universität
wien

MASTERARBEIT / MASTER’S THESIS

Titel der Masterarbeit / Title of the Master’s Thesis

„Learning by Viewing”

Implementing a DMARC aggregate report analysis tool for and with email experts

verfasst von / submitted by

Lukas Pühringer, BSc.

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Diplom-Ingenieur (DI)

Wien, 2018 / Vienna 2018

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 066 935
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Medienformatik
Betreuet von / Supervisor:	Univ.-Prof. Dipl.-Math. Dr. Peter Reichl, Privatdoz.

Contents

Acronyms

ACOnet Austrian Academic Computer Network. 3, 46–48, 85

ADSP Author Domain Signing Practices. 1, 2, 15, 16, 43, 44, 84

API Application Programming Interface. 46

ARC Authenticated Received Chain. 20

CD Contextual Design. 20, 30

CERT Computer Emergency Response Team. 36

COSY Cooperative Systems. 3, 36

CP Concurrent Probing. 33

CSS Cascading Style Sheets. 69

CSV Comma Separated Values. 63

CTA Concurrent Think Aloud. 33

D3.js Data-driven Documents. 69

DANE DNS-based Authentication of Named Entities. 11

DKIM DomainKeys Identified Mail. 1, 2, 5, 7, 8, 13–16, 18, 19, 43, 63, 65, 73, 84–87, 116, 117, 119

DMARC Domain-based Message Authentication, Reporting and Conformance. V, VII, 2–8, 10, 12–20, 24, 25, 27, 31, 34–41, 43–48, 50–56, 59–73, 83–85, 87–101, 111–117, 119, 120

DNS Domain Name System. 1, 2, 7, 8, 10–19, 37, 38, 43, 44, 48, 88, 116

DNSSEC Domain Name System Security Extensions. 11

DOM Document Object Model. 69

FAQ Frequently Asked Questions. 88

GUI Graphical User Interface. 46, 47

HCD Human-centered Design. 22

HCI Human-computer Interaction. 20, 22

HSL Hue, Saturation, Luminance. 62

HTML Hypertext Markup Language. 68, 69

ICANN Internet Corporation for Assigned Names and Numbers. 10

IMAP Internet Message Access Protocol. 8

IP Internet Protocol. 1, 10, 11, 19, 44, 63, 86, 87, 115

IxD Interaction Design. 20

JS JavaScript. 68, 69

M3AAWG Messaging, Malware and Mobile Anti-Abuse Working Group. 70
MTV Model Template Controller. 68
MUA Mail User Agent. 8
NPM Node Package Manager. 69
PD Participatory Design. 20
PDF Portable Document Format. 62, 115
PGP Pretty Good Privacy. 7
PNG Portable Network Graphics. 115
POP3 Post Office Protocol v3. 8
RFC Request For Comments. 2, 8, 10, 11, 13–19
RP Retrospective Probing. 33
RTA Retrospective Think Aloud. 33
RUA Reporting URI for Aggregate Data. 17–19, 38, 44, 48, 88
RUF Reporting URI for Failure Data. 16, 18, 19
Sass Syntactically Awesome Style Sheets. 69
SMTP Simple Mail Transfer Protocol. 1, 7–11, 16
SPF Sender Policy Framework. 1, 2, 5, 7, 8, 11–13, 15, 16, 18, 19, 43, 63, 65, 73, 84–87, 116, 119
STS Sciene and Technology Studies. 72
SUS System Usability Scale. 33, 36, 54, 56, 57, 118
SVG Scalable Vector Graphics. 69
TCP Transmission Control Protocol. 10
TLD Top-level domain. 10
TU Technische Universität. 37
UCD User-centered Design. 3–7, 20, 22–27, 29, 31–33, 35, 36, 39, 48, 49, 52, 53, 59, 67, 71–73
UI User Interface. 20, 25, 30, 31, 34, 38, 51, 52, 54, 57, 59, 67, 69, 115, 119
URI Uniform Resource Identifier. 88
UX User Experience. 20, 25, 33, 34
W3C World Wide Web Consortium. 34
WCAG Web Content Accessibility Guidelines. 34
XML Extensible Markup Language. 2, 19, 87
ZID Zentraler Informatikdienst. 3, 36

Abstract

Phishing is a common term used to describe unsolicited email messages that are sent with the objective of stealing confidential information from the email receiver. By taking advantage of the lack of adequate security mechanisms in the core email protocols, phishers can impersonate email senders that are known and trusted to the email receiver, to make attacks even more effective.

Domain-based Message Authentication, Reporting and Conformance (DMARC) is a new sender authentication technology that has emerged and proliferated in recent years to help combat email sender impersonation. In addition to authenticating email sender domains, DMARC standardizes the report exchange about authentication results between email providers, yielding unprecedented insights into email domain use and abuse.

This Master thesis presents *DMARC viewer*, a fully functional open source DMARC report analysis software, which, in contrast to existing tools, provides complete report data sovereignty and a multitude of related DMARC learning aids to its users. An extensive set of user requirements for the software is assessed by studying the underlying technologies and, more importantly, by including relevant industry experts in the participatory development process.

A broad review of suitable user study methods, the design and execution of field studies and evaluations of prototypes of incremental fidelity, as well as the interpretation of the collected data and their significance for the design of the target system and for user-centered software development in general constitute a secondary contribution of this work.

An important insight from the user studies is that mail administrators will ultimately require more experience with DMARC to assess the full extent of its use and usefulness. The analysis of DMARC reports using the presented software aptly serves that purpose. This will be shown by the results from the corresponding user studies, coining the title of the thesis, *“learning by viewing”*.

Zusammenfassung

Der Begriff *Phishing* beschreibt das Versenden von unerwünschten E-Mail-Nachrichten, mit dem Ziel, Adressatinnen vertrauliche Informationen zu entlocken. Um die Effizienz von Phishingattacken zu steigern, werden häufig Absenderinnenadressen gefälscht, die den Adressatinnen unverfänglich erscheinen. Der Adressenschwindel wird durch einen Mangel an relevanten Sicherheitsmaßnahmen in den zentralen E-Mail-Übertragungsprotokollen ermöglicht. Mit Domain-based Message Authentication, Reporting and Conformance (DMARC) wurde in den vergangenen Jahren eine neuartige Authentifizierungstechnologie zur Bekämpfung von E-Mail-Betrug etabliert. Darüber hinaus standardisiert DMARC den Austausch von Berichten über die Authentifizierungsergebnisse zwischen E-Mail-Providern. Mithilfe geeigneter Analysesoftware lassen sich aus diesen Berichten noch nie dagewesene Einblicke in den Gebrauch und Missbrauch von E-Mail gewinnen.

Im Rahmen der vorliegenden Arbeit wird *DMARC viewer* entwickelt, eine voll funktionsfähige open source Software zur Analyse von DMARC-Berichten, die, im Gegensatz zu bestehenden Programmen, den Benutzerinnen exklusive Datensouveränität gewährt. *DMARC viewer* wird in einem partizipativen, mehrphasigen Softwareentwicklungsprozess für und mit Industrieexpertinnen im Bereich der Mailadministration entwickelt. Dafür werden in einem ersten Schritt relevante Methoden und Methodologien identifiziert, um eine interviewbasierte Feldstudie durchzuführen. Auf Basis der gewonnen Erkenntnisse werden Prototypen erarbeitet und gemeinsam mit den Expertinnen evaluiert und überarbeitet. Die Dokumentation der iterativen und inkrementellen Entwicklung der Zielsoftware sowie eine Beurteilung der verwendeten Methoden stellen eine Hilfestellung für zukünftige Softwareentwicklungsprojekte dar, die sich partizipativer Prozesse bedienen möchten.

Eine für die Entwicklung der Zielsoftware entscheidende Erkenntnis dieser Arbeit ist, dass Nutzerinnen zur Bewertung der Nützlichkeit und Verwendung von DMARC mehr Wissen und einschlägige Erfahrung benötigen. Um DMARC zu verstehen, erweist sich die Aufbereitung von DMARC-Berichten durch adäquate Software als besonders geeignet. Darum stellt *DMARC viewer* umfangreiche Mittel zum Erlernen der zugrunde liegenden Technologien und Terminologien zur Verfügung. Dieser Zusammenhang prägt den Titel der vorliegenden Arbeit: „learning by viewing". [1]

[1] Die vorliegende Zusammenfassung in deutscher Sprache bedient sich des generischen Femininums.

Acknowledgments

Special thanks go to my supervisor Professor Reichl and the rest of the current and former Cosy team, including Christian, Hannes, Ivan, Patrick and Svenja, as well as my mail admins. Thank you for sharing your expertise with me!

Furthermore, I am grateful beyond words to my mentor and friend, Albert, my fellow student and partner in crime, Christoph, my tireless review heroes, Dan, Jeremy and Lois, my wonderful roommates, Felix and Matthias, and Harry and Michi, and my ever-supportive family, above all, Johanna.

Thank you for everything!

1 Introduction

Ever since the infancy of communication networks electronic mail has been one of the most important and most utilized applications. Some of the reasons for its popularity are that it is inexpensive, federated, asynchronous, and its messages can usually be delivered instantaneously [34, pp. 147f]. Moreover, despite the proliferation of social platforms and other messaging services today, email remains the number one communications channel for many businesses, and its use keeps growing [68].

But, as with many other Internet applications, email was not designed with today's vast anonymous and untrusted network in mind. The Simple Mail Transfer Protocol (SMTP), which has been at the heart of email transfer since the early 1980s and still is, has no intrinsic means of verifying whether an entity claiming to be the author of an email really is its author [27]. This makes it easy to forge identities, an activity that is also called *spoofing* and is often used in combination with *spamming* or *phishing* attacks to make them more effective. The *Spamhaus Project*, an international non-profit organization that tracks spam and related cyber threats, defines spam as unsolicited bulk mail [50]. Phishing, on the other hand, is a special kind of spam, where the attacker aims at stealing confidential information, such as banking details or passwords, e.g. by impersonating an email sender known to the receiver [10, pp. 16f]. The financial damage caused to global brands by phishing attacks is estimated at $4.5 billion each year [51, p. 4].

Over the last few decades a series of valuable approaches to efficiently target spam have been developed. This includes content-based spam classification, reputation-based solutions, like IP *black-*, *grey-*, and *white-listing*, and authentication-based solutions. Two popular authentication-based protocols are Sender Policy Framework (SPF) and DomainKeys Identified Mail (DKIM). Both SPF and DKIM use the Domain Name System (DNS) to associate emails with a legitimate sender domain. SPF authorizes IP address ranges as valid email sources for a certain domain. DKIM uses cryptographic signatures, added to the email header, and a related public key that is deposited in the sending domain owner's DNS record. As an extension for DKIM, Author Domain Signing Practices (ADSP) further allows the sending domain owner to request a policy from the mail receiver, which specifies the desired treatment of emails with missing or invalid DKIM signatures [37].

With Domain-based Message Authentication, Reporting and Conformance (DMARC) a successor of ADSP has emerged, using both SPF and DKIM authentication and also providing a DNS-based policy mechanism. Furthermore, DMARC introduces a novel reporting mechanism that allows for inter-domain exchange of authentication attempts. It provides two types of reports: *failure reports*, which are sent immediately upon a failed DMARC authentication; and *aggregate reports*, which are sent periodically, and aggregate results from multiple authentications. As failure reports can drastically increase mail volume, the DMARC community recommends using them only in a narrow range of cases [13, par. "Do I want to receive Failure Reports (ruf=)?"]. Using the more compact DMARC aggregate report, which aggregates only a range of authentication results over a time period specified by the domain owner, does not carry this risk, but still has unprecedented potential to reveal insightful information about a given domain's mailing use and abuse. As aggregate reports are exchanged in an XML format, the need for appropriate parsing and interpretation software arises. The elicitation of the requirements for a DMARC aggregate report analysis software and the subsequent implementation of such software is the primary contribution of this Master thesis project.

1.1 Problem

Prior to the introduction of DMARC, mail administrators' insights into the use and abuse of the email domains they oversee were usually limited to report data generated within their own infrastructure. This still held true with the adoption of DMARC as an authentication protocol and policy enforcement mechanism. However, with the proliferation of DMARC reporting, mail administrators, for the first time, had the opportunity to monitor spoofing activities globally. In particular, detailed information about individual authentication attempts of sender domains collected at the receiver site permitted domain owners to assess the effectiveness of DMARC for identity protection. As the technology has become more robust, these findings have translated into more restrictive DMARC authentication policies.

Due to the novelty of DMARC[1] at the time this work commenced in mid-2014, little knowledge and experience regarding its use and usefulness was available to the related community. Nor was it certain whether DMARC would find broad and sustained acceptance within that community.

[1] DMARC was first released as *Active Internet-Draft* in March 2013 and has been an RFC since March, 2015 [33].

Nonetheless, initial discussions with the postmaster of the Vienna University Computer Center (ZID) showed substantial interest for a thorough investigation of the technology. As a consequence the Cooperative Systems (COSY) research group of the University of Vienna, together with the Zentraler Informatikdienst (ZID), initiated a research project, collecting DMARC aggregate reports for the email domains owned by the University. The goal was to evaluate the technology's potential for preventing identity fraud.[2] In addition to the custom report analysis for the University's domains, the project team also commissioned the development of a generic DMARC aggregate report analysis software. At that time several commercial tools to process and analyze DMARC aggregate reports already existed. However, in spite of the availability of these tools, they were not unconditionally accessible to postmasters, due to costs and data privacy concerns. As a consequence, the need for a free and self-deployable DMARC aggregate report analysis tool arose, one that could provide full report data sovereignty to the target users.

Moreover, in conversations with representatives of the Austrian Academic Computer Network (ACOnet), the idea of hosting a DMARC aggregate report analysis service within the ACOnet emerged. Due to the mutual trust ACOnet members had for each other, a new form of DMARC report analysis was conceived, namely *collaborative analysis*. The vast amounts of DMARC reports pertaining to a multitude of different mailing domains offered a pool of useful data, but commercial tools strictly isolated these reports from each other when presenting them to their customers. As a commercial service provider, any other policy would presumably violate privacy restrictions. However, in a network like ACOnet, where users know and trust each other, and also have the ability to opt in and out of sharing agreements, a collaborative analysis of DMARC aggregate reports could potentially reveal important insights into global domain spoofing activity and underground economics. Both *centralized* and *federative* deployments, offering different flavors of a collaborative setup, could be possible. That is, the analysis tool could be hosted centrally, requiring no deployment efforts for the individual collaborator, however, such an arrangement could reduce data sovereignty. Or, collaborators could run their own federated instances of the service, allowing them to fully control their report data and the extent of their collaboration with other hosters.

In order to appropriately investigate the above issues, the idea of including target users and leveraging their problem domain experience and knowledge throughout all stages of the development process was born. Such software development projects are usually referred to as *user-centered*. User-centered Design (UCD) is and has been thoroughly discussed in the literature. However, despite the availability of uncounted UCD resources "there is no complete published method that novices can pick up and use as is" [11,

[2] The corresponding project is led by Christoph Steindl, BSc. in the course of a Master thesis project.

par. 15.11], with the consequence that software developers often do not involve the user at all [38]. Thus, besides developing a DMARC aggregate report analysis tool, based on the target users' expertise, an additional goal was set: to conceive and to evaluate the execution of a proof-of-concept UCD process that transparently demonstrates the choice and use of the appropriate methodology to the reader of this work.

Since the commencement of this work DMARC has experienced significant, although selective uptake. Today it is used by big email providers and businesses including *Gmail*, *Yahoo*, *Facebook*, *Amazon* and *LinkedIn*, as well as major financial institutions such as *PayPal*, *VISA*, *Bank of America* and *American Express* [29, p. 1]. However, the overall adoption of DMARC still remains low [9, p. 3].

Reasons why DMARC has failed to catch on could include the setup and management of DMARC, which can require a significant investment in training for mail administration personnel, and the need to allocate more time for the additional tasks. Furthermore, in order for DMARC to be effective it needs to be adopted on a broad scale, since domain owners seeking to protect their domains require their communication partners to follow the DMARC protocol. And finally, there needs to be a reliable solution for false positives, that is legitimate mail being classified as illegitimate [25].

It should also be noted that while today new commercial DMARC report analysis tools are available and existing tools have improved, there is still no viable free and open source solution. This might be an explanation for the appreciable response to the release of the DMARC report analysis software created in the course of this work, as measured by the corresponding *GitHub* activity and the number of visitors who have watched the provided live demo page.[3,4]

1.2 Contribution

This work proposes *DMARC viewer*, a fully functional, web-based DMARC aggregate report analysis software program.

DMARC viewer is implemented using *Python Django* and other state-of-the-art web technologies. It provides an *analysis view* editor that allows the user to compose custom analyses, filtering and visualizing any desired aspect of the given DMARC aggregate report data to focus on geographical origin of DMARC-evaluated emails and temporal trends.

[3] https://github.com/dmarc-viewer/dmarc-viewer

[4] https://dmarc-viewer.abteil.org

The open source *DMARC viewer* code base is developed with readability in mind, including detailed documentation of the source code and its usage. This approach enables it to provide a reviewable and extensible tool to the target community.

DMARC viewer is designed for free and autonomous deployment, giving full report data sovereignty to its users. A live demo website is also available, as well as extensive documentation regarding both *traditional* and *containerized* deployments.

Developed for and with email experts, *DMARC viewer* uses a custom multi-stage UCD approach that includes a participatory field study, design testing and prototype evaluation. The fully transparent conception, preparation, execution and evaluation of the applied UCD approach is a secondary contribution of this work. Of particular interest is the characterization and discussion of the users' related expertise, and the appropriateness of the applied method in regard to that expertise.

Moreover, the user studies revealed early on a strong heterogeneity regarding said expertise. In particular, a lack of experience and knowledge related to the problem domain, that is DMARC itself, was observed. This may be attributed to the novelty of the DMARC technology at the time of conducting the user studies. However, as a consequence of this insight and the iterative and incremental development process, it was possible to shift the purpose of the proposed software from its initial tasks of enabling collaborative or federative analysis, to the new task of using DMARC aggregate report analysis as a learning aid for DMARC technology.

In that sense, one of the study participants coined the title of this work, "learning by viewing", as a play on words from "learning by doing." While the study participant referred to the proposed tool's composable analysis views, which not only allow a user to analyze the content of DMARC aggregate reports, but also to playfully examine the underlying technology, the phrase also suits the nature of the participatory development process. That is, by incrementally and iteratively observing (or viewing) the participants' interaction with the proposed system, the author of this work was able to learn about new user requirements. And vice-versa, as the project carried on the study participants learned more and more about the problem domain by walking through (or viewing) the materials provided by the author of this work.

1.3 Structure of this work

The remainder of this thesis is structured as follows: Chapter 2 provides an introduction to related technologies and methodologies fundamental to this work. More precisely, Section 2.1 introduces email and email security protocols, including SPF, DKIM and DMARC. In contrast, Section 2.2 provides a review of relevant UCD methodology, including *field* *study* and *prototyping* methods, such as *semi-structured interviews*, *wireframes*, *cognitive walkthroughs*, *usability analysis* and *think aloud*. The presented techniques are used for the conception and execution of a custom three-phase UCD process, as documented in detail in Chapter 3.

Section 3.1 traces the participatory field study and requirements-gathering sessions, exploring the problem domain of mail administration and identity fraud prevention, in order to collaboratively elicit requirements for a system design of the proposed DMARC aggregate report analysis software. In the course of the incremental expert user-driven development, the design and prototype evaluation sessions are further characterized in Sections 3.2 and 3.3, outlining goals, used methods, and results for each phase of the process, as well as a meta-analysis of the employed methods.

Chapter 4 presents the fully functional prototype implementation of the proposed DMARC aggregate report analysis software as a result of the insights gathered from the UCD sessions, as well as contributions from the related open source community that followed the software release.

In Chapter 5 this thesis concludes with a discussion of the significance of expertise, in particular, technological expertise in regard to UCD, and provides ideas for future work.

2 | Fundamentals and related work

Two types of knowledge are fundamental in order to develop a DMARC aggregate report analysis software in the course of a user-centered development process. That is knowledge about the target domain, including email and email security-related technologies, and knowledge about the related UCD methodologies and methods required to design and conduct user studies. Both thematic areas are covered subsequently.

2.1 Related technologies

This section gives an introduction to email and email security-related technologies that are relevant to this work. It will explain how the core email protocol SMTP functions, how it can be abused for identity fraud or spoofing and how DMARC, DKIM and SPF protect against such malicious activities.

DMARC, DKIM and SPF alike protect an email sender domain against spoofing by publishing information using the DNS. Upon reception of an email message, the receiver can authenticate the sender domain and, to some extent, provide guarantees about the message's integrity by using the information provided in the sender domain's DNS records. This process is mostly transparent to the end-user, as evaluation usually happens on the receiving mail server.

An alternative email authentication mechanism, that additionally guarantees full end-to-end integrity and confidentially is Pretty Good Privacy (PGP). PGP, among other things, differs from aforementioned methods in that it requires increased end-user initiative. That is, email users on both sending and receiving side have to manage and use their own cryptographic keys to sign and encrypt, or verify and decrypt email messages. Moreover, in order to provide those guarantees, PGP users have to make sure that a given key corresponds to the purported identity. Establishing the required trust often involves physically meeting the key owner, or relying on a *web of trust*, where a trusted key owner vouches for the key of a non-trusted key owner, thereby extending trust relationships [34, pp. 751–753]. DMARC, DKIM and SPF do not require offline key exchanges or webs of trust. Instead, they work on the assumption of a reliably accessible and non-compromised DNS, where the domain owner alone has write access to the administered DNS records. In addition, DMARC introduces a unique, inter-domain reporting facility, which is the focus of this work. Before diving into DMARC, DKIM and SPF, a brief introduction to email and DNS will be given subsequently.

2.1.1 Internet electronic mail

Just as ordinary postal mail (also known as *snail mail*), email requires multiple entities that follow well-defined protocols in order to deliver asynchronous messages from the sender to the recipient. At the heart of email transfer lies Simple Mail Transfer Protocol (SMTP), which was first standardized in 1982 and today is defined in RFC 5321 [27]. SMTP is based on a *client-server* model, where a compliant client and a compliant server both follow their respective parts of the protocol in order to exchange email messages.

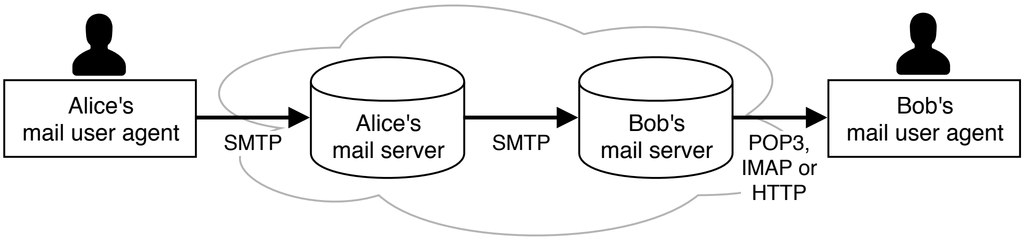


Figure 2.1: Email protocols and their communicating entities (based on [34, p. 154])

An exemplary email exchange can be seen in Figure 2.1. In this example Alice composes an email message and uses her Mail User Agent (MUA) to send the message to her mail server, typically using SMTP. Alice's mail server then attempts to relay the message to Bob's mail server, also using SMTP. Upon reception, Bob's mail server takes the message and places it into Bob's mailbox, identified by the host part of the recipient email address as specified by Alice. Bob finally has several options to fetch the message from his mailbox on the mail server, e.g. by using a user agent and one of the mail access protocols Post Office Protocol v3 (POP3) or Internet Message Access Protocol (IMAP) [34, pp. 153–158]. Benefits of having intermediate mail servers include that neither Bob's nor Alice's user agents need to be able to reliably accept incoming connections.

The corresponding SMTP dialog between Alice's and Bob's mail servers can be seen in Listing 2.1. Lines that originate from Alice's mail server are prefixed with *C:* for *client*, whereas responses from Bob's mail server are prefixed with *S:* for *server*. These prefixes

```
C: HELO alice-server.com
S: 250 ok
C: MAIL FROM: <alice@alice-server.com>
S: 250 ok
C: RCPT TO: <bob@bob-server.com>
S: 250 ok
C: DATA
S: 354 Go ahead
C: From: alice@alice-server.com
C: To: bob@bob-server.com
C: Subject: Testing the SMTP protocol

C: This is a test message
C: .
S: 250 ok
C: QUIT
S: 221 Bye bye
```

Listing 2.1: SMTP dialog (based on [34, p. 151])

are not part of the SMTP dialog but are added for readability.
Before starting an SMTP session, the client has to establish a TCP connection with the server. Once the connection is established, the client indicates that it wants to start an SMTP session by sending the *HELO* word followed by the server's domain name. Next, the client specifies which user the mail is coming from (*MAIL FROM:*) and which user the mail should go to (*RCPT TO:*). The server replies to these lines using an appropriate status code and optional English-language explanation. The *DATA* word indicates the beginning of the actual email message, which includes all lines until the client sends an isolated period. The end of the session is signaled by the *QUIT* word. Independently of the *envelope headers*, taken from the SMTP session (i.e. *MAIL FROM:* and *RCPT TO:*), the email message itself may supply additional header fields. These fields include *From:*, *To:* or *Subject:*, which may be used by the recipient's email reader for structured presentation of the email message. The *Internet Message Format* that defines these headers is standardized in RFC 5322 [56].

Note that the sender can easily use any desired identity, by specifying a spoofed sender email address in the SMTP session. DMARC has the ability to detect and prevent such activities and report them to the legitimate domain owner, which will be discussed further below.

2.1.2 Domain Name System (DNS)

Before discussing DNS-based strategies to mitigate email domain spoofing, it may be useful to briefly outline the functioning of DNS. As seen in Section 2.1.1, humans usually address entities that are connected by the Internet (e.g. mail servers) using their comparably mnemonic domain names. Routers and similar Internet equipment, on the other hand, charged with the task of relaying Internet packets (e.g. emails) across the network, use fixed-length IP addresses. In order to translate domain names to IP addresses and vice-versa, DNS provides a distributed database that can be queried for such purposes. For reasons of scalability, the domain-IP mapping for the broader Internet does not rely on a single DNS server. Instead, the database is distributed over a multitude of hierarchically structured DNS servers, including root, Top-level domain (TLD), authoritative and local DNS servers, delegating responsibilities for different domains or parts of domains.
In order to add new domain-IP mappings to the distributed DNS database, prospect domain owners have to purchase the domain from a so-called *registrar*. Similar to DNS servers, there are many registrars, the ensemble of which is accredited by the Internet Corporation for Assigned Names and Numbers (ICANN). An overview of DNS can be found in [34, pp. 158–172] and its specification is covered in RFC 1034 and RFC 1035 [40],[41]. It has been updated and extended by many additional protocols, most notably, for the purpose

of authentication, Domain Name System Security Extensions (DNSSEC) and DNS-based Authentication of Named Entities (DANE) [1],[21].

2.1.3 Sender Policy Framework (SPF)

SPF is a sender authentication protocol, used to verify whether a mail server's IP address is authorized to send emails using the specified domain. The domain owner authorizes or denies IP addresses by publishing an appropriate DNS *TXT* record [41, par. 3.3.14.]. The DNS record further specifies what the receiver should do with the evaluated email.

When a sending mail server initiates an SMTP session with a receiving mail server, the receiving mail server takes the domain, followed by one of the *HELO*, *EHLO* or *MAIL FROM* words (i.e. envelope headers) and queries the corresponding SPF DNS record. The receiver then compares the sender mail server's IP address with the IP addresses authorized or denied in the DNS entry and, depending on the result of the evaluation, continues or aborts the SMTP session. SPF is defined in RFC 7208 [26].

2.1.3.1 SPF DNS record

There are two main concepts in an SPF DNS record, so-called *qualifiers* and *mechanisms*. Qualifiers specify how to handle an email if the according mechanism matches. Mechanisms specify in what case the according qualifier should be applied. These mechanism-qualifier pairs are called *directives* and are evaluated sequentially if multiple exist. There are four different types of qualifiers: *Neutral (?)*, *Pass (+)*, *Fail (−)* and *Softfail (∼)*. *Neutral* means that the matched sender's legitimacy cannot be assessed, *Pass* means that the matched sender is legitimate, and *Fail* means that the sender is illegitimate. *Softfail* has the same semantics as *Fail* but suggests a generous treatment of the email that is being evaluated. A directive without qualifier defaults to *Pass*.
Furthermore, there are eight different types of mechanisms, including *all*, which always matches no matter what IP address was used to send the email, and the *ip4* and *ip6* mechanisms, which allow for specification of certain IP addresses or address ranges, which have to coincide with the sender's IP address. Additionally, there are the *a* and *mx* mechanisms, which point to the IP addresses specified in the corresponding DNS entries, as well as a *ptr*, an *exists* and an *include* mechanism. More details about qualifiers and mechanisms can be found in RFC 7208 [26, paras. 4.6.2. and 5.].

Listing 2.2 shows an exemplary SPF DNS TXT record. Since DNS TXT records can hold arbitrary text, the mandatory *v=spf1* version tag is used to identify the record as SPF record. The example provides two directives and reads as follows: “Authorize any IP

```
”v=spf1 ip4:192.168.0.0/24 -all”
```

Listing 2.2: Exemplary SPF DNS record

address in the range 192.168.0.0 – 192.168.0.255 to send emails using the given domain, and discard all others.”.

Authentication results The available SPF evaluation results and how they should be treated are defined in [26, paras. 2.6. and 8.] and summarized below.

- *pass* — The sending host is authorized to use the evaluated domain.
- *fail* — The sending host is not authorized to use the evaluated domain. Whether or not to reject the message is a matter of local policy.
- *softfail* — The sending host is believed not to be authorized to use the evaluated domain. However, the message should not be rejected but rather marked appropriately.
- *none* — The receiving host cannot evaluate the sender domain, due to the lack of a syntactically valid domain name or an SPF DNS record. Thus, no conclusion about whether to accept the message or not can be made based on SPF.
- *neutral* — The SPF DNS record of the evaluated domain makes no definite assertion (positive or negative) about the legitimacy of the host. The result must be treated like a *none*-result.
- *temperror* — The evaluation encountered a transient error (e.g. a DNS problem). The message can be accepted or temporarily rejected.
- *permerror* — The evaluation encountered a permanent error (e.g. a syntax error in the SPF DNS record). The message can be accepted or rejected.

Note that DMARC evaluation is partially based on above SPF results and furthermore may propagate these result to the domain owner via DMARC aggregate reports. DMARC will be explained in more detail in Section 2.1.5.

2.1.3.2 Known issues

SPF has been discussed controversially for the following reasons: If an email is forwarded via hosts that are not authorized in the sender domain owner's SPF DNS entry, the receiver's SPF evaluation, which can only examine the last hop in a forwarding chain, will yield a *fail*-result, although the email may have originated from a legitimate sender. The same issue may occur when using mailing lists. Additionally, email readers often display the email address taken from the header fields inside the email message. As a consequence, an email that passes SPF evaluation, using a non-spoofed address in the envelope header, may still present a spoofed address to the user [33, par. 3.1.]. As shown in Section 2.1.5, DMARC finds a remedy for the latter issue by requiring envelope and email headers to align. The former issue is mitigated in that DMARC does not solely rely on SPF evaluation results but on DKIM results as well.

2.1.4 DomainKeys Identified Mail (DKIM)

In contrast to path-based SPF authentication, DKIM bases its evaluation on the content of the message. It does so by signing an email's content or parts of it, associating the signature with a domain. That is, the sender mail server creates a cryptographic signature over the hash of selected header fields and the email message using a private key. The signature is then added as additional header to the email. The corresponding public key, used to verify the signature, is published in the DNS entry controlled by the sending domain. Upon reception, the receiving mail server looks up the public key from the sender's DNS entry, takes the signed hash and compares it to a newly computed hash of the message and header fields, and thus verifies the signer's identity and the integrity of the message. An email can be signed repeatedly by every host along the transmission path. As long as the content is not altered, each signature can pass validation. DKIM is defined in RFC 6376 [32]. Additionally, RFC 8301, which was published in early 2018, proposes an update to the used cryptographic algorithms [36].

2.1.4.1 The DKIM-Signature header and DKIM DNS record

An exemplary DKIM signature, as it may be added to an email using the custom *DKIM-Signature* header field, can be seen in Listing 2.3. The content of the header field is tag-value formatted and includes the following tags: The *v*=-tag designates the used DKIM version and the *a*=-tag the signing algorithm. The *h*=-tag indicates which header fields were signed. The only field that has to be signed is the *From* header. The *bh*=-tag holds a hash of the body part of the message and can be used as verification shortcut to avoid

```
DKIM—Signature: v=1;
a=rsa—sha256; d=dmarctest.info; s=q7xa;
bh=61Hmn0i17JOGUn3cPtWA5+S8jc8q7dWcEgYBvgxQPOQ=;
b=ltVvmwpZ35GmZ1z5DWIkjc0xNAaNKjaAST5qrKcF0qf1W+lkyJSJXdJ
QxJBqKZsMS3ryalVo1ct52+A1pdZ49MlSfqkDIRp1WlWpbn1eQwpXbDaO
OW1mZmfAzgjZldTbwgeJKtQ0d8UX5AleDQXRMoEjz7OLOKDIsxXdXC9k=
h=Subject:From:From:Subject;
```

Listing 2.3: Exemplary DKIM signature header

```
”v=DKIM1\; k=rsa\;
p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCpKphnOdsdIw9VZQc
WTc1phF6Ire+yzK5zIy+CxGfsbCkqrbxR/4w4KTvBnokFYpPZInPoat3v
yplj6BhaB1PVpKm272wS/sIfZGWN3Y2jBjQ/myWIUDepGmTGTedzOn/Ao
DqtO5Z1mH82MduoNcnthCmBKS1k1LodZdp0P0X40QIDAQAB”
```

Listing 2.4: Exemplary DKIM DNS record

computationally expensive cryptographic signature verification. That is, only if the hash in *bh*= matches the hash computed by the recipient, the signature contained in *b*= is verified. The address for the public key DNS lookup can be constructed using the domain found in the *d*=-tag and the selector in the *s*=-tag. In the given example the derived domain is *q7xa._domainkey.dmarctest.info*. An exemplary DKIM DNS TXT record that may be found under the given domain can be seen in Listing 2.4. It also has a tag-value format, where the public key data is identified by the *p*=-tag. Both formats are defined in RFC 6376 [32, paras. 3.5. and 3.6.].

2.1.4.2 Authentication results

The DKIM signature verification algorithm is described in RFC 6376 and yields one of three states: *SUCCESS*, *PERMFAIL* or *TEMPFAIL* [32, paras. 3.9. and 6.]. However, the RFC recommends using the codes described in RFC 5451 instead, when communicating DKIM results [32, par. 6.2.]. These are also the codes that are included in DMARC aggregate reports. As this work examines DKIM, where it is relevant to DMARC and especially

DMARC aggregate reports, the authentication results as defined in RFC 5451 are outlined here [31, par. 2.4.1.].

- *pass* — The evaluated signature(s) passed verification.
- *fail* — The evaluated signature(s) did not pass verification.
- *none* — There were no signature(s) in the email.
- *policy* — The evaluated signature(s) passed verification, but an additional local policy check, such as domain alignment, failed.
- *neutral* — The evaluated signature(s) contained syntax errors or could otherwise not be processed.
- *temperror* — The evaluation encountered a transient error (e.g. a DNS problem).
- *permerror* — The evaluation encountered a permanent error (e.g. a missing header field that is required for signature verification).

