SIMARGL

D 5.2 User interface

Work Package 5: Design and Development of an intelligent and holistic SIMARGL toolkit (services)

Document Dissemination Level

P	Public	☒
CO	Confidential, only for members of the Consortium (including the Commission Services)	☐

Document Due Date: 31/10/2020
Document Submission Date: 30/10/2020

This work is performed within the SIMARGL Project – Secure Intelligent Methods for Advanced RecoGnition of malware and stegomalware – with the support of the European Commission and the Horizon 2020 Program, under Grant Agreement No 833042

Document Information

Deliverable number:	**5.2**
Deliverable title:	User interface
Deliverable version:	1.0
Work Package number:	5
Work Package title:	Design and Development of intelligent and holistic SIMARGL toolkit (services)
Due Date of delivery:	31/10/2020
Actual date of delivery:	30/10/2020
Dissemination level:	PU
Editor(s):	netzfactor (deliverable responsible)
Contributor(s):	netzfactor (deliverable responsible) RoEduNet
Reviewer(s):	FUH ITTI
Project name:	Secure Intelligent Methods for Advanced RecoGnition of malware and stegomalware
Project Acronym	SIMARGL
Project starting date:	1/5/2019
Project duration:	36 months
Rights:	SIMARGL Consortium

Version History

Version	Date	Beneficiary	Description
0.1	11/03/2020	netzfactor	Template applied, table of contents
0.2	03/04/2020	netzfactor	Added first content for chapter 2
0.3	28/04/2020	netzfactor	Added further content for chapter 2
0.4	14/05/2020	netzfactor	Added further content for chapter 2
0.5	03/09/2020	netzfactor	Added content for chapters 3, 4, 5
0.6	17/09/2020	netzfactor	Added content for chapters 3, 4
0.7	30/09/2020	netzfactor	Added content for chapter 4
0.8	09/10/2020	netzfactor	Added content for chapters 3, 4, 5
0.9	15/10/2000	netzfactor	Added content for chapters 1, 4, 5
0.10	21/10/2020	netzfactor	Add content for chapters 1, 3, 4
0.11	22/10/2020	RoEduNet	Add content for chapter 5
0.12	23/10/2020	netzfactor	Added content for chapters 4, 6

0.13	27/10/2020	netzfactor	Improved content in chapters 2, 3, 4, 5 after first review
1.0	29/10/2020	netzfactor	Final version

Abbreviations and Acronyms

ACRONYM	EXPLANATION
API	Application Programming Interface
CSV	Comma separated values
CURL	Client for URLs, also known as, *Curl* URL Request Library
ELK	Elastic Stack
ES	Elasticsearch
FOSS	Free and Open Source Software
FTP	File Transfer Protocol
FTPS	File Transfer Protocol over SSL
GUI	Graphical User Interface
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
ISP	Internet Service Provider
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
KQL	Kibana Query Language
LEEF	Log Event Extended Format
MIB	Management Information Base
MIME	Multipurpose Internet Mail Extensions
NPMD	Network Performance Monitoring and Diagnostic
PING	Packet Internet (Inter-Network) Groper
REST	Representational State Transfer
SFTP	SSH (Secure Shell) File Transfer Protocol
SIEM	Security Information and Event Management
SNMP	Simple Network Management Protocol
SOC	Security Operation Centre
SSH	Secure Shell
SSL	Secure Sockets Layer
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UX	User Experience
WS	Web Services
WSS	Web Services Security
WWW	World Wide Web
YAML	YAML Ain't Markup Language

Table of Content

Table of Figures

Table of Tables

1. Introduction

This document focuses on the development of the user interface for the SIMARGL toolkit. It describes the current state of the art in user interface design for tool- and web-based software solutions and data visualization, especially in the context of cyber security. It summarises the different user roles with their specific tasks and requirements, typical use cases along with the main challenges for the UI/UX development in SIMARGL and presents the developed UI concept for the SIMARGL toolkit, derived from all preparatory work.

1.1 Overview

This document presents the UI concept for the SIMARGL toolkit and is structured as follows:

- Section 1: Introduction
- Section 2: Related Work
- Section 3: Data visualization challenges in SIMARGL
- Section 4: UI/UX concept for SIMARGL toolkit
- Section 5: Evaluation
- Section 6: Conclusions

In the first section the document's overview, scope and contents are described. The second section focuses on related work and summarises the current state of the art in UI/UX design with a specific view on cyber security. In the third section, all preparational steps are summarised and explained. This includes a description of the chosen approach and the applied methodologies during the development process, the main use cases and the important aspects along with the challenges to be considered. The fourth section presents the developed UI/UX concept for the SIMARGL toolkit with all of its functions and features. It focuses on the globally applied design principles, the main dashboard for the overall overview of the system and of the specific dashboards for the investigation of incidents. A special section is also dedicated to the integration and connection of different tools and to the UI for the configuration. In the fifth section, an overview of the evaluation plan for the developed UI is provided. The last section presents this document's conclusions.

1.2 Scope

The scope of this document is the presentation of the UI concept for the SIMARGL toolkit. It serves as the guideline for the further implementation and integration of the UI in the following steps of the project. This document depends partially on the collected use cases and requirements for SIMARGL in D2.2: "SIMARGL requirements and use cases", on the architecture of the SIMARGL toolkit described in D2.5: "Final version of the SIMARGL toolkit architecture", on the available data sources and types in SIMARGL summarised in D4.2: "SIMARGL Data Production" and on the development of the data fusion module presented in D5.1: "Data Fusion Module".

2. Related Work

In this section the state-of-the-art of UI and UX design is considered (section 2.1) based on the work from Nielsen [1], Krug [2, 3], and Ash [5]. Additionally, the user experience methods AttrakDiff [10, 11, 12] and CUE model [13, 19, 20] are described. Section 2.2 contains roles and task for cyber security operators, requirements for UI in the cyber security context, and a description of work flow support. Then, section 2.3 depicts typical data types for visualization, an overview on forms for visualization based on Grafana [14] and Kibana [15], and a summary for UI/UX as success factor for cyber security software solutions.

2.1 State-of-the-art of UI/UX design

2.1.1 Usability for tool- and Web-based user interfaces

Goal of usability investigation is to make tool- and Web-based user interfaces intuitively usable. This requires many properties that are described in the following.

2.1.1.1 Usability heuristics of Nielsen

Nielsen provided 10 usability heuristics that are also known as usability principles [1]. These usability heuristics represent basic characteristics for usable interfaces:

1. Simple and Natural Dialogue: The system should use natural words and not technical or system-oriented terms, and express itself in natural phrases.
2. Speak the users' language: Dialogues should be adapted to the users (and not vice versa). For example, dialogues for lawyers or medical doctors should not contain technical terms, however for computer scientist they might enabling a more accurate expression.
3. Minimize the user memory load: If the user has to learn by heart the menu structure or the location of specific contents, then navigation may become effortful and tedious.
4. Consistency: Different words, actions, and situations that mean different things should be unique for users.
5. Feedback: The system should inform the user about each interaction and what will happen. Furthermore, the user should get feedback, when the system performed actions, whether they are successful or not. An action may be the deletion of a file.
6. Clearly marked exits: Dialogue boxes should have a clear exit button. A further feature is the ESC button to quit an action. This provides the feeling that the user is in control.
7. Shortcuts: Many users like shortcuts, e.g., Strg-Alt-Del for a quick screen lock. When mouse hovering is enabled, then the shortcuts should be displayed, if the mouse cursor reaches a corresponding region on the display.
8. Good error messages: Error messages should be provided in clearly, understandable phrases. They should not contain cryptic error codes. Better is to provide also a short description of how to fix or prevent the repetition of the error.
9. Prevent errors: Error situations should be prevented, e.g., spelling of names and addresses may lead to errors (and is tedious). Thus, for names should nicknames be used, and for addresses auto-completion.
10. Help and documentation: A ubiquitous help function, like the F1 button for context-sensitive help from Microsoft, should be contained in the system. The help text should provide also proposals for further resources in the Internet.

2.1.1.2 Laws of Krug

Krug developed ten so-called laws for Web design [2] and [3]. These laws are guard rails for design, and describe how to think during design:

1. Usability means that something, e.g., a process, in an easy way. Any average human should be able to use it, without being frustrated.
2. The former point implies that Webpages should be self-explanatory and the content should be apparent. This is described in the following bullets.
3. Pivotal is the requirement “Don’t make me think!”. Humans don’t like to think about how to use something. Additionally, the brain may add information (Figure 1).

Figure 1: The brain may add information [1].

4. Web users intend to save time, especially, when using the Internet. This requires well-structured content and processes.
5. The most used feature of a Webpage is the “back” button. Users prefer to press the back button, if they need to investigate the current Webpage, in order to understand how to use the Webpage.
6. Web users are browsing always in the same manner that is useful for them. They only change their behaviour, i.e., improving or optimizing the way how they are browsing, by accident. Users are merely not looking actively for improvements.
7. In the Internet are “Happy Talk” or “Small Talk” inconvenient. As example, consider a search form: Users are familiar with the button “Search”. A button “What are you looking for?” might be politer, but this kind of “Small Talk” is odd in the Internet. A Webpage should contain as less “Small Talk” as possible.
8. The search function should not be underestimated: Many users apply immediately the search form on the landing page. Thus, the search button should be easy to find, or always at the same position (Figure 2).

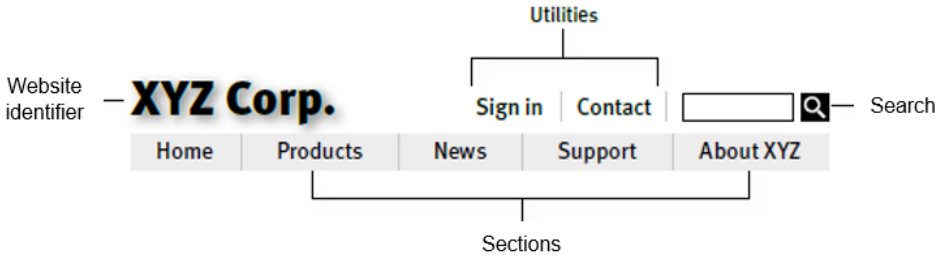


Figure 2: Typical elements of a Webpage and their positions [1].

9. Within Webpages it is difficult to know where you are. Users try to construct mentally a sitemap, similar to moving through a city. The sitemap construction should be supported by an easy navigation, i.e., maximum depth of three, or long pages instead of jumps.
10. The most viable button is the "home" button. Users should be able to find and use the home button intuitively (Figure 2).

2.1.1.3 Guideline from Ash

Pivotal for Ash's research is the optimization of the landing page [5]. The underlying method is A/B testing, aka split testing: The Webpage is presented in two versions to users. The origin version (A) and a variant (B). As quantitative quality measure the conversion rate is determined. The conversion rate is the number of all conversions divided by the number of user visits. As example consider a shop with 10 orders and 100 customer visits during the 10 orders. Then, the conversion rate is 100 divided 10, 10%. This is determined for the versions A and B, and the one with the higher conversion rate is chosen. In order to create a variant B, a hypothesis is required, e.g., a purchase button with a higher contrast like red is increasing the number of purchases (Figure 3).

Figure 3: Two Webpages: Variant A (origin, left) and variant B with red button (right) [6].

During a certain time phase the number of customer visits and purchases are measured for both variants. The one with the higher conversion rate is chosen. This quantitative measure provides a numerical evaluation. Further Web metrics are provided in [7], which forms also the basis for Ash's work.
The challenge is to find the right hypothesis. Initially, the users, respectively customers, need to be understood. Helpful methods are, e.g., Web analytics to get insights into the traffic and Webpage usage and the social behaviour of users [8]. The latter one requires a classification of user types. A common and widely adopted model is the Myers-Briggs type indicators (MBTI) [16, 9]. The MBTI determines the individuals innate behavioural style based on four different dimensions: I (Introversion) – E (Extraversion), N (Intuition) – S (Sensing), F (Feeling) – T (Thinking), J (Judging) – P (Perceiving), resulting in 2 x 2 x 2 x 2 = 16 types. These types are based on work of Keirsey [17]. In order to increase the reliability of the MBTI, it might be useful to repeat the classification. Understanding the users and knowing their social behavioural styles enables a user-centred design with a successful user experience also within technical areas like cyber security.

2.1.2 User Experience

In the following the user experience evaluation method AttrakDiff and the structured method CUE model to gather user experience are described.

2.1.2.1 AttrakDiff

Tractinsky and Hassenzahl state that user experience should comprise records of three dimensions [11, 12]:

- The perceived pragmatic quality
- The hedonic quality
- The attractiveness of a Web tool, interface, product, Web service

Hassenzahl summarises this as the hedonic, emotional and experimental perspective [10]. As an example consider Figure 4: A prototype of a Web service P has been evaluated with respect to its hedonic and pragmatic quality (green rectangle in Figure 4). The Web see P evaluation shows that P fulfils both perspectives as desired, and thus, there is little need for improvement. The green rectangle is also called confidence rectangle according to user consensus [18]. The rectangle is placed for 2/3 in the square desired, and for 1/3 in the square too self-oriented. This example indicates a unique result for the evaluation with AttrakDiff.

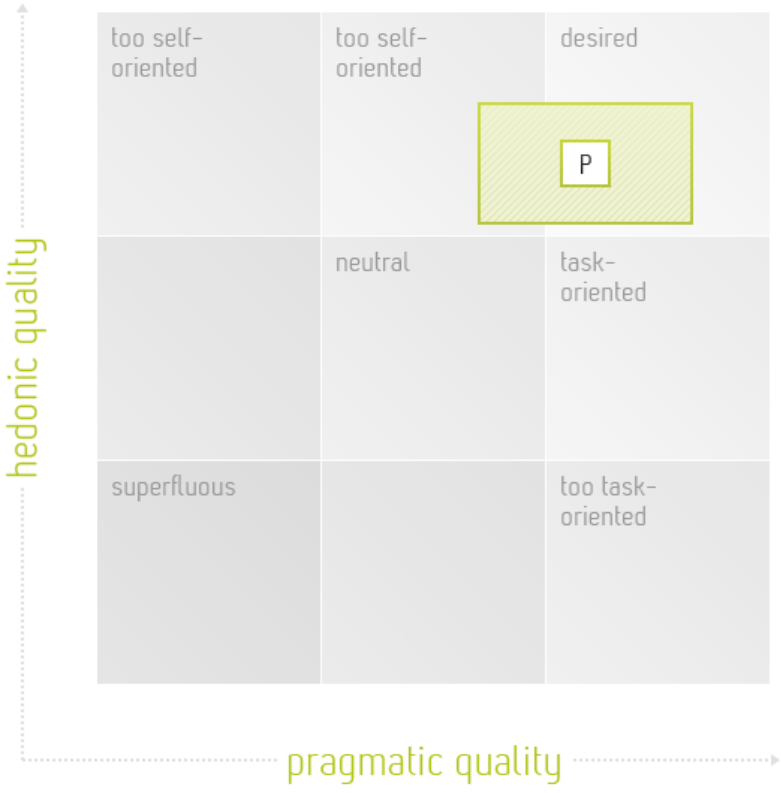


Figure 4: Example for a single evaluation with hedonic and pragmatic perspective [18].

The work model of AttrakDiff is based on the influence of hedonic and pragmatic qualities on the subjective perception of attractiveness. In total 28 measurements are aimed to perform on semantic level like “confusing - clear” and “unusual - ordinary”. The measurements provide evaluations of the following model aspects: The design, i.e., the intended product quality, the subjective perception of quality, the pragmatic and hedonic qualities, and the behavioural and emotional consequences [10, 11, 12].

2.1.2.2 CUE model

The preference of people depends heavily on factors like aesthetic quality and emotional experiences [19]. Minge and Thüring postulate that "The concept of user experience can be characterized as a multidimensional phenomenon that comprises both, the perception of different product qualities as well as emotions that arise while a user interacts with a product…" [20]. The authors implemented an analytical component model of user experience by a structured questionnaire, the so-called CUE model [13].

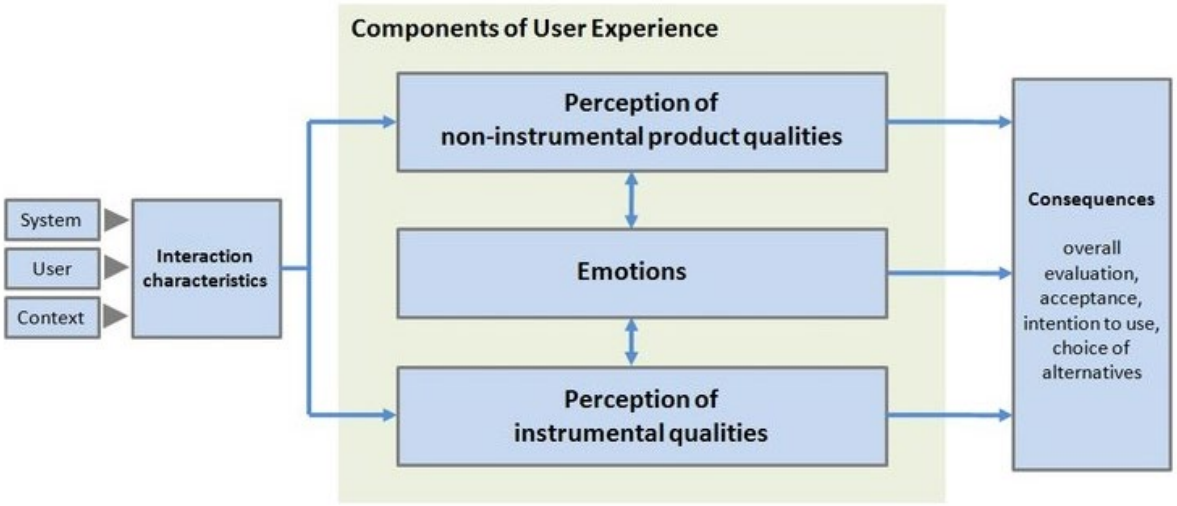


Figure 5: The CUE model [13].

