

Interaktive Webseiten für effiziente Kooperation auf Basis offener Standards

Klaus Herberth, Daniel Kaiser, Daniel Scharon, Marcel Waldvogel

Fachbereich Informatik und Rechenzentrum
Universität Konstanz
Universitätsstr. 10, 78457 Konstanz, Deutschland
<Vorname>.<Nachname>@uni-konstanz.de

Abstract: Homepages von Forschern, Informationsseiten der Verwaltung, Support- und Beratungsseiten, Webmail oder sonstige Groupware begleiten uns bei der täglichen Arbeit im akademischen Umfeld. Unmittelbare Rückfragen zu den Inhalten oder Interaktionen sind jedoch weiterhin nicht möglich; eine Integration von Direktkontakten wäre häufig angenehm und hilfreich. Swoosch, unsere Javascript-Bibliothek auf Basis der offenen Standards XMPP, HTML5 und WebRTC, bietet hier Abhilfe. Auf einfachste Weise lassen sich so bestehende Webseiten und -anwendungen transparent um Funktionen für Instant Messaging inklusive Videokonferenz erweitern und neue Beratungs- und Kommunikationskanäle eröffnen, bei denen die Privatsphäre durch Ende-zu-Ende-Verschlüsselung gesichert ist. Die Erfahrungen in den Testinstallationen zeigten nahtlose Integration, niedrige Eintrittsschwelle, Geschwindigkeit und Benutzerfreundlichkeit. Durch die inhärente Föderation von XMPP ist der Dienst auch über die Organisationsgrenzen hinaus einsetzbar. Dank seiner offenen Standards ist Swoosch beinahe beliebig integrier- und erweiterbar.

1 Einleitung

Webseiten dienen an Universitäten und Hochschulen als Frontends für diverse Dienste: So ist es für *Studieninteressierte* möglich, sich über das Studienangebot zu informieren oder Tests durchzuführen, die bei der Wahl des Studiengangs helfen; *Studierende* können sich über das Vorlesungsangebot informieren, sich für Vorlesungen bzw. Klausuren registrieren und Ergebnisse einsehen; *Mitarbeiter und Forscher* können Kollaborationstools verwenden, Publikationslisten verwalten und finden Informationen zu Verwaltungsvorgängen. Für *alle* Mitglieder der Institution stehen in der Regel Webfrontends für Dienste wie E-Mail und Filesharing zur Verfügung.

Es ist unmöglich, alle diese Angebote so zu dokumentieren, dass die Nutzer rasch die benötigten Informationen finden, was eine persönliche Beratung oft unumgänglich macht. Aufgrund der hohen Benutzerzahlen und knappen Ressourcen sind gut organisierte Beratung und Support in allen Bereichen der Hochschulen nötig. Dies gilt sowohl für technische Fragen, z.B. wenn ein Nutzer Probleme bei der Benutzung eines Dienstes hat, als auch für Beratung, z.B. wenn sich Studieninteressierte bei der Wahl des Studiengangs beraten lassen möchten. Für den Fall des Supports zu einem Dienst findet sich meist eine E-Mail-Adresse, an die die Frage gestellt werden soll. Auf eine Antwort muss der Hilfesuchende

oft länger warten. Wenn es Rückfragen gibt, kommt es jeweils zu einer weiteren Iteration dieses Prozesses, die zu weiteren Wartezeiten führt. Deshalb wird auch häufig zum Telefon gegriffen, was einen Medienbruch und eine Unterbrechung der aktuellen Tätigkeit nach sich zieht.

Für die Studienberatung kommt neben E-Mail zum Teil auch Videotelefonie zum Einsatz. Dabei werden proprietäre Tools wie Skype genutzt, die den User zwingen, Client-Software zu installieren, ein Benutzerkonto einzurichten und die entsprechende Beratungsstelle als Kontakt hinzuzufügen. Ganz abgesehen von den offenen Fragen zur Privatsphäre [Bee13] ist dieser Medienbruch umständlich für den Benutzer und führt zu einer weiteren Fehlerquelle, um die sich der Support kümmern muss.