2.1.4.3 Known issues

While DKIM is more robust to false positives due to email forwarding or distribution over mailing lists than SPF, it can still fall victim to legitimate content modification along the transmission path, which is likely to break the DKIM signature. Such modifications include annotations made by mailing list servers, e.g. a link to the receiver's mailing list settings affixed in the email body, or the name of the mailing list prepended to the subject header. These issues and how to become a "dkim-friendly" email intermediary are discussed in detail in RFC 6377 [30, paras. 2.4. and 3.].

Another shortcoming of DKIM is that it does not provide a mechanism to inform the receiver about the sender's signing practice, i.e. whether emails from the sender's domain require a DKIM signature at all, or how to preferably react to a given DKIM evaluation result. ADSP, a precursor to DMARC that never received widespread adoption [13, par. "What happens if a sender uses DMARC and ADSP?"], tries to solve this issue by introducing a separate DNS TXT record that prescribes one of three signing practices: *unknown*, which indicates that the domain can author emails that might be signed or not signed; *all*, which indicates that all emails from this domain are supposed to be signed; and *discardable*, which is equivalent to *all* but encourages recipients to discard messages upon a DKIM *fail*-result [37, par. 4.2.1.]. Other known attack surfaces of DKIM, such as DNS or key compromises, are discussed in RFC 4686 [15].

2.1.5 Domain-based Message Authentication, Reporting and Conformance (DMARC)

DMARC is a combined authentication mechanism that uses the results of the authentication protocols SPF and DKIM, and extends the ADSP standard as a DNS-based policy-publishing mechanism for conformance. Additionally, DMARC provides a novel mechanism for inter-domain reporting, which is the focus of this work. It is defined in RFC 7489 [33].

For a DMARC compliant mail flow the sending domain owner should have either an SPF DNS record, or a working DKIM signature with the according DKIM DNS record, or both. If none is provided DMARC authentication will necessarily produce results of type *fail*, as it is based on these mechanisms and requires at least one of the two checks to be valid in order to produce a result of type *pass*.

A typical DMARC mail flow is illustrated in Figure 2.2.[1] It shows how the sending mail server uses a private key to create a DKIM signature for an email message that was submitted by a client. The email, together with the signature, is sent to the receiving mail server. The receiver then verifies the DKIM signature and the SPF authorization of the sender domain, both using information extracted from the corresponding DKIM and SPF DNS records. Subsequently, the DMARC verification routine compares the sender domain evaluated by DKIM, i.e. the domain in the *d*=-tag of the signature, and SPF, i.e. the domain taken from the envelope header, with the domain in the message header (see Section 2.1.1 for an explanation of the different *from* headers and Section 2.1.3.2 for issues due to non-alignment). Only if at least one of DKIM or SPF aligned domain evaluation passes will DMARC produce a result of type *pass*, i.e. successfully authenticate the sender domain.

Following the evaluation, the receiving mail server queries and applies the policy published in the corresponding sender domain owner's DNS entry, according to the evaluated DMARC result. Available policies are *none* (the sender domain owner requests no specific action no matter the evaluation result), *quarantine* (depending on the capabilities of the mail receiver, this can mean "place into spam folder", "scrutinize with additional intensity", and/or "flag as suspicious"), and *reject*, which should terminate the SMTP session.

The applied policy is called *disposition*. If the DMARC evaluation passes or the sending domain owner has requested a *none*-policy, the email is delivered to the receiver mailbox. In case of *quarantine* or *reject*, and if the sender domain owner has specified a Reporting

[1] This figure is derived from joint effort with Christoph Steindl.

URI for Failure Data (RUF), the receiving mail server must immediately send a failure report to the sender domain owner. Independently of the evaluation, the receiving mail server should temporarily store the results in order to later send an aggregate report to the owner of the purported sender domain, but only if the sender domain owner has provided a Reporting URI for Aggregate Data (RUA) in the corresponding DMARC DNS record. The aggregate report contains all DMARC evaluation results for the evaluated domain and within the requested aggregation interval, which is typically a day. An overview of the general DMARC flow can also be found in RFC 7489 [33, par. 4.3.].

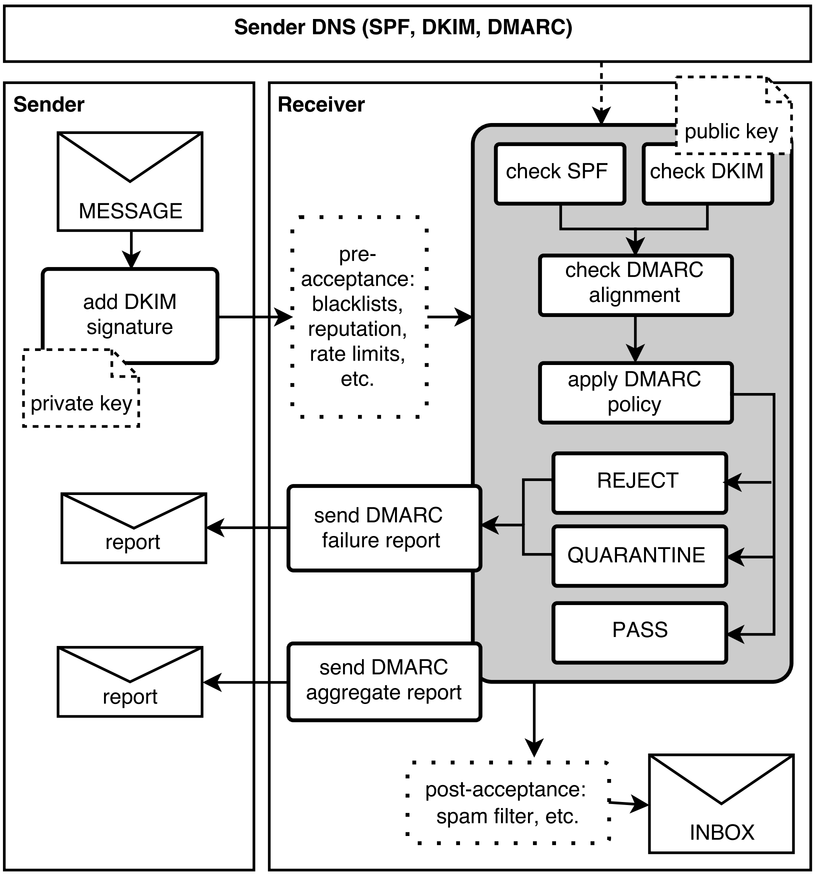

Sender DNS (SPF, DKIM, DMARC)
Sender
MESSAGE
add DKIM signature
private key
report
report
Receiver
public key
check SPF
check DKIM
check DMARC alignment
apply DMARC policy
REJECT
QUARANTINE
PASS
pre-acceptance: blacklists, reputation, rate limits, etc.
send DMARC failure report
send DMARC aggregate report
post-acceptance: spam filter, etc.
INBOX

Figure 2.2: DMARC mail flow

```
”v=DMARC1\;  p=reject\;
rua=mailto:re+iw36i2xvuub@dmarc.postmarkapp.com\;
mailto:x9qrezxv@ag.dmarcian.com,
mailto:dmarc_rua@dmarctest.info ,
ruf=mailto:dmarc_ruf@dmarctest.info\;”
```

Listing 2.5: Exemplary DMARC DNS record

2.1.5.1 DMARC DNS record

DMARC DNS TXT records follow the same extensible tag-value syntax as DKIM and SPF. An exemplary DMARC DNS record can be seen in Listing 2.5. The record consists of two required fields, that is the version field *v=* and the policy field *p=*. If a sending domain owner wishes to receive aggregate reports, a RUA-tag (*rua=*) must be specified. Failure reports, on the other hand, are sent to the address specified in the RUF-tag (*ruf=*). Using the *adkim=*- and *aspf=*-tags, alignment modes for SPF and DKIM can be set to either *strict* or *relaxed*. In strict mode, the sender domain evaluated with SPF or DKIM must fully align with the sender domain taken from the message's *From* header. In relaxed mode, only the organizational domains, i.e. top-level domain plus rightmost part of a domain, must align. Other tags include:

- *ri=* — Aggregate report reporting interval (default is 86400, i.e. 24 hours)
- *fo=* — Failure report options
- *rf=* — Failure report format
- *pct=* — Percentage of emails to which DMARC policy should be applied (used for slow roll-out)
- *sp=* — Similar to the *p=*-tag, but applied to sub-domains.

The full DMARC DNS TXT record format is specified in RFC 7489 [33, par. 6.3].

2.1.5.2 Reports

Of particular interest for this work is the unique inter-domain reporting facility provided by DMARC. Two different reporting strategies are supported: aggregate reports and

failure reports.
A DMARC compliant receiving mail server should send reports depending on whether the sending domain owner has specified RUA and/or RUF addresses in his or her DMARC DNS record. Failure reports are sent on a per-message basis. The report format as well as options when to send failure reports can be parametrized via the corresponding DNS record. The DMARC community advises to be careful when requesting failure reports, as they may contain an entire copy of the email, which can lead to an enormous increase in email volume [13, par. "Do I want to receive Failure Reports (ruf=)?"]. Additionally, mail administrators might be reluctant to receive an entire copy of their users' emails due to privacy concerns. Aggregate reports, on the other hand, are sent at a requested interval and contain substantially fewer data, of which only sender mail server IP addresses are classified as personal. The XML Schema of DMARC aggregate reports is defined in RFC 7489 [33, appx. C] and an exemplary report can be found in Listing 5.1 on page 120 of the appendix. DMARC aggregate reports contain the following information:

- **Metadata** — Information about the report sender (i.e. receiving email server) and the date range for which the report aggregates domain authentication attempts.
- **Policy** — Information about the policies specified by the sending domain owner in the corresponding DNS entry at the time of email reception.
- **DMARC evaluation results** — Information about DKIM and SPF authentication as well as DMARC alignment and disposition aggregated by email sender's IP address.

It is noteworthy that the *eco Competence Group Email* of the German *Association of the Internet Industry* has assessed the exchange and use of DMARC aggregate reports as acceptable in terms of data privacy, under the condition that it is used to detect and prevent spam and phishing [28, pp. 19f]. However, although privacy data law experts have classified DMARC aggregate reports as insignificantly privacy sensitive, domain owners may still be reluctant to make their aggregate reports available to third parties such as commercial DMARC analysis providers. This reluctance was one of the key motivations to develop an open source DMARC aggregate report analysis tool, deployable by mail administrators inside their own domains.

2.1.5.3 Known issues

Because DMARC relies on SPF and DKIM, some of the issues pertaining to those technologies affect DMARC as well (see Sections 2.1.3.2 and 2.1.4.3), most notably, the risk of false positives, i.e. to falsely assess legitimate emails as illegitimate, which is usually a consequence of indirect mail flow. The authors of the DMARC standard have been working on a protocol to mitigate this risk. The Authenticated Received Chain (ARC) protocol is designed to preserve email authentication results across the transmission path, by prescribing individual intermediaries to add the evaluation results for the preceding hop to the forwarded email message. A discussion of ARC can be found in the corresponding *Internet-Draft — draft-ietf-dmarc-arc-protocol-16* [2].

2.2 Related methodologies

This section gives an overview of existing User-centered Design (UCD) methodology and methods. A broad review of the related literature was crucial for the author of this work in order to design the UCD studies carried out in the course of developing the proposed DMARC aggregate report analysis tool. It also provides pointers to guiding textbooks and online resources to the interested reader.

2.2.1 User Experience (UX)

UCD is situated within the larger ecosystem of User Experience (UX) design. According to Don Norman, who invented the term, "UX encompasses all aspects of the end-user's interaction with the company, its services, and its products" [48]. That is, UX does not merely refer to a user's experience when interacting with a software product's User Interface (UI), but rather describes the entire *user journey*, including acquiring, owning and even troubleshooting the product. In order to provide a usable product to the users, so that they can carry out their context-dependent tasks in an efficient and pleasurable manner, UX-design often relies on UCD [19, par. "UX Design is User-Centered"].

An overview of the many disciplines and methodologies that are brought together under the umbrella of UX-design can be found in Figure 2.3. Of particular interest for this work are the research field of Human-computer Interaction (HCI) and its more applied cousin-discipline Interaction Design (IxD), which both have laid important theoretical and practical groundwork in terms of UCD. To learn more about how these disciplines have emerged alongside and are based on each other, consider reading the introductory chapters of [35, pp. 1–11] and [58, pp. 2–22].

Different flavors of UCD include Contextual Design (CD) or *customer-centered design*, which is a scaffolding for UCD methods and methodologies developed by the industry practitioners Hugh Beyer and Karen Holtzblatt [4],[22]. Furthermore, in Participatory Design (PD) or *co-design* the process is shaped by the cooperation of all sorts of experts

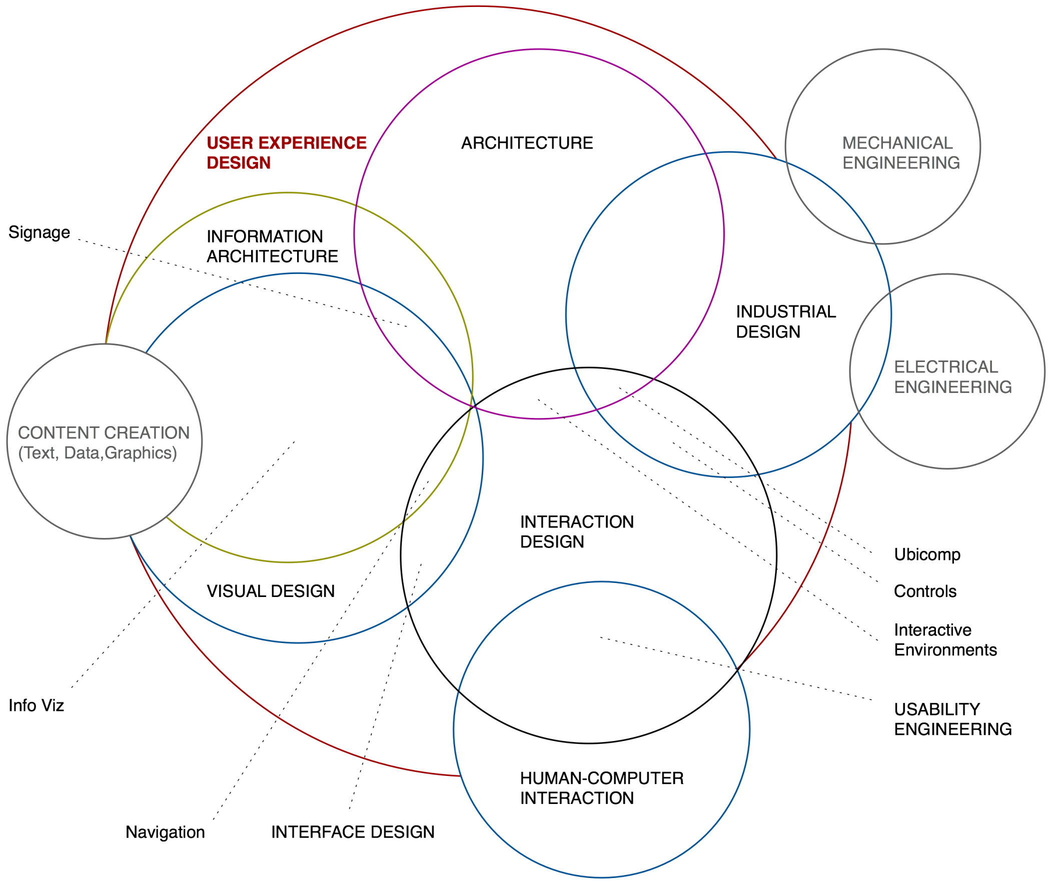


Figure 2.3: The Disciplines of User Experience by Dan Saffer [58, p. 21]

including customers and users, who are perceived as experts of their experience and aren't treated as mere study objects but rather as central co-creators [66],[60].

The following subsection will describe a standardized abstract UCD process followed by an introduction and discussion of a few of the most important UCD methods for data collection, analysis and evaluation pertaining to most UCD processes.

2.2.2 User-centered Design (UCD)

UCD or Human-centered Design (HCD) is an approach to develop interactive systems and to make them useful and usable by including the target user throughout all stages of the development process.

A high-level UCD process is standardized in *ISO 9241-210:2010 Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems* the successor of *ISO 13407:1999: Human-centred design processes for interactive systems* [24],[23]. While the norm does not prescribe a clear structure of a software project or any methods that may be used, it suggests a set of principles, and reasons to follow these principles.

2.2.2.1 Principles of UCD

- The development is based on a broad understanding of the prospective user and his or her work tasks and work environment

- The user is included throughout the development process using methods of user-centric evaluation

- The process is incremental and iterative, that is each step of the process is based on insights and decisions from previous steps and may be repeated

- The process covers the entire user experience

- The development team usually consists of various experts from diverse fields, such as business administration, human resources, psychology, ergonomics, usability, accessibility, HCI, user research, the system's target domain, interface- and visual design, system architecture and software engineering

See *ISO 9241-210:2010* [24, par. 4].

2.2.2.2 Why use UCD?

- The productivity of users and the economic viability of their organizations can be enhanced
- The need for user training and support is reduced
- Enhanced accessibility benefits users with various backgrounds and abilities
- User stress and discomfort are reduced
- It helps to attain sustainability goals

See *ISO 9241-210:2010* [24, par. 3].

2.2.2.3 The activities of UCD

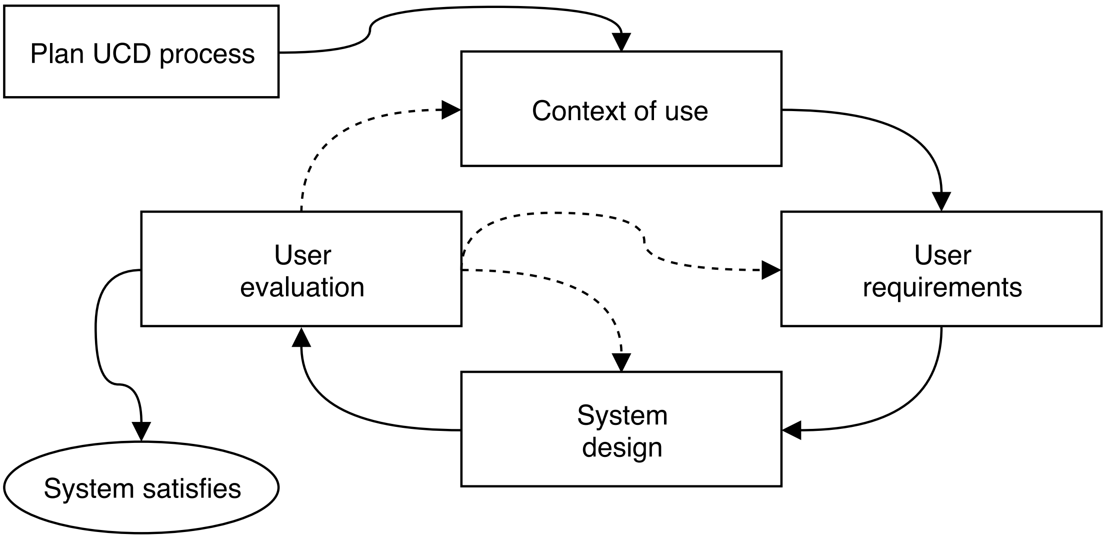


Figure 2.4: Overview of UCD activities (based on *ISO 9241-210:2010* [24, par. 6])

A high-level depiction of the activities of a UCD process, as defined by [24], can be seen in Figure 2.4. It is important to note that user participation does not only happen once a mature prototype of the system is established, but indeed throughout the entire process. Users can and should be included in order to understand the context of use, elicit requirements, and test designs and prototypes. An overview of the methods of user participation and their fitness for the various stages of the UCD process will be given subsequently.

2.2.3 User study methods

Today an endless amount of resources about UCD methods are available. Choosing the appropriate method or set of methods depends on various factors, such as availability and experience of the developer team, time constraints on developer- and user-side, or type and maturity of the system that is being developed.

A few dichotomous paradigms may be used in order to characterize and categorize the available methods. The terms on the left tend to be found at the start of the development process and the terms on the right at its end:

- *explorative* vs. *experimental*
- *qualitative* vs. *quantitative*
- *formative* vs. *summative*

Explorative research is usually conducted early in the development process when the developer team is entering the target domain, with the goal of establishing mutual knowledge of the domain by discussing preliminary requirements or ideas about the system that is being developed. Experimental research, on the other hand, aims at validating or discarding assumptions or hypotheses about the system and usually requires a more mature system. Similarly, formative and summative evaluation relate to the maturity of the system to be evaluated, where formative evaluation aims to gather ideas and requirements of the system that should be developed. By contrast, summative evaluation is after the fact, it evaluates an implemented system focusing on, for example, usability aspects.

Qualitative and quantitative methods can be used throughout all stages of the development process. However, early stages of the process (explorative/formative) tend to favor qualitative studies using, for example, in-situ observations or open-ended interviews, in which subjective descriptions of participants' ideas or difficulties with existing systems provide helpful insights to elicit requirements for the system to be developed. Quantitative methods, on the other hand, are more useful in an experimental or summative setting, e.g. by recording the time a user takes to carry out a task using the system under investigation, or by statistically analyzing post-user study questionnaires about user satisfaction.

The first explorative field study that was conducted in the course of this work revealed a general lack of expertise and experience with the related technology, that is DMARC and especially DMARC aggregate reports. Thus, the negotiation of a mutual understanding between study participants and developer, regarding the use and usefulness of the target system, remained a crucial task throughout the entire process. As a consequence, this work focuses primarily on qualitative research methods in the formative phase (field study and requirements gathering) but also later for summative evaluation (design and prototype tests). A fitting explanation for why the author of this work and researchers in general choose qualitative over quantitative methods can be found in [12, p. 5], where the authors state that qualitative studies take a holistic and comprehensive approach to study areas not yet thoroughly researched and explore how meanings are formed and transformed.

The field of UX borrows much of its methodology and methods from the human and social sciences, therefore textbooks from these disciplines provide good starting points to learn how to design and conduct qualitative and quantitative studies and evaluate the resulting data. Some of the books that were used in order to prepare this work and can be recommended by the author are *Basics of Qualitative Research* [12], *An introduction to qualitative research* [17], and *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler* [6]. Moreover, books including *Research-methods in Human Computer Interaction* [35], *Designing for Interaction* [58], or *Rapid Contextual Design* [22] show how to apply these research methods in the context of UCD-driven software design and development.

In addition to the textbooks, a lot of well-curated online resources are available. These websites provide articles, tutorials and webinars. Of particular help for this work were the publications by *usability.gov* a website managed by the *U.S. Department of Health and Human Services' (HHS) Office of the Assistant Secretary for Public Affairs*, one of the leading resource for UX best practices and guidelines [69].

Furthermore, the *Interaction Design Foundation*, a Denmark-based non-profit organization collaborating with universities and companies, offers extensive design education both as free and paid services [18].

Also worth mentioning is the *UX Cheat Sheet* and other introductory articles and decision aids for UX research methods, provided by the *Nielsen Norman Group*, a UX consulting and training firm, led by UX experts Jacob Nielson, known and taught for his UI heuristics; Don Norman, who coined the UX term; and Bruce Tognazzini, author of many works about UI design and testing, including *The Apple Human Interface Guidelines* [14],[57].

The huge amount of methods may at first seem intimidating to the UCD-novice, but as the authors of *User Interface Design and Evaluation* put it so aptly, "any kind of user testing is better than none" [67, p. 24]. Subsequently, the most common methods are briefly outlined, providing pointers to further readings.

2.2.3.1 Field studies

At the beginning of many UCD processes lies the *field study*. Although the field study is sometimes referred to as a UCD method it can also be seen as an initial explorative stage of the UCD process that can employ many different methods to gather and evaluate data from the target domain. Field studies are sometimes also referred to as *ethnographic studies* or *contextual inquiry* [49]. Some of the methods that are typically used in the field study phase of a software development project are:

- *Observations*: When conducting observations the researcher goes to the user's environment, e.g. his or her work place, to get an idea about daily routines and tasks. The researcher takes notes or makes recordings about the user's behavior relevant to the problem domain, e.g. particularities or difficulties in the interaction with an existing system. As a consequence, observations can give unique insights into the user's work reality that might not be revealed in an interview, because a prepared set of questions can never cover all aspects that are important to develop a proposed system. On the downside, the researcher runs the risk of misinterpreting the observations made, or might be perceived as intrusive by the study participant, which in turn can lead to unnatural behavior of the observed study participant. More details about observations can be found in [6, pp. 262–277],[67, pp. 29–31], [12, pp. 40–42],[17, pp. 221f] and [58, p. 86].

- *Diaries*: Diaries are a form of self-observation, where study participants capture their work reality on their own. Diaries can reveal aspects that are particularly important to the user without the intrusion of a researcher at the time of data collection. On the other hand, keeping a diary requires more original investment by the participant, and to some extent assumes that the participant knows what information might be of interest for the study. More details about diaries can be found in [35, pp. 135f],[16] and [71].

- *Personas* and *task analysis*: Personas or task analysis are effective methods to structure findings from field studies.
  Personas are archetypical user descriptions based on similar behaviors, goals and motivations of users representative for the study. Personas often include fictitious biographical information such as a realistic name and a *head-shot* photo. They are personifications of the target audience and may be used to recruit appropriate participants for usability studies in later stages of the development process. Since personas are an abstraction of actual users and often contain fictitious elements for richer descriptions, the technique has been criticized as providing only a loose

relationship to the target users.
In contrast to personas that describe archetypical user characteristics, task analysis deals with the way personas, or actual users, perform their work. In addition to the structured set of related activities to achieve a goal, the user's knowledge required to perform a given task is examined as well. Task analysis may translate more directly to the design of the target system than personas. However, in order for task analysis to be effective, the target system should relate to existing user tasks. This is not always the case, as for instance in this work, where the participants of the conducted UCD studies had no or very little prior experience with using or analyzing DMARC aggregate reports.

Learn more about persons in [42],[73], [20],[58, p. 106] and [22, pp. 181f], and about task analysis in [22, p. 190], [67, pp. 57 and 625] and [78].

While above methods are presented for the sake of completeness, and to make the election process of employed UCD methods transparent to the reader, they were not used as methods of data collection or evaluation in this work. This was because they were either too time expensive, or required unattainable user investment. Furthermore, task analysis appeared to be more appropriate in contexts, where the target system tries to solve a problem in the current work practice. This was not given due to the novelty of the technology under investigation (DMARC) and the uncertainty about the actual use and usefulness of the proposed system (DMARC report analysis). Instead, a combined field study and requirements-gathering session was conducted using the tried and tested method of interviews.

2.2.3.2 Interviews

Interviews are the primary instrument of qualitative research, and many textbooks dedicate extensive chapters to interviewing methods and techniques [6, pp. 236–262],[17, pp. 149–170],[12, p. 37]. It seems natural that UCD makes use of this method, especially in early explorative phases of the development process like field studies and requirements gathering, but also in later phases, e.g. as a debriefing method after having conducted a usability study [35, pp. 187–228],[67, pp. 33f],[58, p. 87],[44].

Interviews come in many shapes and colors, characterized, for example, by the number of participants in an interview session, or the level of standardization. Standardization refers to the flexibility of the interview and its questions, where interviews with very closed predefined questions become similar to surveys using questionnaires. Interviews with less structure and open questions, on the other hand, are more akin to conversations or discussions. Furthermore, interviews can be one-on-one or involve multiple interviewers or interviewees. While interviews with multiple interviewers or interviewees can leverage the synergy of multiple speakers or listeners, they require a larger research team and more availability and investment of the interviewees. Thus, if recruiting and scheduling individual interviews with interested and knowledgeable participants is already difficult, bringing together multiple such participants at the same time is not feasible. Additionally, interviews with more participants can be subject to unwanted peer effects. As a consequence one-on-one interviews were chosen for this work.

As for standardization, semi-structured qualitative interviews proved to be especially effective for the field study and requirements-gathering phase of this work. This type of interview allows the participant to freely speak about topics they find important, while making it easier for the interviewer to steer the conversation towards thematic areas of interest [12, p. 39]. The authors of [5, par. 52.6.4] describe how to walk through the phases of a semi-structured interview:

- **Arrival** — Put the interviewee at ease.
- **Introduction** — Inform about the purpose of the study and ensure that there is informed consent, that is about recordings of the conversation, usage of data and the option to withdraw at any time.
- **Beginning the interview** — Give the participant confidence and gather background information that can contextualize the rest of the interview.
- **During the interview** — Focus on the thematic areas important to the research.
- **Ending the interview** — Appropriately indicate the end of the interview to avoid the feeling of loose ends.
- **After the interview** — Thank the interviewee, e.g. by additionally handing over (promised) participation incentives, and make him or her leave feeling well. Often interviewees provide interesting information after the recorder is turned off, which may be noted.

More helpful guidelines, tips and tricks on how to plan and conduct interviews can be found in [6, pp. 244–245],[17, pp. 170–173],[35, pp. 210–216] and [22, pp. 88–90].

In [35, pp. 188–189] the authors talk about the pros and cons of conducting interviews in the course of user studies. While they provide the ability to go deep and explore ideas and insights that were not anticipated, in order to increase the understanding about the problem domain, they depend on the participant's ability to recall and articulate useful information. Furthermore, the interviewer is required to bring an extensive skill set, both for planning and conducting the interview but also for analyzing huge amounts of qualitative data. Consequently, some techniques to deal with qualitative data such as those gathered by conducting interviews will now be discussed.

Qualitative data analysis Before collected data can be interpreted, it must be prepared appropriately. Notes taken during an interview should be processed as soon as possible after having conducted the interview. This makes it easier to recall associated information that the notes can be enriched with. Additionally, audio recordings may be transcribed using different approaches. Verbatim transcription gives a full written account of the interview, which may be easier to further analyze using software or to exchange with colleagues. However, transcribing an interview word by word is a cumbersome exercise and might not always be necessary. It may be easier to base the analysis on notes and just resort to the recordings when coming across particularly interesting bits of information in those notes [35, pp. 219–224].

Once the data is prepared, an analysis method can be selected. Data interpretation usually involves some sort of coding, or clustering, by categories that either emerge from the content or were defined a priori. A simple approach that was employed in this work is described in *Interview Analysis for Novices* [35, p. 222]. The basic idea is to isolate individual notes or ideas and place them each on an index card or in a separate line of a text document. Subsequently, these items are annotated with category names. If categories were not defined beforehand, they can be formed by using verbs or nouns taken from the individual items. As the list of categories grows larger, a hierarchy of categories can be formed by looking for connections between the initially identified categories and summarizing and moving items to broader categories or sub-categories. If validity of the categorization is important, it is helpful to consult colleagues.

It is also important to be aware of any bias when analyzing qualitative data. Bias can result in a practice that is referred to as *cherry picking*, where the researcher only filters parts of the data or interprets them in a way that appears opportune to the study. Avoiding cherry picking is especially challenging if the UCD researcher who interprets the data is also the developer who implements the target system. This should be and is usually avoided in UCD projects (see UCD principles in Section 2.2.2.1). Since this work was carried out by a single person, increased sensitivity to this problem was of utmost importance.

More in-depth information about analysis of qualitative data and how to use different coding approaches can be found in [5, par. 52.7],[35, pp. 299–327] and [17, pp. 305–332].

Design implications A crucial link between the analysis of collected data and the design of the target system is the identification of design implications based on the interpreted data. This process is sometimes referred to as ideation [58, pp. 113–126].

The inventors of CD suggest using a sequence of well-described methods, in order to transform the findings from qualitative data into a prototype. These methods include *affinity- or wall walking*, that is collectively going through the categories found during data analysis, generating and recording *hot ideas* and *key issues* that are subsequently used to build *visions* and *storyboards*, which eventually inform the paper prototypes [22, pp. 193–243].

While following such clearly defined protocols may be helpful and effective to generate design implications, the authors of [61] also stress the importance of the designer's creativity and knowledge, e.g. experience or heuristics, in addition to using the right method or set of methods.

2.2.3.3 Prototyping and evaluation

After having explored the target domain and extracted the relevant design implication from the collected data to compile an initial requirements catalog, it is time to create and evaluate prototypes of the proposed system. Prototypes may vary in terms of fidelity, ranging from low fidelity prototypes, e.g. paper mock-ups that sketch out the structure of the user interface and its components, to wireframes, which are mock-ups created using design software, up to high fidelity prototypes that might contain some or all of the functionality of the target system. Low fidelity prototypes are less costly in terms of designer or developer investment, which makes them more suited for studies that encourage substantial changes to the prototype.

Card sorting Card sorting is a prototyping method used to cooperatively determine a UI structure. Here the user is presented a set of unstructured UI components each on a separate index card, and is subsequently asked to group or arrange the individual items. Card sorting is an easy and cheap method to explore the user's mental model of the target system and also helps to identify relevant terminology that is likely to be misunderstood [67, pp. 311 and 550]. The method can vary in its degree of openness, that is whether the items and categories are all or mostly predefined or are created by the study participant. Very open card sorting is akin to the method of user sketching, where the study participants draw mock-ups of the target system or parts of it in the course of a usability evaluation session. As with many UCD methods, card sorting can be employed in various stages of the development process, e.g. in earlier exploration sessions prior to prototyping, where the information architecture is assessed cooperatively [70]. In this work a custom association method, inspired by card sorting, was conducted in the field study and requirements-gathering phase (see Section 3.1.3).

Cognitive walkthroughs The cognitive walkthrough is another method to explore a user's mental model about the target system, which often uses prototypes of low fidelity. In the UCD literature, cognitive walkthroughs usually involve a set of tasks that the study participant is asked to carry out during a prototype testing session. The goal of the cognitive walkthrough is to uncover mismatches between how the user thinks a task is performed and how the designer thinks about the task, determining whether a user can choose the appropriate actions using the proposed interface [67, pp. 71 and 607].

This work utilized a variation of cognitive walkthroughs, in which the focus is shifted from walking through specific tasks to cooperatively exploring the interface and assessing its usefulness and usability for analyzing DMARC aggregate reports. Section 3.2.2 describes how the customized technique was used. In order to acknowledge the role of the study facilitator and the cooperative aspect during the walkthrough session, this method is referred to as an *assisted walkthrough*. Assisted walkthroughs were also used as a method for *competitive usability evaluation*, exploring strengths and weaknesses of existing DMARC analysis tools based on wireframes of their UIs [62].

Scenarios In contrast to paper sketches and mock-ups, high fidelity prototypes allow designers to present a more lifelike representation of the system to study participants, in the hopes of provoking more practical feedback. The authors of [55, p. 193] claim that although computer and paper prototypes are equally suited to detecting usability issues, participants usually prefer to interact with computer-based prototypes. Moreover, evaluating functional prototypes may reveal true human performance data, such as task completion time or error rate [74, par. "High-Fidelity and Low-Fidelity Prototyping"]. According to the authors of [55, p. 188] the best prototype evaluation method is to provide the study participants with representative scenarios, and allow them to carry those out using the target system.

The UCD literature uses the term *scenario* in different contexts. Scenarios may be presented as detailed and narrative descriptions of an already existing user task (*task scenario*), or as a description of a task in the anticipated computer system that is being developed (*use scenario*) [67, pp. 625 and 627]. Both of these may be based on insights collected in early explorative stages of the development process, in addition to personas [22, p. 190]. While scenarios as described above are used in the ideation phase of the target system, i.e. to transfer field study user data to a concrete design, activities that should be carried out by users during prototype evaluation sessions may also be referred to as scenarios. The latter requires a different approach. As outlined in [76, par. "Using Scenarios in Usability Testing"] scenarios that are being presented to the study participant should not include step-by-step instructions of how to accomplish a given task, as that could potentially conceal usability issues. It is nonetheless beneficial for the researcher to prepare reference solutions against which the results of the study can be compared. Preparing these solutions for each task also provides a benchmark for how long a task should take.