Minge and Mahlke argue like Tractinsky and Hassenzahl [11, 12] that user experience is a broad compound of the following three aspects [19]:

1. The perception of instrumental qualities, such as the controllability or the effectiveness of a system,
2. The perception of non-instrumental qualities, such as visual aesthetics or haptic quality, and
3. The user's emotional responses to system behaviour.

These aspects are the main components of user experience and of the CUE model (Figure 5). The questionnaire consists of five modules, in which a Web service can also be any tool, product or interface:

1. The perception of instrumental Web service qualities describes the usefulness and usability.
2. The perception of non-instrumental Web service qualities reflects visual aesthetics.
3. Emotions are distinguished positively and negatively.
4. Consequences like intension of use.
5. The overall evaluation is a global view on the Web service.

In total 34 questions covering the five modules are asked with a discrete scale like "-5" to "5". The authors provide also an Excel file, where the answers of the probands can be filled in and the means and standard deviations can be computed. The consequences lead to a holistic evaluation comprising the acceptance and choice of alternatives (cf. Figure 5).

2.2 User roles in the context of cyber security

Users and experts need to adopt different roles in the context of cyber security. These roles are described in section 2.2.1 and the described tasks imply requirements for user interfaces in the cyber security context (section 2.2.2). The work flow support for cyber security systems is analysed in section 2.2.3 based on the technical frameworks Grafana [14] and Kibana [15].

2.2.1 Roles and tasks

The most effective and worldwide operating cyber security centre [21] consists of three different services for the detection of anomalies, responding and analysing incidents, and tool management for a security operation centre (SOC). The services are summarised in Figure 6.

DETECT	**Level 1 Analyst (seconds to minutes)** Alarm Monitoring ▪ Continuous Analysis of Events ▪ Classification/priorization ▪ Trigger Incidents ▪ If req. escalation to Level 2	**Level 2 Analyst (minutes to hours)** Extended Analysis ▪ Deep dive analysis for indicator of compromises incl. context relevant data ▪ Classification/priorization ▪ Trigger incident or escalation to Level 3	**Security Data Analyst** Overarching analysis independent of pre-defined use cases ▪ Reviews all relevant sources to detect possible attacks ▪ Regular task (not specific to incidents)
RESPOND	**Level 3 Expert (hours to weeks)** Expert Analysis and incident handling ▪ On demand-analysis for critical Incidents ▪ Analysis with support of specialist (e.g. Forensic) ▪ Incident management*	**Incident Orchestrator** Ensures that incidents are properly manag. ▪ Tracking of tickets until defined actions are closed (prevents that incidents that are triggered by Analysts are forgotten)	**Forensic/ Incident Response Specialist** Experts with know-how on specific topics ▪ Supporting Analysts especially in case of incidents
SOC-TOOL MANAGEMENT	**Content Engineer** especially for SIEM ▪ Technical implementation of use cases ▪ Implementation of parsers for different sources ▪ Definition and implementation of alarms	**SOC System Owner** Responsible for Apps of SOC (e.g. SIEM) ▪ Definition and management of new features and Software releases ▪ Main contact to vendor	**SOC Application Monitoring** Functional monitoring of security tools ▪ Control that relevant apps incl. interfaces/sources are functioning ▪ Pre-defined actions (e.g. issue tickets) to establish again target state

Figure 6: Services of a SOC [21].

The first service for the detection of anomalies comprises tasks on level 1 and 2. A real-time analyst monitors alarms on level 1 by continuously analysing and classifying events, the trigger of incidents, and initiates, if required, escalations on level 2. An extended analysis is performed on level 2 that may take minutes to hours. This extended analysis consists of a deep dive analysis for the indicator of compromises based on context relevant data, a classification and prioritization of the event, and, if necessary, the trigger for an escalation on the subsequent level 3. Additionally, a security data analyst performs continuously and independently of predefined use cases an overarching analysis of all relevant sources to detect possible attacks. This task is regularly done, independent of current and known attacks.
The second service for responding and analysing incidents requires on level 3 expert work that may take hours to weeks for the analysis and fully understanding of the event in order to mitigate the incident. This process is twofold: a) an on-demand analysis for the triggered critical incident with specialists, e.g., forensic, and b) the management of the incident including government agencies. The management of these parties requires an orchestration of well-defined actions including a proper tracking of all tickets.
Finally, the applied tools need to be managed and configured, i.e., use cases must be defined, especially for the security information and event management (SIEM) [22]. The use cases need to be technically implemented in the tools, and for the surveillance parsers for different resources are necessary, including the definition and implementation of alarms. This requires the role of a content engineer. A further role

is the SOC system owner who is responsible for the applications of the SOC like the SIEM. This role comprises the definition and management of new features and Software releases, and being the contact to the application vendors. Important is also the monitoring of the functioning of the tools, including the ticketing management with pre-defined actions.

2.2.2 Requirements for UI in the cyber security context

The described tasks in section 2.2.1 are based on big data, and thus, require tools for the various tasks of a SOC:

- Data analysis: the exploration of data to find new attacks and anomalies.
- Incident orchestration: Tracking and escalation of tickets.
- Vulnerability management: Real-time view on vulnerabilities.
- Threat intelligence: Background information on threats and early warnings from different sources.

These tools must fulfill specific requirements for the visualization of data statistics like histograms, or heatmaps for alerts. In order to achieve a good usability and user experience the various roles like the analysis expert on level 3 should be equipped with several monitor displays allowing a simultaneous view on different applications (Figure 7). All applications need to be able to operation in real-time, even the data are assembled from numerous locations in the world.

Figure 7: UI equipment of a real-time operating SOC [21].

The real-time monitoring of separated data sources, and additionally, the orchestration and management tasks require a large screen that can display numerous data and applications for the specific roles, such that experts and managers are able to cooperate with each other. For reasons of practicability should the large screen have a concave curvature. As an example for such a large screen see the top of Figure 7.

2.2.3 Workflow support

The support of workflows, i.e., the automatization of processes like incident ticket handling and visualization is essential for cyber security [23]. The toolbox Grafana [14] contains for this purpose numerous dashboards for alarm boxes, alert lists, analytics panel, various charts and diagrams, data table panels, heatmaps, reports, progress lists, and dashboard lists (Figure 8). The workflows are shared in the Grafana community, and thus, their number grows continuously.

Figure 8: Dashboards of Grafana [14].

Figure 9: Kibana dashboard for data analytics [15].

The alternative toolbox Kibana [15] contains also numerous dashboards (500+) for all purposes. Kibana is available together with Elastic (big data storage) and they provide together so-called “Beat” packages which can be configured for various dashboards like data analytics (Figure 9).
A profound example for the integration of devices and ICT media into a unique workflow for communication is provided in [24]. Here the devices and media build two independent domains that are integrated into one fully integrated system resulting in an automated communication workflow.

2.3 Data visualization for cyber security

Section 2.3.1 contains the scientific background and state-of-the art of data visualization for cyber security. Then, in section 2.3.2 basic types of data are depicted followed by an overview of applied forms for visualization with the toolboxes Grafana [14] and Kibana [15] (section 2.3.3). A summary of UI/UX as success factor for cyber security software solutions is provided in section 2.3.4.

2.3.1 Scientific background

Staheli et al state that there “is little research on what makes a cyber visualization ‘good’…” [35]. The cause is missing human-in-the-loop evaluation for cyber visualization. Evaluation should not only comprise key performance indicators (KPI) like effectiveness, performance and real-time, but also a deep understanding of the following dimensions:

1. Network operations
2. Forensics
3. Threat monitoring
4. Data analysis
5. Decision making
6. Communication
7. Work environment (24/7 operations centre, contested terrains)
8. Work style (individual or collaborative, peer-to-peer or hierarchical)
9. User cognition (experience, mental models, biases)

In this deliverable we focus on specific quantitative and qualitative evaluable dimensions that include user experience and preference, usability and learnability, feature set utility, effect on collaboration, and cognitive workload. KPIs and measures like physical demand, algorithmic efficiency, and component interoperability are not in-scope.
Visualization evaluation could be based on practical methods like Heuristic Evaluation or Cognitive Walkthrough [1, 2, 5]. These methods are mostly applied in industry, since they carve out business demands, and nearly all cyber visualization tools like Grafana and Kibana are designed for real users.
Evaluation for visualization research consists of models like “complex, deep, qualitative, unexpected, and relevant” or the approach of patterns with five pattern classes “exploration, control, generalization, validation, and presentation” [35].
The future direction that we adopt is the evaluation of the data analysis process particularly through the use of case studies as it is also promoted by Gates and Engle [37]. Staheli et al compiled evaluation dimensions from a variety of existing work [35]. From user experience perspective the following five evaluation dimensions are in focus (cf. sections 2.1.1 and 2.1.2):

1. User experience / preference: The overall experience of using an interface, especially in terms of how easy or pleasing it is to use
2. Usability / learnability: The ease of use and learning curve of an interface
3. Effect on collaboration: Does an interface encourage more collaboration (measured in terms of increased communication, shared control of resources, etc.)?
4. Cognitive workload: From a cognitive science perspective: How effectively does the system utilize a person’s working memory? More heuristically: How hard does the person have to think to accomplish their tasks while using the system?
5. Task performance: How well does a person / team perform on a predefined task using this system?

Ma indicates that an iterative drill-down process is required for detecting anomalies using a visualization-assisted analysis of computer network activities [36]. As an example see Figure 10: A drill-down process of analysis can start by looking at the aggregate information about routing changes over a complete period of time, followed by examining routing update messages over a selected period of time including their corresponding statistical values. Then in a subsequent step, particular instances of instability can be visualized in detail. This joint visualization enables the verification of the visual and statistical information for anomaly detection in parallel. Their work includes illustrated examples for drill-down visualizations for port data visualization and scan characterization. Ma concludes with two mentionable remarks: First, Visualization is best for guiding a complex data analysis process since visualization is particularly good for showing an overview of the data, which can direct the analyst’s attention to the aspects of the data that require further investigations. And second, what visual representations should we use to study such heterogeneous data in

a unified manner? What would be the meaningful linkages among the disparate data to facilitate cross-exploration? For the field of information visualization, this new class of problems presents many challenges and open research questions.

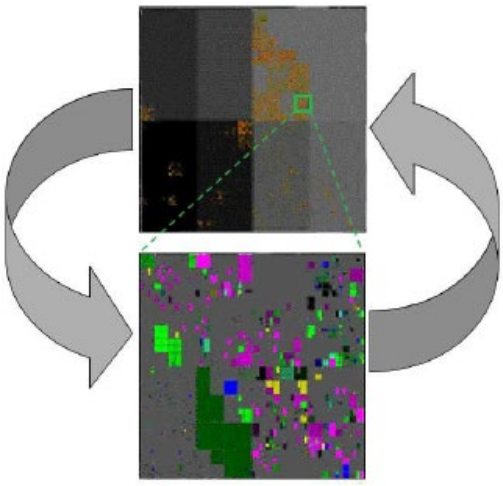


Figure 10: Iterative drill-down process for anomaly inspection [36].

Cyber defenders distrust visualizations that hide or smooth the underlying data: Fink et al [38] discovered that access to the source data is critical, but even with details on demand, cyber defenders seemed more comfortable looking at the actual data line by line. Another problem is that they want to be able to filter, join, and transform the data without losing or altering the original. Visualizations seldom allow flexible manipulation of data, and they can give a feeling of distance and lack of control. Cyber analysts feel exposed to poorly designed visualizations that prejudice them against all visualization. As important usability lack they found out that an anomaly investigation typically involves many windows, and analysts have to multi-task among several open investigations. The typical analyst workstation they saw in their study had one or two moderate-resolution displays with 20 to 40 windows open at a time representing multiple active investigations. This meant that more windows were covered or minimized than were visible at any given time. This may lead to low-level thinking during the investigation. The Mandiant Highlighter is a helpful tool for tasks like this [47]: It focuses on textual representation of log files, but offers helpful interactive features such as highlighting records based on a selected field value, and easily filtering selected records. Fink showed that correlation of data types was more effective at improving analyst performance than visualization alone, but that visualizing correlated data was significantly more effective than either visualization or correlation alone [48]. Ma [36] supports this argumentation. Note, the challenge of handling many windows can be mitigated by using more screens (Figure 11) or a large display (Figure 7). For usability reasons both of them should be curved.

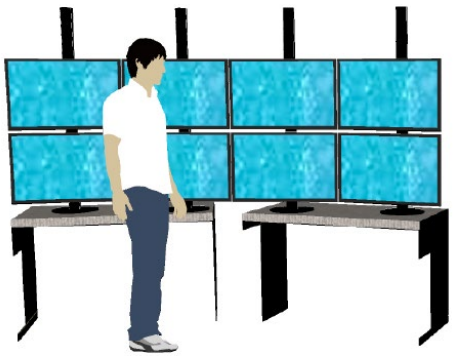


Figure 11: Multiple screens, curved arrangement, standing orientation [38].

Tamassia et al [39] investigate graph drawing techniques for visualization: Nearly all methods allow to zoom into the data in order to get a more detailed view. An additional, and as aforementioned important feature, is the correlation of data. This can be based on statistics, or, e.g., with (semantic) graphs / networks. In computer and network security applications, the input to the visualization system is often a large multi-dimensional and temporal data set. Moreover, the layout needs to support colour, labels, and variable node shape/size and edge thickness. This requires sophisticated graph drawing algorithms using multiple data sources.

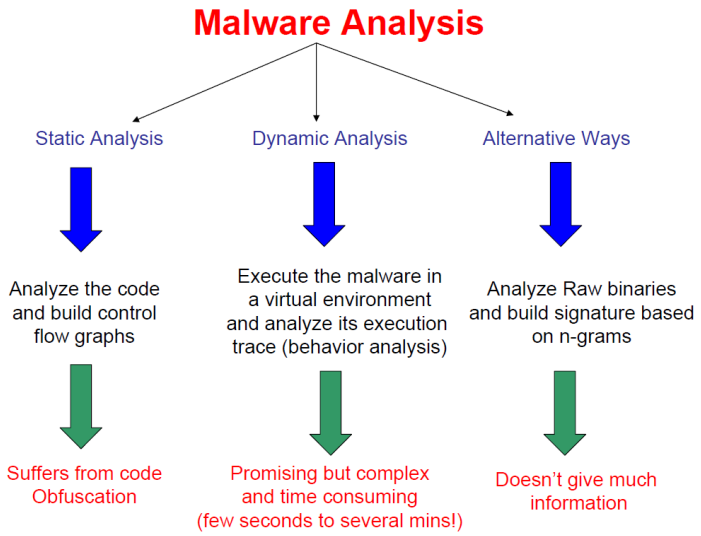


Figure 12: Malware analysis based on visualization [40].

Nataraj et al. [40] underline the importance of visualization during the analysis process (Figure 12): Both a static analysis, and a dynamic analysis with the execution of malware in order get insights into its behaviour are supported by visualization. Also, Ferebee et al [41] state that visualization should integrate multiple data sources like log and alert files into an intuitive visualization.
However, all discussed methods demonstrate that cyber security inspection is a semi-automatic process that requires the incorporation of human experts, and hence, usability and UX should be in the focus of the design of cyber security data visualization.

2.3.2 Basic types of data

Basic types of data vary from one application to others. As an example consider the types of data of a university, here Michigan University [32] (excerpt):

- Federal Information Security Management Act (FISMA) Data: The Federal Information Security Management Act (FISMA) requires federal agencies and those providing services on their behalf to develop, document, and implement security programs for information technology systems and store the data on U.S. soil. This means that, under some federal contracts or grants, information the university collects or information systems that the university uses to process or store research data need to comply with FISMA.
- IT Security Information: IT Security Information consists of information that is generated as a result of automated or manual processes that are intended to safeguard the university's IT resources. It includes settings, configurations, reports, log data, and other information that supports IT security operations.