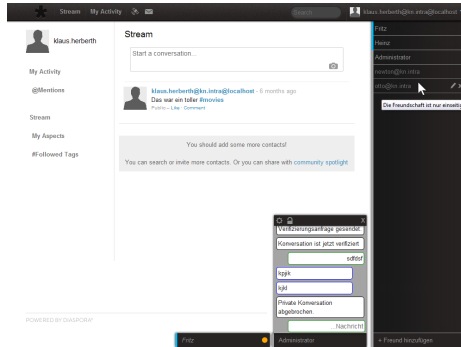
Die meisten, auch angehenden, Akademiker sind reine Webnutzer. Obwohl native Mailclients komfortabler zu bedienen und einfach einzurichten sind, zeigt die intensive Nutzung auch durch regelmäßige Anwender, dass der spielend leichte Zugang wie bei Webmail ein wichtiges Dienstekriterium geworden ist.

Unser Ziel war deshalb, direktere Kommunikation ohne zusätzliche Software in den Browser zu integrieren. Swoosch, unser Framework, macht es möglich eine multitabfähige Chat-Applikation, die auf XMPP [SA11] basiert, Videokonferenzunterstützung bietet und auf OTR [Cyp12] basierende Ende-zu-Ende Verschlüsselung unterstützt, mit sehr geringem Aufwand transparent in bestehende Seiten und Webanwendungen zu integrieren. Das Chatfenster integriert sich nahtlos in bestehende Seiten und kann auf jedem Webfrontend der Institution eingeblendet werden. Falls ein Benutzer sich auf einer Seite befindet, auf der Support oder Beratung angeboten wird, blendet Swoosch automatisch die entsprechenden Links in einer dafür vorgesehenen Kategorie der Kontaktliste ein. Dies ist sowohl für Hilfe suchende Mitglieder der Institution als auch für Gäste eine intuitive Möglichkeit, sich ohne Aufwand an den Support zu wenden.

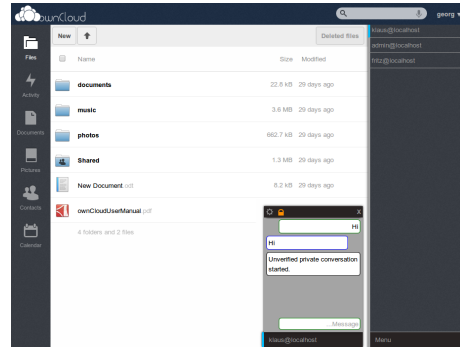
Im Gegensatz zu existierenden Support-Chat-Lösungen, die lediglich synchrone Kommunikation erlauben, ermöglicht es Swoosch dank XMPP Nachrichten bei Abwesenheit auch asynchron zu behandeln, was die Vorteile von Chat-Support und Mail-Support vereint. Gleichzeitig nutzt Swoosch mit XMPP auch die bestehende Infrastruktur und ermöglicht damit den Nutzern den Einstieg in eine plattformübergreifende, föderierte Instant-Messaging-Infrastruktur.

Die Ende-zu-Ende-Verschlüsselung bietet dabei ein hohes Maß an Sicherheit, welches der Benutzer weder mit Bequemlichkeit noch Geschwindigkeit bezahlen muss. Es gibt Formen der Beratung, bei denen der Schutz der Privatsphäre essentiell ist, beispielsweise bei Studien- oder Lebensproblemen. In diesem Anwendungsfall erlaubt es Swoosch Hilfesuchenden, die nicht persönlich erscheinen möchten, die Dienste der Beratung in Anspruch zu nehmen und dabei eine einem Zwiegespräch ähnliche Privatsphäre zu genießen.