Think aloud There are various methods available to collect user data while participants are engaged in a prototype testing session. Given that this work focused on explorative data collection methods throughout the entire study, gathering quantitative data was not a priority. Instead, the think aloud technique was employed in the scenario-based prototype testing study (see Section 3.3.2). Think aloud is a method that asks users to verbalize their thoughts and feelings about the proposed system, while they are carrying out the tasks presented to them in the course of the test session, including little to no intervention of the study facilitator. According to the usability engineering expert Jacob Nielsen it has been "the single most valuable usability engineering method" for the past 25 years, and it still is [45]. In the cited article Nielsen further deliberates on the reasons for the method's importance: Think aloud is cheap, because it does not require any special tracking equipment or testing environment; it is easy to learn and robust against facilitator mistakes (more robust than statistical usability evaluation methods, for instance); and it is flexible enough to work with very low to very high fidelity prototypes, all while revealing rich user data about the usability of the target system. This method does have its drawbacks, however. Users may find it difficult to stick to the think aloud protocol, and they may feel it unnatural to talk aloud while being observed performing complicated tasks. As a consequence, the facilitator might need to repeatedly remind participants to keep their monologue running, or else probe the participants for specific information without biasing their behavior.

A taxonomy of the different moderating techniques that can be used for usability evaluation is provided in [75]. The moderating techniques include Concurrent Think Aloud (CTA), Retrospective Think Aloud (RTA), Concurrent Probing (CP) and Retrospective Probing (RP), which are distinguished by degree of facilitator intervention and whether they are performed during or after users carry out their tasks.

As discussed above, there are trade-offs between using think aloud or the probing technique, and between using a concurrent or a retrospective technique. Both CTA and CP might interrupt the user's train of thought, whereas RTA and RP may miss information due to the participant not recalling important things.

A special case of RP is the *critical incident method*, where study participants are asked to retrospectively name "single outstandingly positive or negative experiences" noted while working with the target system [59].

Questionnaires A low-cost "quick and dirty" way of quantitatively measuring the usability of a system is the use of questionnaires like the System Usability Scale (SUS) [8]. It has the benefit of being reliable even with a small number of participants, and the results can be considered valid in the sense that it can effectively assess if a system is usable or not [77, par. "Considerations when using a SUS"]. However, the SUS does not reveal any tangible issues; it only gives an idea of whether there are important usability issues in the evaluated system or not. The 10-item SUS questionnaire can be found in Table 5.1 on page 118 of the appendix.

Post-session interviews After having conducted a prototype evaluation session it is important to debrief the participants in informal, off-record post-session interviews. Debriefing helps to assure the participants of their positive contribution. Furthermore, these interviews provide space for interesting user thoughts that might not have fit in earlier. Debriefing can also be used to share more in-depth background information about the study and its goals with the participants without biasing their evaluated performance [35, p. 216],[67, p. 621].

As with all stages of the UCD process, prototyping might include several iterations.

2.2.4 Heuristics

While Section 2.2.3 presented *user-centered* or *interaction-centered* UX-design methods, where target users actively participate in the development process, this subsection names popular *system-centered* UX-design methods, such as generic heuristics, principles and guidelines, whose validity in terms of good UX has been proven in the past.

Evaluation sessions based on heuristics are often called *expert reviews* [35, p. 268],[72]. Note that the expert term refers to the reviewer's usability expertise and not to the user's expertise. The latter is discussed in more detail in Section 5.1.

Two popular sets of heuristics are the 10 *Nielsen heuristics* and *Shneiderman's 8 Golden Rules of Interface Design* [47],[63]. Furthermore, Stone et. al present four psychological principles, plus three principles from experience, i.e. visibility, affordance and feedback [67, pp. 90 and 97]. All of these can be considered when designing UIs.

A more extensive variant of heuristic evaluation is the *guideline review*. In this context, guidelines differ from heuristics in that they are a considerably larger set of recommendations. The *Web Content Accessibility Guidelines (WCAG)* compiled by the *World Wide Web Consortium (W3C)* is probably the best-known guideline document and gives directions on how to make web-based UIs accessible to users with disabilities [35, p. 269].

While expert reviews of the target system were not conducted in the course of this work, the developer used and applied principles and guidelines to the best of his knowledge, while designing and implementing the DMARC aggregate report analysis software.

Detailed information about the above principles and guidelines, and how to conduct heuristic evaluation, can be found in textbooks and online resources such as: [67, pp. 525–537],[43] and [79].

3 | User-centered development of a DMARC report analysis tool

Chapter 2 presented work that is fundamental to the user-centered development of a DMARC aggregate report analysis tool. That is, related technologies in Section 2.1, and an abstract UCD process, including a discussion of some of the most relevant methods, and their applicability for the purpose of this work in Section 2.2. This chapter presents the concrete UCD approach as adapted and executed in the course of this work, and furthermore provides a specific evaluation of the employed methods.

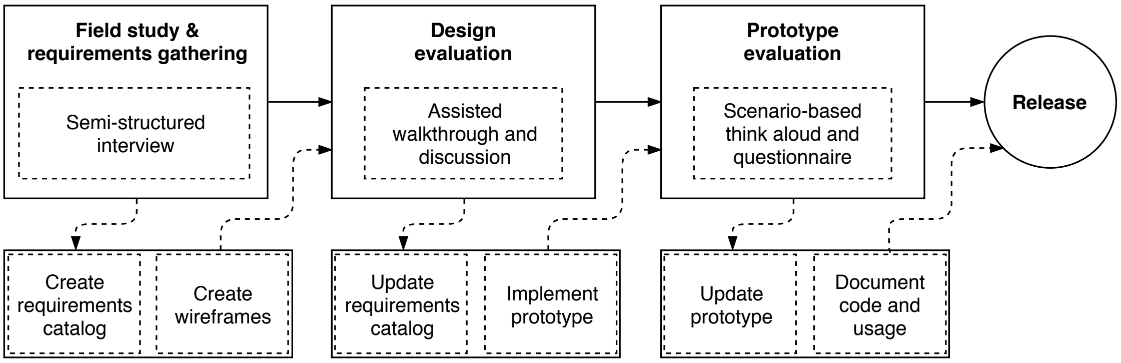

Field study & requirements gathering
Semi-structured interview
Design evaluation
Assisted walkthrough and discussion
Prototype evaluation
Scenario-based think aloud and questionnaire
Release
Create requirements catalog
Create wireframes
Update requirements catalog
Implement prototype
Update prototype
Document code and usage

Figure 3.1: Overview of the user-centered development of *DMARC viewer*

Figure 3.1 gives an overview of the applied three-phase user-centered development process, the methods that were used in each phase, and the artifacts created as a result of the user participation.

The first phase consisted of a combined field study and requirements-gathering session. Its primary objective was to gather insights into the target domain work practice, especially in regard to anti-spamming and anti-phishing, as well as other security-related techniques that mail administrators undertake in order to protect their mailing domains. A focus was put on the study participants' expertise and experience with DMARC and DMARC-related technology. Moreover, preliminary system ideas and analysis possibilities, as conceived by the author, were discussed collaboratively, using semi-structured interviews. The design implications, deduced from interpreting the transcripts and notes of the interview sessions, were transformed into an initial requirements catalog. This requirements catalog, in turn, served as a base for the design of the target system, using the mid-fidelity prototyping technique of wireframes.

In phase 2, design evaluation, the study participants were walked through the prepared wireframes, encouraging them to verbalize their thoughts, especially regarding the clarity of the interface and the adequacy of the proposed help section. The goal of this session was to match the preliminary expectations of the users with the proposed design. The notes taken from these user study sessions were used to make adjustments to the initial requirements catalog, and to implement a prototype of the system.

Eventually, the fully functional prototype of *DMARC viewer* was evaluated by the study participants in phase 3 of the development process. The goal was to assess the acceptance of the proposed system in terms of usefulness and usability. The prototype evaluation session consisted of a partially assisted, scenario-based usability evaluation, using the think aloud protocol. In addition to post-session interviews, the usability was formally tested using the SUS questionnaire. The insights gathered from the prototype evaluation were applied directly to the prototype, which, after adding comprehensive code and usage documentation, was released on *GitHub* and promoted on the *dmarc-ietf* mailing list.

Study participants and setting As outlined in Section 2.1.5, DMARC evaluation as well as report generation is performed at the receiving mail server. While the recipients of DMARC reports do not necessarily have to be the administrators of the purported sending mail server, they are likely to have the largest interest in those reports. Assuming that mail administrators have the responsibility of providing a secure and reputable infrastructure to their mail users, they may use the insights revealed by DMARC reports, in order to make decisions about the provisioning of fraud protection mechanisms. As a consequence, mail administrators were identified as the target users for a DMARC aggregate report analysis tool and thus ideal candidates for the related user studies.

The first attempt to recruit study participants was made in September 2014, when a delegation of the University of Vienna gave a talk at the *cert.at Stammtisch*[1]. In this talk preliminary results of the DMARC research collaboration between COSY and ZID were presented. At the end of the talk the development of a novel DMARC report analysis tool was announced by the author of this work, describing the proposed UCD development process and initial ideas, and inviting the audience to participate. From about 30 Austrian

[1] A regular meeting of the Austrian Computer Emergency Response Team (CERT).

mail administrators in the audience four accepted the invitation. After the talk the invitation to participate was re-issued on the *cert.at* mailing list, where another four mail administrators responded. When it came to coordinating appointments for the interview sessions, only two administrators were willing to actually participate. Another talk with the objective of introducing DMARC and recruiting participants was given at the Technische Universität (TU) Vienna in November 2014. None of the eight mail administrators from the audience were interested in participating in the development process, however, one of them offered to help find participants on the *LUGA* mailing list, with no results. Additionally, all postmasters of *.at*-domains on the *Alexa top 500* list[2] that had a DMARC DNS entry, as well as various academic and non-profit organizations, were inquired individually, to no avail. Eventually, five interested participants were recruited, one of them was already affiliated with the research project, two came from the *cert.at* talk, and two more were engaged via personal contacts.

It shall be noted that while assessing the right number of participants in a given user study has been a topic of lively discussion, various resources suggest that five participants are a reasonable number to start with [67, p. 457],[46].

The user studies and the corresponding incremental development of a first prototype of *DMARC viewer* were conducted over the course of a year between December 2014 and December 2015. All five participants were male between 38 and 48 years old. All but one had domain experience since the mid- or late-90s and varied in their expertise level towards mailing as well as their level of acquaintance with DMARC. All of them were interested in DMARC but at that point did not know how exactly they could or would use the technology. Three of the participants administered mailing infrastructures in academia and research, one came from the private sector and one worked for a political party. The sizes of the managed infrastructures ranged from 2,000 to 100,000 users. One participant was not able to participate in the last phase of the development process. All others participated throughout the entire process. Most of the studies were conducted at the participants' workplaces, accommodating their preferences. One participant, however, preferred to meet and conduct the sessions at a public café. Participants were given gifts, such as cookies or cakes, as a symbolic refund for their efforts.

The remainder of this chapter will outline each phase in more detail, describing the goals, used methods and results pertaining to a given phase. In addition, a meta-evaluation of the corresponding methodologies will be given.

[2] http://www.alexa.com/topsites

3.1 Phase 1 – Field study and requirements gathering

This section describes the first phase of the proposed user-centered development process, i.e. field study and requirements gathering.

3.1.1 Preface

As a precursor to the user studies conducted in this phase, existing DMARC report analysis software was examined in the course of the field study. The evaluation of existing tools helped the author of this work to develop preliminary design ideas that were discussed with the study participants in the course of the first interview sessions (see remainder of this section). Furthermore, wireframes, generated based on the examined UIs, were presented to the users for competitive usability analysis in the course of the second phase of the development process (see Section 3.2).

At the time of conducting the field study, two commercial DMARC tools supporting graphical DMARC report analysis stood out, that is *dmarcanalyzer*[3] and *dmarcian*[4]. For analysis, both tools used a graphical web-interface, offering similar feature sets to their customers. These features included the presentation of aggregate reports as expandable lists and tables, as well as the visualization of authentication activities using time lines and pie charts. Dmarcian, most notably, also performed a basic semantic analysis of the DMARC results, clustering messages and message senders into groups that were annotated using non-DMARC terminology. The used categories included: *Your Domains*, *Third Parties* and *Threat/Abuse/Other*. The classification, however, did not seem transparent to the author of this work.

In order to make the reports available for visualization, both, dmarcian and dmarcanalyzer, provided web file uploads. Moreover, both tools designated a unique RUA to their customers, which they could add to their corresponding DMARC DNS records, in order for the tools to immediately receive DMARC reports on behalf of the domain owner (see Section 2.1.5.1 for more details about DMARC DNS configuration). Both tools provided different pricing models depending on the extent of desired functionality and the expected email volume of the domain owner.

Another commercial DMARC report analysis tool that was available at that time was *postmarkapp*[5]. Postmarkapp, however, did not provide a graphical web-interface, but instead sent weekly digests of authentication results to its customers. Similar to the above-mentioned tools, postmarkapp provided a designated RUA to its users. In addition to the

[3] https://dmarcanalyzer.com
[4] https://dmarcian.com
[5] https://dmarc.postmarkapp.com

above standalone DMARC analysis applications, large email security service providers had incorporated DMARC report assessment into their services. These, however, were not available to the author of this work.[6,7]

3.1.2 Goals

Preliminary ideas for the proposed tool were gathered by examining existing DMARC report analysis tools and through informal discussions with colleagues. The evaluation of these preliminary idea was a primary goal of this phase. In addition, the conducted user studies aimed to assess the extent of DMARC-related knowledge among the study participants and whether the technology and a corresponding analysis software would integrate well into their work practice. The individual goals of the field study and requirement gathering user study are listed below.

- Gather knowledge about the practical work and work environment of mail administrators, focusing on security-related tasks
- Assess and evaluate participants' knowledge about and attitude towards DMARC-related technologies
- If necessary, make participants acquainted with DMARC and encourage its use
- Evaluate preliminary ideas regarding DMARC aggregate report analysis
- Collaboratively develop new ideas regarding DMARC aggregate report analysis

3.1.3 Method

As described in Section 2.2.3.2, semi-structured interviews provide a suitable method for early explorative studies in a UCD process. They give the researcher enough control to drive the direction of the study towards evaluation of existing ideas, while providing room for unexpected input from the study participants. Thus, semi-structured interviews were chosen over other field study methods, such as observations or diaries (see Section 2.2.3.1).

The interview sessions were structured in three main parts. First, gather information about the participants' related education and career, as well as their currently administered infrastructures; second, inquire about their daily work practice, covering thematic areas the researcher deemed important for the field of mail administration; and finally discuss

[6] https://agari.com
[7] https://returnpath.com

DMARC. This discussion was further subdivided into questions regarding the participants' experience with and their thoughts regarding DMARC, DMARC-related technologies and existing DMARC report analysis software. Subsequently, there were questions and scoring tasks on preliminarily developed usage scenarios for the proposed tool. Eventually, the interviews were concluded with a free association exercise, akin to card sorting (see Section 2.2.3.3), using predefined DMARC report parameters that the participants were asked to group into interesting analysis views. The corresponding interview guide can be found in Text listing 5.1 on page 83 of the appendix.

Prior to the interviews, the facilitator was asked to give an introduction to DMARC and DMARC-related topics, by four out of five participants. There was informed consent about audio recordings and the usage of the gathered data for the purpose of this work. Participants were also encouraged to participate in the subsequent phases of the development process.

Data analysis The audio recordings of the interviews were fully transcribed using the *Transcriptions* software.[8] In spite of being a cumbersome exercise, the verbatim transcription of hours of interview recordings allowed for an iterative, in-depth examination of the participants' statements, in order to lay the ground for tangible design implications, which were not apparent at the beginning of the analysis. The individual transcripts were then scanned repeatedly, extracting individual interesting data points and clustering them by the categories and sub-categories defined a priori. The categories can be seen in the interview guide, as titles for individual question packages (see Text listing 5.1 on page 83 of the appendix). The described method is akin to the basic method of qualitative data analysis as outlined in Section 2.2.3.2. Subsequently, a summary of the merged per-category and sub-category results of the interview sessions is given.

The categories included general work areas of mail administration. Of particular interest were tasks such as: reporting, as reports are the main focus of this work; training and learning, to understand if mail administrators theoretically had the opportunity to deploy new technologies such as DMARC; programming, to make decisions about the proposed software, intended as a contribution to the related open source community; security threats, to locate DMARC within existing tool sets; and DMARC-related technologies, to assess the amount of available expertise among the interviewed mail administrators.

[8] https://code.google.com/p/transcriptions/

3.1.4 Results

3.1.4.1 Mail administrator work practice

The following paragraphs outline mail administrators' day-to-day work practice, and specific tasks, deemed relevant to the purpose of this work, as revealed by the interviews.

General work practice One of the participants stated that once you had a running mail infrastructure the two key issues, especially for large infrastructures, were to keep unsolicited mail out, i.e. filtering incoming spam, and to assure that you "get rid of your own mail", i.e. domain owners would need to establish a good and robust global reputation, so that legitimate emails originating from the owned domain would not be rejected by other mail receivers. Apart from that, all participants mentioned regular maintenance work like debugging, patching, updating software and exchanging hardware. Additionally, all five participants agreed that a major part of their work consisted of processing requests and giving support to both co-workers and users.

Professional exchange and training The extent of professional exchange and related training, e.g. to learn about new technologies such as DMARC, varied largely from participant to participant. One of them regularly organized talks and workshops for fellow mail administrators and claimed that these events were very productive in terms of professional exchange, emphasizing the importance of after-work sessions, where colleagues would casually talk about prevailing issues. The other participants mentioned informal exchange with colleagues or friends working in the same domain. Additionally, all the participants named related media sources, where they would receive information about current topics and issues. A particularly popular source of information were mailing lists, blogs and the web in general. One participant further found *usenet newsgroups*, and another one professional journals, essential for their profession. Yet another participant said that learning about current topics would often happen *en passant* when researching for bug fixes. Only one of the five participants admitted that he did not have time to keep up-to-date.

Reporting While none of the participants mentioned that reporting would be demanded by their superiors, all five participants agreed that generating reports was one of the major tasks in mail administration. One participant elaborated that reporting was especially interesting on two time levels, that is, real-time reporting, in order to directly react to identified anomalies, and long-term reporting, in order to recognize trends and plan provisioning.

Programming Two out of five participants stated that programming was part of their work practice, since they used a lot of open source software, which they sometimes had to patch and customize according to their requirements. Among the preferred programming languages were *C*, *Perl*, *Bash* and *Python*. While one participant claimed that basically no programming language would discourage him from participating in an open source project, two other participants said that programming was not among their usual tasks.

3.1.4.2 Security threats

The following paragraphs provide a more specific analysis of how the participants claimed to identify and react to security threats.

Spam All participants agreed that the most important measure against spam was to reject as much and as soon as possible, by using rate limits and evaluating technical formalisms, regarding the used email protocols. One of the participants used the Spaghetti Western movie title *The Good, the Bad and the Ugly* as analogy for the mail traffic he had to deal with. *Ugly* would be mail that could be rejected right away; *bad* was mail that could be filtered through grey-listing, or, in a worst-case scenario, be scored by spam filters such as *SpamAssassin*[9], and *good* was mail that was okay. Two of four participants also mentioned that technical spam, i.e. spam that could be rejected before content-based spam detection, would be a constant but decreasingly relevant nuisance. A more eminent problem, because harder to detect, they said, was highly customized *spear phishing*, where malicious actors craft individual phishing mails, using legitimate free-mail accounts, in order to target a specific person. One of the five participants claimed that he had no spam problems at all, because he would use a *Cisco* anti-spam appliance that was doing all the security-related work for him.

Phishing All five participants said that phishing detection was part of spam detection. Some referred to additional publicly accessible databases, such as *sanesecurity* or *phishtank*. Two participants further pointed out that available detection patterns had to be adopted constantly, in order to detect and prevent highly customized *spear phishing*.

Account abuse All participants agreed that the prevailing security threat was account theft and account abuse, which they usually detected by semi-automatic mail log analysis related to account activity, or by user complaints. All participant had similar protocols they carried out when an account was identified as stolen, which included an immediate

[9] https://spamassassin.apache.org/

lock-down of the corresponding mailbox and requiring the user to manually change the password. Two participants further mentioned that they would hand out questionnaires to users, whose account had been stolen, in order to learn about how the identity was lost. They emphasized that users usually had no idea how the compromise had happened. The participating mail experts, however, assumed that in most cases it was the result of re-using potentially weak passwords, or of successful phishing attacks.

3.1.4.3 Usage of DMARC & DMARC-related technologies

The following paragraphs summarize the results of the participants' expertise and experience related to the fraud protection technologies that are fundamental to this work, i.e. SPF, DKIM, ADSP and DMARC, as well as existing commercial DMARC report analysis software.

SPF One participant pointed out that SPF was a useful and cheap technology, in terms of spoofing protection versus configurational complexity and computational resources. The participant, however, particularized that it was only safe to use restrictive SPF policies, if the served users would not use mailing lists, which, the participant stated, was an inadmissible assumption for large mailing infrastructures.

This claim was in line with the statements of most of the other participants, who also pointed out the inherent danger of rejected legitimate mail, due to SPF false positives. However, four out of five participants mentioned that they would use SPF passively, by positively adjusting received emails' spam scores, according to the availability of a corresponding SPF DNS entry. In order to reduce false positives for relayed emails, two of the five participants said that they would perform sender rewriting. One participant was only vaguely familiar with SPF (see Section 2.1.3 for more details about SPF).

DKIM Only one out of five participants used DKIM. According to the participant, DKIM would be preferable over SPF, due to the absence of policy enforcement. The participant found DKIM especially suited to authenticate legitimate inner-domain traffic, because the existence of a DKIM signature could be anticipated. All other participants did not use DKIM, either due to the difficulty of managing cryptographic keys, or the lack of related expertise.

ADSP Only one out of five participants was familiar with ADSP and had deployed it for a short period, but without seeing its benefits. The participant claimed that akin to DMARC's policy mechanism, ADSP was mainly interesting for brand protection, which, as he stated, was not a major concern in terms of phishing prevention. Modern phishers, the participant elaborated, would instead use non-spoofed sender addresses that sounded similar to the impersonated domain, referred to as *cousin domains* and thereby circumvent spoofing-prevention mechanisms, such as ADSP or DMARC. However, the participant did acknowledge that DMARC had a convincing advantage over ADSP, namely inter-domain reporting.

DMARC Two out of five participants had deployed DMARC in their infrastructures. However, both of them were using it in monitor mode. That is, their corresponding DMARC DNS records specified a *none*-policy and a RUA address (see Section 2.1.5.1). One of the three remaining participants stated that although he found DMARC interesting, he would not use it for his infrastructure. The other participants had not yet thought about using DMARC. The essence of all participants' attitudes towards DMARC was that it was primarily interesting for its reports, more than as a policy enforcement mechanism.

Existing DMARC report analysis tools The two participants, who were already receiving DMARC aggregate reports for their administered domains, mentioned that they had also experimented with one of the existing commercial DMARC report analysis tools. Both of them underlined the aesthetic appeal and the good structure of its web-interface. Moreover, they claimed that it was especially helpful for showing problems in the own DMARC configuration. One of them particularly pointed out the appeal of a threat map, visualizing the origin of illegitimate mail, based on IP addresses. On the other hand, one of the two participants criticized that some presented results were not clear, which he attributed to the complexity of the DMARC specification. Furthermore, all participants agreed that it was uncertain whether their superiors would grant the use of commercial analysis software, which requires the transfer of DMARC reports to a third party.

3.1.4.4 Design implications

The following paragraphs summarize the results from the user study, mapping the participants' implicitly and explicitly stated ideas and wishes, regarding DMARC aggregate report analysis software, to the design of such a system. Design implications were deduced by brain storming on the results of the studies as described above and especially by considering user statements from discussing preliminary ideas for the proposed system at the end of the user study sessions.

Education One of the participants pointed out that DMARC analysis software to him was especially interesting in the beginning, when he started to explore DMARC for the first time, as it helped to better understand the underlying technology. Hence, the proposed system should, in part, focus on users who are less familiar with DMARC and the related technologies. This requirement was underscored by the general lack of familiarity in regard to DMARC that was observed among the study participants. Moreover, as a consequence of the absence of tangible user requirements brought forth by the interviewed mailing experts at the time of the user studies, it was decided to design a system that would be flexible enough to easily extend the analysis facilities, once more knowledge about the use and usefulness of DMARC was available. In addition, a composable analysis software, where the user can generate and modify custom views on the DMARC aggregate report data, may also allow to playfully examine the capabilities of DMARC. However, in order to accommodate the DMARC novice, it was deemed important to provide predefined default analyses, based on which the user can create custom analyses.

In addition, one of the study participants pointed out the importance of a help section that explains frequently asked aspects of the DMARC technology. Another study participant mentioned that it would be helpful to detect misconfiguration in their own DMARC setup, when using the tool. This could be realized by scanning the received and sent reports for anomalies.

Documentation and reporting The interviews further revealed that documentation as well as both real-time and long-term reporting were all important features to have during mail administration. Given that DMARC aggregate reports are usually received once a day, they are not particularly suited for real-time feedback. Hence, the tool will focus on long-term documentation, e.g. to observe trends. One participant, however, pointed out that it was equally important to be able to see detailed snapshots of the data from a given time.

Although the geographic origin of mails was assessed as being of low interest, several participants stated that they found the threat map, as implemented in existing DMARC analysis software, especially appealing.

Usability All participants agreed on the importance of usability and aesthetic aspects in the system design. That is, a well-structured and visually appealing interface, which would be self-explanatory and easy to use. In terms of structure, it was mentioned that analyses should provide multiple degrees of granularity. For instance, a general overview of the data that also provides ways to manually zoom in on the data and apply filters as desired. Furthermore, automation was found to be very important, in particular that reports could be fed into the software's database automatically. As alternative usage, an

API was suggested, where the reports could be retrieved from the software's database, in order to process them with other statistics tools.

Collaboration In principle, the participants welcomed the idea of collaborative DMARC report analysis and also had no major privacy concerns regarding the exchange of DMARC reports with other members of the ACOnet. Nevertheless, the benefits of collaboration were not generally evident. One of the participants suggested postponing the investigation of collaboration to later stages of the development, and to first focus on the use and usefulness of non-collaborative DMARC report analysis. Another participant claimed that collaboration would only be interesting for really big domains. Additionally, a participant pointed out that, although DMARC aggregate reports would not reveal any user sensitive data, the media could easily make an affair out of the sharing of such reports. All participants agreed that, if the tool offered collaborative analysis, they would have to first discuss the juristic aspects of data sharing with their superiors.

The design implications, as outlined above, led to the conception of a DMARC aggregate report analysis software, that will be described subsequently.

3.1.4.5 System overview

The proposed system consists of a database that stores DMARC aggregate reports, a utility that parses reports into the database and a web-interface that makes report data and its analysis available to the user. The target users are mailing administrators with a particular focus on DMARC novices. The tool can be used for incoming reports as well as outgoing reports.[10] The entry point to the system's Graphical User Interface (GUI) provides a general overview and statistics about all stored data, as well as anomaly alerts that are generated by periodic scans of the stored data. In addition, detailed analysis can be performed by using predefined analysis views or by creating custom analysis views. Analysis views are composable, re-usable snapshots of particular DMARC aggregate report data, based on user-configurable filter sets. Each analysis view receives a name and a description, and can be selected from a list of analysis views. The visualization type of an analysis view is one of world map, time line or table. Filter sets pertaining to an analysis view are related to the variables in DMARC aggregate reports. A set of filters is used to retrieve the corresponding data set from the stored reports. The use of multiple filter sets in a given analysis view allows the user to compare different data sets, e.g. as multiple lines in a time line diagram. To distinguish individual filter sets they can be assigned a label

[10] Incoming reports are received by the domain owner. They contain DMARC evaluation results about emails that were sent with the domain owner's domain. Outgoing reports, on the other hand, are generated by the domain owner, based on DMARC evaluation results pertaining to received emails.

and a color. Sets of filters can be stored as templates so that they can be re-used in other analysis views. Views and their associated filter sets can be created and managed on a separate view management page.

In order to enable collaborative analysis, the system provides individual user accounts that are isolated from each other by default. Only upon inter-user negotiation of sharing agreements, which can be performed using the system's web-interface, are reports accessible across user boundaries.

The system's usage and the intricacies of DMARC and its related technologies are described in a comprehensive help section, also available via the web-interface. A full requirements catalog can be found in Text listing 5.2 on page 87 of the appendix.

3.1.4.6 GUI design

The requirements catalog was used to sketch a GUI of the proposed system. Mid-fidelity prototypes, i.e. wireframes, were drawn using the open source prototyping tool *pencil.*[11] The wireframes of the proposed system can be found in Figures 5.1–5.13 on page 89ff of the appendix.

As a comparison of the proposed system's GUI with the GUIs of existing DMARC aggregate report analysis software, which was requested by a study participant, wireframes of three available commercial tools, *dmarcian*[12], *dmarcanalyzer*[13] and *easy solutions*[14], were created. The author of this work preferred the use of wireframes over screen shots, in order to only compare structure and basic functionality of these tools and avoid bias regarding the fidelity of the compared interfaces. The wireframes, based on the appearance of the corresponding tools as per May 2015, can be found in Figures 5.14–5.22 on page 102ff of the appendix.

3.1.4.7 Deployment variants

Initial explorations of collaborative analysis in the scope of a trusted network, such as the ACOnet, revealed mixed results. Generally speaking, the study participants preferred to determine the use and usefulness of non-collaborative DMARC aggregate report analysis first, before further exploring collaboration. As a discussion aid for the subsequent user study sessions, deployment diagrams of three viable deployment scenarios were created. Diagrams for *simple deployment* (Figure 5.23 on page 111), *centralized deployment*

[11] http://pencil.evolus.vn/

[12] https://dmarcian.com

[13] https://dmarcanalyzer.com

[14] https://www.easysol.net/

(Figure 5.24 on page 112) and *federated deployment* (Figure 5.25 on page 113) can be found in the appendix. The three scenarios were evaluated in terms of trade-offs between configurational complexity for the user, data sovereignty and collaboration possibilities.

In the case of simple deployment, the proposed system would be deployed by the user of the system. While simple deployment would require the user to provide the corresponding infrastructure in order to host the system, this scenario would give full data sovereignty to the user. Reports would still be able to be exchanged manually with other domain owners, however, in-system collaboration would not be available.

In a centralized deployment, the system would be hosted on an external infrastructure, e.g. a trusted infrastructure such as the ACOnet. Unlike above, in this scenario the system could provide facilities to collaboratively analyze DMARC aggregate reports. Multiple domain owners could create accounts within the system in order to access their report data either isolated from each other or with additional collaboration. Furthermore, in order to provide report parsing automation, the trusted deployer could provide two mailboxes for each user, designating corresponding email addresses to the users. As a consequence the user could specify one of the report addresses as RUA in the corresponding DMARC DNS record, so that incoming reports are automatically received by the system and associated with the user account. As for outgoing reports, users could add a hook to their report generation routines, in order to automatically send a copy of generated reports using the second designated email address. While a central deployment scenario would require less configuration effort for the individual user, and provide an easy way to collaboratively analyze reports, users would have to make their reports available to a third party.

As a third option, federative deployment combines a local deployment, where data resides within the domain owner's infrastructure, with the collaboration benefits of the centralized deployment. In a federative deployment scenario, each user could be a centralized deployer, inviting other trusted users to send their reports, by e.g. providing two mailboxes as described above for each federated collaborator. However, this scenario would also require an increased effort in terms of configuration and maintenance to the federated deployer.

3.1.5 Lessons learned

The field study and requirements elicitation phase, as presented above, was by far the most laborious part of the proposed UCD process. Both, in terms of preparation and execution of the user studies, as well as data interpretation and identification of tangible design implications. This may be attributed to a mutual lack of related knowledge, regarding the problem domain, but also to missing experience in terms of UCD techniques on the researcher side. When designing the study, the author of this work hoped to receive comprehensive user input, which could directly be mapped to the design of the target system. While the anecdotal evidence from the proposed study suggests that its results were indeed determining for the design of the proposed system, such expectations are unnecessarily ambitious. Especially in a problem domain, where both the researcher and the participants are largely oblivious to the use and usefulness of the investigated technology, the initial phase of user involvement can be an important barrier to entry. This may prevent researchers from moving forward in the project, or cause them to refrain from field studies altogether. Therefore, the author recommends keeping the exploratory phase of the UCD study rather open, and to not expect too many immediate design implications. Instead, it should be seen as an opportunity to find common ground between the developer and the study participants for further user sessions. In terms of methods, semi-structured interviews showed to be appropriate for such goals, but also experimental requirements elicitation techniques may be used early in the UCD process. However, such methods might not work as expected, in terms of revealing relevant quantitative data. Instead, as shown in this study, the employed association exercises and scoring tasks were useful in a sense that they triggered insightful discussions. Thus, it can be recommended that the study facilitator provide a certain degree of flexibility and be open to deviations from the protocol. Finally, the duration of the individual user sessions should be kept to a necessary minimum, especially when investigating a novel and complicated technology. It is difficult to make quantifiable claims in terms of an appropriate duration, as the sessions conducted in the course of this phase varied between 30 minutes and two hours, depending on the amount of investment of the individual participant. However, fatigue was observed among all participants. Hence, it is advantageous to thoroughly review the interview guide prior to the study, in order to identify less relevant parts that may be omitted, or that are suited for separate techniques of data collection, e.g. questionnaires, which users can fill out without supervision.

3.2 Phase 2 – Design evaluation

The main objective of the second phase of the proposed UCD process was to evaluate whether the requirements, based on the results from the preceding participatory field study, were correctly mapped to the proposed design. The remainder of this section provides goals, methods and results as well as a meta-evaluation of the conducted user-centered design evaluation sessions.

3.2.1 Goals

Two specific goals, based on the results of the prior study, were: one, to assess whether the system appeared to be suited for DMARC novices; and two, to re-evaluate the idea of collaborative analysis. A list of the general study goals is presented subsequently.

- Evaluate graphical user interface using mid-fidelity prototypes of proposed software
- Compare functionality and structure using mid-fidelity prototypes of existing software
- Remove unnecessary functionality
- Add missing functionality
- Evaluate and improve help section
- Decide on collaboration and deployment variants

3.2.2 Method

The study consisted of four parts. The first part was an assisted walkthrough over the core analysis pages of the proposed system, using the previously created wireframes (see Figures 5.1–5.8 on page 89ff of the appendix).

Participants were asked to walk through the different pages, verbalizing their thoughts, in regard to their expectations and questions. If they had questions, they were asked to look up the help page and try to find the corresponding answers there. Additionally, they were asked to point out what they found particularly positive or negative. After the first walkthrough, the participants were asked general questions about how they liked specific parts of the proposed interface.

The second part consisted of a discussion about different deployment scenarios for the tool, using the diagrams for *simple deployment* (Figure 5.23 on page 111), *centralized deployment* (Figure 5.24 on page 112) and *federated deployment* (Figure 5.25 on page 113), as found in the appendix. The deployment scenarios were presented in regard to their trade-offs between configurational complexity for the user, data sovereignty and collaboration possibilities.