- Personally Identifiable Information (PII): PII is a category of sensitive information that is associated with an individual person, such as an employee, student, or donor. PII should be accessed only on a strictly need-to-know basis and handled and stored with care.
- Sensitive Identifiable Human Subject Research: Sensitive identifiable human subject research data is regulated by the Federal Policy for the Protection of Human Subjects (also called the "Common Rule"). Among other requirements, the Common Rule mandates that researchers protect the privacy of subjects and maintain confidentiality of human subject data.

Types of data should be described and categorized based on their usage in order achieve a structure. As an example see the Georgia Tech Cyber Security institution with four different categories [33]:

1. Category I – Public Use: This information is targeted for general public use. Examples include Internet website contents for general viewing and press releases.
2. Category II – Internal Use: Information not generally available to parties outside the Georgia Tech community, such as directory listings, minutes from non-confidential meetings, and internal (Intranet) websites. Public disclosure of this information would cause minimal trouble or embarrassment to the Institute. This category should be the default data classification category.
3. Category III – Sensitive: This information is considered private and should be guarded from disclosure; disclosure of this information may contribute to financial fraud and/or violate State and/or Federal law.
4. Category IV – Highly Sensitive: Data which needs to be protected with the highest levels of security, as prescribed in contractual and/or legal specifications.

Then, the e-mail address is category I, the salary information is category II, the social security number is category III, and credit card numbers are category IV.

The basic types of data for cyber security are based on the following network data types [34]:

1. Network telemetry data: This data is usually collected from networking devices at remote locations and transmitted to monitoring systems for off-net processing and analytics, usually around performance management. There are two primary sources for network telemetry data: flow data and SNMP data. Flow excels at tracking near-real-time path data for active notification and isolation of issues due to changes in the network. For its part, data provides a polling methodology for network elements, with a subset of objects through an SNMP management information base (MIB) view. This provides data about devices, interfaces, CPUs, etc. for monitoring and collecting the network infrastructure status. Although it's a good foundation for basic network up/down monitoring, SNMP typically does not provide detailed network information. This is important for analysing the root cause of problems for application performance or many user experience issues such as quality of service (QoS) policies and tunnel performance.
2. Synthetic testing and virtual software agent data: Synthetic testing is a method of understanding a user's experience with an application by predicting behaviour. Cloud applications can lack visibility and performance data, data are needed to detect anomalies like too long latency. By using virtual software agents and collecting data from them, IT can continuously monitor these applications.
3. Application recognition data: Applications running in enterprise networks require different levels of service based on different business requirements. Therefore, having insight and data are vital to maintaining security. Network-Based Application Recognition (NBAR2,) offers a mechanism that classifies and regulates bandwidth for network applications on certain routers. This data allows network administrators to view the mix of applications in use on the network at any given time and decide how much bandwidth to allow each application. The latter aspect is also important for security and the recognition of threats.

4. Application visibility and control data: Application visibility and control (AVC) data, another important source, incorporates several technologies—including application recognition and performance monitoring—into WAN routers. Previously, network traffic could easily be identified using well-known port numbers, such as Port 80 for HTTP. Today, however, many applications are delivered over HTTP. Many applications—such as Exchange, voice, and video—use dynamic ports that are delivered over Real-time Transport Protocol (RTP). This makes them impossible to identify by looking at port numbers. In addition, some applications disguise themselves as HTTP because they do not want to be detected. As a result, identifying applications by checking well-known port numbers is no longer viable. AVC data fills this gap. AVC is tracked with a combination of metric providers, embedded monitoring agents, and Flexible NetFlow. AVC includes both TCP performance metrics—such as bandwidth use, response time, and latency—and RTP performance metrics including packet loss and jitter.
5. APIs and packet capture data: Finally, when capturing data that's useful for isolating root cause, there are two data sources on which network operations teams heavily rely: application programming interface (API) data and packet data. An API is a set of subroutine definitions, communication protocols, and tools for building software. In general terms, it's a set of clearly defined methods of communication among various components. In today's SDN environments, the control plane is typically centralised with a management application and controller to define and push policies and configurations down to the devices and functions. An API integration with the management systems, and access to that data, provide a way for path and APP ID information to know the data class and traffic routing through the SDN environment.

These basic types of a data are also used in network performance monitoring and diagnostic (NPMD) platforms. Especially the diagnosis allows the cyber security analysis with Grafana and Kibana.

2.3.3 Overview of applied forms for visualization

This section contains the description of the visualization and data analytics toolboxes Grafana and Kibana with a focus on visualization.

2.3.3.1 Grafana

The toolbox Grafana [14] consists of three important features (Figure 13): Among visualization for graphs and alerts, also the integration of various data sources.

Figure 13: Three important features of Grafana [14].

The visualization includes the building of heatmaps, histograms, and graphs [26]. Therefore, various data sources, including Elastic storage for big data, can be fully integrated. Additional visualization of alert rules and their visual definition for metrics. Grafana is able to continuously evaluate the alert rules and to send notifications operation monitoring systems. As an application of Grafana see the real-time I/O-monitoring of HPC applications [25].

2.3.3.2 *Kibana*

The toolbox Kibana [15] (for its essentials see [27]) visualizes data that are indexed in the Elastic data storage [31] (Figure 14).

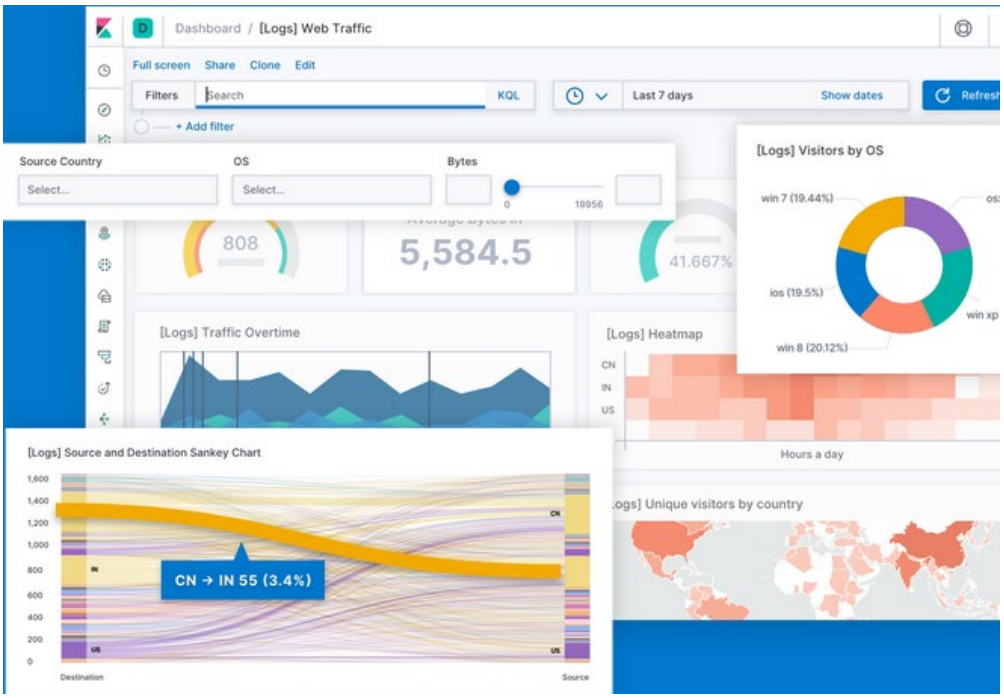


Figure 14: Kibana visualization for data stored in Elastic [31].

Kibana enables the visualization of indexed big data with similar features to Grafana. Through the underlying big data storage Elastic is a monitoring of the visualization effort possible. This feature is important, when several, in the field of cyber security, numerous data source are pulled and the data are stored in Elastic. Applications of Kibana are the building of an IoT data hub [29] and a framework for social media data analytics [30].

2.3.4 Summary: UI/UX as success factor for cyber security software solutions

The former sections contain important requirements for the design of automated cyber security workflows based on dashboards and SOC operation centres, namely

1. Multiple screens with a curved arrangement for cyber security experts (cf. Figure 11), independent of their role and tasks (cf. section 2.2.1), either in standing or sitting orientation.
2. Large curved screens for SOC operation centres (cf. Figure 7).
3. Multiple data sources like log files and alerts need to be displayed into visually combined representations.
4. Visualizations for various inspections and analyses are required like graph-based, (threshold-based) histograms, heatmaps, and (semantic) networks. Examples are implemented in the tools Grafana and Kibana (cf. sections 2.3.3.1 and 2.3.3.2) [14, 15].

Users have individual preferences and methods of working. There is no unique one size-fits-all for processes and visualization [46]. Thus, users, experts, analytics specialists, and managers need configurable and adaptable tools to their personal preferences. The above described tools Grafana and Kibana lack in these

adaptable options as they are called in the research field of end-user development [4]. A further option is a mobile app showing the most important alerts and incidents for experts. Englert and Glass [42] present an architecture for multimodal mobile applications that could serve for this purpose. Furthermore, an approach for an intuitive guidance through user interfaces, the so-called stepping stones model (Figure 15), has been developed by Englert and Joost [43]. The conceptual model consists of the following six steps that are described in detail in [43]:

1. A taxonomy of interfaces
2. A study on the dimensions of personalization
3. A study of user groups
4. A mapping of results from steps 1 – 3
5. Interface and interaction design
6. Implementation and usability testing

The idea of the conceptual model stepping stones is that designers can step back and forth between the stones, e.g., between the dimensions of personalisation (stone 2) and the user's perspective (stone 3) as is shown in Figure 15.

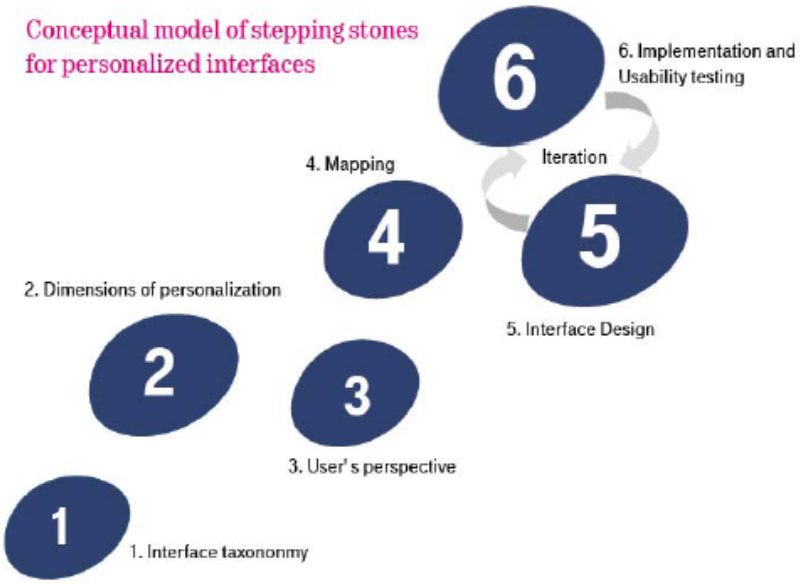


Figure 15: The conceptual model 'stepping stone' [43].

User-centred design methods for cyber security visualizations are also shown in [44] and an example for a framework for visualization design is provided in [45].

3. Data visualization challenges in SIMARGL

This chapter describes the approach, the general use cases and the derived challenges for the UI development of the SIMARGL toolkit. Section 3.1 introduces the chosen approach and the used methods for all preparational steps to design the UI concept. The identified major use cases are presented in section 3.2. Finally, section 3.3 summarises all important aspects and visualization challenges for the SIMARGL UI collected during the preparational stages. The insights described in this chapter build the foundation for the development of the UI concept which is presented in chapter 4.

3.1 Approach and methodology

In order to develop a profound and valid UI concept for the SIMARGL toolkit, it is necessary to follow a solid approach for performing all preparational steps needed. Within the preparational steps, well-known methods, especially literature studies, expert interviews and questionnaires, are used to accomplish the associated tasks. Figure 16 depicts the main steps of the chosen approach for the UI development in SIMARGL, which are described in detail throughout this section.

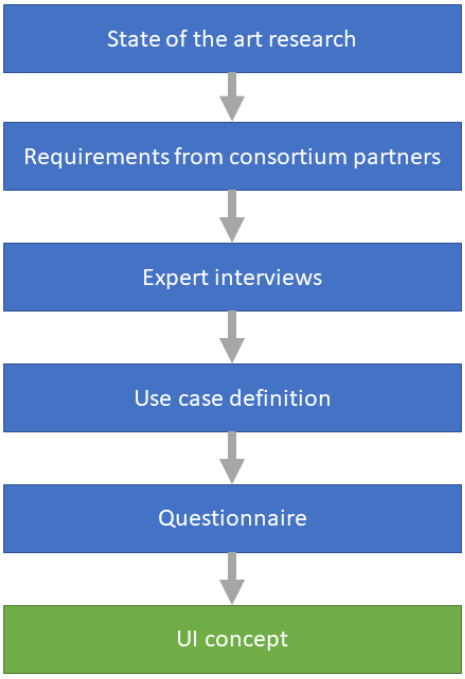


Figure 16: Approach for the preparational steps of UI development in SIMARGL

State-of-the-art research
The first step is to conduct a research to determine and summarise the current state of the art of UI/UX design, user roles in the context of cybersecurity and data visualization for cyber security. The main methodology to use for this step is literature study. The state-of-the-art research delivers the necessary scientific background and serves as a foundation for the whole further development of the UI concept. The results of the research are described and summarised in the previous chapter 2.

Requirements from consortium partners
After completing the state-of-the-art research, the next step is the collection and evaluation of requirements from all consortium partners. To perform this task, all early requirements concerning the UI development are

extracted from deliverable document D2.2, analysed and incorporated for the UI development process. The results of this step are included in the description of the important aspects for the UI in section 3.3.

Expert interviews
For UI development, it is important to gain a good understanding about the target user groups of the system to be developed. Knowing the users, their daily workflows and their needs allows building an optimized UI for them. In addition to the scientific background and to the conceptual requirements described in deliverable document D2.2 at an earlier stage of the SIMARGL project, the qualitative method of expert interviews is used to further enrich, discuss and validate previous findings. Expert interviews are conducted in two domains: cyber security and criminal investigation.

Interviews with cyber security experts
For the preparation of interviews with cyber security experts, the first step is the visit of a real-time operating SOC to observe SOC operators at their workplace, get insights in their daily work and the tools they use. The observations and first insights are then be combined with the results of the previously performed research to develop a question catalogue for the following interviews. Finally, interviews with a team of cyber security experts are performed to discuss their usage of tools, main use cases, requirements, ideas, current problems and possible improvements.

Interviews with police detectives
Like cyber security analysts, police detectives also perform investigations of incidents in their daily work. Beside the fact, that the domain is different, they have to also collect facts and data, further analyse them and discover relations. Hence, they face similar problems like cyber security analysts. Therefore, it is a good idea to also conduct interviews with police detectives. The objective of these interviews is to find similarities, get suggestions for new ideas and maybe transfer methods or approaches from the investigation of criminal incidents to the investigation of cyber security incidents.

Results and key findings of the expert interviews are summarised and described in section 3.3, where the important aspects for the UI in SIMARGL are presented.

Definition of use cases
The examination of all results of the previous steps leads to the definition of the main use cases for the UI of the SIMARGL toolkit. During the definition of the use cases, cyber security experts are involved to validate each use case and to incorporate further suggestions. The use cases build the guidance for the following steps and are described in section 3.2.

Questionnaire
In order to receive important qualitative feedback for the UI development from consortium partners, a questionnaire is prepared and distributed. During the questionnaire, the partners are able to estimate general visualization and interaction concepts and evaluate possible features for the UI of the SIMARGL toolkit. Other important aspects of the questionnaire are the intended usage of the SIMARGL toolkit for each partner and the ability to provide additional suggestions and ideas.

Before launching the questionnaire, a preparational step is the development of first ideas for the UI concept based on the previously defined use cases. In this process, first mock-ups are created to further illustrate and summarise these early ideas. As best practice for a user centred UI/UX design process, these early visualizations help the users, respectively the consortium partners, to gain a better understanding of the first ideas for the UI concept and thus to support the development.

The main results of the evaluation of the questionnaire are described in section 3.3. For the complete questionnaire see Annex A: Questionnaire for UI concept. To summarise, the results of all preparational steps

described above and presented in chapter 2 and chapter 3 lead to the development of the UI concept which will be explained in detail in chapter 4.

3.2 Use cases

After collecting the requirements, a profound starting point for beginning with the design and development of user interfaces for software solutions is the definition of the main use cases, which will be addressed by the solution to be developed. For the SIMARGL toolkit, as a solution for the detection and analysis of cyber security threads, three main use cases have been identified: data visualization (3.2.1), collaboration (3.2.2) and big curved display (3.2.3).

The use cases were derived from expert interviews and developed together with the management and the analysts of the operating team of the cyber security centre of German ISP Telekom. They are also based on the overall requirements and scenarios developed and described in deliverable D2.2. In this context, it is important to note, that the term “use cases” is used in a slightly different meaning throughout this chapter compared to D2.2: Whereas the defined use cases in D2.2 describe concrete and specific cyber threat detection scenarios like detecting new C&C servers addresses (UC-02) or detecting anomalies in files (UC-07), the use cases in this section describe the general interaction scenarios for cyber security experts.