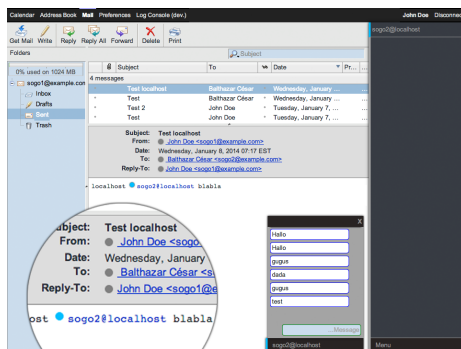
Swoosch ist nicht auf den Einsatz als Beratungstool bzw. als Werkzeug zur Kommunikation mit Mitgliedern der selben Institution beschränkt. Es ist aufgrund von XMPP, früher unter dem Namen *Jabber* bekannt, auch möglich, mit beliebigen XMPP-Nutzern zu kommunizieren. Durch seine breite Unterstützung u.a. bei vielen Freemail-Providern, automatische Föderation und inhärente Erweiterbarkeit ist zu erwarten, dass XMPP in Zukunft noch ein breiteres Feld abdecken wird. Für den Benutzer ist interessant, dass ein Kontakt



(a) Soziale Netze (Diaspora)



(b) Dokumentensharing (ownCloud)



(c) Groupware (SOGO)



(d) WebCMS (Typo3)

Abbildung 1: Einfache Integration in bestehende Systeme

unter der Mailadresse auch für Instant Messaging, Audio- und Videokommunikation zur Verfügung steht.

2 Anwendungsszenarien

2.1 Beratung

Um Beratung in Anspruch zu nehmen, kann auf der jeweiligen Webseite der Beratungsstelle direkt die eingebettete Chatfunktion aktiviert werden. Ob sich die Kommunikation auf reinen Textchat beschränkt oder bspw. Audio- und Videoverbindungen zusätzlich zugelassen werden, kann hierbei dem Wunsch der Hilfesuchenden oder den Voreinstellungen der jeweiligen Webseite überlassen werden.

Falls der Hilfesuchende Mitglied der Institution ist, kann über eine Loginfunktion des Web-CMS die Zuordnung zu einem Konto vorgenommen werden, was es im Fall der Stu-

dierendenberatung einem Sachbearbeiter ermöglicht, die passende Akte einzusehen ohne nach der Matrikelnummer fragen zu müssen. Doch auch für Personen, welche nicht Mitglied der Institution sind oder anonyme Beratung wünschen, kann die Beratung via Swoosch in Anspruch genommen werden, wenn dies die Seite entsprechend vorsieht.

Auf der Seite der Beratenden ist eine Gruppe von Personen angebunden, welche die jeweiligen Hilfesuchenden übernehmen kann; dies ist auch ohne dedizierten Funktionsaccount möglich. Gerade für den IT-Support erleichtert die integrierte Funktion des Desktopsharings die Beratung; hierbei kann die Beraterin dem Hilfesuchenden anschaulich einen Lösungsweg darlegen oder umgekehrt ein Hilfesuchender ein auftretendes Problem vorzeigen.

Der von Swoosch verfolgte Ansatz ist hierbei so flexibel, dass sowohl auf beratender wie auch auf hilfesuchender Seite je nach Nutzerwunsch ein kompatibler Webbrowser oder geeignete Clientprogramme verwendet werden können. Aufgrund der durchgehenden Ende-zu-Ende-Verschlüsselung der Kommunikation ist bei allen Szenarien gewährleistet, dass personenbezogene Daten nur durch die Beratenden selbst eingesehen werden können.

2.2 Zuständigkeitsseiten

Auf vielen akademischen Webseiten sind Kontaktdaten eines Beraters aufgeführt, weil dieser für ein bestimmtes Sachgebiet oder einen Namensraum zuständig ist. Swoosch bietet hier die Möglichkeit, in der jeweiligen Zuständigkeitsseite eine erweiterte Chatfunktion einzubinden, welche direkt zum aufgeführten Berater führt. Auch hier ist persönlicher Login oder anonymen Zugriff wählbar. Die Verfügbarkeit des Beraters kann durch Swoosch direkt beim Aufruf der Webseite angezeigt werden.