Following the discussion about deployment and collaborative analysis, another assisted walkthrough was carried out, using wireframes specific to collaborative analysis (see Figures 5.9–5.13 on page 97ff of the appendix). The procedure was akin to the first walkthrough, although it tended more towards a discussion.

The last part consisted of a competitive usability analysis of related DMARC report analysis software. In order to provide a fair comparison, mid-fidelity prototypes of the related tools were provided (see Figures 5.14–5.22 on page 102ff of the appendix). In this part the participants were asked to walk through the wireframes of the related tools and point out functionality that they particularly liked and had not seen in the wireframes presented earlier.

The related study guide can be found in Text listing 5.3 on page 114 of the appendix. The employed methods are described in more detail in Section 2.2.3.3.

Data analysis The recordings from the design evaluation sessions were intelligently transcribed and enriched with the extensive notes, taken during the test sessions. Intelligent transcription means that the recordings were not transcribed in verbatim — as in the prior phase (see Section 3.1.3) — but rather scanned for both positive and negative critique. Participants tended to only point out positive aspects explicitly. Negative aspects, however, were noted by the study facilitator, by observing the participants' interaction with UI elements. Positive items were seen as validation of the proposed design. Negative items, together with suggestions of the participants, were used to re-phrase existing requirements and to add new ones.

3.2.3 Results

The feedback to the design propositions was generally positive. Especially, the core of the tool, that is re-usable analysis views with associated filter sets, was appreciatively accepted by all study participants. Moreover, it was pointed out that the composition or customization of analysis views would be helpful to playfully investigate the DMARC technology. In that regard, one of the participants coined the term "learning by viewing". However, all participants underscored the importance of descriptive predefined analysis views, which would serve as basis for individual exploration. The participants further agreed that the provision of predefined analysis views, together with the possibility to clone views, would eliminate the need for view and filter templates, which, in turn, would help to keep the interface simpler. Additionally, all participants were satisfied by the proposed analysis view types (world map, time line and table) and agreed that they would sufficiently visualize the reports. One participant said that he would actually only need a time line and a table and referred to the world map as "management porn".

The complexity of the DMARC specification was a re-occurring issue, observed among all study participants. In some cases, core principles about DMARC's mode of operation had to be explained throughout the test session. The participants rarely considered the help section, although its usage was explicitly requested by the session protocol. Instead, several users pointed out that they would prefer contextual help, especially regarding DMARC-specific terminology in the analysis view editor.

The walkthroughs also revealed several issues with non-DMARC-specific terms. For instance, three out of five participants found the *Customize view* button in the deep analysis section confusing. They assumed that the button would change visual settings, where it was actually intended to access the analysis view editor. Also, the difference between *incoming* and *outgoing* reports was unclear to all users. The terminology was introduced by the author of this work, in order to distinguish reports the user receives, and which existing analysis tools usually focus on, from reports about DMARC evaluation results that are generated by the user, which may also provide relevant insights, e.g. about issues in the user's DMARC evaluation setup. As a consequence, the terms identified as ambiguous or unclear were marked for revision or as candidates for contextual help, to be re-evaluated in the final prototype evaluation sessions.

In regard to the deployment scenarios, all participants clearly voted in favor of the simple self-deployment approach. Collaboration, as initially planned was generally rejected. All participants agreed that the benefits of a collaborative analysis would very likely not compensate for the increased configurational effort of a federated system, where the administrators would have to make adjustments to their mailing infrastructure. The centralized approach, on the other hand would carry too many risks regarding data privacy, even if the system were to be deployed in a trusted network. The participants also stated that they would prefer for the developer to put the additional implementation effort into the general analysis features, rather than into user handling or sharing negotiation facilities, which would be required by a collaborative system.

In three out of five cases the competitive usability analysis was entirely or largely skipped, either due to the participant's time constraints or apparent fatigue. The remaining two participants found that the proposed tool had sufficiently adapted those features, they found appealing in the existing DMARC analysis tools. One of them actually preferred the less cluttered UI of the proposed system.

A list of specific UI enhancements and requirement modifications, as revealed by the design evaluation study, can be found in Text listing 5.4 on page 115 of the appendix.

3.2.4 Lessons learned

The user sessions pertaining to the second phase of the conducted UCD process were substantially easier, in terms of execution and interpretation, than the ones in the previous study. As opposed to the field study, which dealt with a vast and uncharted problem domain, the second phase had already narrowed down that problem domain to a tangible design proposition. This also helped the study participants to give concrete feedback.

The employed custom method of assisted walkthroughs, using wireframes, diagrams and post-session discussions, proved to reveal rich insights into the tentative usability of the system. However, similar to the experience from the first study, most of the participants showed fatigue after approximately two-thirds of the session, which took between 30 and 90 minutes. In cases where participants became too fatigued, the competitive usability analysis, using wireframes of existing DMARC report analysis software, was entirely or partially omitted. While the creation of those wireframes may have been of limited value for the user sessions, they were helpful for the developer to draw design inspirations.

3.3 Phase 3 – Prototype evaluation

The final study of the custom three-phase UCD process was performed using a functional web-based prototype of the proposed DMARC aggregate report analysis tool. The software was developed and evaluated based on the insights from the prior two studies. The main objective of this study was to assess its user acceptance. Similar to the preceding two sections, this section will present the goals, employed methods, study results and lessons learned, pertaining to the participatory prototype evaluation study, conducted in the course of this work. The used prototype, including minor changes as suggested by the results of this study, is documented in Chapter 4.

3.3.1 Goals

The goals of the study included usefulness and usability aspects as outlined below.

- Evaluate usefulness in terms of analyzing the managed domain and learning the DMARC technology
- Evaluate usability in terms of appeal and ease of use
- Find additional items required for help section
- Find additional required contextual help items

3.3.2 Method

The prototype evaluation user study consisted of a partially assisted, scenario-based usability evaluation of the functional prototype employing the think aloud protocol, and was concluded by a satisfaction questionnaire and a post-session discussion. The corresponding test session guide and SUS questionnaire can be found in Text listing 5.5 and Table 5.1 on pages 116 and 118 of the appendix.

Before starting the individual user test sessions the facilitator gave a five to ten minute primer, refreshing the participants' minds, in terms of DMARC technology, and briefly described the goals and envisioned test protocol.

The participants were asked to act in the role of the mail administrator of the University of Vienna, using the proposed prototype to explore real ingoing as well as outgoing DMARC aggregate reports, for the domains *univie.ac.at* and *unet.univie.ac.at*[15], which had been collected in an anonymized fashion since the beginning of 2015.

Three scenarios that aimed at covering the entire UI of the proposed system had been prepared (see session guide in the appendix). All scenarios involved the creation and usage of custom analysis views. The first scenario was used as a confidence builder, where the task was to get acquainted with the interface, exploring the different pages and a predefined view, and to make various modifications to that predefined view, using the view editor. The participants were urged to accomplish the given tasks while thinking loudly and describing their corresponding approach and thoughts. Additionally, the participants were encouraged to point out aspects they perceived as particularly positive or negative about the tool's functionality and usability. After having carried out a given task, the participants were further asked to verbalize three questions, whose responses could have helped them to solve the task. At the end of the session, following an off-record debrief, the participants were invited to fill out a SUS questionnaire.

All participants carried out the tasks using the developer's *MacBook Pro 13-inch, Mid 2009*, running *OSX 10.9.5* and *Chrome Version 47.0.2526*. Participants were given the choice to use the built-in trackpad or an external optical mouse. They all preferred the use of the trackpad.

Given that the insights of the sessions were largely based on observations made by the facilitator, audio recordings were omitted. Instead, the facilitator took notes of what the participants remarked positively or negatively and where they had problems.

3.3.3 Results

Four out of five participants were able to complete all proposed tasks with varying degrees of assistance required. The completion time for each task ranged from 15 to 40 minutes, depending on the task and the participant. While a fast completion time might be interpreted as an indicator of good usability, this was not the case in this study.

[15] The University of Vienna uses different domains for their employees (*univie.ac.at*) and students (*unet.univie.ac.at*).

On the contrary, participants, who took more time to complete the tasks, also required less assistance than participants who completed the tasks in shorter time. In order to meaningfully compare completion time, assistance would have to be either omitted entirely, or distributed equally among all participants. However, as in the prior studies, qualitative data, such as user comments, was preferred over quantitative. In one case, where the tasks could not be completed, the participant was still able to use the tool and perform some of the requested actions, such as analysis view creation, duplication, as well as basic interpretation of the data visualizations. In this case the facilitator decided that it would be more insightful to let the participant freely explore the tool and record his comments, instead of assisting him to perform the given tasks.

The overall feedback was exceedingly positive. The participants found the user interface to be visually appealing, and that it provided a very pleasant menu navigation and page structure.

The overview page was perceived as a good starting point, however, the participants felt little need to go back, whereas the deep analysis page, especially the time line chart, was found highly interesting and informative.

The table of the deep analysis page was both criticized for being too long and having too much redundant information, but participants also liked the possibility to go into detail and look for specific domains and authentication results, as filtered by the created analysis views. Participants further appreciated the implemented convenience features in regard to analysis view creation and management, such as drag-and-drop support for analysis view sorting, view and filter set cloning, and auto-completion in the filter input fields of the view editor. Especially the latter was experienced as a valuable learning aid, in terms of the DMARC specification.

One participant pointed out that it was actually fun to create analysis views. Furthermore, all participants positively remarked on the provided user feedback, such as loading wheels, feedback messages upon view modification, and form validation feedback. One participant, however, pointed out that the form validation feedback was unfit for users with vision impairment, as invalid input fields were marked in a red color, with a small warning text below the corresponding input field.

Another issue that was observed repeatedly was that participants were kept from scrolling all the way down on the main analysis page. This was due to the first diagram, i.e the world map, stretching over the entire width of the monitor, capturing the scroll gesture, as soon as the cursor moved over the diagram, in order to zoom in and out of the map. As a consequence, none of the participants were able, at first, to see the other available data visualizations, i.e time line and table.

Another question that was asked by several participants, was whether the time line chart would be zoom-able and what the purpose of the mini-chart below the time line was.

The usage of the mini-chart in order to zoom in on the time axis of the time line chart was explained in a description text, located above the chart. However, it was observed that these description texts, of which there were several throughout the pages, were generally ignored by the participants. In reaction to pointing participants at the help texts, they claimed that clearly identifiable *tooltips*, revealing contextual help on hovering or clicking, would attract more attention. The need for contextual help, as it had already been revealed by the prior stage of user testing, was again prevalent, especially on the analysis view editor page. Additionally, it was suggested to add a diagram to the help sections, depicting the entire DMARC work flow.

One participant re-emphasized the significant complexity of the DMARC specification, but also was very optimistic about the proposed software, helping to get and stay acquainted with the technology.

A list of specific fixes and enhancements, gathered from the participants' statements and from observing their interaction with the proposed prototype can be found in Text listing 5.6 on page 119 of the appendix.

SUS results The total result of the system usability scale questionnaires was calculated as proposed by its author [8] and yielded a result of 71.5, which is the mean of the score of each contribution. It has a margin of error of 10.75 at a confidence level of 95%. Bangor et al. have associated adjectives — *worst imaginable*, *awful*, *poor*, *OK*, *good*, *excellent* and *best imaginable* — with SUS scores. Scores around 71.4 are given the adjective *good*. It may be noted that the total mean SUS score of $1,433$ web-interfaces, evaluated in the course of the cited work, was 68.2 [3].

3.3.4 Lessons learned

The prototype testing sessions were attended with notable enthusiasm by all participating users, more than the previous participatory studies, i.e. field study and design evaluation. This may be attributed to the quality of the implemented system, but also to the nature of the study.

In contrast to the previous studies, where the participants — as domain experts — were required to contribute a large amount of creative ideation, the sessions pertaining to the prototype evaluation study provided clearly defined tasks, which the participants — as users — could carry out mostly intuitively.

A few pitfalls, however, were identified. Given lower facilitator interference than in the previous studies, it was very important to repeatedly encourage the study participants to loudly communicate their thoughts, regarding positive or negative incidents. The right amount of facilitator interference remains a topic for discussion. For this work, it was the intention of the researcher to find a balance between enough assistance to help the participants to cover all relevant parts of the UI and prevent user frustration, but not too much so that issues would be masqueraded by the facilitator’s comments.

In some of the test sessions, it was observed that the participants needed to be reminded repeatedly that not being able to complete a given task was a failure of the UI, rather than their own.

Moreover, while the employed SUS questionnaires revealed a quantifiable acceptance measure, they were also intended to evoke additional qualitative user comments, especially in case of extreme scores. However, the given ratings aligned well with the predominantly positive feedback gathered during the test session.

4 Prototypical realization

This chapter introduces the latest version of the prototype of *DMARC viewer*, a freely available, self-deployable, open source web tool, to analyze DMARC aggregate reports. The proposed software is a direct result of the custom iterative and incremental UCD process described in the previous chapter. Moreover, feature requests and contributions from the related open source community have been integrated since its release in May 2018. The remainder of this chapter will give an overview of its functionalities mostly from a UI perspective. The employed technologies, the release process, and the activities following the release will be outlined as well.

4.1 User interface & functionality

DMARC viewer allows the user to parse and visually analyze incoming and outgoing DMARC aggregate reports, providing unique insights into how mailing domains are used and abused. Moreover, *DMARC viewer* lets the user create and store custom analysis views to visualize reports filtered by any desired report aspect. The general usage is described as follows:

1. **Deploy *DMARC viewer*** — Documentation for containerized and non-containerized deployment is available on the source code repository of the software.[1]

2. **Parse DMARC aggregate reports** — A custom command line interface is provided to import incoming and outgoing DMARC aggregate reports into the tool's database and to perform GeoIP lookups.

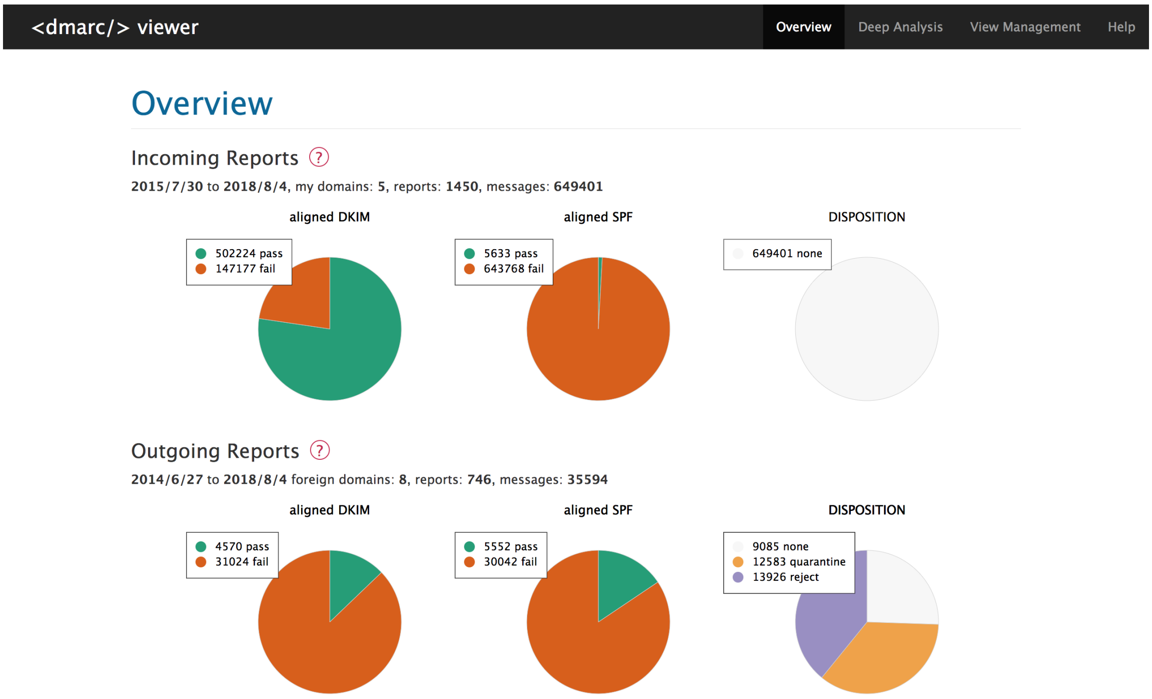
3. **Create analysis views** — Basic data interpretation is available on the *overview* page, however, the core analysis functionality is provided by composable *analysis views*, which can be created in the *analysis view editor*, or imported using a custom command line interface.

[1] https://github.com/dmarc-viewer/dmarc-viewer

4. **Analyze reports on deep analysis page** — The report data, retrieved from the database by applying the filters associated with a given analysis view, is visualized on the *deep analysis* page as a world map, a time line chart and a dynamic report record table.

The individual pages of *DMARC viewer* are described in detail subsequently.

4.1.1 Overview



Overview

Incoming Reports ?

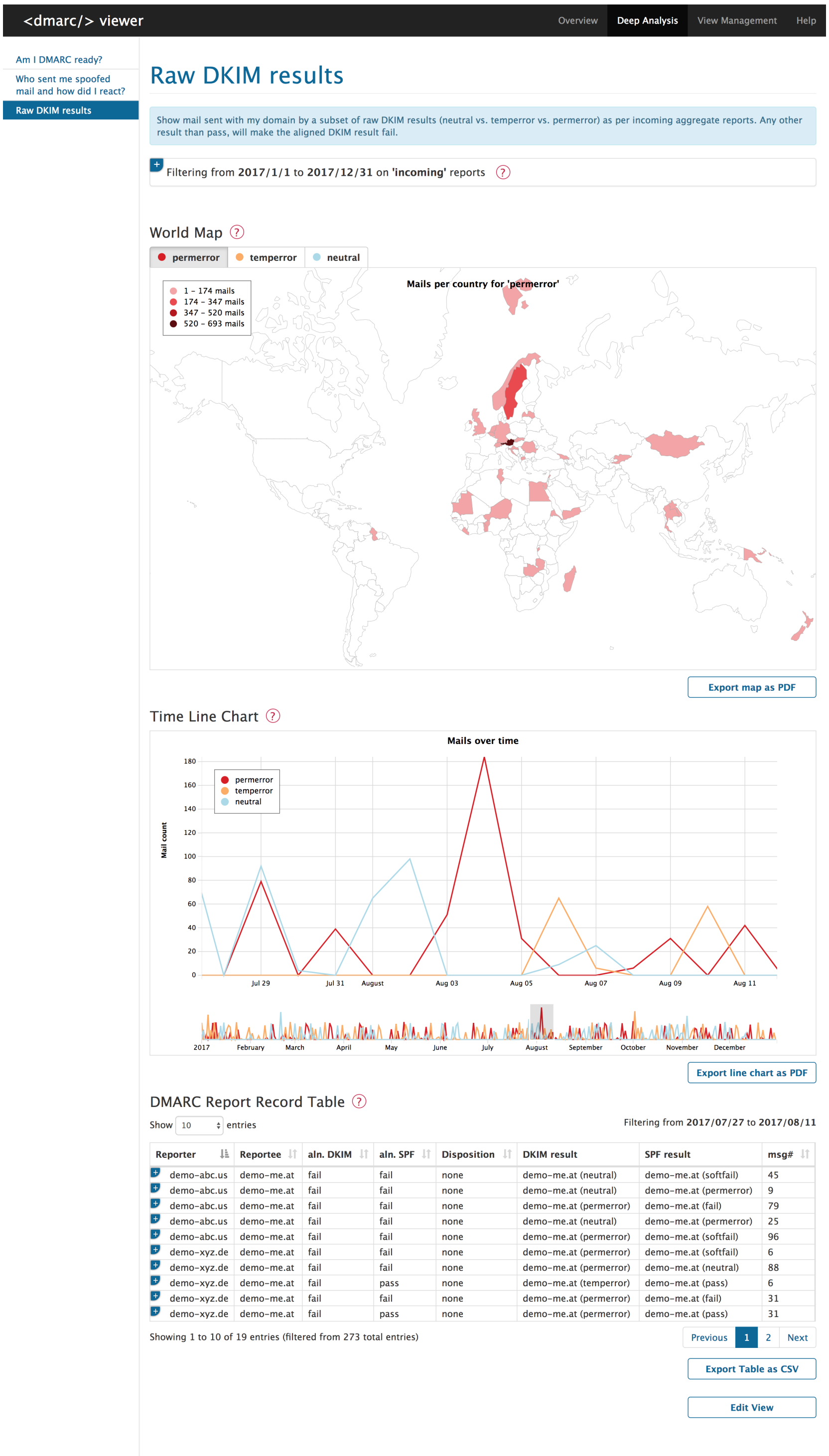
2015/7/30 to 2018/8/4, my domains: 5, reports: 1450, messages: 649401

Outgoing Reports ?

2014/6/27 to 2018/8/4 foreign domains: 8, reports: 746, messages: 35594

Figure 4.1: *DMARC viewer* – Overview

The *overview* page provides a starting point for further data exploration, showing general information about all incoming and outgoing reports stored in the database. For both types it displays the date range covered by all available reports, as well as the amounts of corresponding report receiver domains, reports and messages. It also shows pie charts, comparing the total amount of messages that failed or passed DMARC authentication and how they were treated. An example overview page can be seen in Figure 4.1.



Raw DKIM results

Show mail sent with my domain by a subset of raw DKIM results (neutral vs. temperror vs. permerror) as per incoming aggregate reports. Any other result than pass, will make the aligned DKIM result fail.

Filtering from **2017/1/1** to **2017/12/31** on **'incoming'** reports

World Map

Time Line Chart

DMARC Report Record Table

Show 10 entries

Filtering from **2017/07/27** to **2017/08/11**

Reporter	Reportee	aln. DKIM	aln. SPF	Disposition	DKIM result	SPF result	msg#
demo-abc.us	demo-me.at	fail	fail	none	demo-me.at (neutral)	demo-me.at (softfail)	45
demo-abc.us	demo-me.at	fail	fail	none	demo-me.at (neutral)	demo-me.at (permerror)	9
demo-abc.us	demo-me.at	fail	fail	none	demo-me.at (permerror)	demo-me.at (fail)	79
demo-abc.us	demo-me.at	fail	fail	none	demo-me.at (neutral)	demo-me.at (permerror)	25
demo-abc.us	demo-me.at	fail	fail	none	demo-me.at (permerror)	demo-me.at (softfail)	96
demo-xyz.de	demo-me.at	fail	fail	none	demo-me.at (permerror)	demo-me.at (softfail)	6
demo-xyz.de	demo-me.at	fail	fail	none	demo-me.at (permerror)	demo-me.at (neutral)	88
demo-xyz.de	demo-me.at	fail	pass	none	demo-me.at (temperror)	demo-me.at (pass)	6
demo-xyz.de	demo-me.at	fail	fail	none	demo-me.at (permerror)	demo-me.at (fail)	31
demo-xyz.de	demo-me.at	fail	pass	none	demo-me.at (permerror)	demo-me.at (pass)	31

Showing 1 to 10 of 19 entries (filtered from 273 total entries)

Figure 4.2: *DMARC viewer* – Deep analysis

4.1.2 Deep analysis

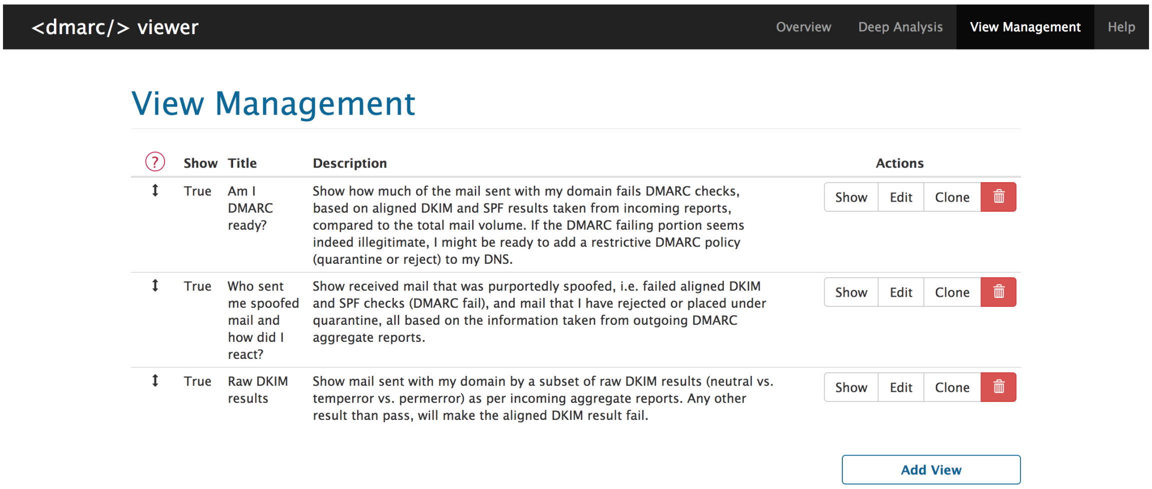
The so-called *deep analysis* page, depicted in Figure 4.2, provides the core analysis functionality of *DMARC viewer*. It gives access to all predefined and user-defined analysis views. When opening an analysis view, associated filters are applied on the available report data query, in order to present the corresponding parts to the user. Moreover, each analysis view may define multiple filter sets, in order to compare different aspects of the data to each other, most notably, as different lines in the time line chart. Visualizations of the report data include a world map, showing where evaluated mail originated from; a time line, showing when mail was evaluated; and a dynamic table, providing detailed information about the related reports. By default, all available visualization types are displayed in a given analysis view. However, a user can choose to enable or disable different view types as desired. Besides a custom view title and description, the list of all applied filters, binding specific report data to an analysis view, is presented at the top of the analysis view page. The three basic visualization types are described in more detail in the subsequent paragraphs.

World map The world map is visualized as choropleth. As it was found unsuited to display multiple filter sets in one given map, individual choropleths for each filter set are generated and accessible via a tab menu on top of the map canvas. The countries on the map are shaded according to the amount of DMARC-evaluated mail matching the applied filters of the corresponding filter set. The color range is dynamically created around the filter set's configured color, by converting the base color to the Hue, Saturation, Luminance (HSL) color space and stepping the lightness value as suggested in color guidelines for mapping and visualization by C. Brewer [7]. The individual countries on a map reveal its name and per-country message emergence, when hovering over the map. A given map may be downloaded as PDF document.

Time line chart The time line chart plots the amount of DMARC-evaluated mail on a time axis. It benefits most notably from the provision of multiple filter sets, as it shows one time line per filter set, using the configured colors. Additionally, the time line chart provides a mini map that can be used to zoom on the time axis, which also narrows down the displayed rows in the DMARC report record table. The time line chart can be downloaded as PDF document.

DMARC report record table The report record table shows all distinct DMARC report records matched by the combined filter sets pertaining to a given analysis view. The table may be used to explore the matched reports in detail. The columns of the table represent the attributes of a record of a DMARC aggregate report, including aligned and raw DKIM and SPF authentication results and domains, message count and mail sender IP addresses. In addition, general information about the report, the record belongs to, are displayed in each row. This information includes report sender (*reporter*), report receiver (*reportee*), report date range, and report ID. Depending on the screen size some columns may be hidden. The user can then display the additional data by clicking on the first cell of the expandable table row. Furthermore, to reduce latency when dealing with large amounts of table rows, the table is paginated loading pages asynchronously. The report record table can be exported as CSV document.

4.1.3 Analysis view management

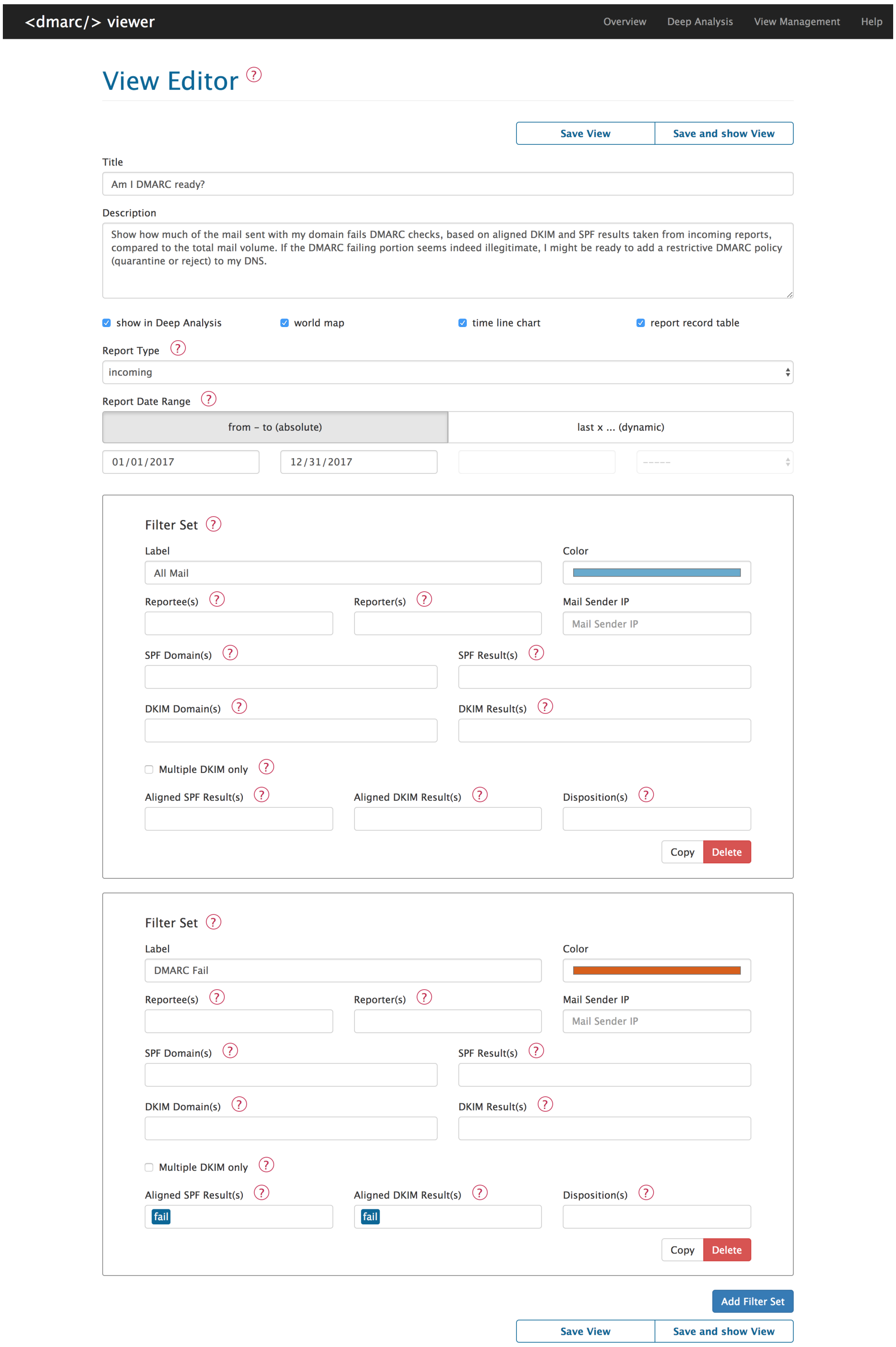


View Management

?	Show	Title	Description	Actions
↕	True	Am I DMARC ready?	Show how much of the mail sent with my domain fails DMARC checks, based on aligned DKIM and SPF results taken from incoming reports, compared to the total mail volume. If the DMARC failing portion seems indeed illegitimate, I might be ready to add a restrictive DMARC policy (quarantine or reject) to my DNS.	Show Edit Clone 🗑
↕	True	Who sent me spoofed mail and how did I react?	Show received mail that was purportedly spoofed, i.e. failed aligned DKIM and SPF checks (DMARC fail), and mail that I have rejected or placed under quarantine, all based on the information taken from outgoing DMARC aggregate reports.	Show Edit Clone 🗑
↕	True	Raw DKIM results	Show mail sent with my domain by a subset of raw DKIM results (neutral vs. temperror vs. permerror) as per incoming aggregate reports. Any other result than pass, will make the aligned DKIM result fail.	Show Edit Clone 🗑

Figure 4.3: *DMARC viewer* – Analysis view management

On the *analysis view management* page users can create, edit, clone and delete analysis views. Additionally, the order in which analysis views are presented on the deep analysis page's sidebar can be modified by dragging the arrow handles in the left-most cell of the analysis view overview table. By clicking the *Add View* button or the *Edit* button for a particular analysis view, the user can access the view editor. An example analysis view management page, showing three available analysis views, can be seen in Figure 4.3.



View Editor ?

Title
Am I DMARC ready?

Description
Show how much of the mail sent with my domain fails DMARC checks, based on aligned DKIM and SPF results taken from incoming reports, compared to the total mail volume. If the DMARC failing portion seems indeed illegitimate, I might be ready to add a restrictive DMARC policy (quarantine or reject) to my DNS.

☑ show in Deep Analysis ☑ world map ☑ time line chart ☑ report record table

Report Type ?
incoming

Report Date Range ?
[✓ from – to (absolute)] last x ... (dynamic)
01/01/2017 12/31/2017 ------

Filter Set ?

Label: All Mail
Color: 🟦
Reportee(s) ?
Reporter(s) ?
Mail Sender IP: Mail Sender IP
SPF Domain(s) ?
SPF Result(s) ?
DKIM Domain(s) ?
DKIM Result(s) ?
□ Multiple DKIM only ?
Aligned SPF Result(s) ?
Aligned DKIM Result(s) ?
Disposition(s) ?

Filter Set ?