3.2.1 Data visualization

A major task for the application of the SIMARGL toolkit and for other cyber security tools is the analysis of incidents and alerts to detect or classify different cyber security threats, as described in the various scenarios through deliverable document D2.2. In addition, section 2.3.1 on scientific background illustrates that cyber security inspection is a semi-automatic process that requires the incorporation of human experts. Both aspects arise the need for a good visualization that supports a human expert during his analysis. Therefore, one general use case for the UI of a cyber security toolkit like SIMARGL is data visualization.

With the help of data visualization, a security analyst is able to generate, show and inspect various graphical representations of the data, which belong to or might be relevant for the current investigation of an incident or of suspicious behaviour. A security analyst wants to apply different types of graphs like charts, histograms, heatmaps (see 2.3.3) or even custom defined graphs, be able to scroll through visualized data, inspect one single graph or show two or more graphs in parallel, and visualize data at different levels of detail.

As the investigation of malware often requires collecting data from various input sources and operating on different types of data, it is important to incorporate data from different connected data sources into the process of visualization. According to the various types of data like e.g. NetFlow, logs, execution traces or summarised statistical data, as described in 2.3.2, different forms of visualization have to be used. The same applies for the visualization of the various results from the different detection tools and detection algorithms provided by consortium partners for the SIMARGL toolkit.

3.2.2 Collaboration

Another common use case throughout the analysis of cyber security incidents is the collaboration between two or more experts. In a cyber security centre, there are typically different roles defined and different tasks assigned for users and experts, as described in detail in section 2.2.1. For the detection of anomalies and the investigation of incidents, a real-time analyst on level 1 performs first step analysis and, in case a complete investigation is not possible within a short period of time, initiates escalations on level 2. The same approach is also used from analysts on level 2, who might trigger escalations to level 3. At least, an analyst at level 3 may involve specialists with specific knowledge and expertise to perform a deep analysis of a potentially new cyber threat. And in special cases, like critical incidents, experts might work together in small teams.

An important aspect for the escalation process and for the involvement of specialists is the handover between involved experts. To support collaboration, experts want to be able to share their previous results and insights. Two aspects are vital for this handover process:

1. Time series of the observed attack currently in investigation
2. Time series of the previous steps already performed during the analysing process (history)

Ideally, both aspects are visualized on a timeline and combined together. This way, other experts can easily see important findings of previous investigation steps. In addition, a time-independent representation of previously found correlations might further support the collaboration between experts.

To trigger the escalation process and to keep track of handling alerts and incidents, security experts use a ticket system or some similar system for workflow support. Using these systems, an expert has an overview with organized information over the actual status of incident handling and can perform necessary actions like triggering an escalation or adapting the status of a single incident.

Another aspect and a different form of collaboration is threat intelligence. Threat intelligence describes a structured collection of information and evidence-based knowledge about threats and threat actors which might be used during an analysis of a security threat. In contrast to the previously described direct collaboration, this form of collaboration is indirect because it incorporates common knowledge collected from many security analysists.

3.2.3 Big curved display

In a security operations centre (SOC) it is an important task to have an overview over the global status of monitored networks and over ongoing alarms and incidents. As described in section 2.2.2 and section 2.3.4, big curved displays are used for this purpose in order to enable experts and managers to cooperate with each other. Big displays visualize the most important information, filtered and aggregated from the underlying big data of the monitored systems, and display the most critical alerts. In addition, a big curved display can be split to display some typical relevant overview information in a KIOSK mode in one separate area.

In the context of a SOC one can distinguish three main classes of displays:

- Cinema: A large curved display for global overview (big picture) in a SOC
- Medium: Medium-sized displays for collaborative work of small expert teams in meeting rooms
- Normal: Normal-sized displays for cyber security experts at the workstation, typically, multiple screens with a curved arrangement

The big curved display can serve as starting point for the investigation of alerts or abnormal behaviour, especially in cases of critical alerts. Beginning with the information displayed in the overview of the big display, an expert can start an investigation by easily accessing and importing relevant data from the big display to his workstation screens. From that point on, the expert can perform an iterative drill down process and further analyse the incident on the workstation screens.

Exchanging data in the opposite direction from a screen of a workstation to the big curved display is typically not necessary, because it is not the purpose of the big overview to share detailed information of a special analysis. But in order to work together with a small expert team on a critical incident, it is sometimes desired to share the data of the workstation screens of an analyst with a medium-sized display of a meeting room.

3.3 Important aspects for visualization in SIMARGL

As identified throughout the various preparational steps described in section 3.1, the most important aspects for the visualization are a meaningful and clear approach for the exploration of data, an easy collaborative work, the ability to orchestrate incidents, and support for the prevention of future incidents. The handling of these aspects within the SIMARGL toolkit is described in detail below.

3.3.1 Usability and user experience

Basic requirements for a good user experience derive from section 2.1 and the former deliverable document D2.2. The aspects that are important from the usability point of view are listed as followed:

1. The functionality of the UI should be immediately obvious to the users.
2. The navigation inside the applications should be linear and straightforward.
3. Using general meaning pictograms and traffic light colours. For user interface controls use terms and labels that are familiar to users.
4. Minimum status information for each module: protected, connected, last updates.
5. Visual timing measurement for actions that take more than 10 seconds

These aspects for usability and user experience have been confirmed in the questionnaire, which also stated out, that a clear and uncrowded list – especially when it comes to important information such as health or performance status – is highly necessary. An overall view for occurred notifications, but also a depended history is indispensable, too. The full questionnaire is listed in Annex A: Questionnaire for UI concept.

3.3.2 UI layers

As the purpose of the SIMARGL UI is to ease the use of multiple tools, a multilayered and iterative drill-down process into data needs to be implemented for a fast and efficient analysis of the data. This approach already takes different user roles into account, by delivering information over several layers that correlate with the skills and needs of the different user roles. A survey within the consortium was used to determine the importance of various tool overviews and entry points. It turned out that this is strongly user-dependent and therefore requires a very flexible solution. Starting with a configurable dashboard layer that gives overview over the most important systems, events and incidents will guarantee this requirement and will thus increase the usability for each user. From there the users will be able to move layers down into deep data analysis or use direct collaboration tools, such as a ticket-based system for escalating issues to the next support levels that are described as user roles in section 2.2.1. From this entry layer, the analyst shall be able to drill down into specific network nodes that contain aggregated data of the integrated tools for further investigation and analysis. A deep dive into the specific UI of the otherwise integrated tools forms the lowest analysis layer, which is further described in section 3.3.5. The available depth of data analysis layers is determined by the skills of the user and therefore allows for a fine-grained control of information flow.

3.3.3 Interactivity and data visualization

From interviews with SOC managers that are described in section 3.1, the requirement was raised to increase the interactivity and the efficiency within the analysis process. Being able to dynamically combine multiple data sources and to correlate them over the time dimension, is the key feature that lacks in many other tools and thus fulfils the request. The data itself therefore needs to be organized into multiple data type or source dependent lanes, which can then be correlated over timelines. The data types will be determined by the different tools that are integrated and configured into the SIMARGL UI as described in section 4.5.2. The ability of swiftly composing diagrams and overviews over important events will further increase the requested interactivity. This defines the deeper layers of the data analysis that will be available depending

on the user role. In these layers the users shall have multiple tools to allow for long-term analysis and incident handling. The users shall be able to expand and collapse data lanes to identify relationships between various data lanes. Once relationships or anomalies within the data are found, the user needs tools to mark and highlight these for further inspection and for reconstruction of the course of events that lead to the incident. Additionally, notes and other highlighters shall be attachable to marked data snippets. Referring to an interview with a German police detective, an overview of relations between suspicious subjects and events is a very helpful tool. Transferred to SIMARGL this would be another analysis layer, that visualizes the relations between data and sets these again in relation with other marked anomalies. The usefulness and need of such graph-based tools is underlined by software like Maltego and FUNDUS that is already used in SOCs and criminal departments to highlight data relations.

On big displays or on a multi-display setup the different layers shall be visualized next to each other, which further increases the efficiency of the SIMARGL UI. Further, a KIOSK mode, that can be utilized on big curved displays, shall be implemented, to offer a uniform and central dashboard for all SOC analysts, as described in section 3.2.3. Since there is the demand for increased work pace, the UI must respond quickly to interactions and updated data.

As the questionnaire stated out also, the possibility to offer customizable and highly interactive graphs, next to the well-known standards, is very important. The set of known graphs consists of gauge, bars, timelines, heatmaps and geolocation maps, but also simple tables and lists are important, too. Navigate quickly through the graphs, combine multiple sources – also from different tools, and create a role-based access to the result and data in general are also a frequently asked requests by security experts in the consortium.

3.3.4 Collaboration

Various collaborative operations will be integrated in multiple layers, as already indicated in the previous sections. On the first dashboard layer, whole incidents shall be handed over to other analysts or escalated to higher support levels. Also, there shall be integrations of external or internal systems, that could handle complex workflows, like the ticketing system OTS Storm or the workflow management tool TheHive. The integration of these tools could be done via API calls or referral links. The internal collaboration workflow shall support the assigning of single incidents to other analysts, which will then see the incident on their dashboard. An overall overview shall be granted by tracking all alerts, notifications and actions in a history that is visible to the corresponding users. On deeper layers of the data analysis, users shall be able to directly collaborate inside the focus of a single incident. As previously described, analysts can highlight and mark data with notes. This information can be shared with multiple users and analysts, so that they can directly see the investigation progress. All changes of notes and marked data need to be listed in the history, so that analysts can trace the course of events on the smallest possible information snippet. Even the breakouts to specific tools and the results of these breakouts could be stored in the history for a complete progress overview.

3.3.5 Tool integration

One of the biggest challenges of the SIMARGL UI is the integration of the different tools, not only for their data, but also for their UI that allows an in-depth analysis with the highest precision. For the dedicated data, as well as for the seamless integration of the single tools, these tools should provide an extensive API, that allows for handing over specific actions, extra data and information. The handover forms the lowest layer of the data analysis and thus is performed inside the analysis layers, not inside the dashboard layer. Depending on the data type, multiple suitable tools might be suggested to the analyst, so that the user can easily reach all relevant tools. After the investigation in an external tool is finished, the tool should ideally report any findings back to the SIMARGL UI. That way, the UI could automatically mark relevant data with additional information that is also tracked in a uniform history for a better overview.

As result from the questionnaire, the usage of the same toolset within the whole network cluster or a subnetwork cluster is as important as using single tool instances independently on a single node. This means, that the SIMARGL UI toolkit needs to take care of both structures. But not only the tool integration within the SIMARGL environment, also the exportability of obtained results to be used in other tools as well – such as MISP or TheHive – has been stayed out as an important point. Furthermore, offering a REST application programming interface to connect external tools with the SIMARGL environment and the underlying data sources is also an important aspect.

3.3.6 Threat intelligence

By providing summarised information and notifications from the specialised tools in one single SIMARGL UI, the analysts and users can efficiently keep track of threats and incidents. This also includes a structured collection of results from previously performed analysis of threats and incidents available from all tracked actions in the SIMARGL toolkit. To improve short response times, the SIMARGL UI shall integrate and distribute notifications from each integrated tool to multiple receiver channels. For instance, channels like SMS, E-Mail or messenger apps should be taken into consideration. Another notification channel of course is the SIMARGL UI itself, that must display all information and ongoing notifications to the corresponding users and analysts. As indicated by the laws of Krug in section 2.1.1.2 and the usability heuristics of Nielsen in section 2.1.1.1, information shall be presented in a consistent and self-explanatory way. Therefore, a uniform notification process via the SIMARGL system is an essential aspect for a good and efficient workflow.

Furthermore, the users shall be able to create custom alerts that are independent from the integrated tools and analysis algorithms, so that the users can benefit from their individual experience. The custom alerts can be based on thresholds or key events that are applied to the correlated data of the integrated tools.

4. UI/UX concept for SIMARGL toolkit

This chapter describes all aspects of the developed UI/UX concept for the SIMARGL toolkit. The first important aspect and subject of section 4.1 is the definition of the target user groups with their associated tasks which are mainly supported by the UI. The subsequent section 4.2 takes care of the overall UI environment, including but not limited to the main dashboard and surfaces of the SIMARGL frontend system. This section also already introduces the workflow for notifications and incidents, which is further described in section 4.3 along with the complete UI for the investigation of incidents, including the storyboard investigation dashboard and a special focus on collaboration.

Another important aspect of the SIMARGL environment is the connection between the frontend toolkit and the different tools provided by the consortium partners. Section 4.4 is dedicated to this topic by showing the provided interfaces and the general architectural integration between those systems. Additionally, subsection 4.4.3 also describes the decision for the visualization platform the SIMARGL UI should be based on by comparing the two most used and known environments on a technically background.

Finally, since the SIMARGL toolkit should be flexible and adaptive, section 4.5 describes the whole available configuration options and their support throughout the UI.

4.1 Addressed user roles and tasks

According to the defined use cases and the important aspects for the SIMARGL UI, the primary user roles with their associated tasks have been identified for the UI development. Table 1 lists the addressed user roles and tasks. The roles were derived from typical role definitions of an operating cyber security centre. For a complete overview of the user roles in the context of cyber security compare Figure 6 in section 2.2.1.

Table 1: Addressed user roles and tasks for SIMARGL UI

User role	**Associated tasks**
Security analyst	• Continuous analysis of events • Deep dive analysis and investigation • Classification • Escalation and collaboration
Manager/incident orchestrator	• Supervision of overall threat status • Management of incidents
Tool administrator	• Configuration • Monitoring of system health

The SIMARGL toolkit and its UI focuses mainly on the user role and tasks of a security analyst. The most important feature of the SIMARGL system is the support for investigations and the notification about alerts or suspicious behaviour. Therefore, the SIMARGL UI provides all relevant views and actions to enable a security analyst to fulfill his tasks in an optimal manner. Within the SIMARGL UI an analyst will be able to observe the current status of the connected networks, see alerts and perform an iterative drill-down process for deep analysis in detail views. In addition, the UI provides support for collaboration and escalation processes.

The second user role in focus is the role of a manager, respectively incident orchestrator. While a security analyst requires views for supporting deep investigations, an incident orchestrator needs summary views with aggregated information of the most relevant data, alerts and threats. The overall summary of the SIMARGL UI also integrates an overview for the observation of ongoing incidents and their handling.

In order to easily configure the SIMARGL toolkit and monitor its health status, the user role of a tool administrator is also directly addressed and supported by the UI. Therefore, the UI provides views, which show the overall health status of the system and the connected tools. In addition, separate screens are available to assist with the configuration of the SIMARGL toolkit.

The SIMARGL toolkit provides general mechanisms to support the definition and management of different user roles in the system:

- Configurable role-access management
- Role-based task association
- Role-based dashboards, views and actions
- Extensive management-access for administrators

The UI realization and its features for the selected user roles with their associated tasks will be described in detail throughout the following sections of this chapter.

4.2 Overall UI and main dashboard

The overall dashboard is the first entrance point and also the main work surface of the SIMARGL environment. From the overall dashboard the user is able to quickly view the current state of the whole SIMARGL toolkit with all relevant information and to easily reach the main functions and features. In general, the overall UI fulfils two requirements:

1. Creating a general overview about the whole observed network cluster and the installed tools
2. Managing the occurred notifications and the incidents based on them

To achieve this, the SIMARGL environment creates an overall dashboard combining different type of views: The network view, which contains also a highly interactive graphical representation as well as a tabled list, as described further in section 4.2.1. The notification and incident view, which creates an easy and native surface for managing and handling all kind of incidents and notifications, which is topic of section 4.2.2. The switch between both views is designed flowless and non-interrupting. Anyways, the user can freely decide which one of the two dashboards should appear as default.

In general, the whole interface is completely configurable by an administrator as well as customizable by every single user. This way, it can be adapted to individual and personal preferences and needs to improve and support the preferred workflow itself.

Regardless of the used dashboard view, the main design splits always into a three-column layout. While the first view, the left sidebar container, shows the overall system status and the health of the installed tools, as further described in section 4.2.4, the main container as well the right sidebar is dashboard-depended. The size of each container is fully adaptable and both sidebars can also be collapsed and expanded as well.

Following the three dimensions of the AttrakDiff model, as described in section 2.1.2.1, the overall UI follows a clear and functional yet visually pleasing design. This ensures a high perceived pragmatic quality, while maintaining a basic hedonic quality and attractiveness of the platform. That is why the icons are also designed distinguishable and therefore neither abstract nor three dimensional. The icons keep a flat unblurred appearance, which makes it easy to learn and understand them. As the SIMARGL UI mainly displays different status, the colours in the overall UI will be oriented at the usual colours for status indication, such as green, yellow or red. Decorative and other elements will mainly be kept in shades of grey, even though the logo of SIMARGL is red, in the UI it will be turned white or black, since red is typically used as a warning colour.