2.3 Groupware

Swoosch ist natürlich nicht auf Beratungsszenarien beschränkt. Genauso ist z.B. eine allgemeine Integration in eine bestehende Groupwarelösung denkbar. So bietet SOGo [SOG] Webmail, Calendaring und Kontakte in einer einzigen Weboberfläche sowie deren Synchronisation mit vielen Geräten über offene Standards. Mit der Integration von Chat in dieselbe Oberfläche ist die bestehende Groupware um eine sinnvolle Komponente ergänzt. Für Endnutzer bietet sich somit eine Sicht an, die sie bereits von großen, kommerziellen Anbietern wie Gmail oder Yahoo gewohnt sind. Auch hier wird die Kontaktliste der Chatpartner zentral, d.h. in unserem Anwendungsfall bei der akademischen Institution vorgehalten. Zudem ist dank der Föderierung des zugrundeliegenden XMPP-Protokolls der Benutzer bei der Auswahl seiner Chatpartner nicht auf die Mitglieder seiner Institution beschränkt, sondern kann mit sämtlichen Personen kommunizieren, deren Organisation einen XMPP-Server bereithält, welcher für die Verbindung mit anderen XMPP-Servern geöffnet ist. Dazu gehören nicht nur große Freemailanbieter wie GMX, Web.de und Gmail, sondern bereits eine Reihe von akademischen Einrichtungen [JUN].

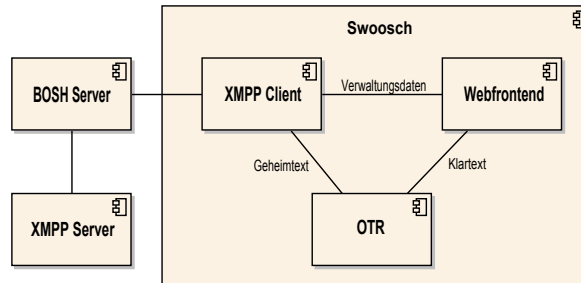


Abbildung 2: Swoosch-Komponenten

3 Funktionsweise

3.1 Allgemeines

Swoosch ist als eine JavaScript-Anwendung konzipiert, welche ausschließlich im Browser ausgeführt wird. Damit ist sie unabhängig von der eigentlichen Serveranwendung und kann in jede beliebige Webanwendung integriert werden.

Beim Laden einer Swoosch-unterstützten Seite kontrolliert die Anwendung zuerst, ob Swoosch bereits in einem anderen Tab aktiv ist und damit ressourcensparende Arbeitsteilung möglich ist. Danach wird überprüft, ob eine BOSH-Verbindung [PSSAM10] wiederverwendet werden kann, oder ob eine neue aufgebaut werden muss. Wenn eine Verbindung steht, werden eingehende Nachrichten dargestellt und ausgehende ggf. verschlüsselt und versendet (Abbildung 2).

Alle Informationen, die über die Darstellungsdauer einer einzelnen Webseite hinaus verfügbar sein müssen, werden im Web Storage [Hic13] des Browsers gespeichert. Dazu zählen die letzten 10 Nachrichten pro Kontakt, der private Schlüssel und Informationen über den Status des OTR-Protokolls.

HTML5 bringt neben Web Storage aber noch eine Vielzahl weiterer Möglichkeiten mit. Genutzt wird insbesondere die *Notification API* [GvK13], durch welche Benutzer über eingehende Nachrichten informiert werden können, selbst wenn der Browser-Tab nicht im Vordergrund ist. Dazu wird ein kleines Fenster eingeblendet, wie man es unter anderem von seinem Email-Client kennt.

Um eine neue Verbindung aufzubauen, wird die Jabber-ID und das zugehörige Passwort benötigt. Diese Daten kann Swoosch direkt vom Benutzer erfragen oder – falls die Webseite dies vorsieht – deren Anmeldeinformation mitnutzen (Abbildung 3).