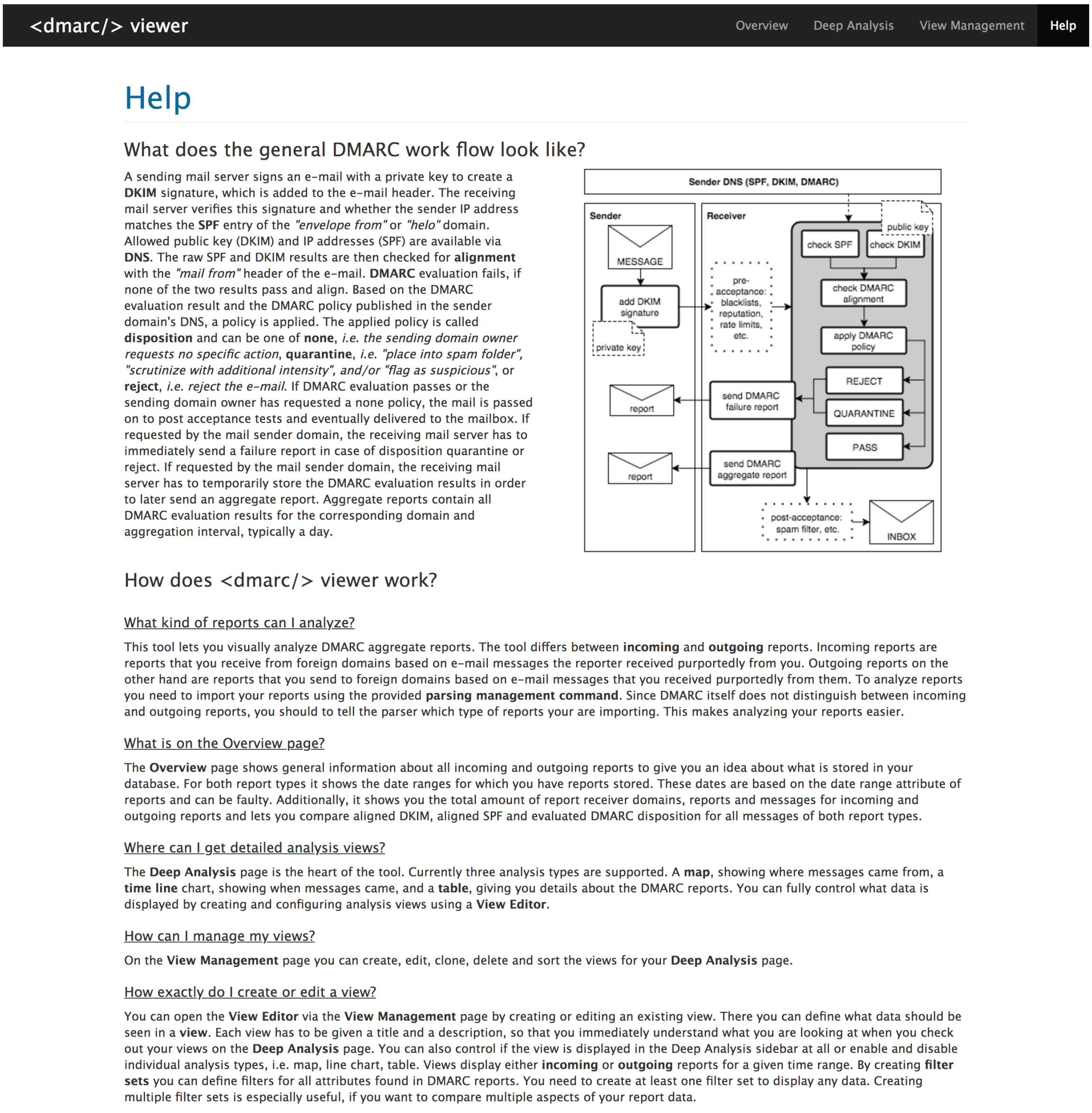
Label: DMARC Fail
Color: 🟧
Reportee(s) ?
Reporter(s) ?
Mail Sender IP: Mail Sender IP
SPF Domain(s) ?
SPF Result(s) ?
DKIM Domain(s) ?
DKIM Result(s) ?
□ Multiple DKIM only ?
Aligned SPF Result(s) ?: fail
Aligned DKIM Result(s) ?: fail
Disposition(s) ?

Figure 4.4: *DMARC viewer* – Analysis view editor

4.1.4 Analysis view editor

The *analysis view editor*, as shown in Figure 4.4, consists of a set of nested forms that allow the user to configure a given analysis view. Each view has a mandatory title and a description, so that the user can immediately grasp what she or he is looking at when visiting the corresponding analysis view on the deep analysis page. Furthermore, the user can parametrize which visualization types should be used, and whether the analysis view should be displayed at all. The latter is useful to keep unused views in the database, while not cluttering the deep analysis page sidebar. In addition to above settings that control descriptiveness and presentation, a view consists of filters that are applied to the database query when retrieving report data sets for an analysis view. Some filters are applied to all data set queries of a given analysis view. These filters include fixed and dynamic date range fields that narrow down report data based on their reporting time range as well as a report type option, to choose either incoming or outgoing reports. In addition to view-wide filters, the user can define dynamic sets of filters that correspond to the remaining attributes available in DMARC aggregate reports. For each defined filter set, a matching report data set is retrieved. A given analysis view can have multiple filter sets, in order to compare different report data sets to each other, most notably, as different time lines in the time line chart. Filter sets can be added, cloned or removed dynamically by the user. Every filter set has to be given a label and a color to distinguish it from other filter sets on the deep analysis page. Convenience widgets for the color filter, i.e. a color picker, and auto-complete multi-select filters are available. Non-exhaustive options, such as report receiver (*reportee*) and report sender (*reporter*) domains as well as available SPF and DKIM domains are loaded asynchronously and are fuzzy-searchable by scanning the stored DMARC reports.

4.1.5 Help



Help

What does the general DMARC work flow look like?

A sending mail server signs an e-mail with a private key to create a **DKIM** signature, which is added to the e-mail header. The receiving mail server verifies this signature and whether the sender IP address matches the **SPF** entry of the *"envelope from"* or *"helo"* domain. Allowed public key (DKIM) and IP addresses (SPF) are available via **DNS**. The raw SPF and DKIM results are then checked for **alignment** with the *"mail from"* header of the e-mail. **DMARC** evaluation fails, if none of the two results pass and align. Based on the DMARC evaluation result and the DMARC policy published in the sender domain's DNS, a policy is applied. The applied policy is called **disposition** and can be one of **none**, *i.e. the sending domain owner requests no specific action*, **quarantine**, *i.e. "place into spam folder", "scrutinize with additional intensity", and/or "flag as suspicious"*, or **reject**, *i.e. reject the e-mail.* If DMARC evaluation passes or the sending domain owner has requested a none policy, the mail is passed on to post acceptance tests and eventually delivered to the mailbox. If requested by the mail sender domain, the receiving mail server has to immediately send a failure report in case of disposition quarantine or reject. If requested by the mail sender domain, the receiving mail server has to temporarily store the DMARC evaluation results in order to later send an aggregate report. Aggregate reports contain all DMARC evaluation results for the corresponding domain and aggregation interval, typically a day.

How does <dmarc/> viewer work?

What kind of reports can I analyze?

This tool lets you visually analyze DMARC aggregate reports. The tool differs between **incoming** and **outgoing** reports. Incoming reports are reports that you receive from foreign domains based on e-mail messages the reporter received purportedly from you. Outgoing reports on the other hand are reports that you send to foreign domains based on e-mail messages that you received purportedly from them. To analyze reports you need to import your reports using the provided **parsing management command**. Since DMARC itself does not distinguish between incoming and outgoing reports, you should to tell the parser which type of reports your are importing. This makes analyzing your reports easier.

What is on the Overview page?

The **Overview** page shows general information about all incoming and outgoing reports to give you an idea about what is stored in your database. For both report types it shows the date ranges for which you have reports stored. These dates are based on the date range attribute of reports and can be faulty. Additionally, it shows you the total amount of report receiver domains, reports and messages for incoming and outgoing reports and lets you compare aligned DKIM, aligned SPF and evaluated DMARC disposition for all messages of both report types.

Where can I get detailed analysis views?

The **Deep Analysis** page is the heart of the tool. Currently three analysis types are supported. A **map**, showing where messages came from, a **time line** chart, showing when messages came, and a **table**, giving you details about the DMARC reports. You can fully control what data is displayed by creating and configuring analysis views using a **View Editor**.

How can I manage my views?

On the **View Management** page you can create, edit, clone, delete and sort the views for your **Deep Analysis** page.

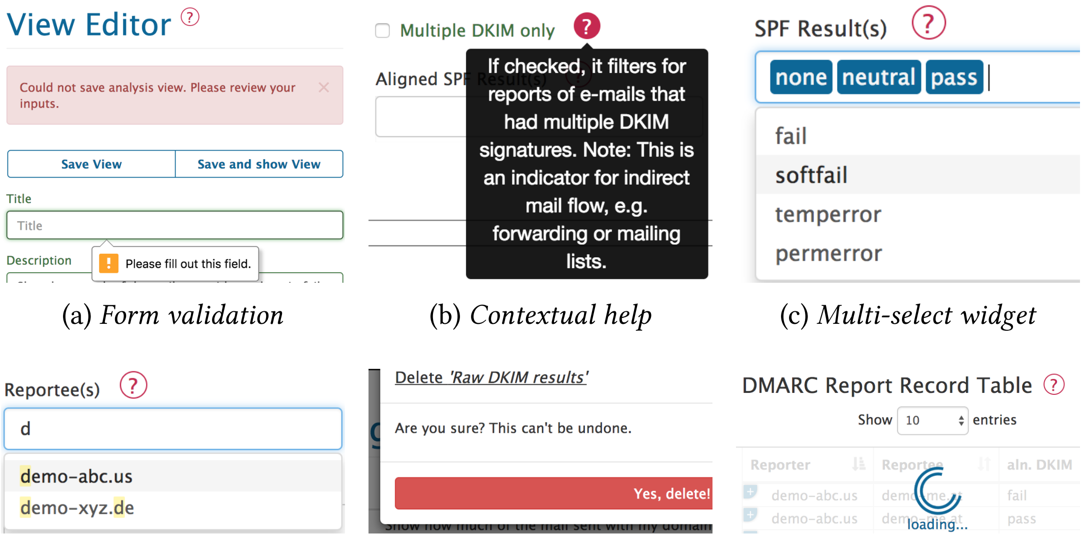
How exactly do I create or edit a view?

You can open the **View Editor** via the **View Management** page by creating or editing an existing view. There you can define what data should be seen in a **view**. Each view has to be given a title and a description, so that you immediately understand what you are looking at when you check out your views on the **Deep Analysis** page. You can also control if the view is displayed in the Deep Analysis sidebar at all or enable and disable individual analysis types, i.e. map, line chart, table. Views display either **incoming** or **outgoing** reports for a given time range. By creating **filter sets** you can define filters for all attributes found in DMARC reports. You need to create at least one filter set to display any data. Creating multiple filter sets is especially useful, if you want to compare multiple aspects of your report data.

Figure 4.5: *DMARC viewer* – Help

A key requirement for the presented DMARC report analysis software was the provisioning of adequate learning aids, in order for DMARC novices to not only gather insights from their reports, but at the same time to better understand the underlying technologies. Thus, a comprehensive help section was created. The help section describes the behavior of the software and provides a visual and verbal description of the general DMARC work flow. The evaluation and enhancement of the help page was a key objective of the second phase of the proposed UCD process (see Section 3.2). The resulting help page is depicted in Figure 4.5.

The design evaluation studies revealed that users strongly welcomed the idea of contextual help, especially regarding the intricacies of the DMARC specification. To accommodate the user needs, numerous contextual help tooltips were added, explaining both the UI and complex DMARC-specific terminology. Locating appropriate places for contextual help and assessing their content was an important part of the participatory prototype evaluation sessions as outlined in Section 3.3. An example use of contextual help is depicted in Figure 4.6b. Moreover, especially the latter phases of the conducted user studies (see Chapter 3), as well as a review of relevant UI heuristics and best practices (see Section 2.2.4), revealed further UI components, beneficial for the overall user interaction. These components are depicted in Figure 4.6. More specifically, form validation and user feedback in Figure 4.6a; a multi-select widget, providing DMARC-specific terminology, in Figure 4.6c; fuzzy-searchable and asynchronously loaded report domains in Figure 4.6d; a confirm dialog for a deletion operation in Figure 4.6e; and progress feedback for a latency-prone UI element in Figure 4.6f.


Could not save analysis view. Please review your inputs.

Please fill out this field.

If checked, it filters for reports of e-mails that had multiple DKIM signatures. Note: This is an indicator for indirect mail flow, e.g. forwarding or mailing lists.

Are you sure? This can't be undone.

(a) *Form validation* (b) *Contextual help* (c) *Multi-select widget*

(d) *Async fuzzy search* (e) *Confirmation dialog* (f) *Loading wheel*

Figure 4.6: *DMARC viewer* – UI components

4.2 Used technologies

This section presents the web technologies that were used to implement *DMARC viewer*. The used technologies were chosen with regard to their popularity and state-of-the-artness. Most notably, Python, a popular programming language known for its readability, was chosen in order to encourage uptake by the open source community. For this purpose the source code was extensively documented and aligned with the *PEP8* code style guidelines.[2] Other reasons for the choice of particular technologies include the developer's related experience (see [53], [52] and [54]) and the availability of usability and accessibility features, above all, in the used front-end libraries.

Back-end The back-end of the presented software was implemented using the *Python Django* Model Template Controller (MTV) framework in version 1.11 LTS[3]. Django provides a convenient object-relational mapping between Python models and the connected database. Moreover, Django ships with a powerful template engine that was used to create HTML pages for the basic site structure. In particular, Django's built-in form processing features significantly simplified the creation, validation and processing of HTML forms based on data models. In addition to Django's core functionality, third-party Django packages were used, some of which are outlined below. The full list of required third-party packages and their pinned versions can be found in the corresponding *requirements.txt* file available in the source code repository.[4]

- *django-bootstrap3*[5] — Bootstrap (front-end component library) integration for Django templates
- *django-formset-js*[6] — JavaScript (JS) helper for complex nested HTML forms based on Django models
- *django-debug-panel*[7] and *django-debug-toolbar*[8] — Django performance auditing tools
- *django-dmarc*[9] — Basic Django implementation of a DMARC aggregate report parser

[2] https://www.python.org/dev/peps/pep-0008/
[3] https://www.djangoproject.com/
[4] https://github.com/dmarc-viewer/dmarc-viewer/blob/master/requirements.txt
[5] https://github.com/dyve/django-bootstrap3
[6] https://pypi.python.org/pypi/django-formset-js
[7] https://github.com/recamshak/django-debug-panel
[8] https://pypi.python.org/pypi/django-debug-toolbar
[9] https://github.com/alan-hicks/django-dmarc

While *PostgreSQL*[10] v9.3.5 was used as database back-end during development, as well as in production for a live demo, *DMARC viewer* is fully database agnostic and can be used with a wide range of relational, as well as non-relational database systems.[11]

Front-end Presentational logic was mainly implemented using *jQuery* v3.2.1[12] and the front-end framework *Bootstrap* v3[13], providing a responsive grid layout and generic UI components. Custom styles were implemented using the CSS pre-processor *Syntactically Awesome Style Sheets (Sass)*[14]. Moreover, the JS task runner, *Gulp.js*[15], was used to automate front-end development tasks, such as Sass compilation, as well as asset minification and concatenation, and source mapping for a smooth integration with browser developer tools. UI colors were chosen using *colorbrewer*[16] and evaluated for accessibility.[17] Below some of the used third-party libraries are briefly mentioned. The full list of front-end requirements can be found in the corresponding Node Package Manager (NPM) file *package.json*, available in the source code repository.[18]

- *Data-driven Documents (D3.js)*[19] — Data visualization library using JS, HTML, SVG and CSS
- *DataMaps*[20] — Map visualization library using D3.js and *topojson*
- *DataTables*[21] — Responsive HTML tables with column sorting, pagination, and asynchronous data requests
- *Selectize*[22] — Multi-select widgets with fuzzy searching and asynchronous auto-completion
- Sortable [23] — Drag-and-drop support for re-ordering DOM elements

[10] http://www.postgresql.org/
[11] https://docs.djangoproject.com/en/1.11/ref/databases/
[12] https://jquery.com/
[13] http://getbootstrap.com/
[14] http://sass-lang.com/
[15] http://gulpjs.com/
[16] http://colorbrewer2.org/
[17] https://www.toptal.com/designers/colorfilter/
[18] https://github.com/dmarc-viewer/dmarc-viewer/blob/master/package.json
[19] http://d3js.org/
[20] http://datamaps.github.io/
[21] https://www.datatables.net
[22] http://selectize.github.io/selectize.js/
[23] http://rubaxa.github.io/Sortable/

4.3 Release

DMARC viewer was released publicly under the *MIT* license in May 2018 on the code hosting platform *GitHub*. Prior to its release, extensive deployment and usage documentation was created, which is also available on the corresponding source code repository.[24]

The release was promoted on the *dmarc-ietf* mailing list and received favorable feedback. Most notably, one of the primary authors of the DMARC standard invited the author of this work to a conference held by the Messaging, Malware and Mobile Anti-Abuse Working Group (M3AAWG), to discuss DMARC and DMARC report analysis.

In addition to the release of the *DMARC viewer* source code repository, a publicly accessible instance of *DMARC viewer* was deployed.[25] It demonstrates its basic functionality using random report data, seeded with DMARC aggregate report data generation software that was implemented specifically for that purpose also by the author of this work.[26]

As measured by the used visitor tracking software *Matomo*[27], the live demo page attracted approximately 70 global visitors in the first month after its release, originating from various countries in North America, Asia and Europe, and performing 200+ page views in total.

Since the release of *DMARC viewer*, non-affiliated members of the DMARC community have contributed *GitHub issues* and *pull requests* to the project. Most notably, a tentative *docker*[28] deployment recipe was submitted and adopted subsequently by the author of this work.

[24] https://github.com/dmarc-viewer/dmarc-viewer
[25] https://dmarc-viewer.abteil.org/
[26] https://github.com/dmarc-viewer/dmarc-demo-data
[27] https://matomo.org/
[28] https://www.docker.com/

5 | Conclusion

This Master's thesis documents the evolution of *DMARC viewer*, a fully functional web-based DMARC aggregate report analysis tool that has been developed for, and with input from, mailing experts. By providing a multitude of visualization and interpretation facilities *DMARC viewer* allows its target users to gather detailed insights about their mailing domain's global use and abuse (see Chapter 4).

DMARC viewer was implemented in the course of a custom three-stage UCD process that involved input from mailing experts throughout each phase. The complete participatory development process is documented in Chapter 3 of this thesis. This process includes the transparent conception, justification of the methods used (a comprehensive review of the related methodologies can be found in Section 2.2), facilitation and interpretation of the corresponding user studies, identification of tangible design implications, and a meta-review of the employed methods.

The UCD process started off with a participatory field study in which requirements were gathered by conducting semi-structured qualitative interviews with mail administrators (see Section 3.1). The main objective of the initial exploratory phase was to understand the target domain of DMARC aggregate report analysis, which was largely uncharted at that time, and to evaluate a few preliminary ideas conceived by the author of this work from, among other things, a review of the related technology (see Section 2.1). In addition, new ideas were introduced during the studies based on the experience and expertise of the mail administrators interviewed.

In the second phase of the study, a wide array of concrete design ideas were collaboratively evaluated, based on insights drawn from the previous study. The evaluation of these insights was conducted using mid-fidelity prototypes, i.e. wireframes, and additional usage diagrams. The objective of the design evaluation studies was to narrow down the many ways of analyzing DMARC aggregate reports, in order to implement a prototype (see Section 3.2).

Eventually, the prototype was realized using state-of-the-art web development technologies, and then evaluated in scenario-based usability testing sessions. These sessions tested for user acceptance of the target system, and aimed at identifying any remaining usability issues (see Chapter 3.3), which were addressed before the software was released to the open source community.

Both the feedback from the prototype evaluation and from the broader DMARC community in response to the release were perceivably positive (see Sections 3.3.3 and 4.3).

Using this incremental and iterative development process, which was driven by the expertise of the participating target users, allowed the author of this work to dismiss some preliminary ideas early in the process. This saved him from wasting developer hours on concepts due to fail. Most notably, collaborative report analysis, where users correlate insights from their own domains with reports from other domains, turned out to be of low interest for a first prototype of the system. Instead, the user studies revealed that the system requires integrated support from which users at all levels of DMARC expertise can benefit. The *expert* term encountered in this work remains a topic for discussion.

5.1 Discussion and outlook

In general terms, experts are referred to as persons "who are particularly competent as authorities on a certain matter of facts" [17, p. 165]. The field of Sciene and Technology Studies (STS), which, among other things, examines power dynamics related to expertise, e.g. in regard to decision-making, suggests a characterization of the term with the distinction that "genuine experts on a topic have knowledge that non-experts lack" [64, p. 180].

In UCD, expertise is relevant on both the researcher- and study participant-side. For the researcher, at least two aspects of expertise can be identified: expertise related to available UCD methodology, and knowledge about the investigated problem domain. The former is generally described as a required asset, which determines the success of a given research project [5, par. 52.4.3],[67, p. 21]. As a consequence UCD projects are usually carried out in teams of experts with various backgrounds in diverse fields (see Section 2.2.2.1).

In contrast to methodological expertise, the extent of domain-specific knowledge required engenders some controversy. It is noted that an increased awareness of the specificities of a given domain may turn the researcher into an invested practitioner rather than an outside observer of the target user [5, par. 52.4.3]. Holtzblatt et al. characterize the relationship between researcher and research subject as akin to an apprenticeship model, where the researcher is the apprentice, who should learn from the user, who, in turn, is the expert of the investigated problem domain [22, p. 86]. This also aligns with the general stance of participatory design, where all users are seen as experts, that is "experts of their experiences" [65].

However, in order to investigate a problem domain adequately, it is important for the researcher to assess the extent and quality of a target user's expertise relevant to the target domain [67, p. 55]. In that regard, Meuser et al., who have conducted comprehensive studies on qualitative research methods and their suitability for expert participants, have pointed out an important pitfall, that of seeming experts who turn out to be non-experts [39].

For the purpose of the presented work, mail administrators were identified as target users of the proposed DMARC aggregate report analysis tool. As such, the development drew on their experience and expertise in the field of mail administration in general, and especially in regard to identity fraud detection and prevention. The assessment of the extent and quality of specific technological knowledge — DMARC, SPF and DKIM — was an important early objective of the studies of the proposed three-phase UCD process. A consequential finding was that the amount of knowledge, the author of this work had hoped to tap, varied significantly among the study participants. This was mainly attributed to the complexity and novelty of the related technologies at the time the studies were conducted. The heterogeneity and, most notably, the lack of relevant expertise determined the design of the target system. As a result the work pursued a DMARC aggregate report analysis software that could not only provide insights into the user's mailing domain, but could serve as an aid for teaching the underlying technologies.

Outlook The proposed DMARC aggregate report analysis tool may be used to conduct future user studies in order to grow the knowledge base about expert-based UCD. More specifically, with the anticipated proliferation of the highly extensible and customizable *DMARC viewer*, domain experts may come up with interesting usage scenarios. By testing these scenarios with separate user groups that are clearly defined by their prior measured DMARC knowledge, correlations between domain expertise and the ability to carry out relevant tasks could be identified. This, in turn, may allow to not only draw general conclusions about usability, but also about the link between usability testing and domain expertise.

Eventually, one of the preliminary objectives of this work, i.e. developing a collaborative DMARC report analysis tool, may be revived. Based on the results of the user studies conducted for this thesis, the idea of collaborative analysis was rejected. However, based on the assumption that the usefulness of individual analysis can be proved if the proposed software is tested over an extended period, future work could investigate how that usefulness may be established in the scope of collaborative DMARC aggregate report analysis.

Bibliography

[1] Donald E. Eastlake 3rd. *Domain Name System Security Extensions.* RFC 2535. Mar. 1999. doi: 10.17487/RFC2535. url: https://rfc-editor.org/rfc/rfc2535.txt.

[2] Kurt Andersen, Brandon Long, Seth Blank, Murray Kucherawy, and Tim Draegen. *Authenticated Received Chain (ARC) Protocol.* Internet-Draft draft-ietf-dmarc-arc-protocol-16. Work in Progress. Internet Engineering Task Force, July 2018. url: https://datatracker.ietf.org/doc/html/draft-ietf-dmarc-arc-protocol-16.

[3] Aaron Bangor, Philip Kortum, and James Miller. "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale". In: *J. Usability Studies* 4.3 (May 2009), pp. 114–123. url: http://dl.acm.org/citation.cfm?id=2835587.2835589 (visited on 08/16/2018).

[4] Hugh Beyer and Karen Holtzblatt. *Contextual Design: Defining Customer-Centered Systems.* 1 edition. San Francisco, Calif: Morgan Kaufmann, Sept. 1997.

[5] Blandford, Ann. "Semi-structured qualitative studies". In: *The Encyclopedia of Human-Computer Interaction.* Ed. by Soegaard, Mads, Dam, and Rikke Friis. 2nd Ed. The Interaction Design Foundation, 2014.

[6] Jürgen Bortz and Nicola Döring. *Forschungsmethoden und Evaluation: für Human- und Sozialwissenschaftler.* German. 4., überarb. Aufl. 2006 edition. Heidelberg: Springer, Oct. 2006.

[7] Cynthia A Brewer. "Color use guidelines for mapping and visualization". In: *Visualization in modern cartography* 2 (1994), pp. 123–148.

[8] John Brooke. *SUS: A quick and dirty usability scale.* 1996.

[9] AGARI Email Threat Center. *Email Fraud and DMARC Adoption Trends, Second Half 2017*. 2017. URL: https://3l9nb01u2hkg4cz5053evqwi-wpengine.netdna-ssl.com/wp-content/uploads/2018/02/Agari_Threat-center-report.pdf (visited on 08/16/2018).

[10] Shalendra Chhabra. "Fighting Spam, Phishing and Email Fraud". Master's Thesis. University of California Riverside, Dec. 2005.

[11] Cockton, Gilbert. "Usability Evaluation". In: *The Encyclopedia of Human-Computer Interaction*. Ed. by Soegaard, Mads, Dam, and Rikke Friis. 2nd Ed. The Interaction Design Foundation, 2014.

[12] Juliet Corbin and Anselm Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. 4th Edition. Revised. Los Angeles: Sage Publications Ltd., 2015.

[13] DMARC.org. *DMARC wiki — Frequently Asked Questions*. URL: https://dmarc.org/wiki/FAQ (visited on 08/16/2018).

[14] Susan Farrell. *UX Research Cheat Sheet*. 2017. URL: https://www.nngroup.com/articles/ux-research-cheat-sheet/ (visited on 08/16/2018).

[15] Jim Fenton. *Analysis of Threats Motivating DomainKeys Identified Mail (DKIM)*. RFC 4686. Sept. 2006. DOI: 10.17487/RFC4686. URL: https://rfc-editor.org/rfc/rfc4686.txt.

[16] Kim Flaherty. *Diary Studies: Understanding Long-Term User Behavior and Experiences*. 2016. URL: https://www.nngroup.com/articles/diary-studies/ (visited on 08/16/2018).

[17] Uwe Flick. *An Introduction to Qualitative Research*. SAGE, Feb. 2009.

[18] Interaction Design Foundation. URL: https://www.interaction-design.org/ (visited on 08/16/2018).

[19] Interaction Design Foundation. *User Experience (UX) Design*. URL: https://www.interaction-design.org/literature/topics/ux-design/ (visited on 08/16/2018).

[20] Aurora Harley. *Personas Make Users Memorable for Product Team Members*. 2015. URL: https://www.nngroup.com/articles/persona/ (visited on 08/16/2018).

[21] Paul E. Hoffman and Jakob Schlyter. *The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA*. RFC 6698. Aug. 2012. DOI: 10.17487/RFC6698. URL: https://rfc-editor.org/rfc/rfc6698.txt.

[22] Karen Holtzblatt, Jessamyn Burns Wendell, and Shelley Wood. *Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design*. San Francisco: Morgan Kaufmann, 2004.

[23] *ISO 13407:1999: Human-centred design processes for interactive systems*. ISO/TC 159/SC 4. Norm. 1999.

[24] *ISO 9241-210:2010: Human-centred design for interactive systems*. ISO/TC 159/SC 4 Ergonomics of human-system interaction. Norm. 2010.

[25] Nigel Johnson. *Low DMARC adoption should not be surprising*. 2017. URL: https://www.zixcorp.com/resources/blog/august-2017/low-dmarc-adoption-should-not-be-surprising/ (visited on 08/16/2018).

[26] D. Scott Kitterman. *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1*. RFC 7208. Apr. 2014. DOI: 10.17487/RFC7208. URL: https://rfc-editor.org/rfc/rfc7208.txt.

[27] Dr. John C. Klensin. *Simple Mail Transfer Protocol*. RFC 5321. Oct. 2008. DOI: 10.17487/RFC5321. URL: https://rfc-editor.org/rfc/rfc5321.txt.

[28] eco Kompetenzgruppe E-Mail. *Gutachten zur Vereinbarkeit von DMARC mit dem deutschen Recht*. 2015. URL: https://www.eco.de/wp-content/uploads/2015/04/dmarc_rechtsgutachten.pdf (visited on 08/16/2018).

[29] Christos Koutroumpas. *CERT-EU Security Whitepaper 17-001, DMARC —– Defeating E-Mail Abuse*. 2017. URL: http://cert.europa.eu/static/WhitePapers/Updated-CERT-EU_Security_Whitepaper_DMARC_17-001_v1_2.pdf (visited on 08/16/2018).

[30] Murray Kucherawy. *DomainKeys Identified Mail (DKIM) and Mailing Lists*. RFC 6377. Sept. 2011. DOI: 10.17487/RFC6377. URL: https://rfc-editor.org/rfc/rfc6377.txt.

[31] Murray Kucherawy. *Message Header Field for Indicating Message Authentication Status*. RFC 5451. Apr. 2009. DOI: 10.17487/RFC5451. URL: https://rfc-editor.org/rfc/rfc5451.txt.

[32] Murray Kucherawy, Dave Crocker, and Tony Hansen. *DomainKeys Identified Mail (DKIM) Signatures.* RFC 6376. Sept. 2011. DOI: 10.17487/RFC6376. URL: https://rfc-editor.org/rfc/rfc6376.txt.

[33] Murray Kucherawy and Elizabeth Zwicky. *Domain-based Message Authentication, Reporting, and Conformance (DMARC).* RFC 7489. Mar. 2015. DOI: 10.17487/RFC7489. URL: https://rfc-editor.org/rfc/rfc7489.txt.

[34] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach.* 5th Ed. Boston, Mass.: Prentice Hall, 2009.

[35] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. *Research Methods in Human-Computer Interaction.* 1 edition. Chichester, West Sussex, U.K: Wiley, Feb. 2010.

[36] John R. Levine. *A new cryptographic signature method for DKIM.* Internet-Draft draft-ietf-dcrup-dkim-crypto-14. Work in Progress. Internet Engineering Task Force, June 2018. URL: https://datatracker.ietf.org/doc/html/draft-ietf-dcrup-dkim-crypto-14.

[37] John R. Levine, Mark Delany, Eric P. Allman, and Jim Fenton. *DomainKeys Identified Mail (DKIM) Author Domain Signing Practices (ADSP).* RFC 5617. Aug. 2009. DOI: 10.17487/RFC5617. URL: https://rfc-editor.org/rfc/rfc5617.txt.

[38] Ji-Ye Mao, Karel Vredenburg, Paul W. Smith, and Tom Carey. "The State of User-centered Design Practice". In: *Commun. ACM* 48.3 (Mar. 2005), pp. 105–109. ISSN: 0001-0782. DOI: 10.1145/1047671.1047677. URL: http://doi.acm.org/10.1145/1047671.1047677 (visited on 10/08/2014).

[39] Michael Meuser and Ulrike Nagel. "ExpertInneninterviews — vielfach erprobt, wenig bedacht". German. In: *Qualitativ-empirische Sozialforschung.* VS Verlag für Sozialwissenschaften, Wiesbaden, 1991, pp. 441–471. DOI: 10.1007/978-3-322-97024-4_14. URL: https://link.springer.com/chapter/10.1007/978-3-322-97024-4_14 (visited on 06/29/2018).

[40] P. Mockapetris. *Domain names - concepts and facilities.* RFC 1034. Nov. 1987. DOI: 10.17487/RFC1034. URL: https://rfc-editor.org/rfc/rfc1034.txt.

[41] P. Mockapetris. *Domain names - implementation and specification.* RFC 1035. Nov. 1987. DOI: 10.17487/RFC1035. URL: https://rfc-editor.org/rfc/rfc1035.txt.

[42] Nielsen, Lene. "Personas". In: *The Encyclopedia of Human-Computer Interaction.* Ed. by Soegaard, Mads, Dam, and Rikke Friis. 2nd Ed. The Interaction Design Foundation, 2014.

[43] Jakob Nielsen. *10 Usability Heuristics for User Interface Design.* 1995. URL: https://www.nngroup.com/articles/ten-usability-heuristics/ (visited on 08/16/2018).

[44] Jakob Nielsen. *Interviewing Users.* 2010. URL: https://www.nngroup.com/articles/interviewing-users/ (visited on 08/16/2018).

[45] Jakob Nielsen. *Thinking Aloud: The #1 Usability Tool.* 2012. URL: https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/ (visited on 08/16/2018).

[46] Jakob Nielsen. *Why You Only Need to Test with 5 Users.* 2000. URL: https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/ (visited on 08/16/2018).

[47] Jakob Nielsen and Rolf Molich. "Heuristic Evaluation of User Interfaces". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* CHI '90. New York, NY, USA: ACM, 1990, pp. 249–256. DOI: 10.1145/97243.97281. URL: http://doi.acm.org/10.1145/97243.97281.

[48] Don Norman and Jakob Nielsen. *The Definition of User Experience (UX).* URL: https://www.nngroup.com/articles/definition-user-experience/ (visited on 08/16/2018).

[49] Kara Pernice. *Ethnographic Studies Training.* URL: https://www.nngroup.com/consulting-field-studies/ (visited on 08/16/2018).

[50] The Spamhaus Project. *The Definition of Spam.* URL: https://www.spamhaus.org/consumer/definition/ (visited on 08/16/2018).

[51] Proofpoint, Inc. *Getting Started With DMARC.* URL: https://www.proofpoint.com/sites/default/files/pfpt-uk-eb-getting-started-with-dmarc.pdf (visited on 08/16/2018).

[52] Lukas Pühringer. "Try Repy! A web-based Development and Execution Environment for Restricted Python". Bachelor's Thesis. Universität Wien, 2011.

[53] A. Rafetseder, F. Metzger, L. Pühringer, K. Tutschku, Y. Zhuang, and J. Cappos. "Sensorium — A Generic Sensor Framework". In: *PIK - Praxis der Informationsverarbeitung und Kommunikation* 36.1 (Feb. 2013), p. 46. DOI: doi:10.1515/pik-2012-0061.

[54] A. Rafetseder, L. Pühringer, and J. Cappos. "Practical fog computing with seattle". In: *2017 IEEE Fog World Congress (FWC).* Oct. 2017, pp. 1–7. DOI: 10.1109/FWC.2017.8368519.

[55] *Research-Based Web Design & Usability Guidelines.* U.S. Dept. of Health and Human Services, 2006.

[56] Pete Resnick. *Internet Message Format.* RFC 5322. Oct. 2008. DOI: 10.17487/RFC5322. URL: https://rfc-editor.org/rfc/rfc5322.txt.

[57] Christian Rohrer. *When to Use Which User-Experience Research Methods.* 2014. URL: https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/ (visited on 08/16/2018).

[58] Dan Saffer. *Designing for Interaction: Creating Innovative Applications and Devices.* 2nd ed. Berkeley, CA: New Riders, Aug. 2009.

[59] Markus Salo, Matthias Baldauf, Peter Fröhlich, and Stefan Suette. "Peak Moments of Physical Mobile Interaction Techniques". In: *Proceedings of the 19th Americas Conference on Information Systems (AMCIS).* 2013.

[60] Elizabeth Sanders. "From user-centered to participatory design approaches". In: *Design and the Social Sciences: Making Connections.* Apr. 2002, pp. 1–7. DOI: 10.1201/9780203301302.ch1.

[61] Corina Sas, Steve Whittaker, Steven Dow, Jodi Forlizzi, and John Zimmerman. "Generating Implications for Design Through Design Research". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* New York, NY, USA: ACM, 2014, pp. 1971–1980. DOI: 10.1145/2556288.2557357. URL: http://doi.acm.org/10.1145/2556288.2557357 (visited on 10/15/2014).

[62] Amy Schade. *Competitive Usability Evaluations: Learning from Your Competition.* 2013. URL: https://www.nngroup.com/articles/competitive-usability-evaluations/ (visited on 08/16/2018).

[63] Ben Shneiderman.
Designing the User Interface. Strategies for Effective Human-Computer Interaction.
3rd. Reading, Mass: Addison-Wesley Longman, Amsterdam, July 1997.

[64] Sergio Sismondo. *An Introduction to Science and Technology Studies.* 2nd Edition. Chichester, West Sussex, U.K.; Malden, MA: John Wiley and Sons Ltd, Nov. 2009.