4.2.1 Network view

The central panel provides the main and global network cluster observation, which can either be viewed as graphical and fully interactive interface, as depicted in Figure 17, or also just as tabled list, as shown in Figure 18. Both available representations contain all available and required information, allow to interact and control the single nodes and systems, but also provide very useful information about the single tools and occurred notifications. However, the whole global network view must be configured and included by respective SIMARGL administrators as further described in section 4.5.1.

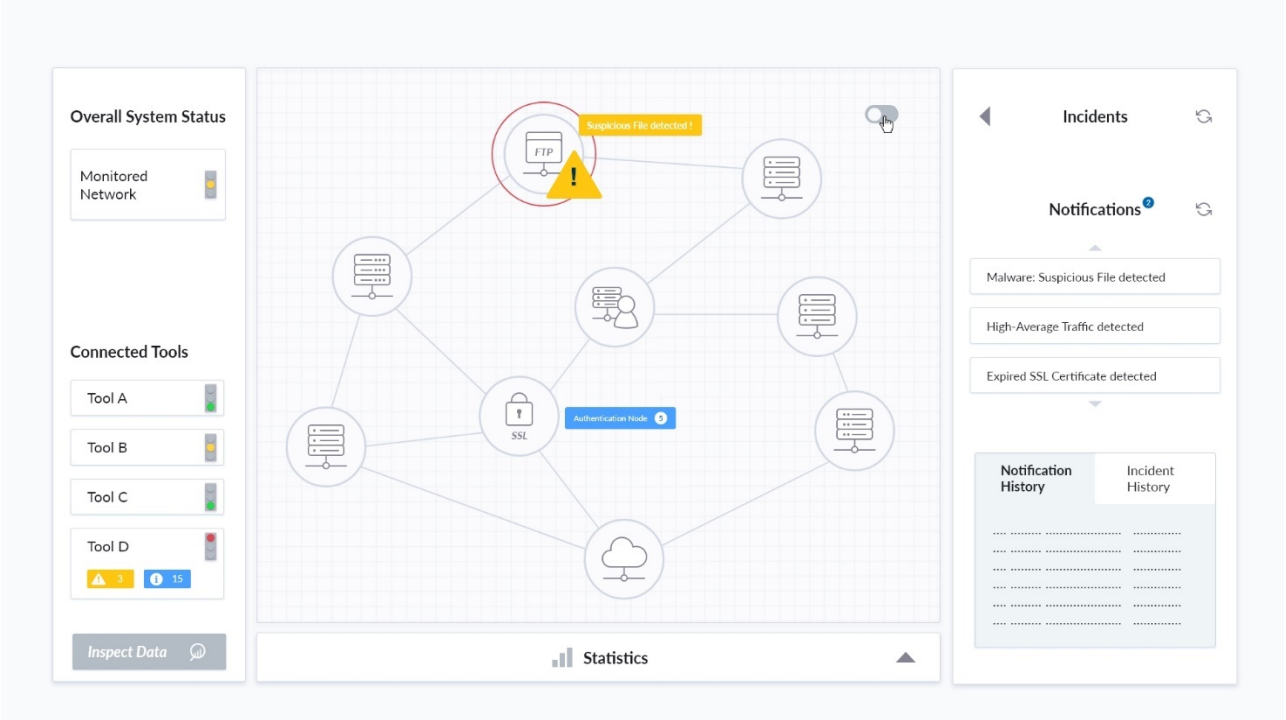


Figure 17: The graphical and highly interactive network observation interface

The graphical interactive interface shows all single network nodes as small configurable icons describing the type of the node. The unique provided icon-set contains a collection of well-known and familiar symbols, containing no text, except known and standardized protocols and abbreviations, to allow an easy understanding about the desired meaning. Furthermore, the single icons and nodes are connected with pulsing lines, which show the connections to each other. The display of lines between the nodes can be optionally disabled by a SIMARGL administrator in general or by single SIMARGL operators. An overlying and clickable box on each single node displays the number of notifications as well as open incidents. Further symbols, like a warning or cross icon, show the status and health of the whole node, too. Tooltips, which appear when the mouse courser is over the desired element, additionally describe the single icons and areas as well. This view also allows summarised nodes together as a sub network cluster, which will be presented to the user, if he zooms in or opens it by clicking on the respective item. Excluding nodes, searching for one or more nodes or filtering the whole presentation is also possible. Such features help the user of the SIMARGL toolkit to keep a clear overview also on big network systems.

Furthermore, the interactive network container can be controlled and used with a mouse, keyboard or the native touch abilities of a respective display. The mouse interaction consists of the mouse wheel to zoom in and out, the movement itself with clicked and held left mouse button as well as a right-clicked depending

context menu which offers a list of interactive options. A similar set of controls is also available on the keyboard, using defined shortcuts which are described below the main dashboard in a small expandable container. Native and well-known gestures on a touchscreen display are available and as well aimed to support touch-enabled notebooks or computer screens. Additional buttons on the screen, for example to reset the whole view, are available as well.

Even if this graphical network observation interface is built to be as minimalistic as possible and provides functions for searching and filtering, many operators may still prefer a simple list. This list, as shown below on Figure 18, can be toggled with the switch on the upper right corner. Each single operator also has the possibility to set the tabled list within the user configuration as the default view. This avoids the need to switch the views at any time. However, the list also takes care of the configured groups and sub-networks and is displayed level-based. This means, the user can walk through the single networks and sub-networks without getting confused by too many nodes and elements. The shown information and meta data are the same as on the graphical user interface.

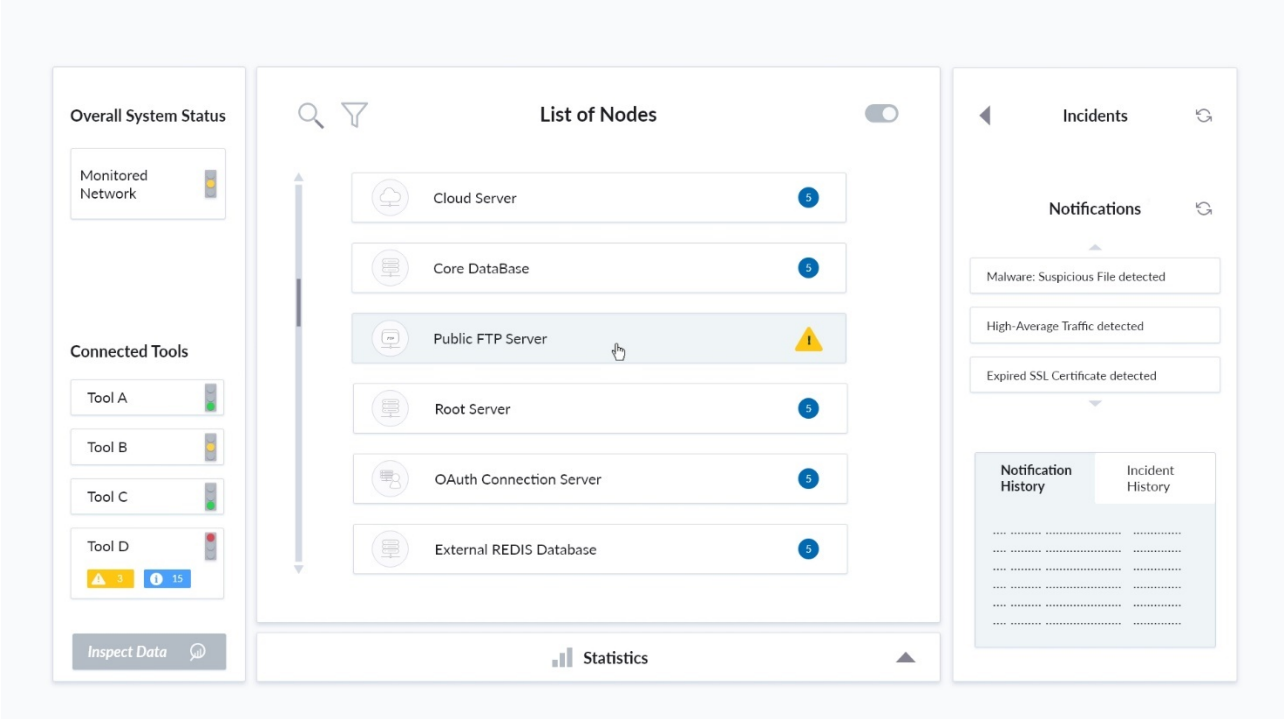


Figure 18: List view of nodes

Each node can also be selected, which lead to a list of available tools depending on them and which will also automatically filter the notification and incident list, as described below. A new button, which allows to directly switch to the respective website of the desired tool is available as well.

The right side of this dashboard view, next to the interactive network cluster container, contains the user-assigned incidents, including a button to create a new one, and the unprocessed notification list. The items on both lists are shown as small boxes, which directly allow a set of interactions, and can be quickly filtered using the respective icons on top of the lists itself as well. A small alert box also notifies the user about new assigned incidents or notifications, which appeared after the interface has been fully loaded. A click on this alert box will directly show all new ones on top of the respective list. This way, the user does not get interrupted on his current work, but still get notified about new content. Section 4.2.3 describes the whole behaviour and design of this lists in more detail.

4.2.2 Notification and incident view

The second available dashboard view dispenses the graphical representation of the network cluster and focus completely on the assigned incidents and occurred notifications, as depicted in Figure 19. Therefore, the main view container on the centre first lists the incidents within a tab-pane view allowing to just show user-assigned, generally opened and closed incidents in this order. Additionally, a button allowing to create a new incident appears as well, as further described in section 4.2.3. A similar interface is also available on the second list below, which lists the notifications. The second tab-pane allows the user to just show all opened, already assigned or generally dismissed notifications. Both lists can be expanded as well as fully collapsed.

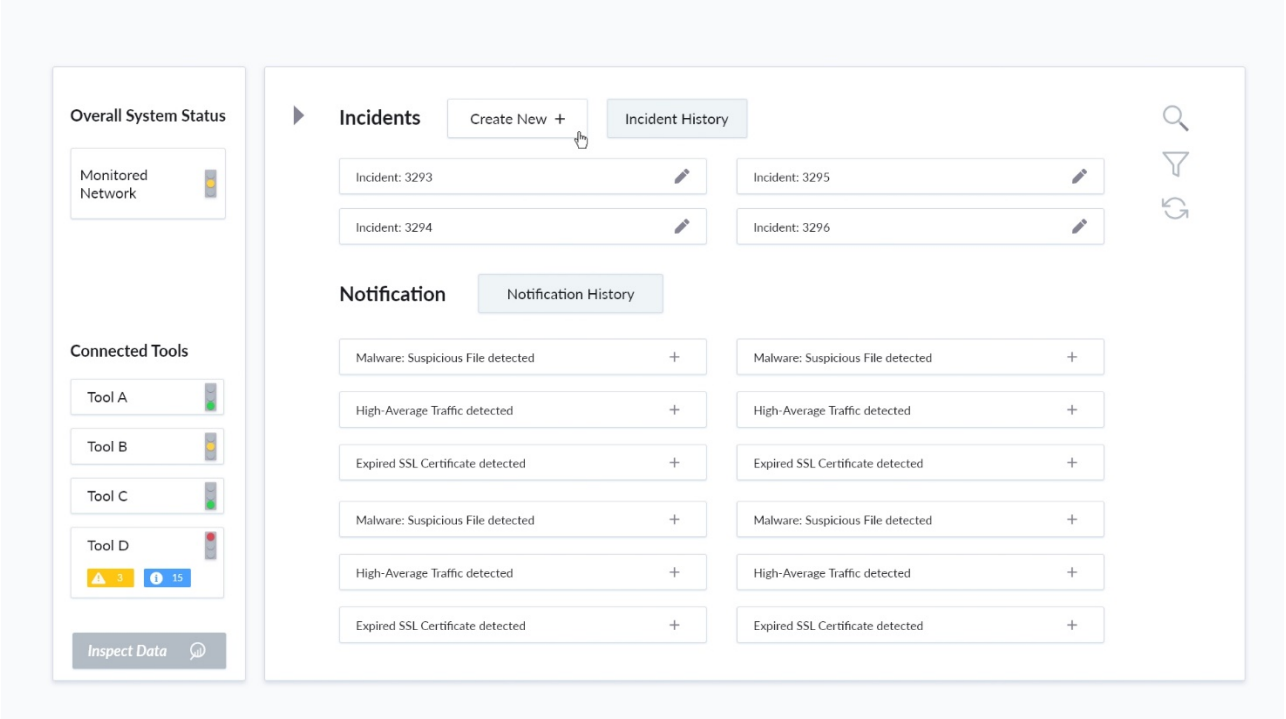


Figure 19: The second dashboard view, showing the incidents and notifications

To create a better overview about all this information, the right side of the main panel contains a few actions which allow to sort, filter and search the whole list of incidents and notifications. The main filter action allows to select one or more specific nodes or whole sub networks, assigned roles and users as well as the status of the single notifications and incidents directly. The search field is directly applied to the title, description and provided meta information of the single incidents, notifications and tools as well as nodes from which they originate. All sorting, filtering and searching actions are dynamically performed, meaning without a redirect, forward or reload of the current page, to support a non-interrupting workflow.

The keyboard and touch supported surface also shows a small information and alert bubble, when a new notification – after the first load of the page – has been occurred or, if new incidents have been assigned to the current user. This behaves similar to the right sidebar on the graphical network dashboard. More information about the notification boxes itself and the way, how an incident can be created from those, are part of the next section 4.2.3.

4.2.3 Handling notifications and incidents

Both dashboards present a unique view and also allow to apply some unique actions, but still offer and support the same possibilities when it comes to interacting and handling of occurred notifications. A

notification itself is shown as a small box, containing a set of information, such as the title, a small description, the depending tool and node, the level of the notification, a human readable and friendly datetime expression, the connected incident (if available) as well as some actions. In general, the main available actions allow to dismiss a notification or to create an incident based on them.

The dismiss option allows to hide the notification from the user's screen, but will not hide them for all users directly. Dismissed notifications can of course be called back by using the corresponding filter function depending on the used dashboard.

However, creating a new incident can be achieved by clicking on the respective action on the notification box itself or by using the general available "Create Incident" button on the used dashboard view. Both will prepend two new containers side-by-side, on the top of the main centred view as depicted in Figure 20. The first one, which may already contain the first notification the user clicked on (unless he used the general button), is used to collect all notifications the user wants to merge into an incident. The right container contains a list of recommended notifications, depending of the already selected ones, which may be important or generally relevant for the incident. Both, the recommendations as well as the general available notifications, can be added by clicking on the respective button on the notification box itself, using the available keyboard shortcut or by using the native drag and drop interaction gesture. When the user collected all notifications for creating the desired incident, he can press the respective button to create one, which will redirect the user to the further described investigation dashboard for the upcoming investigation itself.

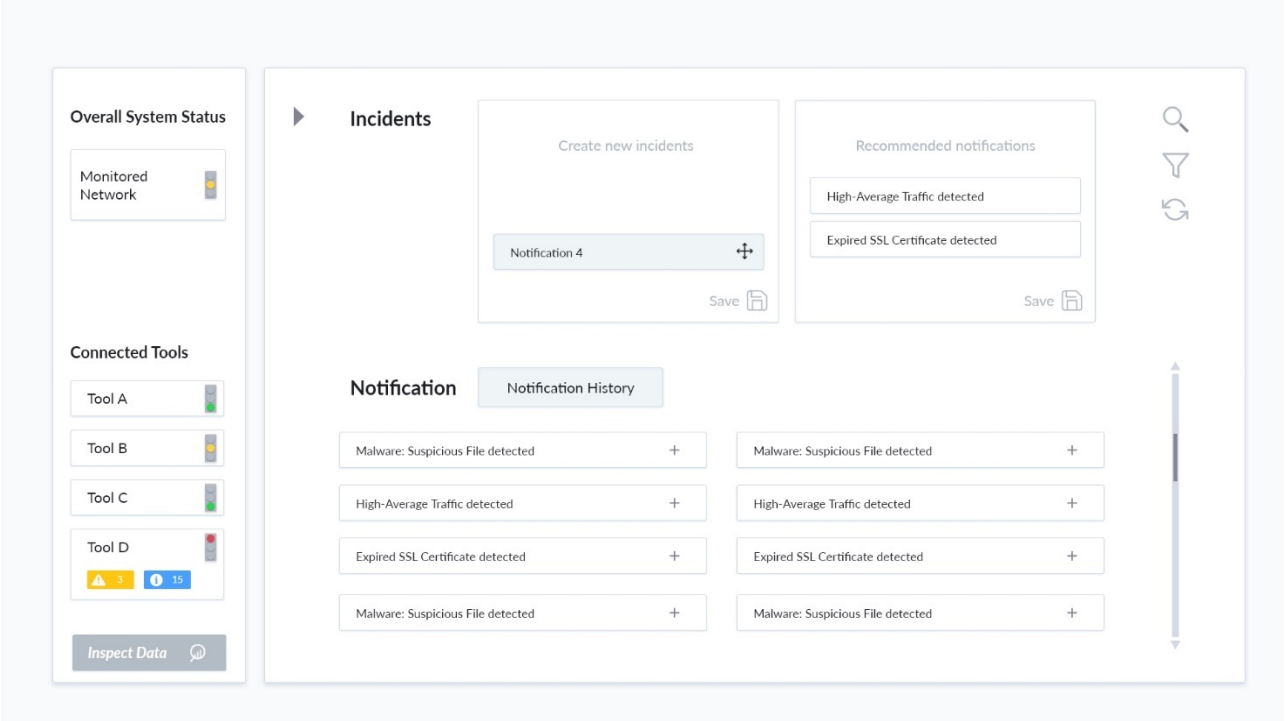


Figure 20: Creation of a new incident, based on one or more notifications.