Bei jedem weiteren Laden einer Swoosch-Seite werden diese Anmeldeinformationen zur Fortsetzung der Kommunikation mit dem XMPP-Server genutzt und der Zustand der Benutzeroberfläche wiederhergestellt. Für den Benutzer scheint das Chatfenster damit vom Browsen nicht betroffen zu sein, solange auch die neue Seite Swoosch unterstützt. Details zur Implementierung befinden sich in [Her13].

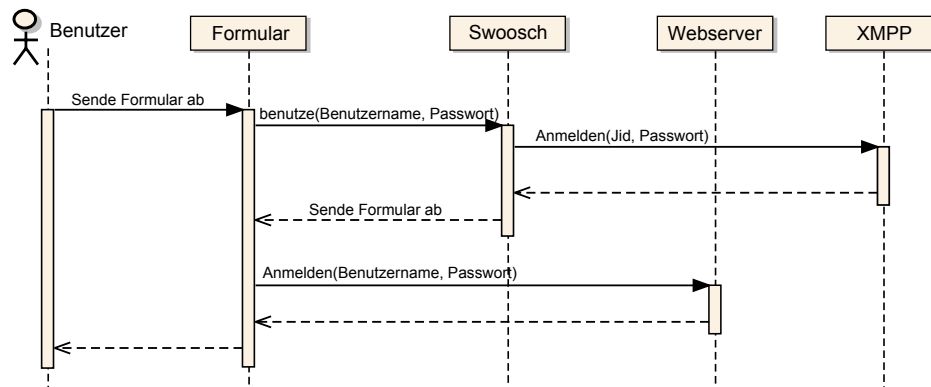


Abbildung 3: Doppelnutzung des Logins zwischen Swoosch und der eigentlichen Anwendung. Die Anmeldedaten werden zuerst für die Anmeldung beim XMPP-Server genutzt. Erst danach geschieht die Anmeldung am eigentlichen Webserver.

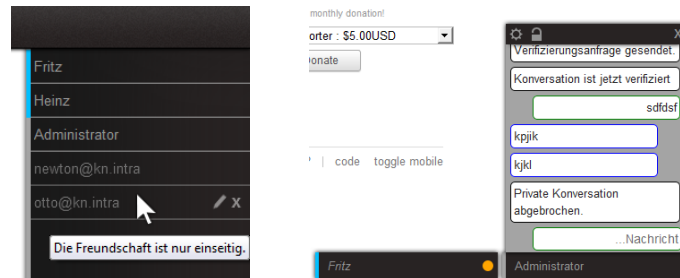
3.2 Privatsphäre mit OTR

Bei Beratungen aber auch bei vielen anderen Anfragen ist Vertraulichkeit unabdingbar. Dazu wird das OTR-Protokoll [Cyp12] eingesetzt. Seine technischen Eigenschaften wurden privaten Diskussionen nachempfunden, d.h. Dritte dürfen nicht mithören und die Abstreitbarkeit gegenüber Dritten bei gleichzeitiger Sicherstellung, dass es sich beim Gegenüber um die gewünschte Person handelt, muss gewährleistet sein. Durch die Nutzung temporärer Schlüssel sind früher versandte Nachrichten auch bei einem späteren Leck des privaten Schlüssels nicht entzifferbar (*forward secrecy*). Diese Eigenschaften sind einer der Gründe, warum bei Chatanwendungen OTR gegenüber PGP oder S/MIME bevorzugt wird [BGB04].

Erhält der Benutzer bei Swoosch eine Anfrage für eine private Unterhaltung oder möchte selbst eine initiieren, wird diese automatisch gestartet und alle Parameter zur Wiederherstellung des aktuellen Zustands werden gespeichert. Dazu gehören unter anderem der aktuelle und frühere Schlüssel, die *secure session ID* sowie die Positionen mehrerer Zustandsautomaten. Dadurch kann dem Nutzer ermöglicht werden, den Chat unabhängig von unterliegenden Seitenaktivitäten zu verwenden.