[65] Froukje Sleeswijk Visser, Remko van der Lugt, and Pieter Jan Stappers.
"Participatory design needs participatory communication: New tools for sharing user insights in the product innovation process". In: (July 2005).

[66] M. Steen, M. a. J. Manschot, and N. De Koning.
"Benefits of co-design in service design projects".
In: *International Journal of Design 5(2)2011, 53-60* (2011). ISSN: 1991-3761.
URL: http://resolver.tudelft.nl/uuid:eefaaa3c-cc7d-408e-9e00-883c6f2ccb03 (visited on 07/17/2018).

[67] Debbie Stone, Caroline Jarrett, Mark Woodroffe, and Shailey Minocha.
User Interface Design and Evaluation.
Amsterdam ; Boston, Mass: Morgan Kaufmann, Apr. 2005.

[68] The Radicati Group, Inc. *Email Statistics Report, 2017-2021.* 2017.
URL: https://www.radicati.com/wp/wp-content/uploads/2017/01/Email-Statistics-Report-2017-2021-Executive-Summary.pdf (visited on 08/16/2018).

[69] usability.gov. URL: https://www.usability.gov/ (visited on 08/16/2018).

[70] usability.gov. *Card Sorting.* URL: https://www.usability.gov/how-to-and-tools/methods/card-sorting.html (visited on 08/16/2018).

[71] usability.gov. *Diary Study.* URL:
https://www.usability.gov/what-and-why/glossary/diary-study.html
(visited on 08/16/2018).

[72] usability.gov. *Heuristic Evaluations and Expert Reviews.*
URL: https://www.usability.gov/how-to-and-tools/methods/heuristic-evaluation.html (visited on 08/16/2018).

[73] usability.gov. *Personas.* URL:
https://www.usability.gov/how-to-and-tools/methods/personas.html
(visited on 08/16/2018).

[74] usability.gov. *Prototyping.* URL: https://www.usability.gov/how-to-and-tools/methods/prototyping.html (visited on 08/16/2018).

[75] usability.gov. *Running a Usability Test.* URL: https://www.usability.gov/how-to-and-tools/methods/running-usability-tests.html (visited on 08/16/2018).

[76] usability.gov. *Scenarios.* URL: https://www.usability.gov/how-to-and-tools/methods/scenarios.html (visited on 08/16/2018).

[77] usability.gov. *System Usability Scale (SUS).* URL: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html (visited on 08/16/2018).

[78] usability.gov. *Task Analysis.* URL: https://www.usability.gov/how-to-and-tools/methods/task-analysis.html (visited on 08/16/2018).

[79] Euphemia Wong. *Shneiderman's Eight Golden Rules Will Help You Design Better Interfaces.* 2018. URL: https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces/ (visited on 08/16/2018).

Appendix

Text listing 5.1: Semi-structured interview guide for field study and requirements gathering

Facilitator notes: Introduce to project and different phases of project. This session gives the opportunity to contribute ideas and wishes for a DMARC aggregate report analysis tool. It would be highly appreciated if participants take their time for the next two sessions as well, however they can withdraw from the project at any time. This session consists of three parts:

1. General questions about participants and work practice
2. Questions about the participants' usage of related technologies
3. Thoughts about requirements for a new DMARC report analysis tool

General questions

- Outline your vocational history.
- What is your current main occupation?
- How many hours do you work per week/how many as mail administrator?
- Since when have you been working as mail administrator?
- Which domains do you administer?
- Describe the mailing services you offer, in terms of:
  - Type (private/commercial)
  - User amount and mail volume
  - Do users use their own outgoing server?
  - Are mailing lists important for your user?
  - Do you administer mailing lists? If yes, do you use mailing list software that can cope with DMARC?

Current occupation

Configuration

- How often do you make changes in your infrastructure? Most recent changes?

Research

- How much of your work is keeping up with current developments?
- Where do you retrieve your information?

Networking

- How much of your work is communication with other mail administrators?
- Do you engage in any community activity? How?
- Which topics dominate the discussion lately?
- Are there any admins/infrastructures that you use as role models? Positive/Negative? Why?

Programming

- How much of your work is programming?
- Which programming languages do you know?
- Do you or have you participated in open source software projects? Which?

Security

- How much of your work is detection and resolving of security problems?
- How do you recognize spam, phishing, domain spoofing? How do you cope with it?
- What were the most critical threats lately? How did you cope with it?

Reporting

- Do you have to deliver regular reports about the infrastructure to you superiors?
- What do those reports look like?

Other tasks

- What else do you deal with as mail administrator?

Usage of technologies Do you use one or more of the following technologies? Why? Why not? Do you plan on using them?

- SPF
- DKIM
- ADSP
- DMARC

Do you request or send DMARC forensic or aggregate reports? How do you analyze aggregate reports?

Have you used any of the following tools?

- Dmarcian
- Dmarcanalyzer
- Postmarkapp

- Have you used other DMARC analysis tools? Which?
- What was especially positive/negative about the tools you used?

Requirements for a new tool *Facilitator notes:* Show which data a report contains. We will now talk about what a new DMARC report analysis tool could look like. Do you have any ideas on what the tool should be able to do?

Privacy

- If you look at the data of a DMARC aggregate report, do you intuitively have privacy concerns? Why?
- Are you equally concerned for the data of incoming and outgoing aggregate reports? Why?

Semantic analyses *Facilitator notes:* By combining SPF and DKIM results, DMARC allows to draw semantic conclusions. Please rate the following interpretations on a scale from 1 (very interesting) to 6 (not interesting).

- Legitimate mail originating from your domain and infrastructure
- Legitimate mail originating from your domain but not your infrastructure (mailing list, forwarding)
- Direct mail to you domain from legitimate senders
- Indirect mail to you domain from legitimate senders (mailing list, forwarding)
- Mail that probably spoofs your domain
- Mail that probably spoofs another domain
- Temporal evolution of mail authentication
- Temporal evolution of report exchange
- Geographical origin of mail
- Relationships/dialogs between you and a foreign domain
- DMARC/SPF/DKIM misconfiguration in your infrastructure
- DMARC/SPF/DKIM misconfiguration in foreign infrastructure

Collaborative analysis *Facilitator notes:* Collaborative analysis means that user are not only able to analyze the DMARC reports associated with their administered domain but can also compare their data with other domains.

- Could you imagine to use the tool for collaborative analysis? If so, as a centralized service in the ACOnet, or rather as self-hosted service, where you can invite other domain owners to share the data?
- How important is it to chose who you can share your data with?
- How important is it to chose what data to share?
- Would the privacy settings – open, anonymous, closed – be sufficient?

Usage You might have some imagination about the tool by now. Please rate the following statements on a scale from 1 (I agree) to 6 (I disagree).

- I would look up the information the tool reveals daily.
- I would look up the information the tool reveals weekly.
- I would look up the information the tool reveals monthly.
- I would look up the information the tool reveals in case of need.
- I want the tool to notify me in case of special events.
- I want to see interesting information at first view without configuring a lot of parameters.
- I have no concrete idea about the usefulness of the tool so far.

Parameter association task Please group some of the following parameters to interesting analysis scenarios.

- Time
- Report sender domain
- Report receiver domain
- IP of mail sender
- Report count
- Mail count
- DKIM signature count
- DKIM authentication results
- SPF authentication results
- Incoming reports
- Outgoing reports
- Disposition reject
- Disposition quarantine
- Disposition none
- Aligned DKIM fail
- Aligned DKIM pass
- Aligned SPF fail
- Aligned SPF pass

Closing questions

- Do you have any other ideas?
- A demographic closing question: How old are you?

Text listing 5.2: Initial requirements catalog

Functional requirements

- Provide a database to store incoming and outgoing DMARC aggregate reports
- Provide a parser to automatically import XML-formatted DMARC aggregate reports
- Provide a web-interface to visualize DMARC aggregate reports

Requirements specific to basic report analysis

- Show general statistics and anomaly alerts on an overview page
- Show in-depth analysis views on a deep analysis page
- Provide management page to create, delete, clone, import, export analysis views
- Provide a view editor
- Analysis views are assigned a name, a description and an analysis type, which is one of world map, time line, or table
- Provide a sidebar on the deep analysis page to select analysis views
- Provide a toggle for analysis views in the deep analysis sidebar
- Provide ordering for analysis views in the deep analysis sidebar
- Provide predefined analysis views
- Allow comparison of report aspects within a given analysis view, using customizable filter sets
- Provide a filter set editor as part of the view editor
- Provide predefined filter templates that can be used for any analysis view
- Filter sets are assigned a name and a color and filters related to DMARC aggregate reports

Filter variables of a filter set include: Report type, report date range, report sender domain, report receiver domain, mail sender IP, DKIM domain, DKIM result, SPF domain, SPF result, aligned DKIM result, aligned SPF result, disposition
predefined filter sets include combinations of filters in order to show:

- Presumably spoofed mails sent to the user's domain(s)
- Legitimate mails originating from the user's domain(s) and infrastructure
- Direct legitimate mails sent to the user's domain(s)
- Mails that presumably spoof the user's domain(s)
- Legitimate mails originating from the user's domain(s) sent via foreign infrastructure
- Indirect legitimate mails sent to the user's domain(s)

predefined views include combinations of filter sets in order to show:

- Time line of legitimate mails originating from the user's domain(s) and infrastructure vs. mails that presumably spoof the user's domain(s)
- Table of legitimate mails originating from the user's domain(s) and infrastructure vs. mails that presumably spoof the user's domain(s)
- World map of mails that presumably spoof the user's domain(s)
- World map of presumably spoofed mails sent to the user's domain(s)

Requirements specific to user handling and collaboration

- Support varying user numbers and varying per-user report emergence
- Provide isolation of report data for different users
- Provide account creation page for new users
- A user account can have multiple associated mailing domains
- Incoming as well as outgoing reports can be associated with a domain
- Designate a custom unique DMARC *ruaI* (Reporting URI for incoming aggregate reports) to each user, so that users can make their incoming DMARC reports available to the proposed tool. The *ruaI* address may be added to the user's DMARC DNS record as RUA address.
- Designate a custom unique DMARC *ruaO* (Reporting URI for outgoing aggregate reports) to each user, so that users can make their outgoing DMARC reports available to the proposed tool. Users may add a hook to their DMARC aggregate report generation facility to automatically send reports to the corresponding *ruaI* address.
- Users can query other users of the software and their domains
- Users can make sharing agreements with other users of the software
- Users can access other users' DMARC aggregate reports according to the corresponding sharing agreement
- Data sharing must be in accordance with juristic requirements

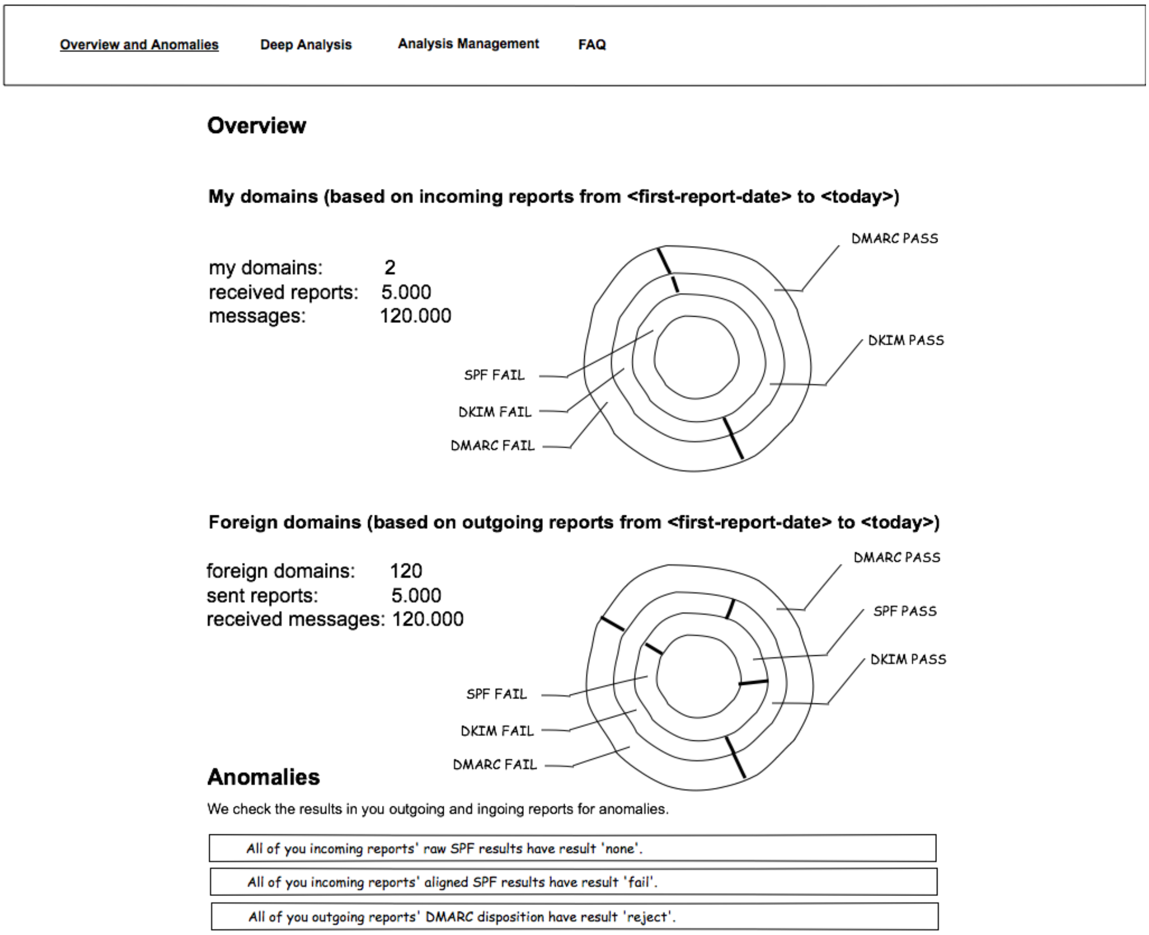
Requirements specific to education and guidance

- Show DMARC configuration problems based on anomalies in DMARC reports
- Help users to better understand the intricacies of DMARC aggregate reports by making the composition of analysis views completely transparent
- Provide predefined analysis views to help novices create their own analysis views
- Provide a comprehensive Frequently Asked Questions (FAQ) section

Non-functional requirements

Usability and hedonistic aspects

- Provide a web-interface that is fun, easy to use and self-explanatory, and appealing in terms of structure and color
- Due to the analysis of large amounts of report data, waiting time for user is expected. This waiting time should be minimized. Provide progress feedback, where waiting time is inevitable



Overview and Anomalies Deep Analysis Analysis Management FAQ

Overview

My domains (based on incoming reports from <first-report-date> to <today>)

my domains: 2
received reports: 5.000
messages: 120.000

Foreign domains (based on outgoing reports from <first-report-date> to <today>)

foreign domains: 120
sent reports: 5.000
received messages: 120.000

Anomalies

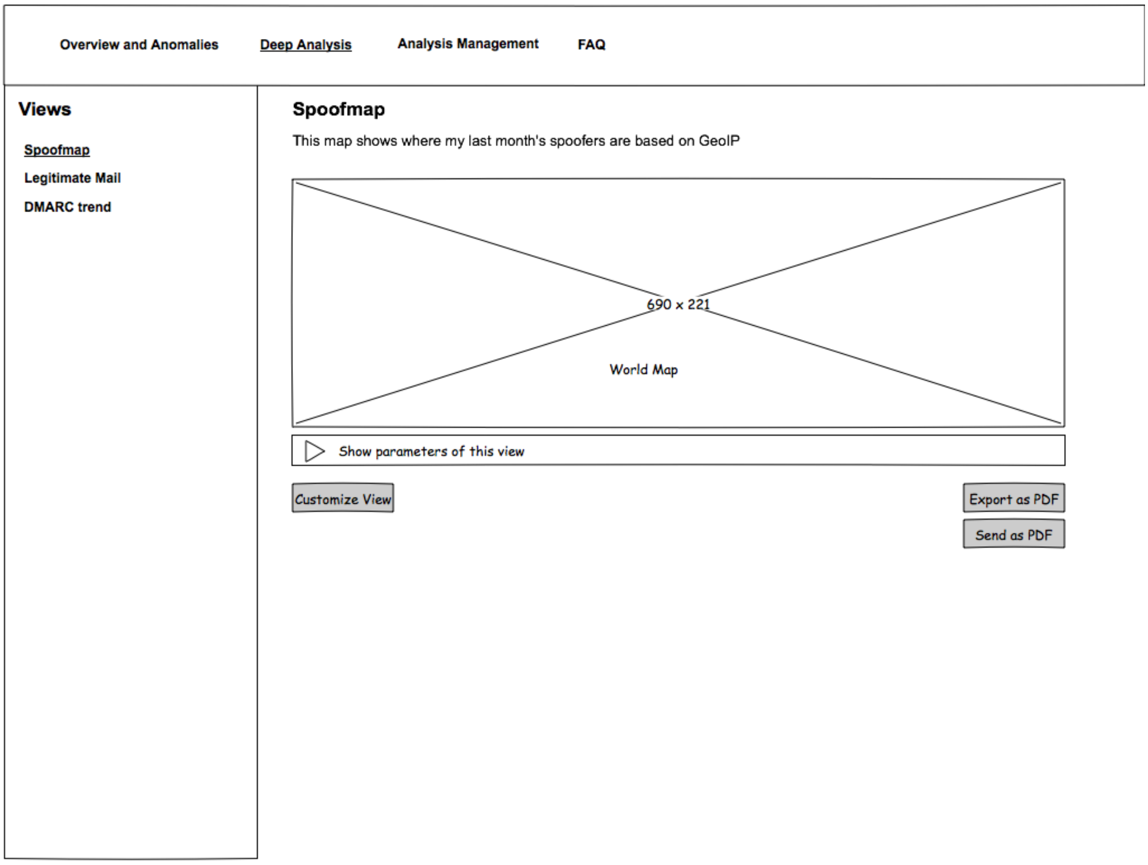
We check the results in you outgoing and ingoing reports for anomalies.

All of you incoming reports' raw SPF results have result 'none'.

All of you incoming reports' aligned SPF results have result 'fail'.

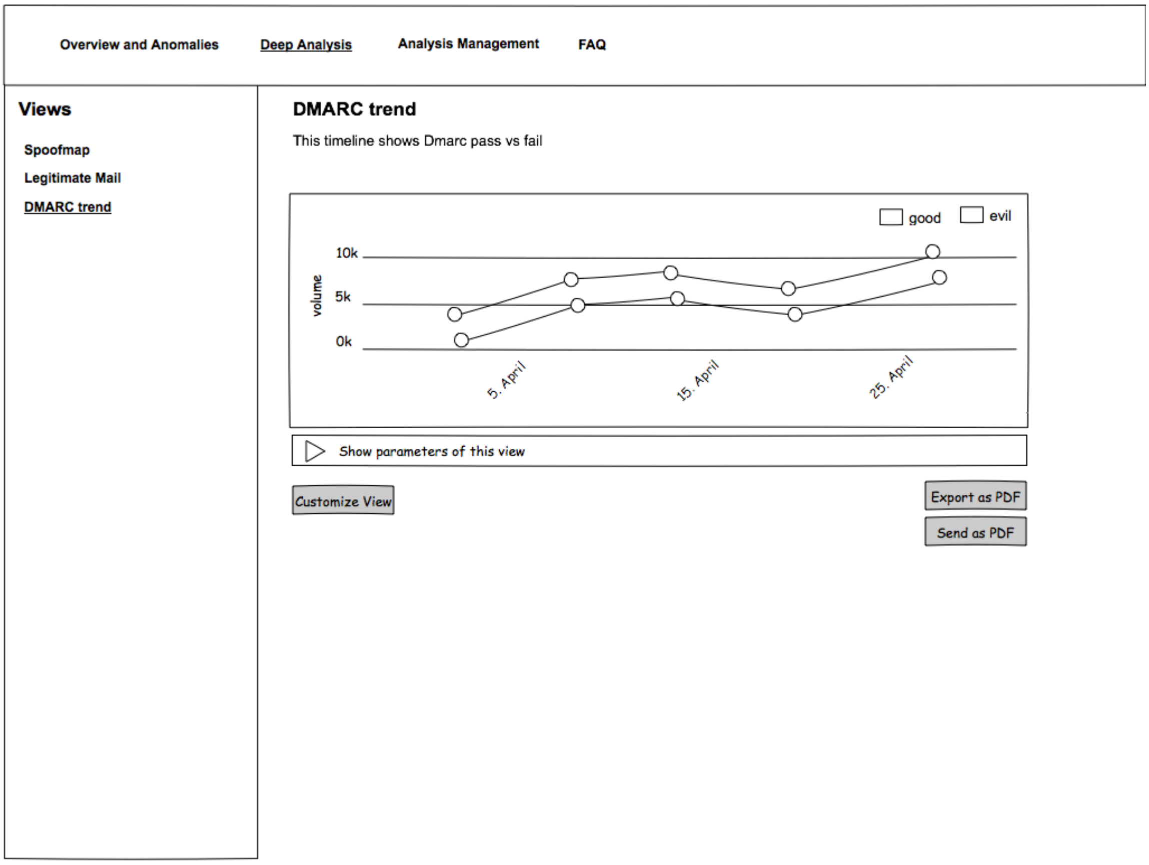
All of you outgoing reports' DMARC disposition have result 'reject'.

Figure 5.1: Wireframe – *DMARC viewer* – overview and anomalies



This map shows where my last month's spoofers are based on GeoIP

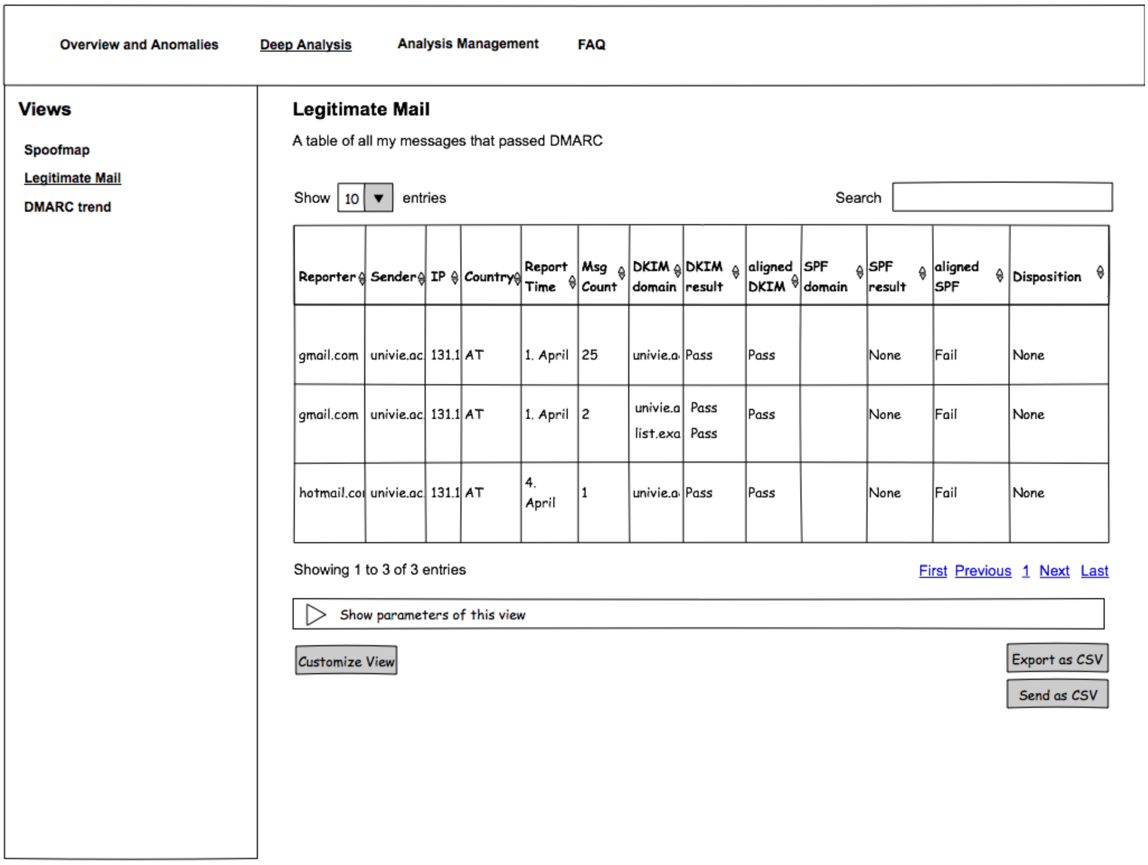
Figure 5.2: Wireframe – *DMARC viewer* – deep analysis – world map



DMARC trend

This timeline shows Dmarc pass vs fail

Figure 5.3: Wireframe – *DMARC viewer* – deep analysis – time line



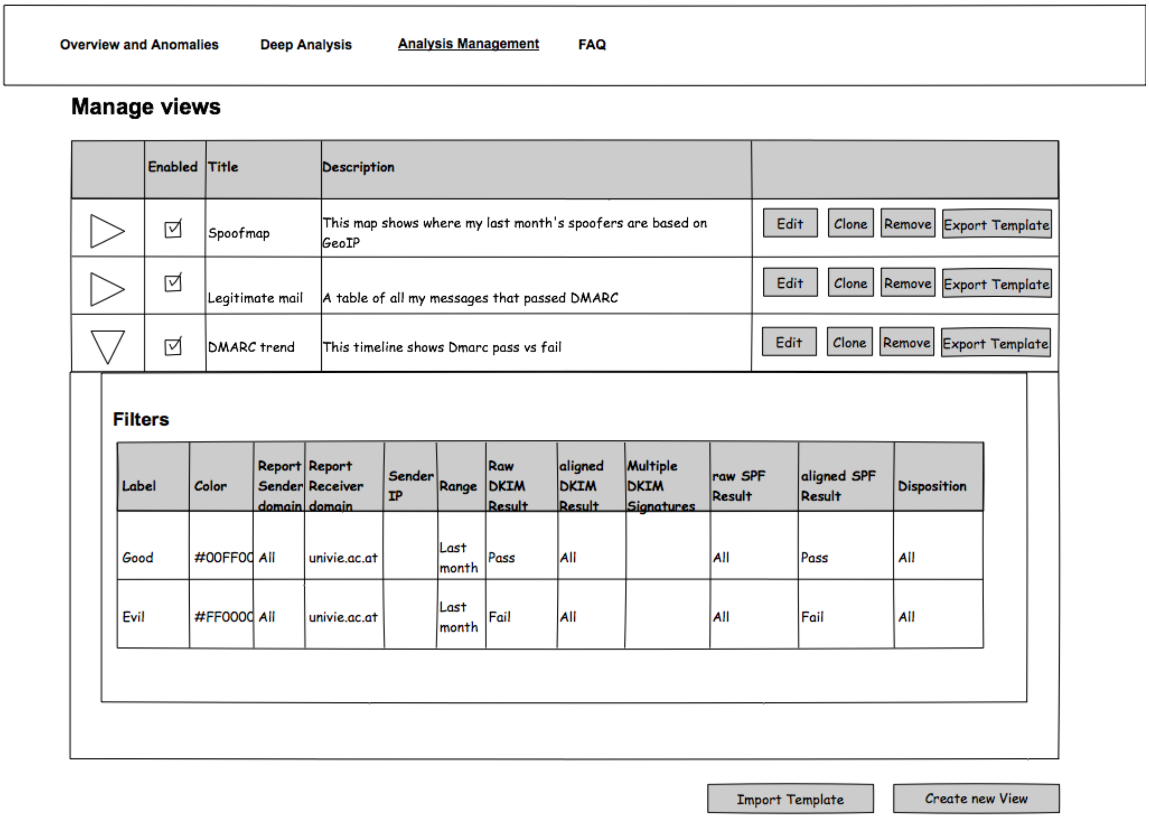
Legitimate Mail

A table of all my messages that passed DMARC

Reporter	Sender	IP	Country	Report Time	Msg Count	DKIM domain	DKIM result	aligned DKIM	SPF domain	SPF result	aligned SPF	Disposition
gmail.com	univie.ac	131.1	AT	1. April	25	univie.a	Pass	Pass		None	Fail	None
gmail.com	univie.ac	131.1	AT	1. April	2	univie.a list.exa	Pass Pass	Pass		None	Fail	None
hotmail.co	univie.ac	131.1	AT	4. April	1	univie.a	Pass	Pass		None	Fail	None

Showing 1 to 3 of 3 entries

Figure 5.4: Wireframe – *DMARC viewer* – deep analysis – table



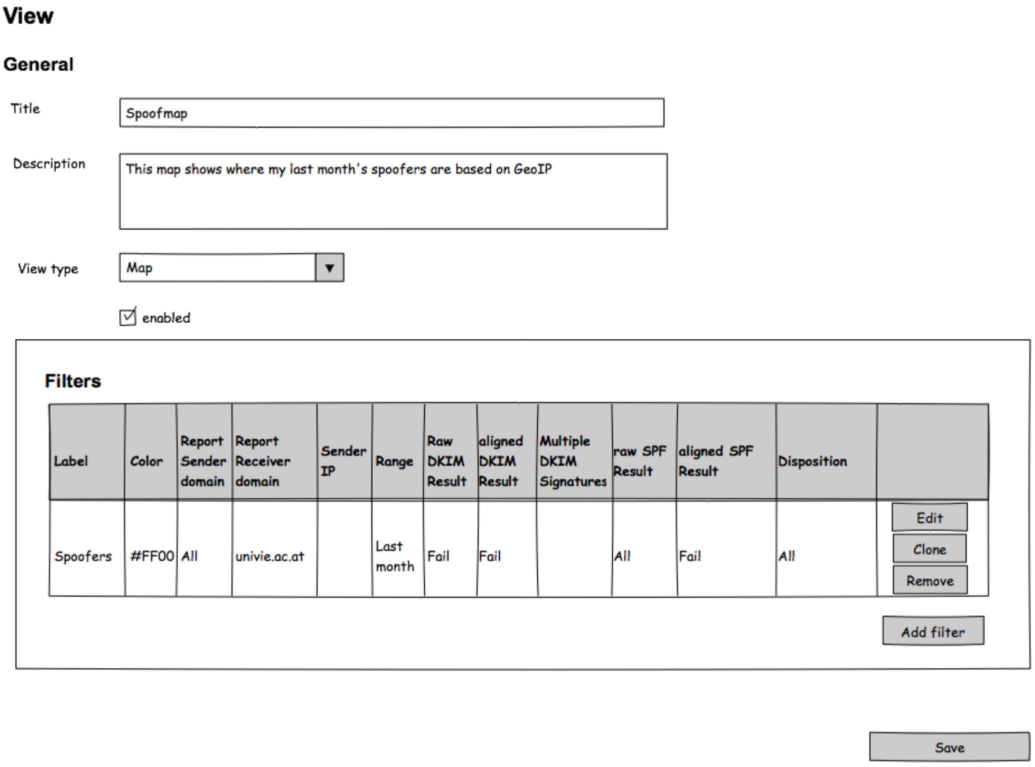
Manage views

	Enabled	Title	Description	
▷	☑	Spoofmap	This map shows where my last month's spoofers are based on GeoIP	Edit Clone Remove Export Template
▷	☑	Legitimate mail	A table of all my messages that passed DMARC	Edit Clone Remove Export Template
▽	☑	DMARC trend	This timeline shows Dmarc pass vs fail	Edit Clone Remove Export Template

Filters

Label	Color	Report Sender domain	Report Receiver domain	Sender IP	Range	Raw DKIM Result	aligned DKIM Result	Multiple DKIM Signatures	raw SPF Result	aligned SPF Result	Disposition
Good	#00FF00	All	univie.ac.at		Last month	Pass	All		All	Pass	All
Evil	#FF0000	All	univie.ac.at		Last month	Fail	All		All	Fail	All

Figure 5.5: Wireframe – *DMARC viewer* – view management



View

General

Title: Spoofmap

Description: This map shows where my last month's spoofers are based on GeoIP

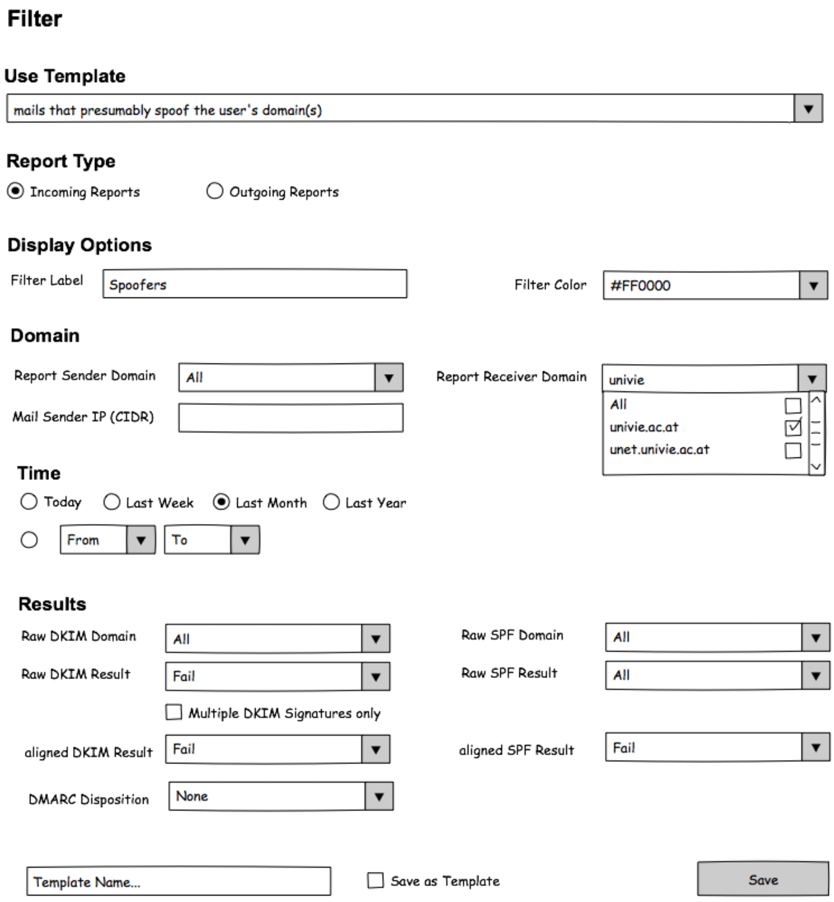
View type: Map ▼

☑ enabled

Filters

Label	Color	Report Sender domain	Report Receiver domain	Sender IP	Range	Raw DKIM Result	aligned DKIM Result	Multiple DKIM Signatures	raw SPF Result	aligned SPF Result	Disposition	
Spoofers	#FF00	All	univie.ac.at		Last month	Fail	Fail		All	Fail	All	Edit Clone Remove

Figure 5.6: Wireframe – *DMARC viewer* – view editor



Filter

Use Template

mails that presumably spoof the user's domain(s) ▼

Report Type

● Incoming Reports ○ Outgoing Reports

Display Options

Filter Label: Spoofers

Filter Color: #FF0000 ▼

Domain

Report Sender Domain: All ▼

Mail Sender IP (CIDR):

Report Receiver Domain: univie ▼
All □
univie.ac.at ☑
unet.univie.ac.at □

Time

○ Today ○ Last Week ● Last Month ○ Last Year

○ From ▼ To ▼

Results

Raw DKIM Domain: All ▼

Raw DKIM Result: Fail ▼

□ Multiple DKIM Signatures only

aligned DKIM Result: Fail ▼

DMARC Disposition: None ▼

Raw SPF Domain: All ▼

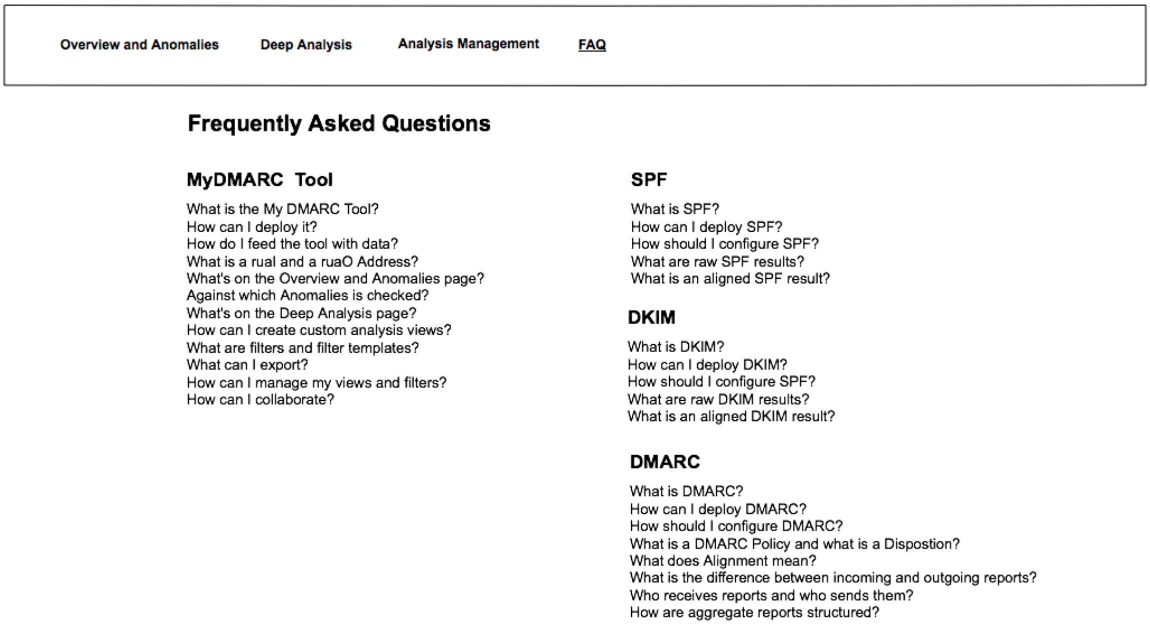
Raw SPF Result: All ▼

aligned SPF Result: Fail ▼

Template Name...