The newly created incident will now also be shown on the respective list unless the user changes the assigned roles or users, or closes and archives it directly on the investigation dashboard. All notifications, which have been used to create a new incident, are signed with the associated incident ID. This occurs on all user dashboards directly, thus each user will be informed about the already created incident and is able to either dismiss it, which will remove it from the respective view, or add it to an own incident as well. Incidents, which are based on one or more of the same notifications can also be merged together. However, this process is further described in section 4.2.3.2.

4.2.3.1 Level of notifications

Notifications in the SIMARGL UI are divided into five different level, which not only represent their meaning but also their importance. The notification level is provided by the data source itself, which push the notification into the SIMARGL toolkit. However, the different levels are:

- **Debug**: Used for development purposes, for example to show some meta information.
- **Info**: Just a reminder or any kind of additional information for the end user.
- **Warn**: A maybe important information about the data source result or the tool itself.
- **Alert**: An important warning about the data source result or the tool itself.
- **Error**: A highly important warning about the data source result or the tool itself.

The SIMARGL administrator can configure, which notification level are available and which user or role should receive which level of notification. This way, it is possible to show notifications on the debug level, for example, just to the respective tool developers, while info messages are shown to all users.

Each notification also contains the respective ID on which node as well as tool, tool output or data source it relates on. This is important to connect the respective information, show them on the SIMARGL UI and allow a seamless and working interaction between the incident creation and the depending data.

4.2.3.2 Merge incidents based on the same notifications

One of the greatest difficulties posed here is the possibility, that more than one user creates an incident with at least one of the same notifications in the same or an immediate time. This situation is not preventable on big SOCs with dozens of employees working on the same platform at the same time. Therefore, the SIMARGL toolkit allows merging different incidents, which have been created with one or more of the same notifications. This merge-process includes the notifications used for creating the incident, as well as the comments and marker as described in section 4.3.

The user, who has created the incident at a later point in time will receive a special information about the already created incident process from the earlier user with the option to either dismiss his own incident or starting the merge process. This merge process allows the user to select all marker, comments and notifications, which he selected while he created the incident. The earlier user instead will only receive a respective notification, if a merging process has been made with the action to reload his current view and loading the newly added notifications and set marker and comments. This might lead to duplicate comments, time selections or set marker, but puts aside the issue that multiple incidents are based on almost the same notifications.

4.2.3.3 User-forced reload

Once the SIMARGL UI has been fully loaded, upcoming notifications and newly assigned incidents will not be shown directly to the current user. Instead, he will receive an alert box about new ones with the inline action to force reload the respective list. This allows to keep the user on track without interrupting the current work and flow by just pushing new notifications and incidents on the current list.

The notification about new content will not only be shown as information bubble on the respective list, but also as optional enablable browser notification and HTML title update. This way, the user gets also informed, even if the SIMARGL web application is currently in the background or opened in another tab within the browser.

4.2.4 Overall system status and statistics

The overall system status, which is always shown on the left side in the layout of the three dashboard views, represents the health and performance of the whole network cluster, including the configured sub networks. Besides that, it also lists the health status and performance of the installed and available tools, including a quick symbolized overview of the occurred notifications. To indicate status information, the well-known colours and symbols from traffic lights have been applied. The always visible bottom of this sidebar contains an action button, which allows to investigate the data output, generated by the different tools, of one or more nodes regardless of a notification or incident. As depicted in Figure 21, clicking this button will open a tabled list containing all available nodes, ordered and sorted within their configured sub networks, if available. The user has now the possibility to select one or more nodes and open an investigation dashboard, as further described in section 4.3.

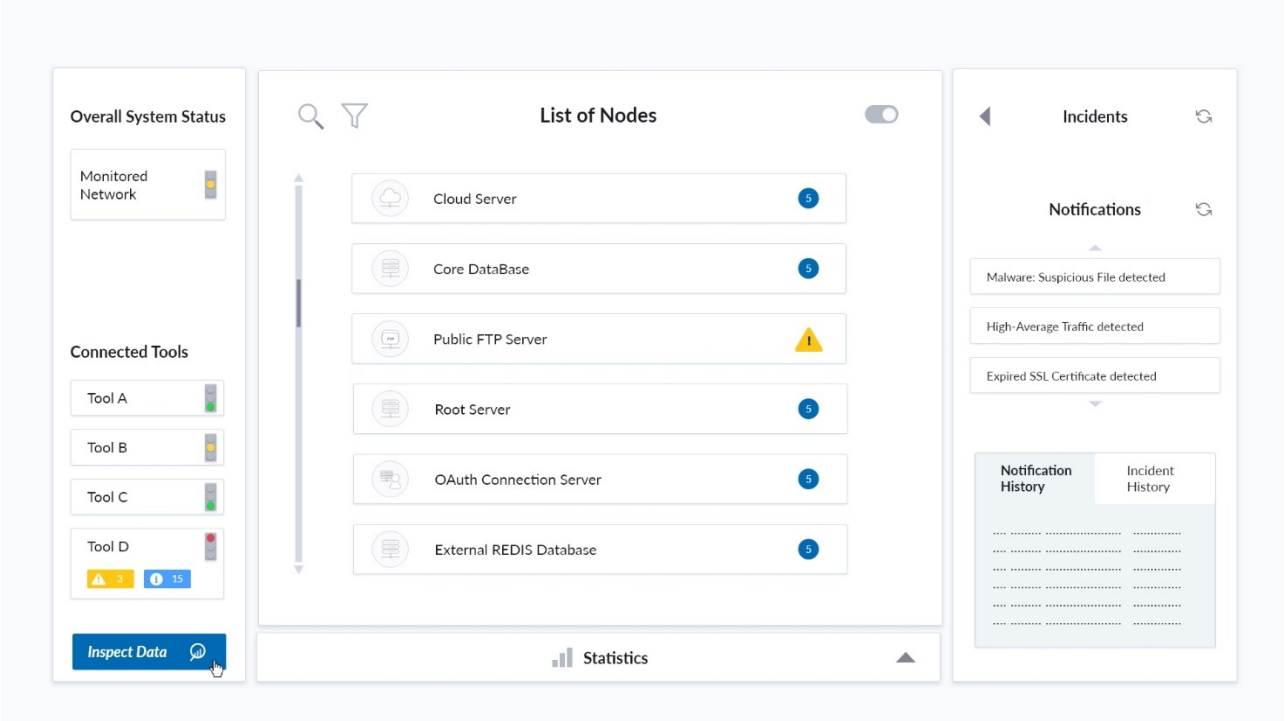


Figure 21: Inspect data from the overall system panel without notification

The listing of the tools is not only informative, but also interactive. Clicking on one tool shows the single nodes, where the respective tool is installed and generally used. A second click on one of the available nodes will show the occurred notifications depending on this selection, which visual representation differs depending on the current dashboard view.

An additional button on the overall dashboard allows the user to open a special statistics page. This page does not only contain information about the number and amount of connections, nodes and tools, but also a lot of statistics about the general notifications, incidents and the user's work. The data sets are calculated per default for the last 24 hours, 7 days and 30 days but the range can be configured directly on this additional page.

The user statistics also shows the number of notifications a user has worked on as well as the number of incidents he has created, closed or been assigned to. The calculation range is configurable as well. However, this information is only shown to the current logged-in user itself and cannot be viewed by other employees to prevent control and performance measurement about single employees.

4.3 UI for investigation of incidents

The SIMARGL UI provides a separate investigation dashboard for the analysis of incidents which can be easily accessed from the overall dashboard described in the previous section. After selecting an incident from the incident list in the overview of the main dashboard, one can choose to start the investigation by a simple button click.

The investigation dashboard supports three steps for performing analysis of incidents. The first step is the sources overview where all relevant data sources possibly related to an incident will be listed. In this step, a security expert can preview, filter and select relevant data sources for further analysis as described in section 4.3.1. The second step is the storyboard where an analyst is able to scroll through visualizations of the previously selected sources, inspect these visualizations and set marker at suspicious positions or segments in the data as described in section 4.3.2. The last step is the deep inspection of marked segments of the data sources where an analyst is able to zoom into the data and inspect as many details as available. Depending on the type of data, even inspection of corresponding entries line by line will be provided, which is further described in section 4.3.3.

It is important to emphasize, that there is no separation between the three steps and no fixed sequence. All steps are seamlessly integrated in the single investigation dashboard and an analyst is able to move forward and backward between the three steps and perform the steps in parallel as needed. For example, during setting marker in the storyboard step, one can easily add or remove another data source for further inspection and take a deep look into its data. In addition, collaboration and workflow support is available throughout all of these steps. These aspects are topic of section 4.3.4.

4.3.1 Sources overview and selection

After beginning an investigation in the main dashboard, the investigation dashboard will be opened. As depicted in Figure 22, the investigation dashboard consists of a main panel where all relevant information for an incident in investigation will be listed and a per default collapsed panel to the right, where additional information, like status or history, and basic actions are available.

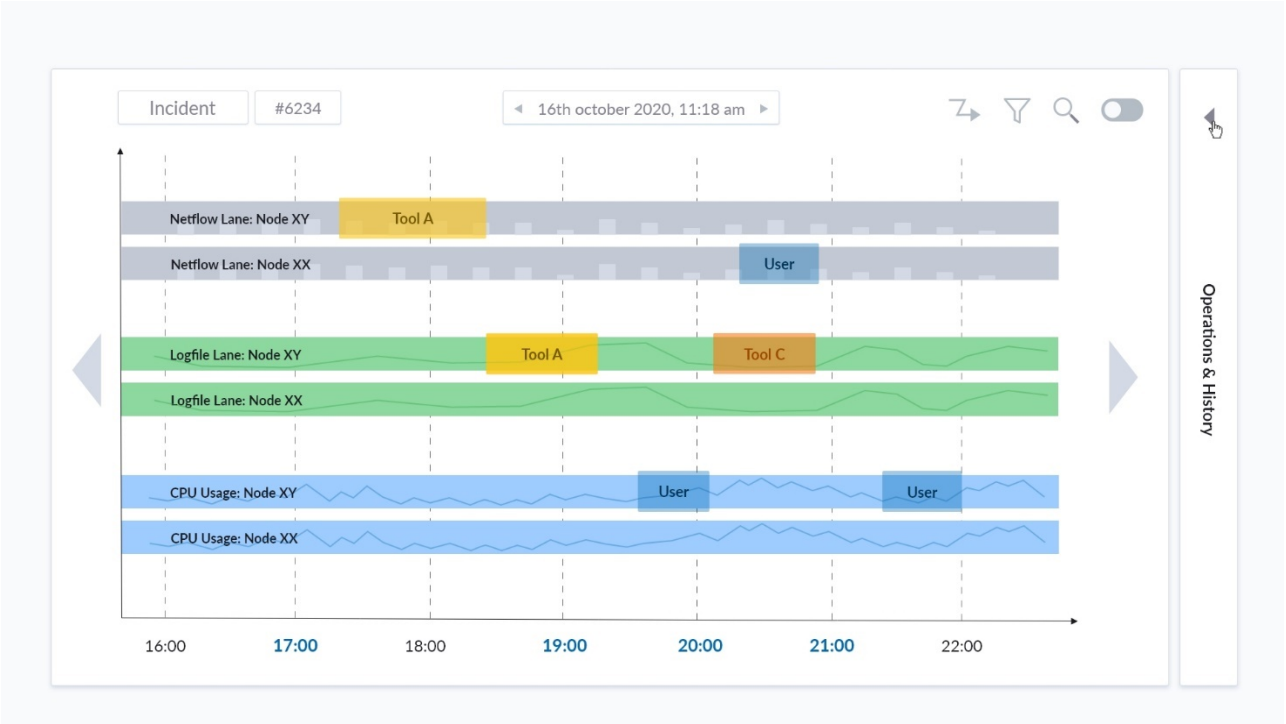


Figure 22: Investigation dashboard

The main panel is the central view for each investigation. At the beginning, it provides an overview over all data sources which are available in the SIMARGL toolkit and might be relevant for the current investigation. The SIMARGL UI uses two elements, lanes and marker, as basic graphical abstractions for the visualization of data sources. Each available data source is either displayed in an own separate lane or as a set of markers on related lanes. This depends on the characteristics and the type of the data source and will be further explained in section 4.3.1.1. Each lane contains a graph or, to be more general, a suitable visualization for the underlying data, whenever possible. All lanes are correlated to the time dimension and the corresponding timeline is always shown at the bottom of the visible lanes in the overview.

The panel allows to scroll horizontally through the overview of lanes according to the timeline to overlook all data sources at different points in time. As there might be more data sources available than can be displayed at the same time on the screen, the panel also provides vertical scrolling to view more lanes. With the usage of these actions, an analyst is able to get a good overview of the corresponding data and to plan the next steps of the investigation. Equipped with the information gained from this first overview, the analyst can start to select interesting data sources in relation to the incident in investigation which is described in section 4.3.1.2.

4.3.1.1 Data sources, lanes and marker

As briefly described, all available data sources of the SIMARGL toolkit are either depicted in the main panel as a lane or as a set of markers. To understand the difference, it is necessary to describe at first, how a data source is defined in the context of the UI.

In general, every data source originates from a data provider and identifies one specific data output of the provider which is available in the data layer or the core layer of the SIMARGL toolkit. In this context, the notion of a data provider summarises all different components which produce their own output data. A data provider might be:

- a data collector which collects records like log files from a server and pushes entries into the SIMARGL data layer
- a data import module which imports data collected from other sources into the SIMARGL toolkit
- a data transformation module which pre-processes, cleans and converts collected data
- a data fusion module which combines different data sources and generates aggregated output
- an analysing tool which takes available data sources as input for an analysis and generates results
- a tool or module which combines two or more of the previously listed functions

Each data provider can provide one or more data sources and might be integrated as a connected and monitored tool in the SIMARGL UI, if appropriate. But the integration is not required, in this case, the data source will be available in the UI without any reference to a tool. Integrated tools, like the tools of the SIMARGL consortium partners, are able to access available data sources, either provided by other modules or collected on their own, and perform analysis on them.

There are three main criteria for the UI to decide, whether a data source will be visualized as a lane or as a set of markers:

1. Output of analysing to tool
2. Type of data
3. Relation to other data sources

The first criterion indicates that only output data of analysing tools, like events for alerts, are suitable for being visualized as a set of markers. Output data of other data providers, for example simple data collectors or data transformation modules, will be directly visualized as lanes, because these data do not contain any indicators for suspicious segments. Instead, they are often only a stream of recorded data like NetFlow or performance measurements.

But outputs of analysing tools do not get automatically visualized as markers. The second criterion, type of data, is also important. To be visualized as a maker, the output has to contain discrete time-related results of an analysis e.g. identified data segments with unusual DNS traffic.

At least, the third criterion, relation to other data sources, has to be considered, too. The result of an analysing tool has to be assignable to at least one lane which visualizes a data source used for the analysis and therefore contains the cause for the result. For example, a list of events with suspicious requests can be presented as a set of markers on the lane for the corresponding access log of the observed web server. There exists one special case: If a tool produces analysing results of a suitable type for being displayed as a marker, but these results are not related to any other data source, respectively lane, this marker will be displayed in a separate lane for the tool which only contains marker and no further data.

To summarise, every data source is visualized as a separated lane, if not all criteria for marker match. When the corresponding data provider is integrated and connected as a tool in the SIMARGL toolkit, this lane is also assigned to that tool. This concept allows flexible data handling in the SIMARGL UI depending on the specific environment where the SIMARGL toolkit will be deployed and is configurable for the available data sources and tools, as further described in section 4.5. For details on the different data source and the different types of data in SIMARGL see deliverable D4.2 on data production.