3.3 Mehrere Tabs

Jede clientseitige Webanwendung muss berücksichtigen, dass der Benutzer die gleiche Seite in mehreren Tabs geöffnet hat und entsprechend darauf reagieren. Die herkömmliche Vorgehensweise war bisher den Server als Vermittler einzusetzen um die Tabs synchron zu halten. Bei einer Anwendung, die wie Swoosch eine Ende-zu-Ende-Verschlüsselung bereitstellt, ist dieses Vorgehen aber unangebracht. Daher findet die interne Kommunikation



(a) Beim Überfahren werden die entsprechenden Optionen sichtbar

(b) Ungelesene Nachrichten werden mit einem orangenen Punkt gekennzeichnet

Abbildung 4: Detailansicht des Rosters und der Chat-Fenster

komplett im Browser statt. Glücklicherweise unterstützt *Web Storage* nicht nur Persistenz, sondern löst optional in jeder Tab-Instanz einen Event aus, wenn ein Wert geändert wird. Damit ist für Seiten der gleichen Domain Kommunikation zwischen den Tabs möglich.

Mit dieser Kommunikation kann nun bei der Existenz mehrerer Tabs ein Master-Tab bestimmt werden, der als einziger die BOSH-Kommunikation mit dem XMPP-Server aufrecht erhält. Dies schont nicht nur die Ressourcen von Browser und Server, sondern erlaubt es auch, dem Nutzer eine konsistente Sicht auf den Chatverlauf anzubieten.

Die Existenz genau eines Masters wird durch ein Keepalive-Signal des Masters sichergestellt, welches alle Slaves beruhigt. Falls der Master-Tab geschlossen wird, bleibt das Signal aus und ein neuer Master wird erkoren. Dabei stellt die Nutzung eines einzigen Schlüssels im Webstorage, der für Keepalive und die verschiedenen Nachrichten im Wahlprozess zuständig ist, sicher, dass nur ein Master-Tab existiert. Details dazu finden sich in [Her13].

3.4 Videochat

HTML5 bietet mit WebRTC [BBJN13] die Möglichkeit einer direkten verschlüsselten Kommunikation zwischen Browsern inklusive NAT-Traversierung mit STUN und TURN. Swoosch verwendet dies zur Übermittlung von Bild- und Tonsignalen für Videochats. Durch die XMPP-basierte Signalisierung mittels *Jingle* [LBSA⁺09] ist auch ein Videoanruf mit nativen Clients wie *Jitsi* möglich.

4 Installation und Nutzung

Die Installation geschieht durch Einbindung der entsprechenden JavaScript- und CSS-Datei in das bestehende Layout. Danach ist die Installation abgeschlossen und Swoosch



Abbildung 5: Durch verschiedene Symbole und Farben werden die unterschiedlichen Verbindungsstadien visualisiert.

muss nur noch konfiguriert werden. Dazu zählen Angaben für Dual Login, das Layout und den BOSH-Server. Die Benutzungs- und Gestaltprinzipien orientieren sich an XMPP-Desktopclients, was eine intuitive Bedienung sicherstellt (Abbildungen 1, 4 und 5).

5 Resultate

Da heutzutage mobile Browser fast genau so weit verbreitet sind wie ihre Desktop-Pendants, wurde nicht nur Chrome 28, Firefox 23, Safari 5 (die neueste Version unter Windows) und der Internet Explorer 10 untersucht, sondern auch zwei weit verbreitete Android-Browser, Dolphin und Opera Mobile. Die Desktoptests liefen unter Windows 7 mit einem Intel Core i5-2500K in einem MSI Z68A-GD65-Motherboard bestückt mit 16GB DDR3-Arbeitsspeicher (1600MHz), die mobilen Tests auf einem Huawei Ascend Y200 unter Android 2.3.6 auf einem mit 800 MHz getakteten ARMv7.