□ Save as Template

Figure 5.7: Wireframe – *DMARC viewer* – filter editor



Frequently Asked Questions

MyDMARC Tool

What is the My DMARC Tool?
How can I deploy it?
How do I feed the tool with data?
What is a rual and a ruaO Address?
What's on the Overview and Anomalies page?
Against which Anomalies is checked?
What's on the Deep Analysis page?
How can I create custom analysis views?
What are filters and filter templates?
What can I export?
How can I manage my views and filters?
How can I collaborate?

SPF

What is SPF?
How can I deploy SPF?
How should I configure SPF?
What are raw SPF results?
What is an aligned SPF result?

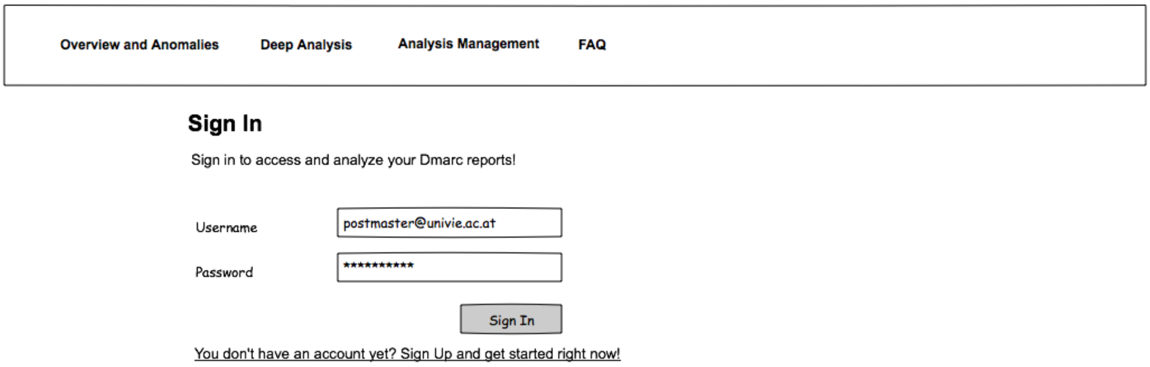
DKIM

What is DKIM?
How can I deploy DKIM?
How should I configure SPF?
What are raw DKIM results?
What is an aligned DKIM result?

DMARC

What is DMARC?
How can I deploy DMARC?
How should I configure DMARC?
What is a DMARC Policy and what is a Dispostion?
What does Alignment mean?
What is the difference between incoming and outgoing reports?
Who receives reports and who sends them?
How are aggregate reports structured?

Figure 5.8: Wireframe – *DMARC viewer* – help section



Sign In

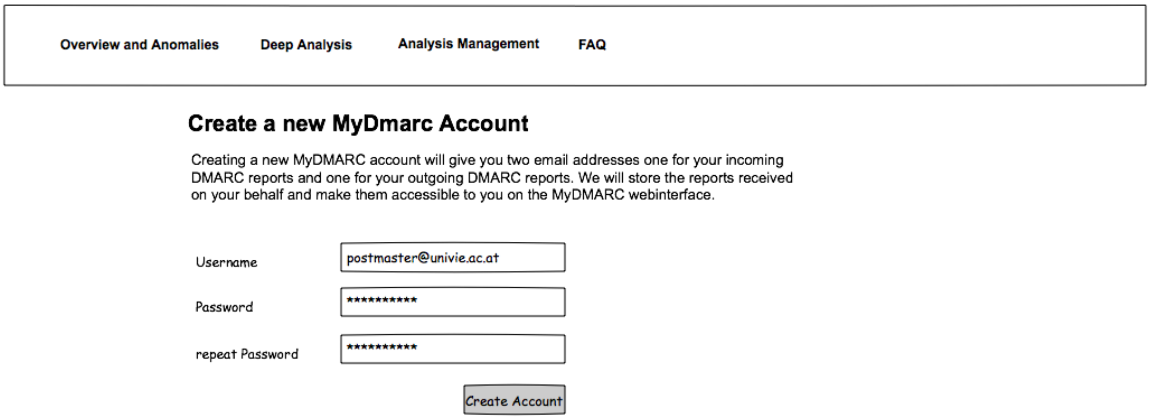
Sign in to access and analyze your Dmarc reports!

Username postmaster@univie.ac.at

Password ***********

You don't have an account yet? Sign Up and get started right now!

Figure 5.9: Wireframe – *DMARC viewer* – sign in



Create a new MyDmarc Account

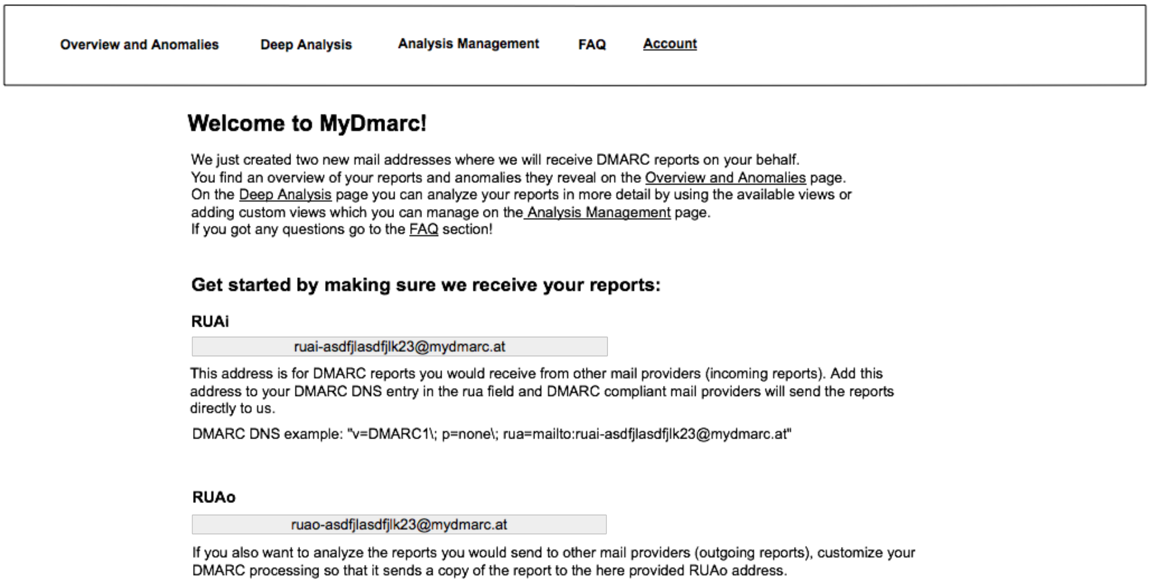
Creating a new MyDMARC account will give you two email addresses one for your incoming DMARC reports and one for your outgoing DMARC reports. We will store the reports received on your behalf and make them accessible to you on the MyDMARC webinterface.

Username: postmaster@univie.ac.at

Password: **********

repeat Password: **********

Figure 5.10: Wireframe – *DMARC viewer* – create account



Welcome to MyDmarc!

We just created two new mail addresses where we will receive DMARC reports on your behalf.
You find an overview of your reports and anomalies they reveal on the Overview and Anomalies page.
On the Deep Analysis page you can analyze your reports in more detail by using the available views or adding custom views which you can manage on the Analysis Management page.
If you got any questions go to the FAQ section!

Get started by making sure we receive your reports:

RUAi

ruai-asdfjlasdfjlk23@mydmarc.at

This address is for DMARC reports you would receive from other mail providers (incoming reports). Add this address to your DMARC DNS entry in the rua field and DMARC compliant mail providers will send the reports directly to us.

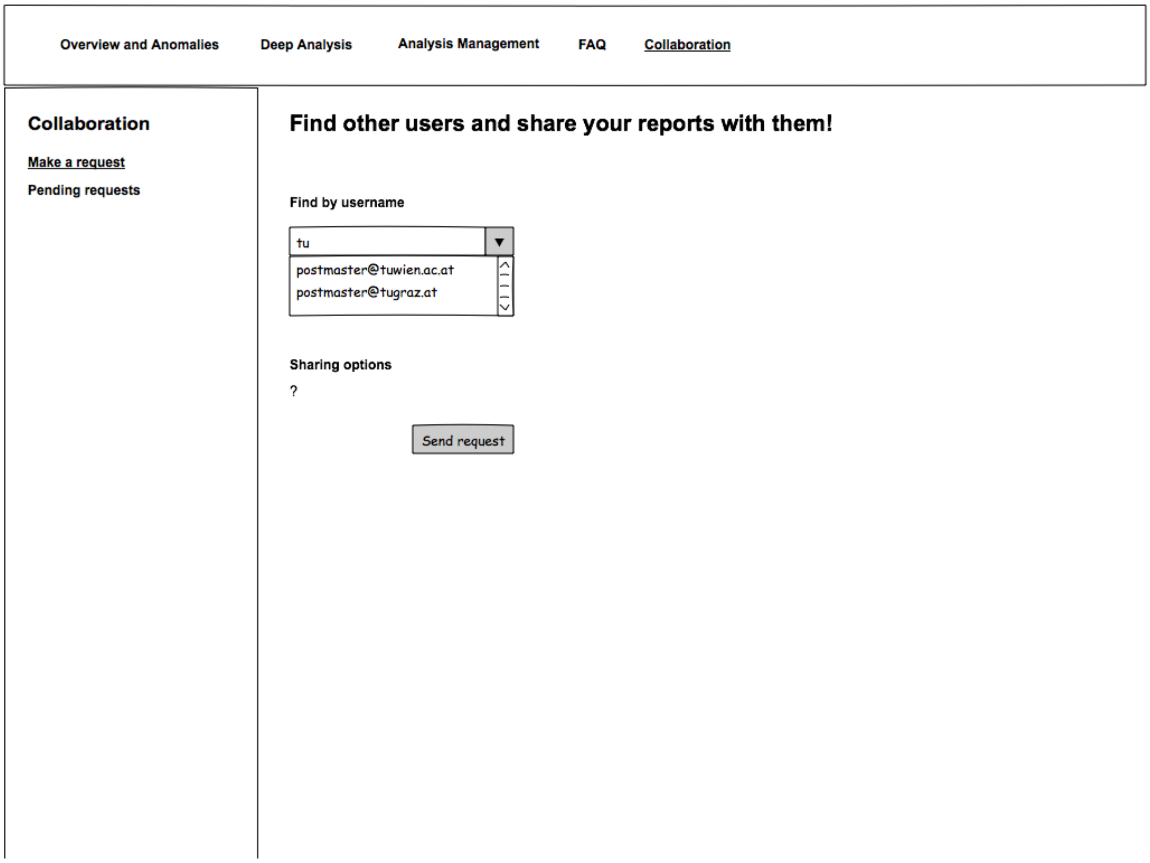
DMARC DNS example: "v=DMARC1\; p=none\; rua=mailto:ruai-asdfjlasdfjlk23@mydmarc.at"

RUAo

ruao-asdfjlasdfjlk23@mydmarc.at

If you also want to analyze the reports you would send to other mail providers (outgoing reports), customize your DMARC processing so that it sends a copy of the report to the here provided RUAo address.

Figure 5.11: Wireframe – *DMARC viewer* – welcome page



Overview and Anomalies Deep Analysis Analysis Management FAQ Collaboration

Collaboration

Make a request

Pending requests

Find other users and share your reports with them!

Find by username: tu

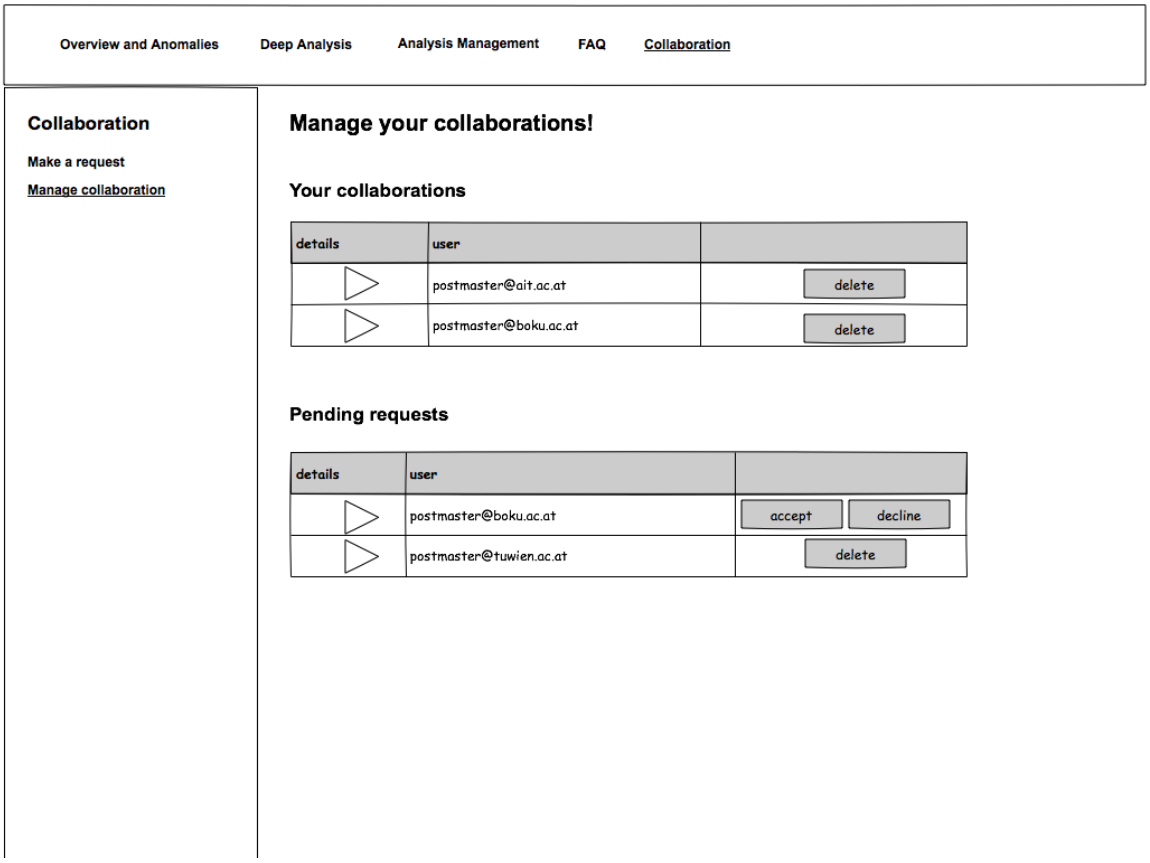
postmaster@tuwien.ac.at

postmaster@tugraz.at

Sharing options

?

Figure 5.12: Wireframe – *DMARC viewer* – request collaboration



Manage your collaborations!

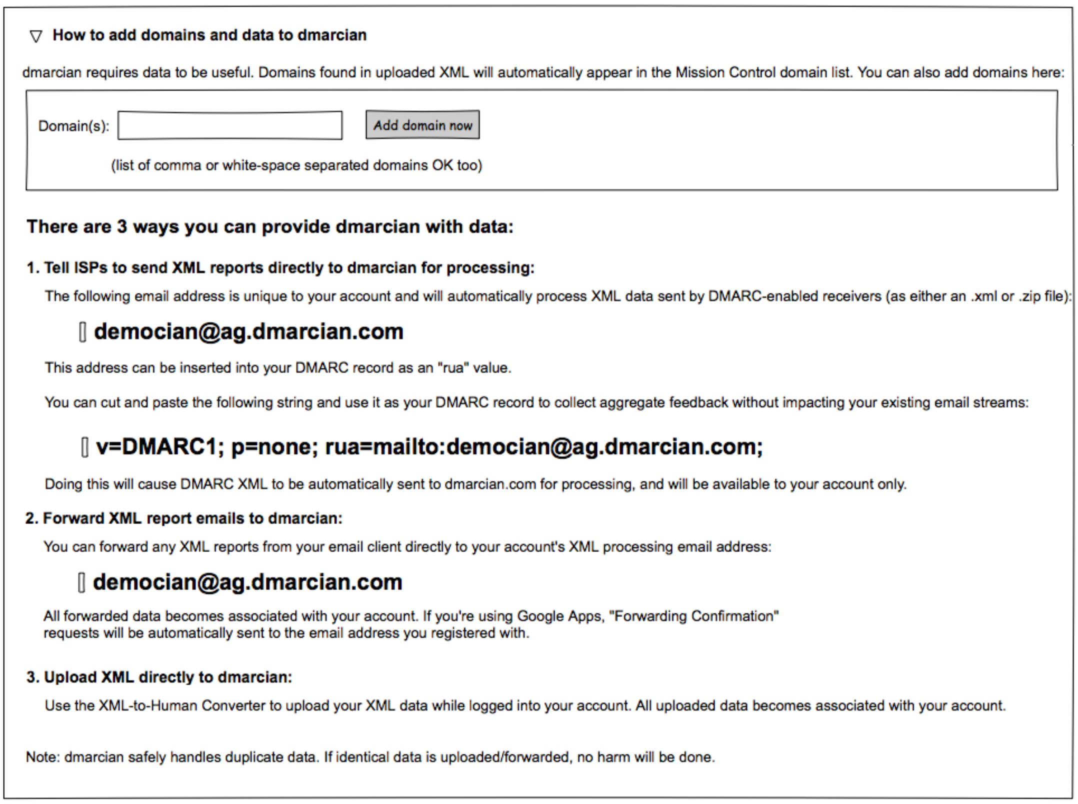
Your collaborations

details	user	
▷	postmaster@ait.ac.at	delete
▷	postmaster@boku.ac.at	delete

Pending requests

details	user	
▷	postmaster@boku.ac.at	accept decline
▷	postmaster@tuwien.ac.at	delete

Figure 5.13: Wireframe – *DMARC viewer* – manage collaboration



▽ How to add domains and data to dmarcian

dmarcian requires data to be useful. Domains found in uploaded XML will automatically appear in the Mission Control domain list. You can also add domains here:

Domain(s): [ ] Add domain now

(list of comma or white-space separated domains OK too)

There are 3 ways you can provide dmarcian with data:

1. Tell ISPs to send XML reports directly to dmarcian for processing:

The following email address is unique to your account and will automatically process XML data sent by DMARC-enabled receivers (as either an .xml or .zip file):

▯ **democian@ag.dmarcian.com**

This address can be inserted into your DMARC record as an "rua" value.

You can cut and paste the following string and use it as your DMARC record to collect aggregate feedback without impacting your existing email streams:

▯ **v=DMARC1; p=none; rua=mailto:democian@ag.dmarcian.com;**

Doing this will cause DMARC XML to be automatically sent to dmarcian.com for processing, and will be available to your account only.

2. Forward XML report emails to dmarcian:

You can forward any XML reports from your email client directly to your account's XML processing email address:

▯ **democian@ag.dmarcian.com**

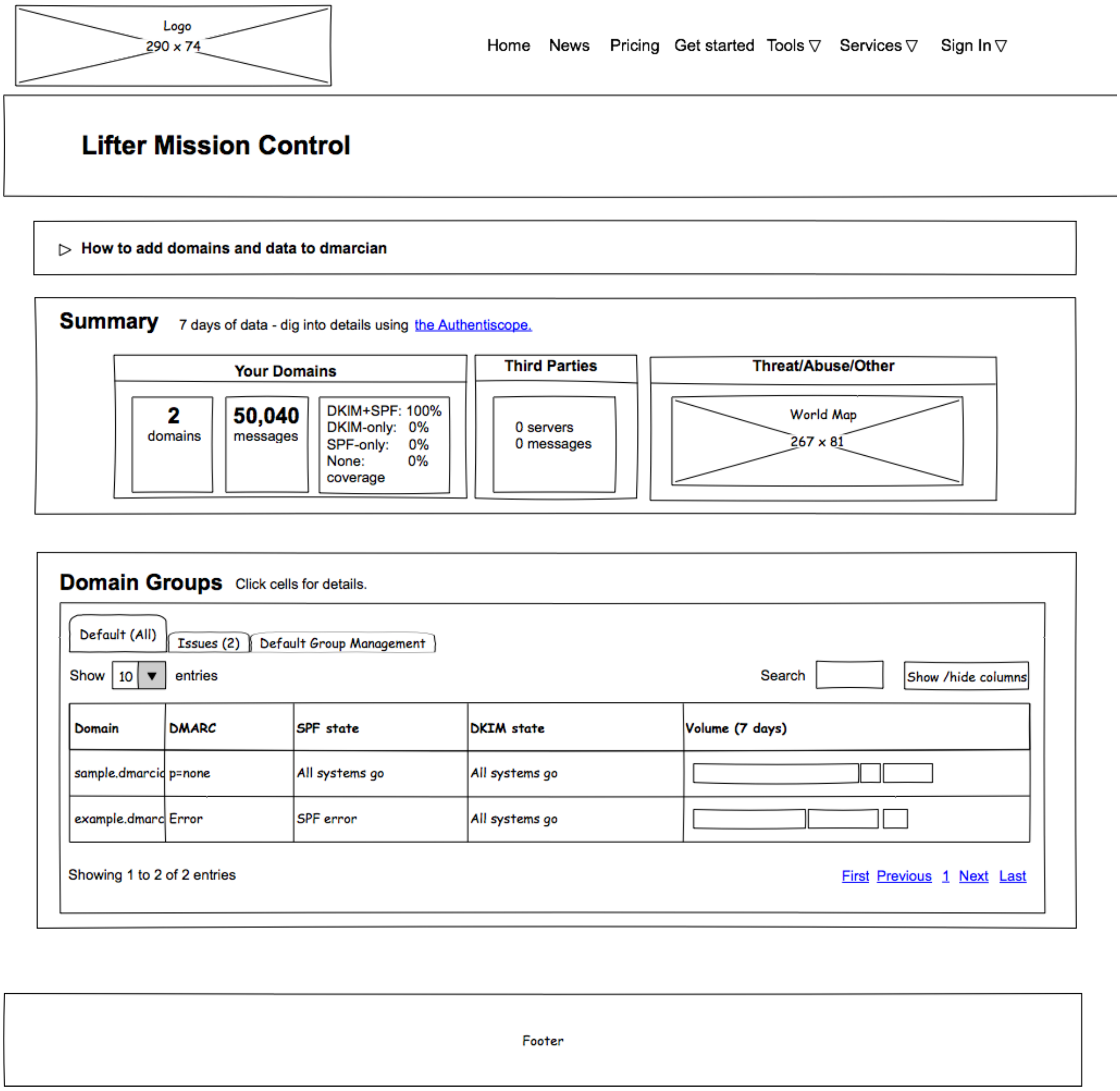
All forwarded data becomes associated with your account. If you're using Google Apps, "Forwarding Confirmation" requests will be automatically sent to the email address you registered with.

3. Upload XML directly to dmarcian:

Use the XML-to-Human Converter to upload your XML data while logged into your account. All uploaded data becomes associated with your account.

Note: dmarcian safely handles duplicate data. If identical data is uploaded/forwarded, no harm will be done.

Figure 5.14: Wireframe – dmarcian – add new domain



Lifter Mission Control

▷ How to add domains and data to dmarcian

Summary 7 days of data - dig into details using the Authentiscope.

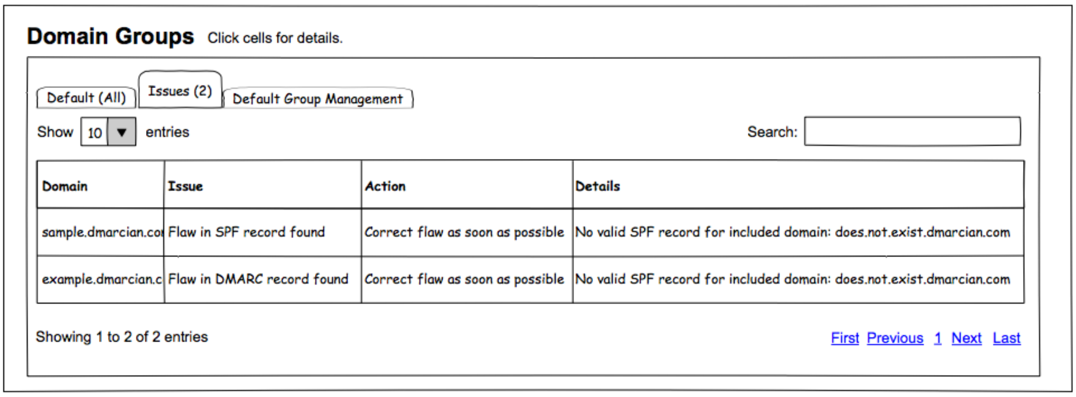
DKIM+SPF: 100%
DKIM-only: 0%
SPF-only: 0%
None: 0%
coverage

Domain Groups Click cells for details.

Domain	DMARC	SPF state	DKIM state	Volume (7 days)
sample.dmarcid	p=none	All systems go	All systems go	
example.dmarc	Error	SPF error	All systems go	

Showing 1 to 2 of 2 entries

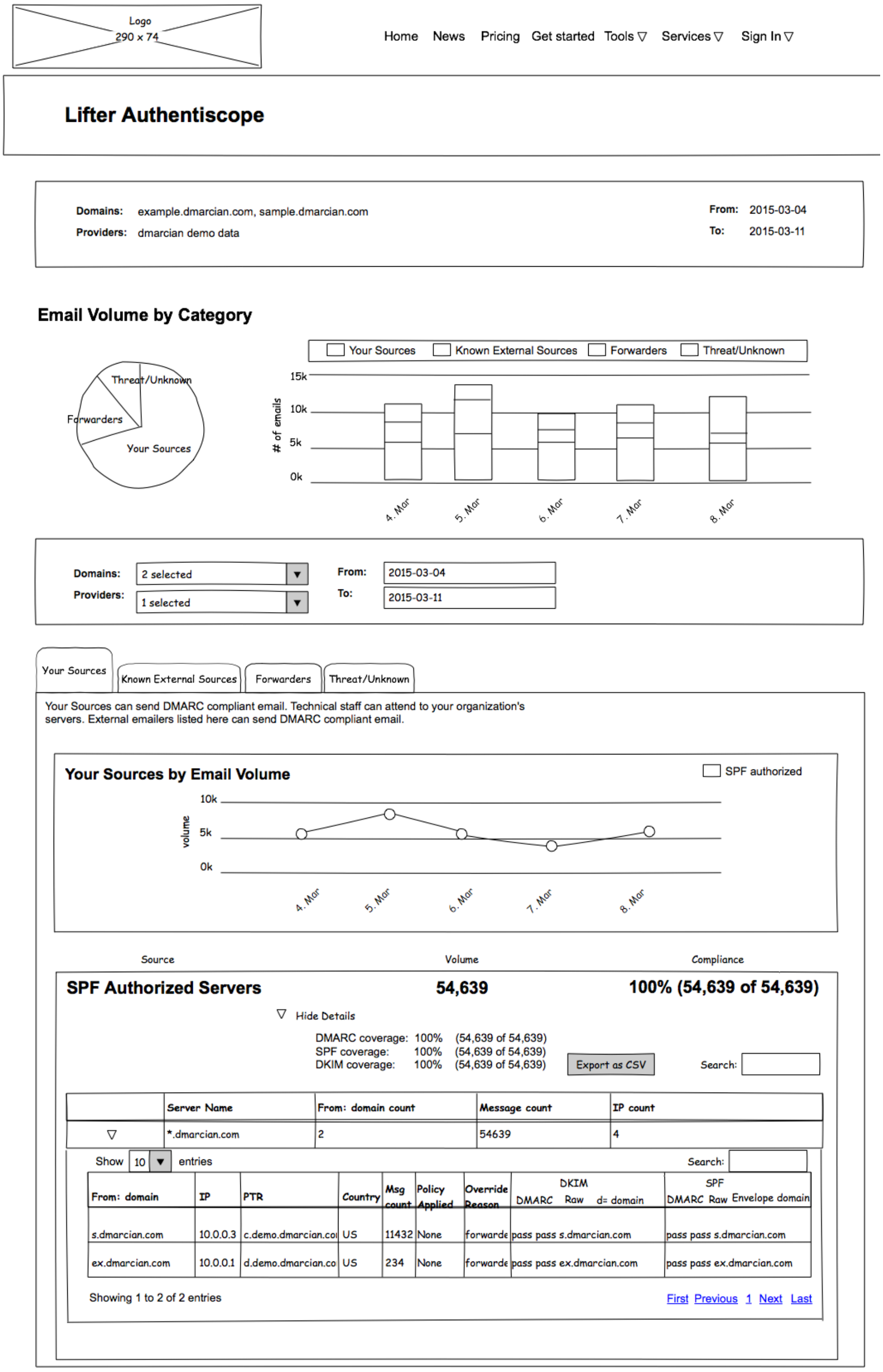
Figure 5.15: Wireframe – dmarcian – mission control



Domain	Issue	Action	Details
sample.dmarcian.co	Flaw in SPF record found	Correct flaw as soon as possible	No valid SPF record for included domain: does.not.exist.dmarcian.com
example.dmarcian.c	Flaw in DMARC record found	Correct flaw as soon as possible	No valid SPF record for included domain: does.not.exist.dmarcian.com

Showing 1 to 2 of 2 entries

Figure 5.16: Wireframe – dmarcian – mission control – issues



Domains: example.dmarcian.com, sample.dmarcian.com
Providers: dmarcian demo data
From: 2015-03-04
To: 2015-03-11

Domains: 2 selected ▼
Providers: 1 selected ▼
From: 2015-03-04
To: 2015-03-11

Your Sources can send DMARC compliant email. Technical staff can attend to your organization's servers. External emailers listed here can send DMARC compliant email.

Source	Volume	Compliance
SPF Authorized Servers	54,639	100% (54,639 of 54,639)

▽ Hide Details

DMARC coverage: 100% (54,639 of 54,639)
SPF coverage: 100% (54,639 of 54,639)
DKIM coverage: 100% (54,639 of 54,639)

Search:

	Server Name	From: domain count	Message count	IP count
▽	*.dmarcian.com	2	54639	4

Show 10 ▼ entries

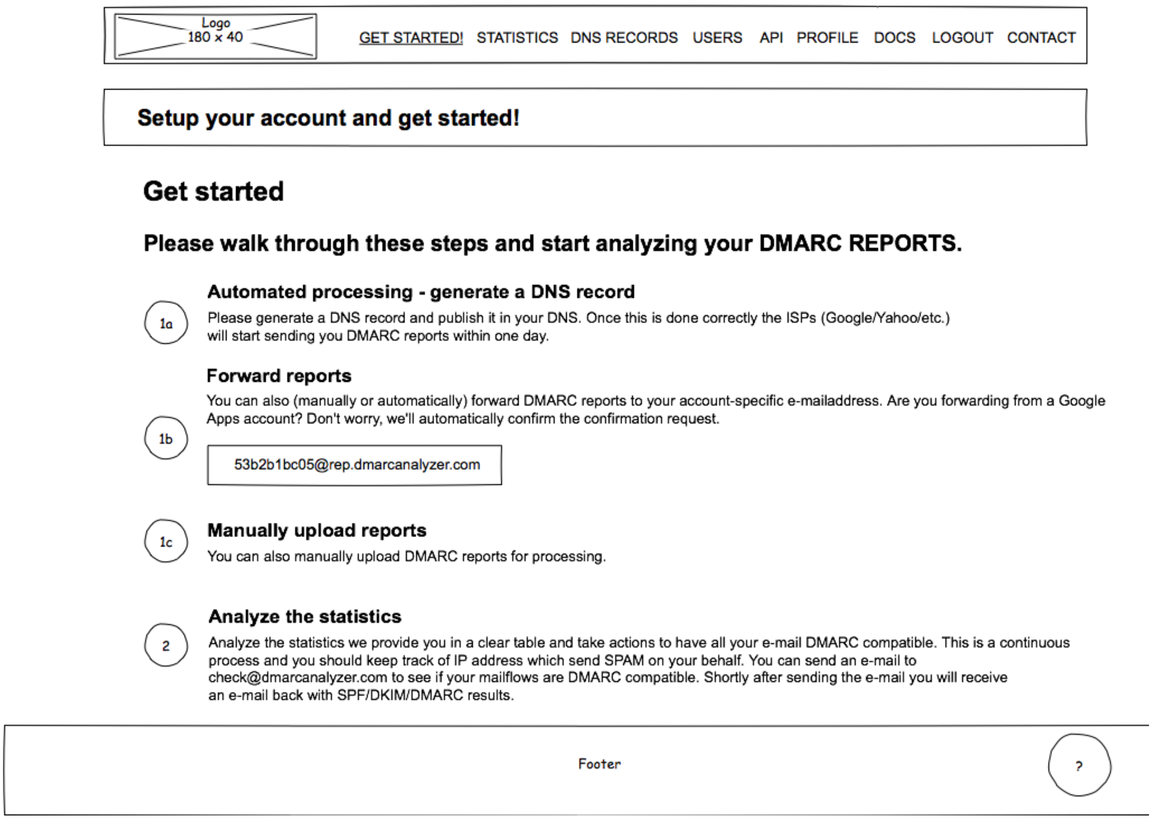
Search:

From: domain	IP	PTR	Country	Msg count	Policy Applied	Override Reason	DKIM DMARC	DKIM Raw	DKIM d= domain	SPF DMARC	SPF Raw	SPF Envelope domain
s.dmarcian.com	10.0.0.3	c.demo.dmarcian.coı	US	11432	None	forwarde	pass	pass	s.dmarcian.com	pass	pass	s.dmarcian.com
ex.dmarcian.com	10.0.0.1	d.demo.dmarcian.co	US	234	None	forwarde	pass	pass	ex.dmarcian.com	pass	pass	ex.dmarcian.com

Showing 1 to 2 of 2 entries

First Previous 1 Next Last

Figure 5.17: Wireframe – dmarcian – authentiscope



Setup your account and get started!