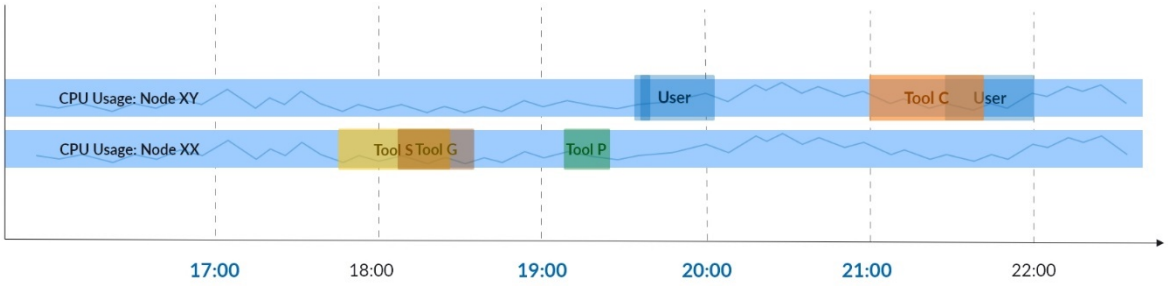


Figure 23: Data sources depicted as lanes or marker in the SIMARGL UI

The excerpt in Figure 23 depicts data sources visualized as lanes and marker. While lanes provide a suitable visualization for the underlying data correlated to the time dimension, a marker clearly indicate suspicious segments or points in data sources according to analysing results of available tools. This helps an analyst to quickly overlook and identify interesting positions in the data. To further assist an analyst, lanes might optionally be grouped according to the type, e.g. NetFlow or access log, or the origin of the data, which might be a node or cluster in the network as well as a connected tool. If there are more lanes available than displayed on the current screen, additional lanes will be lazily loaded and displayed on scroll down. Marker and lanes provide basic actions and allow access to details of the underlying data or results. At least, even a combination of two or more lanes and marker in a single view is available based on the data sources overview. These functions and features will be described in section 4.3.2 on the storyboard and in section 4.3.3 on the inspection of source details.

4.3.1.2 Data source selection

As described above, all available data sources are listed as lanes or marker in the overview of the main panel for a first view. During and after this brief inspection, it is important for an analyst to select the data sources, respectively lanes, which are interesting and might be important for further investigation. Therefore, the SIMARGL UI provides functions and mechanisms to assist a cyber security expert during this selection process.

The first mechanism is the automated smart preordering of lanes, respectively data sources, in the main panel according to notification sources of the current incident. Lanes, which belong or can be assigned to a notification for the incident, are listed at first. For example, a tool has identified a suspicious segment in a NetFlow and raised a notification, which lead to an incident. In this case, the lane, which displays the corresponding NetFlow data and contains a marker from the specific tool, is listed on the first position. To preorder further lanes, additional smart rules can optionally be applied, like listing other lanes with marker from the same tool or lanes with marker at nearly the same time on the following positions. In addition, rules might be learned from previous usage or practice. If, for example, in previous investigations often the output from tool A and tool B has been selected together and combined by an analyst, the lane or lanes from tool B might then be automatically listed on the following positions. The usage of smart rules is optional and depends on the preferences of a user. Therefore, it can be configured in the user settings, where, in addition, a specific fixed preorder can be set as another option, too.

Based on the automated preorder, the main panel provides functions for sorting, filtering or searching data sources as depicted in Figure 24.

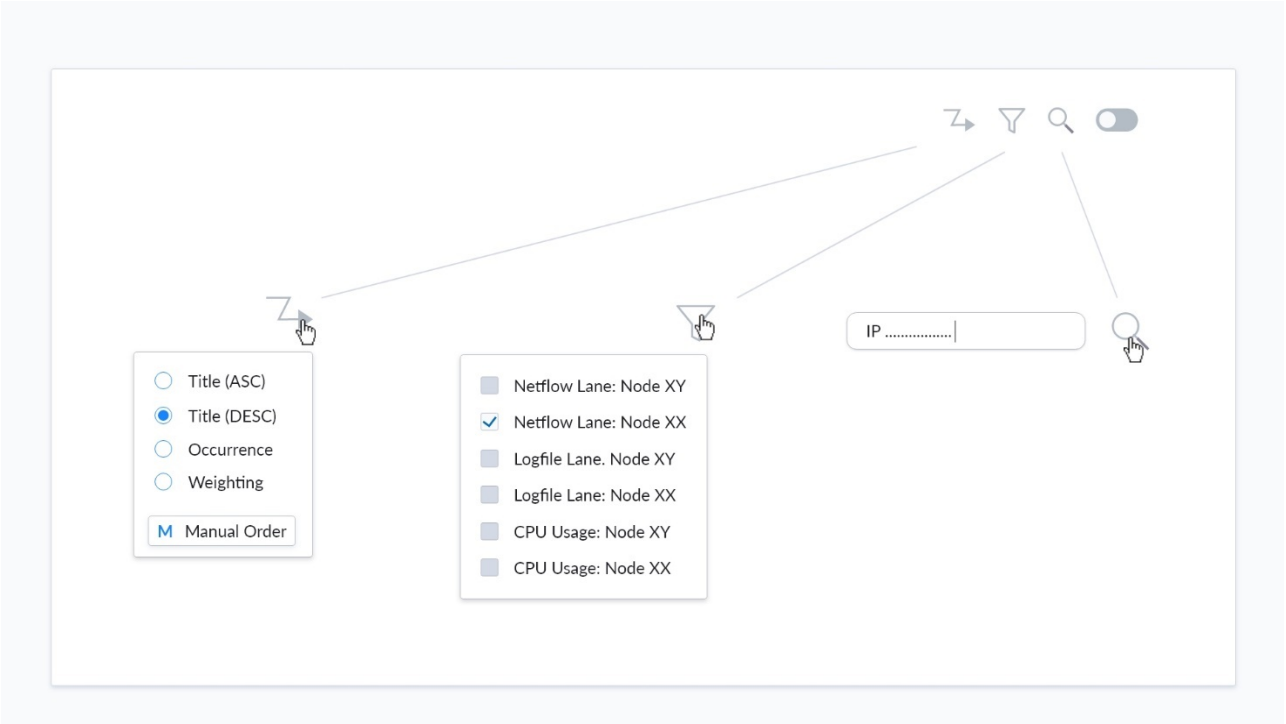


Figure 24: Functions to assist a cyber analyst during the selection of sources

Sorting can be performed either by using a separate box, which is accessible from the respective icon in the top menu of the main panel, or by using drag and drop gestures directly on the lanes. As a result, the lanes will be displayed in the desired order. In addition, this and the other two functions can further be accessed via keyboard shortcuts.

In order to select sources for further investigation, an analyst is able to use the two functions filtering and searching. Both functions are also available from the top menu of the main panel. The filter is accessed from a dropdown field and contains a list of all available data sources with associated checkboxes. By checking or unchecking the checkbox of a data source, the corresponding lane will be added or removed from the view in the main panel. The search function provides a full text search and can easily be used by entering a keyword or search term like an IP address, a name or a note. According to the search results, only lanes which contain matching data will be listed in the main panel. If a filter has been applied previously, the search only includes currently visible lanes per default. But one can choose to expand the search to all available data sources. Additional matching lanes will then be added below the already visible lanes from the filter. After inspecting search results and possibly applying actions like adding or removing lanes in the filter, an analyst is able to reset the search. In effect, all previously visible lanes will be shown again.

With the provided functions and mechanisms described above, a cyber security analyst is able to identify and select the important data sources during the investigation.

4.3.2 Storyboard

The second step, which is accessible in the investigation dashboard, is the storyboard step. Based on the visualization of lanes and marker, the storyboard, as depicted in Figure 25, enables an analyst to compose interesting segments on the data sources for a deep investigation.

Supporting this investigation operation, the storyboard view does not only offer the visual representation of the different data sources, which are or might be related to the occurred notifications, but also allows to dig deep into single log files and provided datasets itself. This data is also a direct and unmodified representation

of the stored collections within the Elasticsearch engine and contains all that information provided by the single tools and input sources. Collecting and loading the complete list of these datasets into one flow would lead to high loading and waiting times, which will cause a break in the user experience. To avoid such behaviours, the SIMARGL environment will first load the most important and related datasets for building the first visualizations and load the rest in the background.

Figure 25: Storyboard, showing the context-sensitive right-click menu

4.3.2.1 Graph visualizations

The entry-point and default view in the storyboard for the executive operators will be the visualization of the selected or depending data sources. Each single data source will be separated onto one lane, using the configured graph system and visual representation, showing the marker which are set by the respective tools and executive operators. The timeframe is based on the corresponding notifications of the incident or on a manually set timestamp, if the storyboard investigation does not directly depend on an incident as already described in section 4.2.4. However, the shown timeframe is shared between all lanes and can be changed at any time in the storyboard itself. The user is able to move forward and backward to past and future times and events. These functions will be executed on all shown lanes and graphs at the same time. The same parallel interaction also applies, if the operator zooms in or out to show a smaller or larger time period. This can also be achieved by using the mouse wheel or the options provided on the top-right aligned dropdown menu, where the user also finds additional operations.

As described in section 4.3.1.2, the pre-ordered positions of the single data sources, or at least of their single visual representations, are based on the used notifications, a smart learning system and on the preferences of the operator. However, the executive operator has always the possibility to change the order, toggle their visibility in general and to filter and search through the different lanes. These functionalities are available using native drag and drop features, different dropdown and modal menus, the context-sensitive right-click context menu and visual options and interfaces on the lanes itself.

The visual representation of the single graphs depends on the configuration, where the data source and tool has been registered onto the SIMARGL frontend environment but can also be adapted by the user. During

the configuration process, an administrator of the SIMARGL toolkit can choose either one of the default-offered visualizations or develop a completely own visualization using the abilities and interfaces of the SIMARGL UI and the related libraries and packages. However, the default visualizations include bars- and line charts, heat- and geolocation maps, tables and columns, gauge and pie charts as well as various other similar graph types. Programming and designing own visualizations using the provided libraries are further described on section 4.5.3.

Another provided feature allows a SIMARGL operator to combine and overlap different visualizations of the same or of different graph types. This can be a useful feature to check or verify different input sources, which might belong together or are based on the same or similar systems or environments.

4.3.2.2 Marker from tools and users

The visual representation of the graphs itself does not only include the lanes but also the, already above mentioned, marker, which are set by the tools and might be part of the underlying notifications, at least if the storyboard is based on an incident. The executive operator has always the ability to create own marker and to edit and delete marker and comments created by other users, too. Furthermore, a marker can also be applied on multiple lanes as well and is not only connected to a single source. This is a very useful feature, because it allows an analyst to mark suspicious or interesting segments for deep inspection and to provide, like a storyboard, an overview for directing the investigation. In addition, it supports collaboration by presenting all marker including added comments to other users. Markers, which apply to multiple lanes, are shown quite differently on each single graph itself. Since the markers are very important, it is also possible to display them solely without the background visualizations in a compact view mode, as shown in Figure 26.

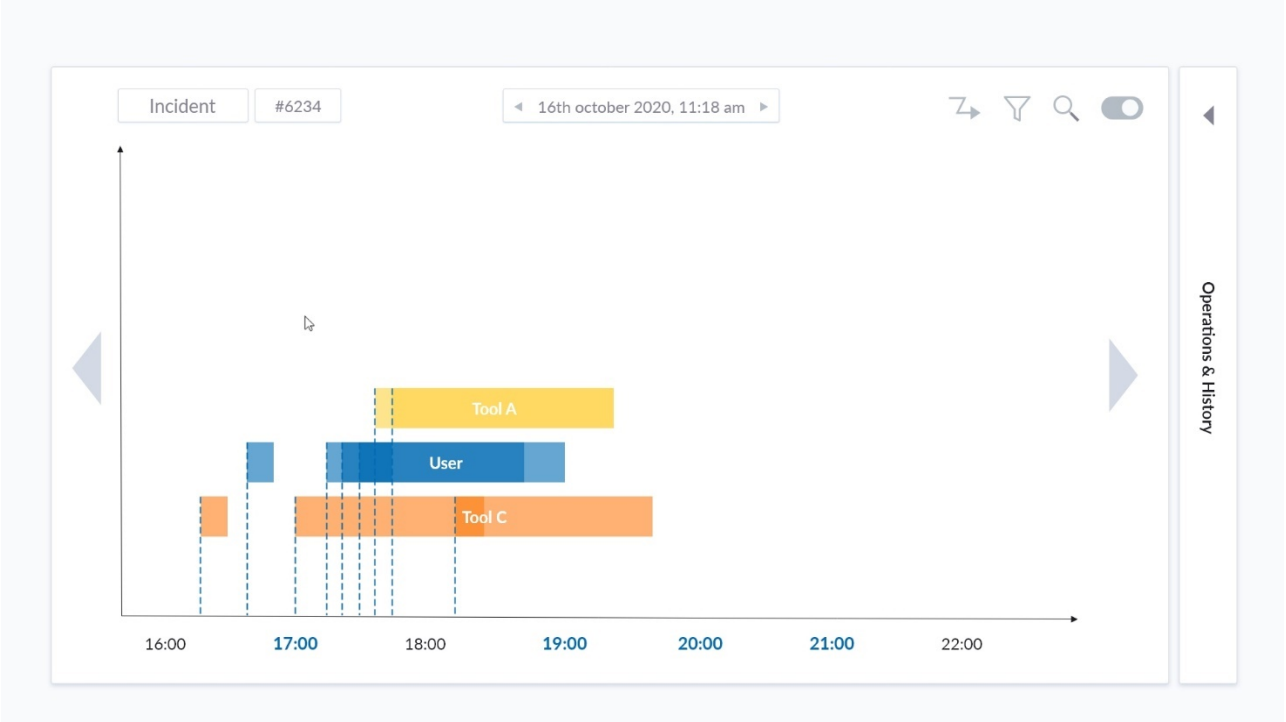


Figure 26: Marker only mode of storyboard

In this case, the markers are sorted by tool and by user regardless of the appeared data source, which may lead to a heatmap similar design when multiple marker overlap themselves within the same timeframe. As in the normal, main view mode of the storyboard, this also contains the specific starting points, showing the initial timestamp of the marker itself on the main timeline. This supports the main overview, allowing to directly see and interact with a marker as well.

Each change made, e.g. if an operator edits or deletes a marker, will be included in the incident history where it can also be reverted as well. The main goal of this history is to list and show all performed operations directly, especially for incidents which are pushed to other operators, groups or levels of investigation. Thus, the newly assigned operator can quickly find out which operations and actions have already been performed and where one should focus on.

4.3.2.3 Context-based operations and options

The whole storyboard view offers a switchable sidebar as well as a sensitive right-click context menu, which both offer available operations and additional options. The sidebar-menu provides more globally affected options, according to the based incident or to the non-incident-based investigation mode. These options contain the possibility to close and mark an incident, for example as “False Positive”, to request or assign additional operators or roles and to push an incident to another investigation level. The sidebar menu also contains the incident history, which shows and lists all operations already performed and actions made by each single operator. Another sidebar action contains a global comment and discussion system, which does not only allow an analyst to keep notes and ideas, but also to talk about occurred issues with assigned colleagues.

The right-click context menu instead offers operations based on the visualization or marker, where the click has been made. On graphs, respectively lanes, the operator is able to configure the shown timeframe, for example to zoom in and out, and to change the used colours and graph scheme. Creating new comments and marker on the clicked position is possible as well. The context menu on single marker is assigned to operations such as the ability to edit, expand, delete or comment on it, inspect the based dataset and log lines in the further described detail views and to zoom in to the affected timeframe, too. Tool-based marker also offer the option to investigate and inspect the selected issue on the tool’s web interface directly, if this is provided by the tool and configured in the SIMARGL UI toolkit as well.

Anyway, the available operations and options can be partly configured by SIMARGL administrators and an additional programming interface also allows to change and add further operations as well. This offers the possibility to directly include operations, which are based on or need to interact with the tools environment which will additionally lead to a better user experience, because a SIMARGL operator does not need to switch to the tool itself. However, this functionality is optional, and must be configured by the respective tool provider and the related SIMARGL administrator.

4.3.2.4 Detail view

Showing the visual representation of the single datasets and the tool- or operator-based marker are perfect to see the timeframes and the deflections of the occurred issues. But inspecting single log lines and detailed results are indispensable for a deep investigation process. Thus, the frontend UI allows to show the logged information, line per line, based on the selected timeframe by the user or the respective marker. This detailed information is presented within a popup-similar modal, which overlays the graphs and is described in more detail in section 4.3.3.

4.3.3 Inspection of source details

According to important aspects from state-of-the-art research, as described in section 2.3, an analyst should have the ability to deeply inspect source data. Therefore, the SIMARGL UI supports the inspection of the data in detail, even at the level of single entries line per line in the corresponding data like logfiles, if available. Furthermore, the environment also supports to search, filter, join and transform the source data directly without losing or altering the original content as stated out by Fink et al [38].

The access to these source details is provided directly on the single lanes in the storyboard of the main investigation dashboard as depicted in Figure 27. The amount and range of the source details can either be defined by the marker, made by the tools, already marked and selected areas by an analyst or by selecting own parts directly on the visualized graph itself. This selection can also be defined with high precision by zooming into the graphs and lanes itself, which expands the respective timeline into smaller time intervals and periods. The UI also offers the possibility to create multiple selections on the same or on different lanes and showing them altogether in the detail view.

Figure 27: Selection of time period for inspection of detailed source data in normal mode

The storyboard also supports a marker-only mode, which will hide the single source lanes and puts all marker, made by the tools and the users, into one single view row per row as described in section 4.3.2. As depicted in Figure 28, this view should assist the user to create a selection together with multiple marker or between them as well.