5.1 Funktionsumfang

Um den Funktionsumfang der einzelnen Browser zu überprüfen wurde eine Liste von Funktionen in jedem Browser getestet. Im Einzelnen wurde kontrolliert ob die Anmeldung am XMPP-Server funktioniert, der Roster lädt und man Kontakte umbenennen kann. Weitere Punkte waren das Öffnen eines Chat-Fensters, die Generierung eines neuen Schlüssels und Versand und Empfang von verschlüsselten sowie unverschlüsselten Nachrichten. Die Tauglichkeit bei mehreren geöffneten Tabs und die Authentifizierung mittels SMP [Yao82] wurden geprüft; weiter floss die Bedienbarkeit, Größe und Positionierung aller Elemente ein, was vor allem bei Mobilgeräten relevant ist.

Detaillierte Ergebnisse finden Sie in [Her13]. Zusammenfassend lässt sich sagen, dass alle getesteten Desktopvarianten wie gewünscht funktionieren, mit Ausnahme des Internet Explorer, der Storageevents nicht standardgemäß implementiert und deshalb mehrere Tabs nur eingeschränkt unterstützt.

Die Unterstützung für mobile Browser ist unzureichend, was jedoch kein Argument gegen Swoosh ist, da auf kleinen Displays ein zusätzlicher Chat-Bereich im Browser stören würde. Daher wurde Swoosh bewusst für Geräte mit größerem Bildschirm optimiert. Auf mobilen Geräten ist die Nutzung von nativen XMPP-Clients eine Ausweichmöglichkeit.

	Desktop				Mobil	
	Chrome	Firefox	Safari	IE	Dolphin	Opera
Minimum	2,4	0,8	7,6	1,4	123	86
Mittelwert	4,1	2,4	9,0	2,9	146	150
Maximum	6,5	7,7	89,5	3,9	189	219

Tabelle 1: Wartezeit für Schlüsselerzeugung [s] ($N = 5$). Der Mittelwert wurde ohne die beiden Extremwerte bestimmt.

5.2 Geschwindigkeit

Um die Geschwindigkeit aller Browser zu testen eignet sich die Laufzeit der komplexen Berechnung des DSA-Schlüssels am Besten. Aus diesem Grund wurde in allen Umgebungen fünfmal ein DSA-Schlüssel berechnet und die benötigte Zeit gestoppt (Tabelle 1). Neben Multitasking ist ein wichtiger Grund für die unterschiedlichen Laufzeiten innerhalb eines Browsers auch das ‘Würfelglück’ beim Erzeugen der Primzahlen.

In erster Linie hängt die Dauer von der Rechenleistung des entsprechenden Prozessors ab, da die des Mobiltelefons um einiges schwächer ist als die des PCs, sind die Zeiten auch um ein Vielfaches höher. An den Laufzeiten erkennt man auch, dass die Entwicklung an den JS-Engines schnell voran geschritten ist. So ist die Rechenleistung des älteren Safari 5 weniger als halb so hoch wie die des nächstbesseren Chrome (Safari 6 und 7 wurden bisher nicht auf Windows portiert).

Die Wartezeit bei der Schlüsselgenerierung ist nur beim allerersten Login mit dem entsprechenden Browser spürbar. Bei späteren Logins steht der Chat sofort zur Verfügung.

6 Schlussfolgerungen und nächste Schritte

Der Einbau von Kommunikationsmöglichkeiten in bestehende Webseiten ist der natürliche Schritt zur Verbesserung des Servicegedankens an Hochschulen, eignet sich aber auch hervorragend für die Ergänzung existierender Kollaborationsplattformen. Die Nutzung von JavaScript ermöglicht nicht nur den nahtlosen Einbau von zusätzlichen sicheren Kommunikationsmitteln, es erlaubt auch die automatische Ergänzung der Webseite um zusätzliche Interaktionsmechanismen. Die Erkennung von Mailadressen und Ergänzung um einen farbigen Punkt (Abbildung 1(c)), der den Präsenzstatus der Person angibt und als Aktionsanker für den Verbindungsaufbau genutzt werden kann, ist nur ein erster Schritt. Weitere Möglichkeiten in der Webseite wie auch automatische XMPP-Bots sind nur durch die Fantasie limitiert.

