supporting many different (including WebRTC-based) software solutions. Consider publishing a Campus Best Practice on this topic.

From Section 3.8: WebRTC2SIP Gateway***:***

- Gain access to functionality similar to a WebRTC2SIP gateway to link real time audio communication via the web browser to your existing SIP/telephony solution. This will help you with flexible and cheap audio conferences, click-to-call buttons, etc.

From Section 3.10: Community Engagement with the TF-WebRTC

- Leverage the TF-WebRTC for building your knowledge of, and position on WebRTC.

From Chapter 2 WebRTC Technology***: the data channel:***

- WebRTC supports a data channel. We did not have a chance to give this a closer look, and the market is also still in the process of out what it can best be used for. We have seen a number of interesting use cases indicating good potential for R&E use cases, especially with sensors and instruments. We recommend to closely track market developments for the WebRTC data channel.

4.3 Closing Words

The transnational nature of the SA8 Task 2 team emphasised the need for RTC solutions to support communication and collaboration between team members. Our dependency on RTC technologies, to assist in our work on RTC technologies, made the Microsoft idiom “to eat your own dog food” (when products developed by an organisation are also used within it), exceptionally fitting. Native WebRTC tools have been consumed in copious amounts while executing the work in Task 2 and writing this report. For the first time in the authors' experience, there was a near- frictionless real-time communication experience, allowing us to fully focus on the work at hand. Not even the latency between Trondheim and Adelaide could stop native WebRTC tools from being a joy to work with.

In summary, WebRTC technology allows for a significant improvement in usability of real-time communication solutions. In addition, it represents a significant leap towards pervasive real- time communication with its potential for contextual communication in the dominant application deployment framework of today; the World Wide Web.

There is no reason to wait with adopting WebRTC.

It's mature enough.

Use it.

References

[ADOBE]	http://blogs.adobe.com/adobeconnect/2015/07/the-adobe-connect-html5-roadmap.html
[appear.in]	https://appear.in/
[BROWSER]	http://iswebrtcreadyyet.com.
[CHRISTENSEN]	http://www.claytonchristensen.com/key-concepts/
[eduGAIN]	http://services.geant.net/edugain/Pages/Home.aspx
[GARTNER]	http://www.gartner.com/newsroom/id/3165317
[GetUserMedia]	http://w3c.github.io/mediacapture-main/
[IBMIoT]	https://www-03.ibm.com/press/us/en/pressrelease/47660.wss)
[IETF]	www.ietf.org
[IETFRTCWEB]	https://tools.ietf.org/html/draft-ietf-rtcweb-overview-15
[JitsiAPI]	https://github.com/jitsi/jitsi-meet/blob/lib-jitsi-meet/doc/API.md
[OpenWebRTC]	http://www.openwebrtc.org/
[ORTC]	http://ortc.org/
[RENDEZ-VOUS]	https://rendez-vous.renater.fr/
[RFC5245]	Protocol for ICE https://tools.ietf.org/html/rfc5245
[RFC5389]	Protocol for STUN http://tools.ietf.org/html/rfc5389
[RFC5766]	Protocol for TURN https://tools.ietf.org/html/rfc5766
[RTCFEATURES]	http://iswebrtcreadyyet.com.
[RTCPeer]	https://w3c.github.io/webrtc-pc/
[RTCData]	https://w3c.github.io/webrtc-pc/
[SCORECARD]	http://iswebrtcreadyyet.com
[SKYPE]	https://blogs.office.com/2015/09/18/enabling-seamless-communication-experiences-for-the-web-with-skype-skype-for-business-and-microsoft-edge/#mmLkguQyVt0dO56l.97
[STARWARS]	https://www.google.com/starwars/
[TNC2015]	https://tnc15.terena.org/
[TNC2016]	https://tnc16.geant.org/
[TF-WebRTC]	https://wiki.geant.org/display/WRTC/GN4-1+WebRTC+Roadmap
[TF-WebRTC-INV]	http://www.geant.org/Innovation/SIG_TF/Pages/TF-WebRTC.aspx
[TF-WebRTC-WIKI]	https://wiki.geant.org/display/WRTC/TF-WebRTC+Task+Force+on+WebRTC
[TWILLO]	www.twillio.com
[TWILLIOBLOG]	https://www.twilio.com/blog/2016/03/chrome-vs-firefox-webrtc-stats-api-with-twilio-video.html
[VIDYO]	http://www.vidyo.com/company/news-and-events/press-releases/vidyo-delivers-webrtc/
[W3C]	https://www.w3.org/

[W3CWGCHARTER]	https://www.w3.org/2015/07/webrtc-charter.html
[WEBKIT]	http://www.webrtcinwebkit.org/
[WEBKITSPEC]	https://webkit.org/status/#specification-webrtc
[WebRTC]	https://webrtc.org/
[WebRTCARCH]	https://webrtc.org/architecture/
[WebRTCDATA]	https://tools.ietf.org/html/draft-ietf-rtcweb-data-channel-13
[WebRTCSPEC]	https://www.w3.org/TR/webrtc/
[WebRTCSTATS]	https://webrtchacks.com/usage-stats/
[WebRTCSTATSAPI]	https://www.w3.org/TR/webrtc-stats/

Glossary

API	Application Program Interface
BoF	Birds of a Feather
CPU	Central Processing Unit
ICE	Interactive Connectivity Establishment
IETF	Internet Engineering Task Force
GN4-1	(GÉANT Network 4, Phase 1) project part-funded from the EC's Horizon 2020 research and innovation programme under Grant Agreement No.691567
MCU	Multipoint Conferencing Unit
MOOC	Massive Open Online Course
NAT	Network Address Translators
NREN	National Research Educational Network
P2P	Peer to Peer
R&E	Research and Education
ROI	Return on Investment
RTC	Real-Time Communications
SA	Service Activity
SDP	Session Description Protocol
SFU	Selective Forwarding Unit
SIP	Session Initiation Protocol
SIG	Special Interest Group
STUN	Session Traversal Utilities for NAT
T	Task
TF	Task Force
TF-WebRTC	Task Force on Web Real-Time Communication
TURN	Traversal Using Relays around NAT
VLE	Virtual Learning Environment
WebRTC	Web Real-Time Communications
W3C	World Wide Web Consortium