Get started

Please walk through these steps and start analyzing your DMARC REPORTS.

1a Automated processing - generate a DNS record

Please generate a DNS record and publish it in your DNS. Once this is done correctly the ISPs (Google/Yahoo/etc.) will start sending you DMARC reports within one day.

1b Forward reports

You can also (manually or automatically) forward DMARC reports to your account-specific e-mailaddress. Are you forwarding from a Google Apps account? Don't worry, we'll automatically confirm the confirmation request.

53b2b1bc05@rep.dmarcanalyzer.com

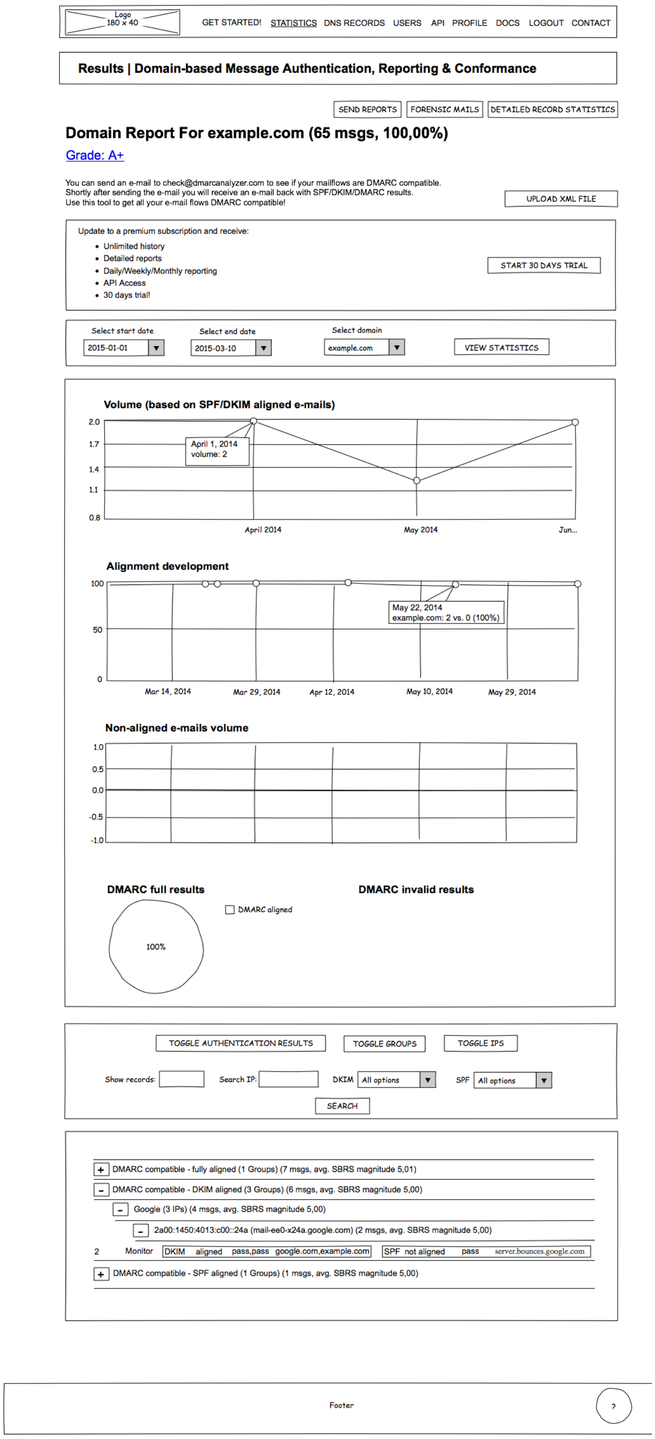
1c Manually upload reports

You can also manually upload DMARC reports for processing.

2 Analyze the statistics

Analyze the statistics we provide you in a clear table and take actions to have all your e-mail DMARC compatible. This is a continuous process and you should keep track of IP address which send SPAM on your behalf. You can send an e-mail to check@dmarcanalyzer.com to see if your mailflows are DMARC compatible. Shortly after sending the e-mail you will receive an e-mail back with SPF/DKIM/DMARC results.

Figure 5.18: Wireframe – dmarcanalyzer – get started



Results | Domain-based Message Authentication, Reporting & Conformance

Domain Report For example.com (65 msgs, 100,00%)

Grade: A+

You can send an e-mail to check@dmarcanalyzer.com to see if your mailflows are DMARC compatible.
Shortly after sending the e-mail you will receive an e-mail back with SPF/DKIM/DMARC results.
Use this tool to get all your e-mail flows DMARC compatible!

Update to a premium subscription and receive:

- Unlimited history
- Detailed reports
- Daily/Weekly/Monthly reporting
- API Access
- 30 days trial!

Select start date: 2015-01-01 ▼
Select end date: 2015-03-10 ▼
Select domain: example.com ▼

Volume (based on SPF/DKIM aligned e-mails)

April 1, 2014
volume: 2

Alignment development

May 22, 2014
example.com: 2 vs. 0 (100%)

Non-aligned e-mails volume

DMARC full results

□ DMARC aligned

DMARC invalid results

Show records:
Search IP:
DKIM: All options ▼
SPF: All options ▼

+ DMARC compatible - fully aligned (1 Groups) (7 msgs, avg. SBRS magnitude 5,01)

– DMARC compatible - DKIM aligned (3 Groups) (6 msgs, avg. SBRS magnitude 5,00)

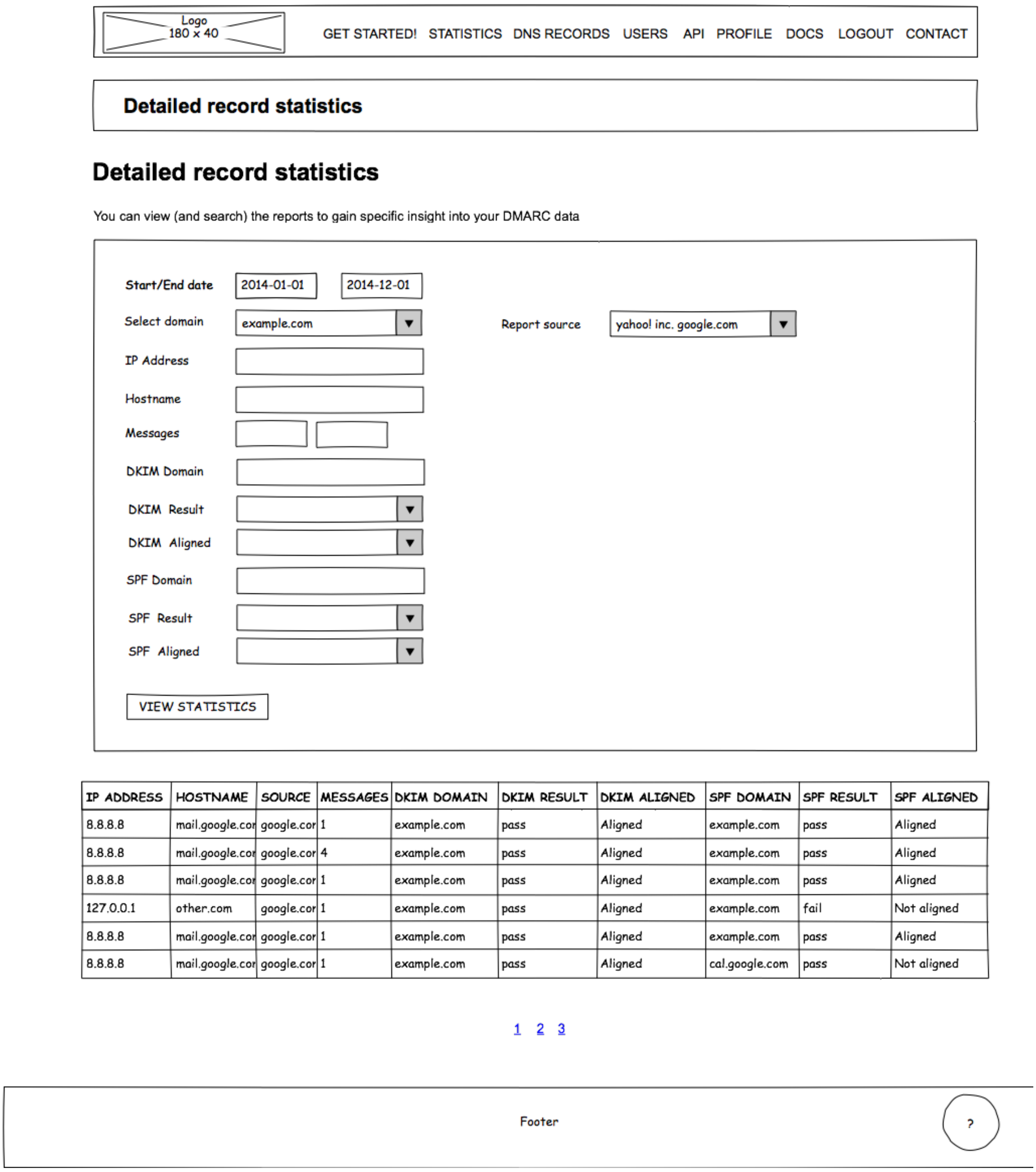
– Google (3 IPs) (4 msgs, avg. SBRS magnitude 5,00)

– 2a00:1450:4013:c00::24a (mail-ee0-x24a.google.com) (2 msgs, avg. SBRS magnitude 5,00)

2 Monitor | DKIM aligned pass,pass google.com,example.com | SPF not aligned pass server.bounces.google.com

+ DMARC compatible - SPF aligned (1 Groups) (1 msgs, avg. SBRS magnitude 5,00)

Figure 5.19: Wireframe – dmarcanalyzer – statistics



Detailed record statistics

Detailed record statistics

You can view (and search) the reports to gain specific insight into your DMARC data

Start/End date: 2014-01-01 2014-12-01

Select domain: example.com

Report source: yahoo! inc. google.com

IP Address

Hostname

Messages

DKIM Domain

DKIM Result

DKIM Aligned

SPF Domain

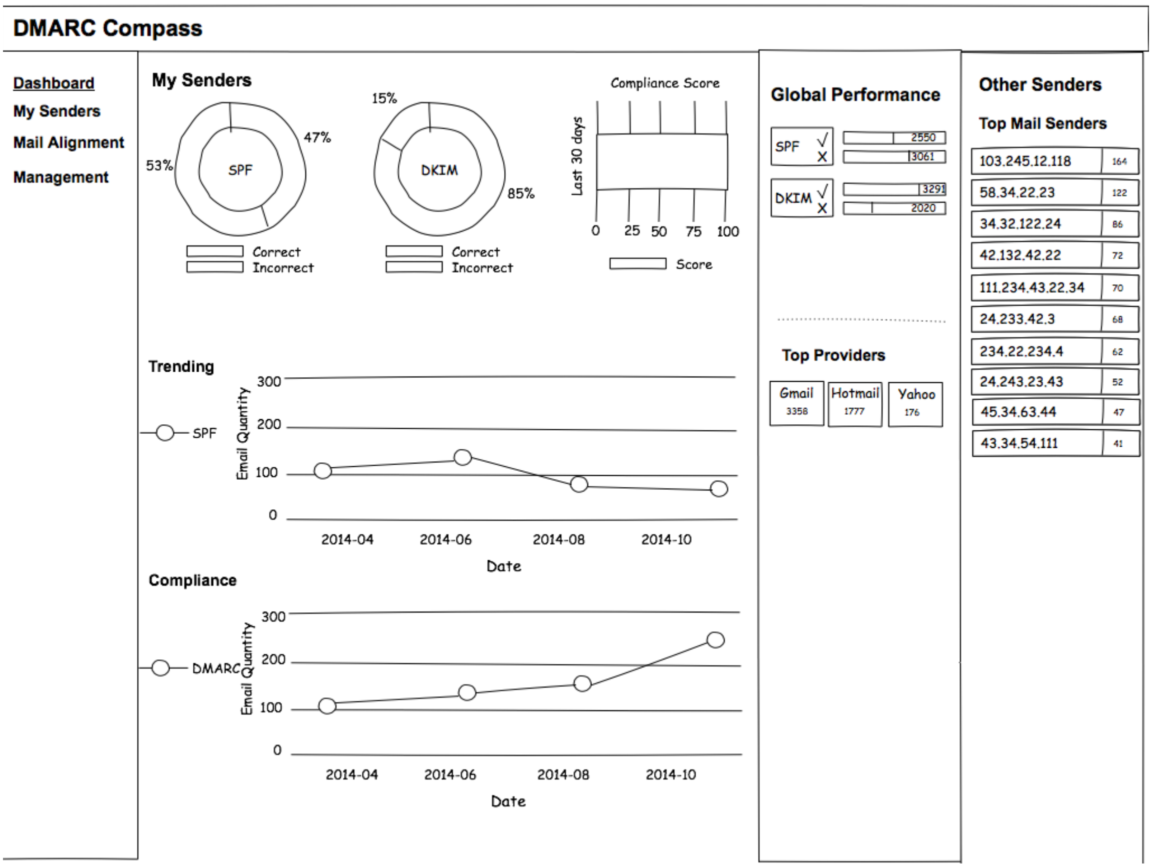
SPF Result

SPF Aligned

IP ADDRESS	HOSTNAME	SOURCE	MESSAGES	DKIM DOMAIN	DKIM RESULT	DKIM ALIGNED	SPF DOMAIN	SPF RESULT	SPF ALIGNED
8.8.8.8	mail.google.cor	google.cor	1	example.com	pass	Aligned	example.com	pass	Aligned
8.8.8.8	mail.google.cor	google.cor	4	example.com	pass	Aligned	example.com	pass	Aligned
8.8.8.8	mail.google.cor	google.cor	1	example.com	pass	Aligned	example.com	pass	Aligned
127.0.0.1	other.com	google.cor	1	example.com	pass	Aligned	example.com	fail	Not aligned
8.8.8.8	mail.google.cor	google.cor	1	example.com	pass	Aligned	example.com	pass	Aligned
8.8.8.8	mail.google.cor	google.cor	1	example.com	pass	Aligned	cal.google.com	pass	Not aligned

1 2 3

Figure 5.20: Wireframe – dmarcanalyzer – detailed record statistics



103.245.12.118	164
58.34.22.23	122
34.32.122.24	86
42.132.42.22	72
111.234.43.22.34	70
24.233.42.3	68
234.22.234.4	62
24.243.23.43	52
45.34.63.44	47
43.34.54.111	41

Figure 5.21: Wireframe – easy solutions – dashboard

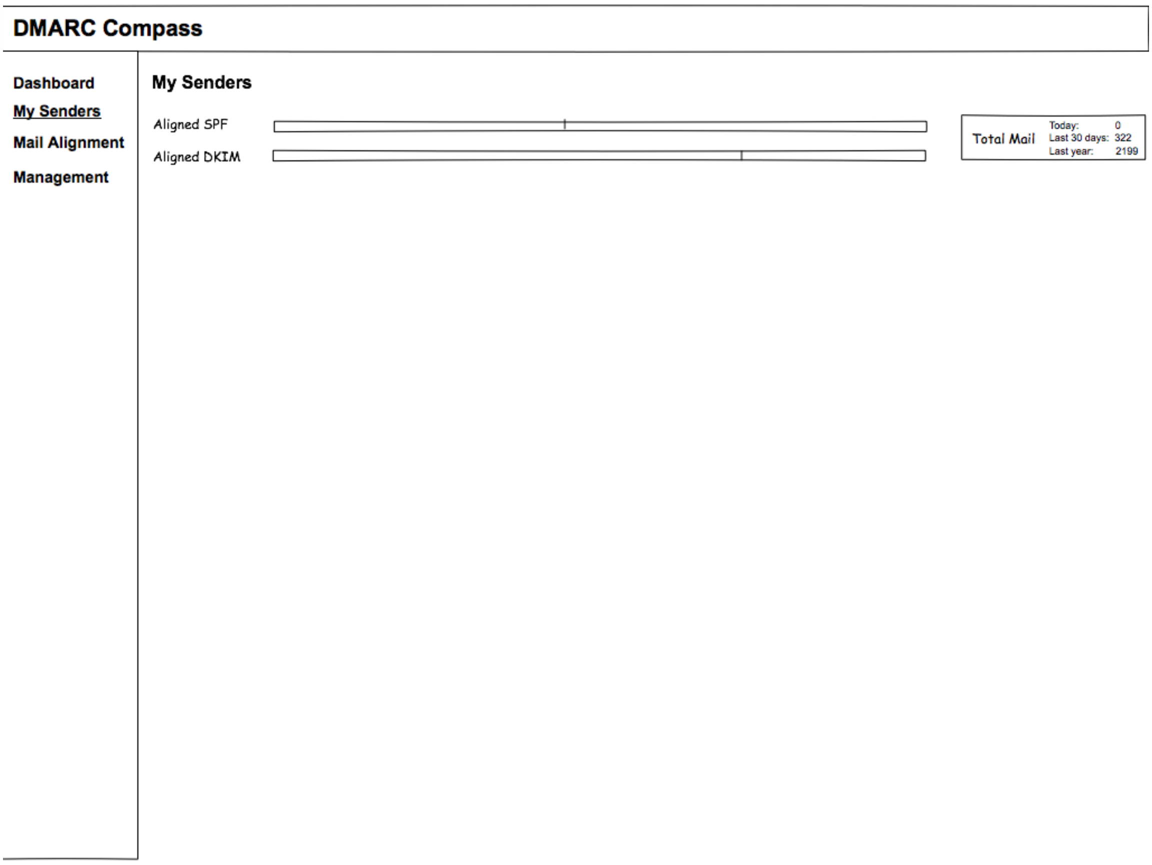


Figure 5.22: Wireframe – easy solutions – my senders

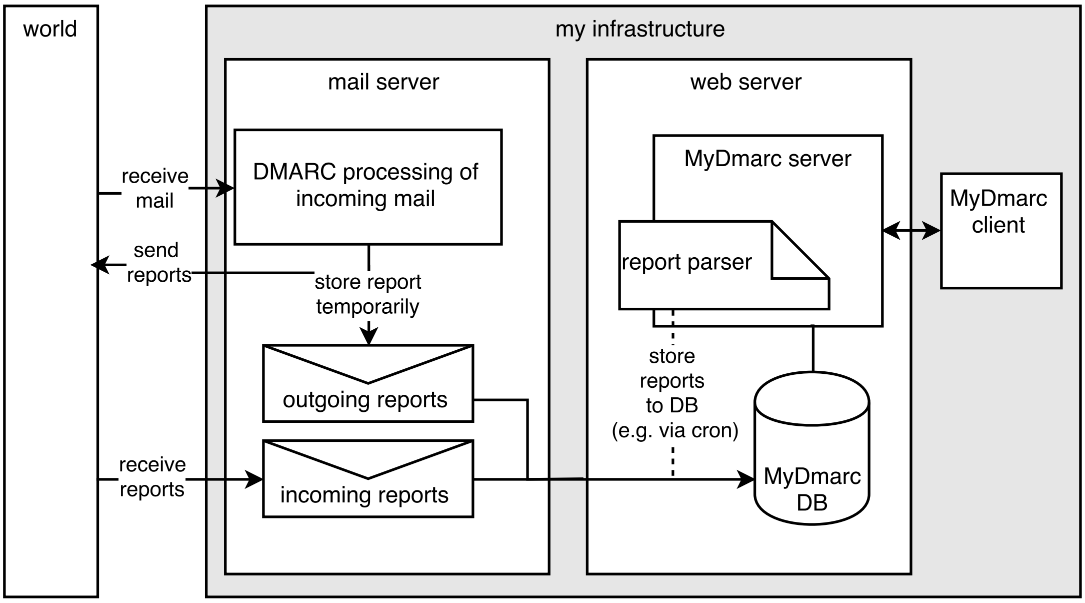


Figure 5.23: *DMARC viewer* – simple deployment

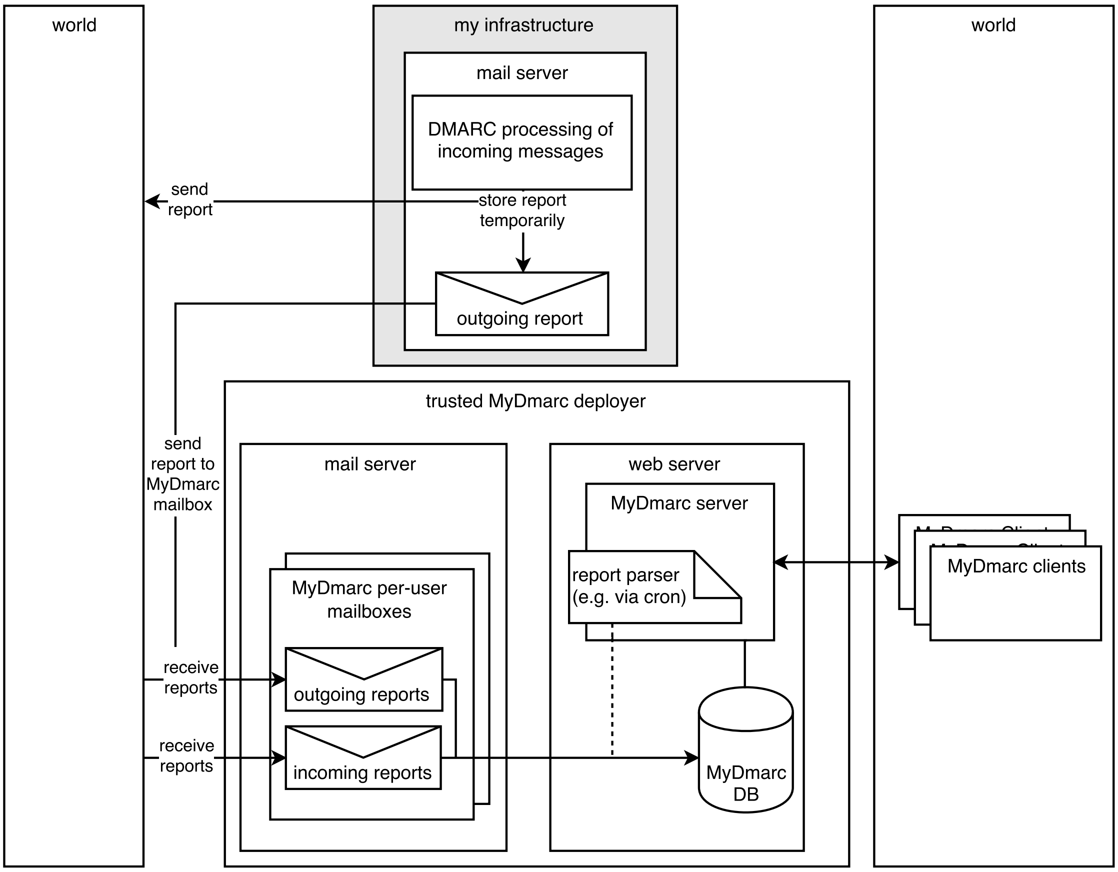


Figure 5.24: *DMARC viewer* – centralized deployment

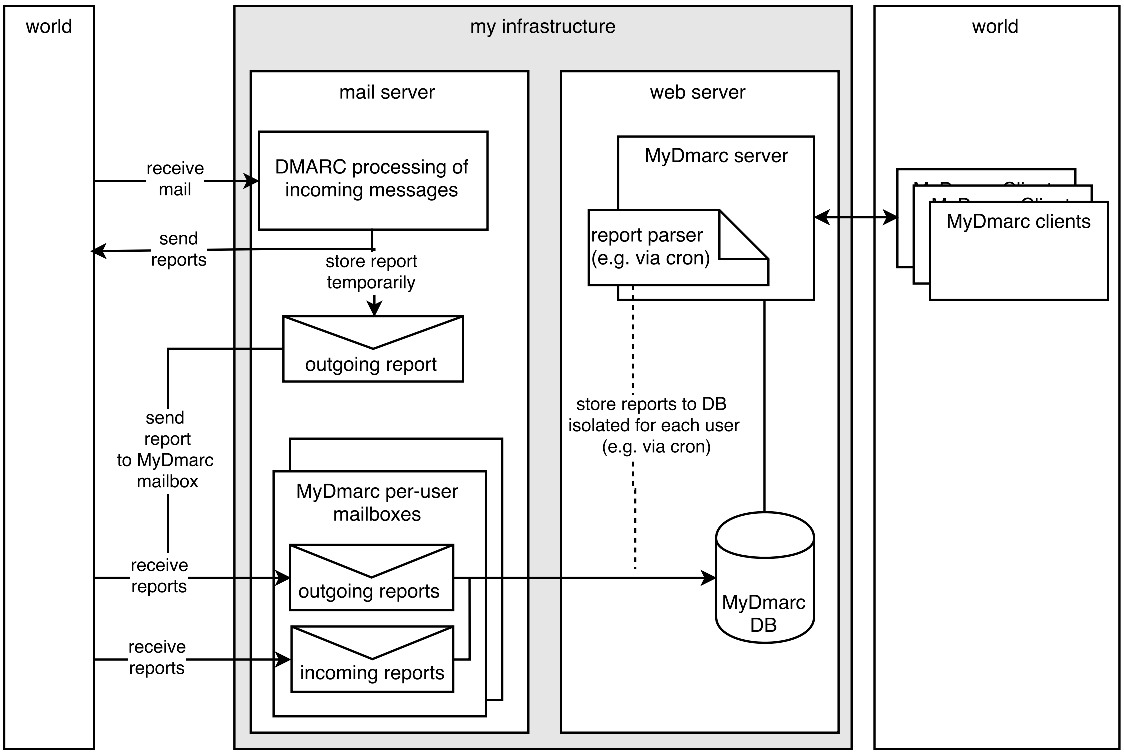


Figure 5.25: *DMARC viewer* – federated deployment

Text listing 5.3: Design evaluation study guide

Based on your statements in the first test session, I have decided that the tool should revolve around the following requirement blocks – education, custom documentation, usability and collaboration. Today's session's goal will be to validate these requirement blocks and to see if they are satisfied by the proposed design.

Walkthrough – analysis pages Let's suppose you have fed the tool with incoming (those you receive for mail that was sent with your domain) and outgoing (those you send based on mail you received) aggregate reports. Let's walk through the different wireframes and look how the data is being displayed. *Facilitator notes:* Show wireframes for pages: overview and anomalies, deep analysis, analysis management and view and filter editors

1. What would you expect on this page?
2. What is unclear?
3. If something is unclear, please check if you can find the question in the help section.
4. Is there something you would add to this page?
5. What did you find particularly positive/negative on this page?

General questions

- Is it helpful to offer predefined analysis views?
- Is it helpful to offer predefined filter sets?
- Is the separation between views and filters clear? Would you change something about it?
- How do you like the possibility to create views and filters?
- Would it be useful to directly compare multiple views of different or same view type? How could this be achieved?
- How do you like the work flow from creating views/filters to actually seeing them?
- Are the view types – time line, table, map – sufficient?
- Are there elements on the pages that would need a direct link to the help section?

Deployment scenarios Currently, I see three different deployment scenarios for the proposed tool. Each of it has different advantages and disadvantages regarding configuration complexity, data privacy and collaboration. *Facilitator notes:* Show deployment diagrams

- Which of the following deployment scenarios would you prefer? Why?
- Do you think other mail administrators would agree with you?

Walkthrough – collaboration For a federated or centralized scenario the tool would need a user handling and a way to agree with other users if and what data should be shared. *Facilitator notes:* Show wireframes for pages: sign in, create account, welcome, request collaboration and manage collaboration

1. Comment on those pages like above.

Comparison with Other Tools I have also sketched wireframes for the user interfaces of existing DMARC analysis tools in order to evaluate them regarding structure and functionality. *Facilitator notes:* Show wireframes for dmarcian, dmarcanalyzer and easy solutions.

1. Is there anything unclear?
2. Is there anything you find particularly positive? How could we integrate this in our tool?

Text listing 5.4: UI enhancements and requirement modifications

- Anomaly alerts are dismissed
- User handling requirements are dismissed
- Collaboration requirements are dismissed
- Analysis view and filter set templates are dismissed
- Every analysis view should integrate all view types (map, time line and table)
- View types can be enabled and disabled in the view editor
- Report time range and report type (incoming or outgoing) are generic filters that apply for all filter sets of a view
- Mail Sender IP address filter must be able to handle IPv6
- DMARC-specific terms should be explained using contextual help (especially in the editor)
- Nested pie charts on the overview page should be less complex
- Save buttons in editor should be on top and on bottom to avoid scrolling
- Time range form widgets should offer a date picker
- Dynamic time range form widget should offer the possibility to input a variable amount of last days, weeks and months
- Multi-select form widgets should be search-able
- It should be possible to export charts as PDF or PNG
- Elements that can be expanded or collapsed should be marked with a plus symbol
- Analysis views on the deep analysis page should give an overview of the used filters
- Charts should be linked to each other and hover-able or click-able to get more information
- Time line charts should have a grid
- Width and amount of table columns should be customizable and able to safe to a profile (not implemented due to time constraints)
- Filter fields should be join-able by different logical operators (not implemented due to time constraints)

Text listing 5.5: Prototype evaluation study guide

Description of test session

1. Three tasks to carry out using the tool (think aloud)
2. Usability questionnaire (System Usability Scale)

Usage scenario Imagine you are a mail administrator at the University of Vienna. Since the beginning of 2015 you have a DMARC DNS record for your domains, i.e. *univie.ac.at* and *unet.univie.ac.at*. You use DMARC in a *none*-policy mode, i.e. DMARC evaluation will not trigger any particular action at the mail receiver's site.
Mail receiver, however, send you DMARC aggregate reports about the DMARC evaluation results of your domain. Additionally, you evaluate DMARC results for mails that are received in your infrastructure, and you generate the according DMARC aggregate reports. Furthermore, mails sent from you infrastructure are DKIM signed and you have no SPF record for your domains. That is, mail receivers can evaluate DMARC results of mails purportedly originating from your infrastructure purely based on DKIM results.

Instructions

- Try to accomplish the following three tasks.
- Think aloud and describe your approach and your thoughts about the tasks and the tool.
- What do you find particularly positive/negative, regarding functionality and usability.
- If questions occur I will note them and we can discuss them later.
- Try to express three questions after each task that would have helped you to solve the task.

Tasks

Task 1 – Check out the system

A Incoming and outgoing DMARC aggregate reports for the year 2015 are stored in the software's database and are ready to be analyzed. In the deep analysis section of the tool there is already a predefined analysis view.
Check out the different sections of the tool and make your self acquainted with the predefined view.
Try to find out how the view is composed *(hint: see view editor)* and describe what you can see in the view.

B You now want to create a separate analysis view, which only shows mail that you would have rejected in 2015. *(hint: disposition == reject)*.
Create a new view based on the existing one, without changing it *(hint: clone)*.

Task 2 – Am I DMARC ready? So far you have operated DMARC in a *none*-policy mode. That is mail receivers should deliver mails originating from your domain, independently of the evaluated DMARC result. Now you want to find out, if you can request a more restrictive policy from you mail receiver, in the case of a negative DMARC result.
Create a new analysis view, that helps you with this decision. *(hint: Comparison of DMARC fail and pass)*

Task 3 – Raw DKIM

A The DMARC evaluation of your infrastructure is purely based on DKIM. That is why you are interested in your raw DKIM authentication results, based on which DMARC's aligned DKIM results are assessed.
Create a view that compares some — *neutral, temperror, permerror* — raw DKIM results for those mails that had *unet.univie.ac.at* as sender domain *(hint: report receiver domain == unet.univie.ac.at).*

B You are especially interested in mails that apparently had syntax errors in the DKIM signature *(hint: raw DKIM Result == neutral).*
Examine the peaks in the time line. Who received these mails?
Create a new view, which shows you whether this reporter regularly finds syntax errors in your mail signatures.

	I strongly agree				I strongly disagree
I think that I would like to use this system frequently.	1	2	3	4	5
I found the system unnecessarily complex.	1	2	3	4	5
I thought the system was easy to use.	1	2	3	4	5
I think that I would need the support of a technical person to be able to use this system.	1	2	3	4	5
I found the various functions in this system were well integrated.	1	2	3	4	5
I thought there was too much inconsistency in this system.	1	2	3	4	5
I would imagine that most people would learn to use this system very quickly.	1	2	3	4	5
I found the system very cumbersome to use.	1	2	3	4	5
I felt very confident using the system.	1	2	3	4	5
I needed to learn a lot of things before I could get going with this system.	1	2	3	4	5

Table 5.1: SUS [8]

Text listing 5.6: UI issues and proposed enhancements per page

- General
  - Don’t highlight analysis view management menu item when the user is in the view editor
  - Use same date format on all pages
- Overview page
  - Move pie chart legend boxes so that they overlap less with the charts
  - Pie chart segments should have the same order in all pies with the same categories
- Deep analysis page
  - Add plus symbol to filter details pane to indicate that it is expandable
  - Add GeoIP source reference to map
  - Disable map zoom so that the cursor isn’t captured
  - Change hover-over color contrast for countries in the map
  - Add configured filter set color to map menu items that show the label of the given filter set
  - Better indicate that the time line chart can be zoomed on x-axis and that zooming also filters the DMARC report records shown in the table
  - Condense table
  - Replace help texts with tooltips
- View editor page
  - Create tooltips for all filter fields
  - Indicate that change of report type removes values from fields: Reportee(s), Reporter(s), SPF Domain(s), DKIM Domain(s)
  - Revise filter field labels (especially time range filter field)
  - There should always be at least one filter set in an empty view editor form
  - Ask for confirmation on view or filter set deletion
- Help page
  - Add diagram of DMARC evaluation flow

Listing 5.1: DMARC aggregate report example [13, par. “I need to implement aggregate reports, what do they look like?”]

```
<?xml version=”1.0” encoding=”UTF—8” ?>
<feedback>
  <report_metadata>
    <org_name>acme.com</org_name>
    <email>noreply—dmarc—support@acme.com</email>
    <extra_contact_info>http://acme.com/dmarc/support</extra_contact_info>
    <report_id>9391651994964116463</report_id>
    <date_range>
      <begin>1335571200</begin>
      <end>1335657599</end>
    </date_range>
  </report_metadata>
  <policy_published>
    <domain>example.com</domain>
    <adkim>r</adkim>
    <aspf>r</aspf>
    <p>none</p>
    <sp>none</sp>
    <pct>100</pct>
  </policy_published>
  <record>
    <row>
      <source_ip>72.150.241.94</source_ip>
      <count>2</count>
      <policy_evaluated>
        <disposition>none</disposition>
        <dkim>fail</dkim>
        <spf>pass</spf>
      </policy_evaluated>
    </row>
    <identifiers>
      <header_from>example.com</header_from>
    </identifiers>
    <auth_results>
      <dkim>
        <domain>example.com</domain>
        <result>fail</result>
        <human_result></human_result>
      </dkim>
      <spf>
        <domain>example.com</domain>
        <result>pass</result>
      </spf>
    </auth_results>
  </record>
</feedback>
```