Dank der offenen Architektur von XMPP sind auch Konferenzen (Text, Audio, Video, Screensharing) möglich. Diese Funktion wird zur Zeit in Swoosch integriert, ebenso wie ein Workaround für die *Web Storage*-Inkompatibilität des Internet Explorer. Bis zum finalen Publikationstermin wird auch die Dokumentation so weit sein, dass weitere Nutzer Swoosch einfach integrieren können.

Durch die Webintegration werden Hürden abgebaut, die es XMPP erlauben sein Potenzial zu entfalten. Trotz weiter Verfügbarkeit fristet es noch ein Nischendasein, da die Nutzer den Komfort mehr schätzen als die Privatsphäre. Hier kann Swoosh eine Alternative bieten, die auf dem Desktop beides uneingeschränkt bietet. Da zu erwarten ist, dass XMPP im Trend zu integrierter Kommunikation eine wichtige Rolle spielen wird, ist damit gleichzeitig auch der Grundstein für eine zukunftsweisende Strategie gelegt.

Literatur

- [BBJN13] Adam Bergkvist, Daniel C. Burnett, Cullen Jennings und Anant Narayanan. WebRTC 1.0: Real-time Communication Between Browsers. Working draft, W3C, September 2013. <http://www.w3.org/TR/webrtc/>.
- [Bee13] Kristina Beer. Vorsicht beim Skypen – Microsoft liest mit, 2013. <http://heise.de/-1857620>.
- [BGB04] Nikita Borisov, Ian Goldberg und Eric Brewer. Off-the-record communication, or, why not to use PGP. In *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*, WPES '04, Seiten 77–84, New York, NY, USA, 2004. ACM. <http://doi.acm.org/10.1145/1029179.1029200>.
- [Cyp12] Off-the-Record Messaging Protocol version 3. Protocol definition, Cypherpunks, 2012. <https://otr.cypherpunks.ca/Protocol-v3-4.0.0.html>.
- [GvK13] John Gregg und Anne van Kesteren. Web Notifications. Working draft, W3C, September 2013. <http://www.w3.org/TR/notifications/>.
- [Her13] Klaus Herberth. Implementierung & Evaluation eines JavaScript-basierten, OTR-fähigen XMPP Clients. Bachelorarbeit, Universität Konstanz, 2013. <http://nbn-resolving.de/urn:nbn:de:bsz:352-257125>.
- [Hic13] Ian Hickson. Web Storage. Recommendation, W3C, Juli 2013. <http://www.w3.org/TR/webstorage/>.
- [JUN] JUNe: Jabber University Network. <http://j-u-n-e.org/>.
- [LBSA⁺09] Scott Ludwig, Joe Beda, Peter Saint-Andre, Robert McQueen, Sean Egan und Joe Hildebrand. Jingle. XEP-0166, XMPP Standards Foundation, Dezember 2009. <http://xmpp.org/extensions/xep-0166.html>.
- [PSSAM10] Ian Paterson, Dave Smith, Peter Saint-Andre und Jack Moffitt. Bidirectional-streams Over Synchronous HTTP (BOSH). XEP-0124, XMPP Standards Foundation, Juli 2010. <http://xmpp.org/extensions/xep-0124.html>.
- [SA11] Peter Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. RFC 6121, Internet Engineering Task Force (IETF), März 2011. <http://tools.ietf.org/html/rfc6120>.
- [SOG] SOGo: Open Source Groupware. <http://sogo.nu/>.
- [Yao82] Andrew C. Yao. Protocols for Secure Computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, Seiten 160–164, Washington, DC, USA, 1982. IEEE Computer Society.