If a selection has been chosen or a custom one has been made, the user is able to apply a set of actions based on it in both view modes of the storyboard panel. One of the available actions allows to open the corresponding source details in a modal view laid over the main panel.

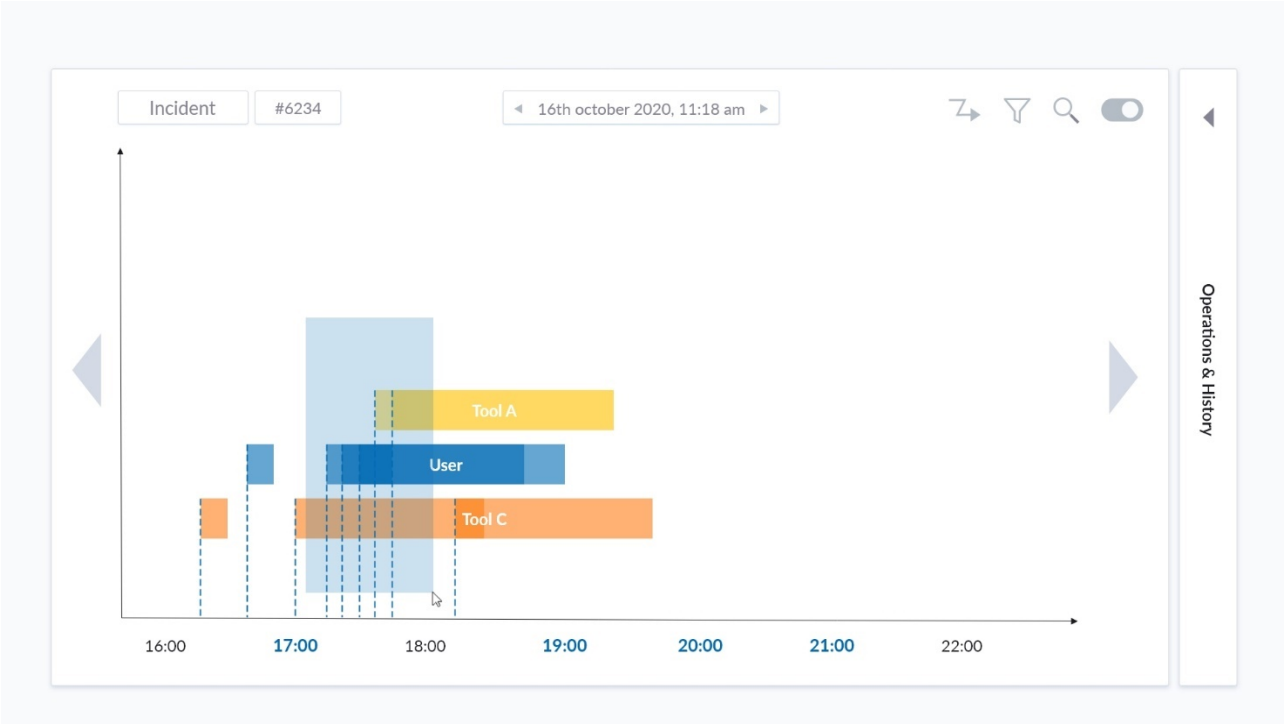


Figure 28: Selection of time period for inspection of detailed source data in marker only mode

The modal view panel for source details is depicted in Figure 29 and shows single log data, line per line, containing at least the exact timestamp, type, the log content itself, as well as the origin lane where it belongs to. Achieving a better overview, the user is able to filter these data per lane, tool, marker and search within the respective content as well. These functions can be easily accessed from the top menu of the panel. However, this view does not only provide a highly deep data inspection, it also allows to perform the same actions as on the storyboard panel itself. This means, that a user is able to mark multiple lines from the different lanes and sources and create comments or own marker using them.

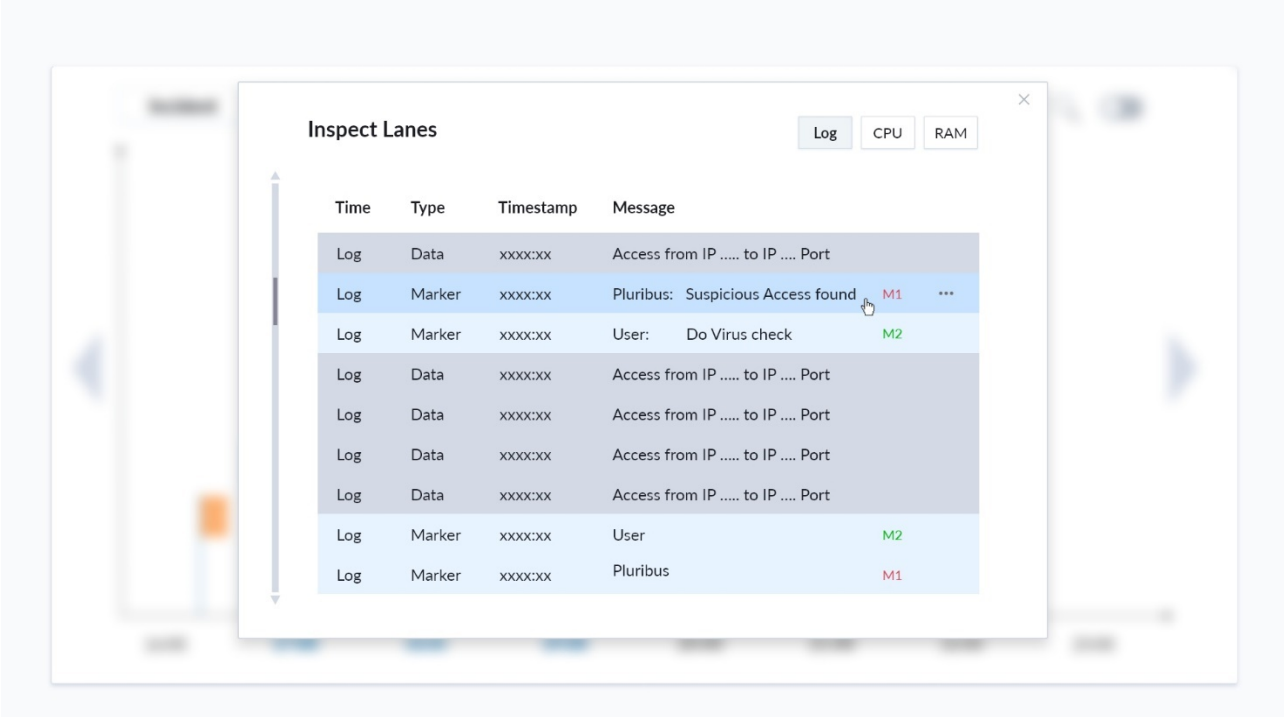


Figure 29: The modal view panel for source details

In general, all markers, regardless if they have been made by a tool or by a user, are also part of the shown source list. The markers always surround their associated log lines as depicted in Figure 30 and are not only shown by their type but also illustrated by a specific highlighting colour as well as a right border visualization, if the mouse hovers them explicitly. This allows the analyst to clearly identify the data entries which belong to the marker. Additional actions and interactions are available, depending on the marker, which are accessible from the three-dot menu on the right side. The most usual action is to toggle the respective visibility or to edit, expand or shrink the number and amount of associated log lines. Marker which are created by a user can also be deleted completely, but with the option to restore them using the incident history, which is further described in section 4.3.4.

Analysis results, which are shown on the lines, can also be opened within the corresponding tool, if available. The respective data get passed to, so the operator does not have to search again for the same alert or notification. This does not only require an accessible web interface provided by the tool, but also a registered call handler within the SIMARGL frontend configuration, as described in section 4.5.2.

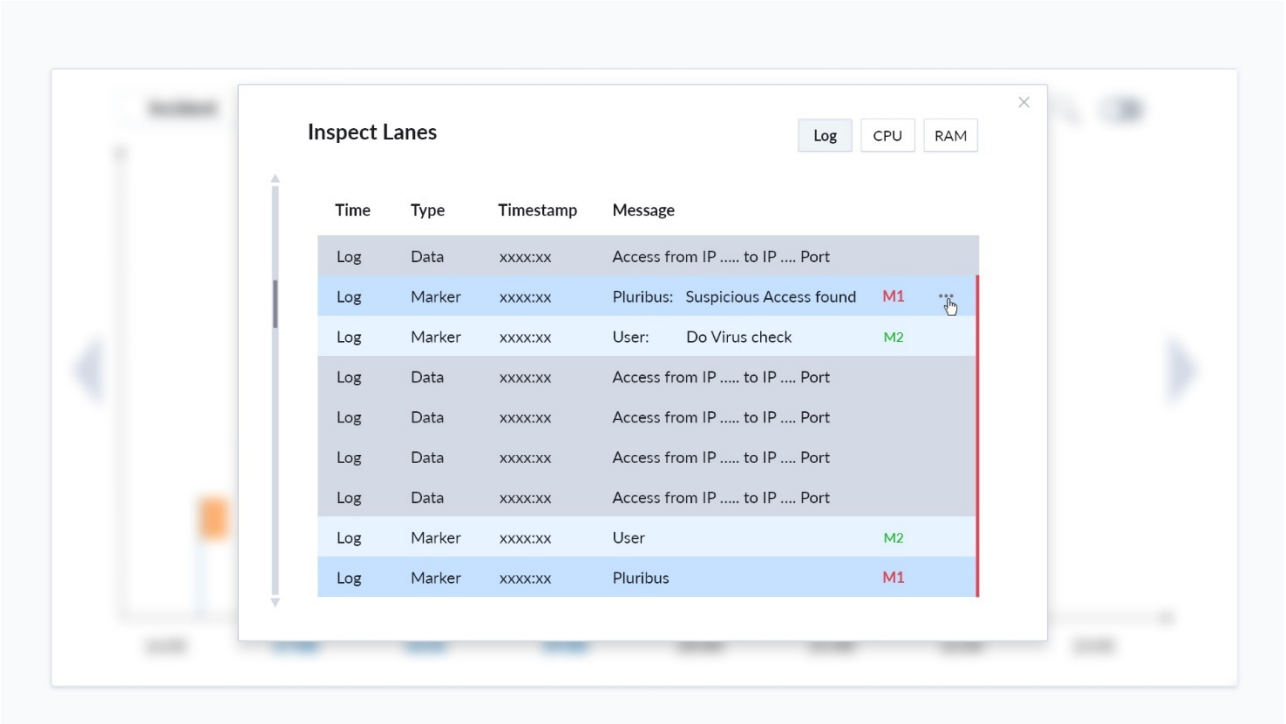


Figure 30: Additional available options on respective marker items.

4.3.4 Support for collaboration

The SIMARGL frontend UI supports collaboration and workflow generally in multiple ways. First, the status and incident information as well as the finished actions, including the ticket escalation. Second way is the inspection of already performed investigation steps, such as handling or deleting marker and comments, namely an incident history. Third way is the collaboration of the SIMARGL toolkit with additional ticket systems, such as OTRS Storm, by exporting and important their resulting data.

The first two ways are provided on the right panel of the investigation dashboard, as depicted in Figure 31, which is separated into two sections: A quick action menu as well as the incident history. The first area allows a SIMARGL operator to handle incident-based actions, such as setting a new incident status (for example “Work in Progress” or “False Positive”), assigning other roles or users or pushing the incident to another level

of investigation. The single actions lead to a popup window, which allows the user to add additional, and may also require, information and data. For example: If the user performs the action to push an incident to another level, he can add an additional description explaining why this action is needed. Or when the user wants to assign additional roles and users, the popup modal offers the respective option showing all available roles and users also including a filter and search ability to find the desired ones quickly.

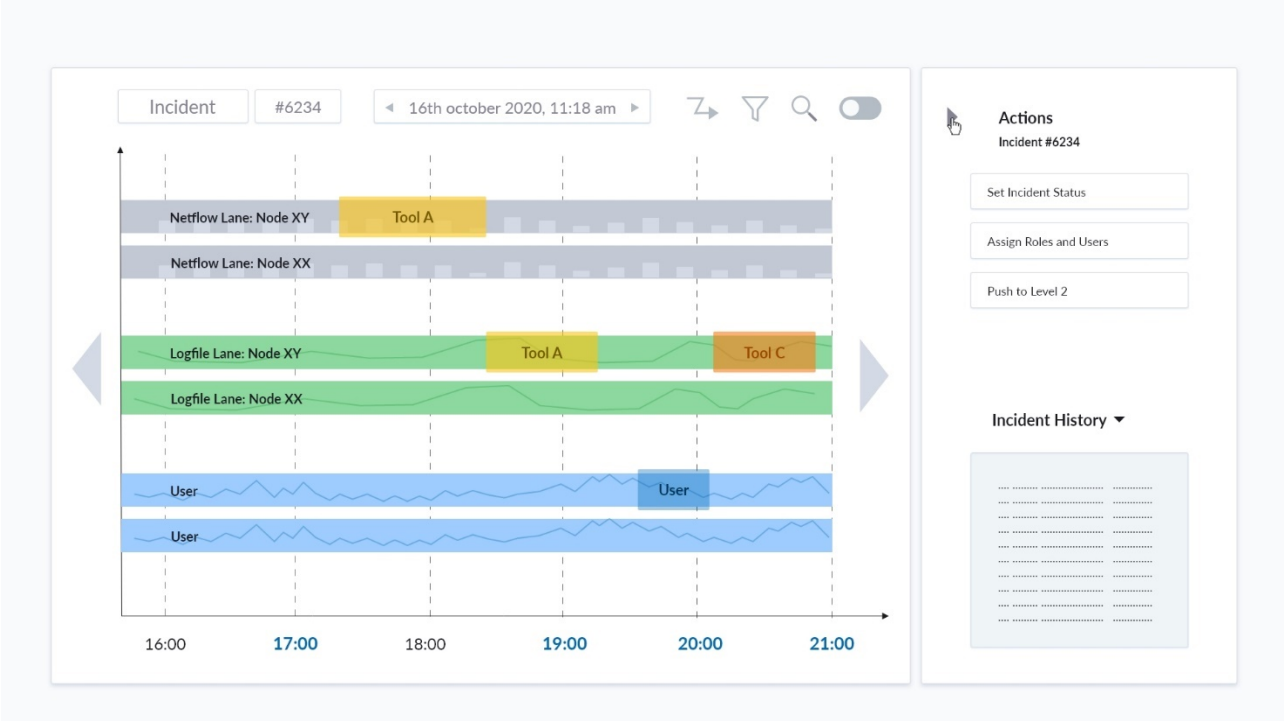


Figure 31: Collapsible right panel for workflow support

The second area shows the incident history, listing all actions already performed, such as the assignment and transfer to another user or the conducted interactions with the graphs and the shown marker. Already performed actions like the editing or removing of a marker or comment, can also be reverted there with a single button click. Every item within this history also provides a timestamp as well as the associated user, too. However, the section appears collapsed per default allowing to focus on the investigation itself on the first view. The user can expand it whenever he needs to, but he can also configure this panel to be expanded per default.

The connection to other ticket systems, such as OTRS Storm, can may be done automatically by providing a respective API or by calling the respective Webhooks from the third-party system. The possibility to import and export the desired data for a manual integration, using well-known formats

The primary content of the investigation dashboard are the source data lanes itself, as depicted in Figure 32, including the marker set by the tools, or the single operators, as well as their comments. Every executive operator is able to see all actions already performed, including his own ones, of course. These comments and marker are specially colorized and always contain the username, role and timestamp. Everything notable can also be viewed together with the same information in the incident history as well.

Figure 32: Inspection of results of investigation from previous analyst

The comments itself are also always shown with the timestamp of its creation, as well as the responsive user. The same applies to the user made marker, which may have been assigned to multiple lines, represented as small dotted pattern on the top and bottom of the respective marker. However, each user and tool action can be commented by each operator, but also directly edited and removed as well. The general visual representation of the main panel can also be highly adapted by each single user on his own, as further described in section 4.5.

The environment supports multiple users to interact and work on the same incident simultaneously, without disturbing each other. This is possible, since each the dashboard interactions such as manipulating the time span or handling the assigned notifications are done within the used browser and each made action are stored on its own. The whole interaction is also designed with the “everyone can do everything” principle, so per default there are no actions nor operations which are not accessible to a specific operator or role. However, of course this can be configured as well by the SIMARGL administrator, to adapt the SIMARGL environment to their individual and specific rules and settings.

To summarise, with its structure and behaviour the investigation dashboard addresses important aspects for the UI in SIMARGL as identified by the state-of-the-art research, the preparational steps like expert interviews and the questionnaire, as summarised in section 3.3. Especially, it allows the selection of sources for a better alert investigation, supports the ability to dynamically combine multiple data sources and tool outputs, to correlate them over the time dimension and provides a deep inspection of the underlying data.

4.4 Connection to specific tools

Another main aspect of the SIMARGL toolkit is a comfortable and interactive connection between the UI of the complete toolkit and the single tools of the consortium partners itself. In this context, the jump to the web interfaces of those tools is indispensable, since actions and executive interactions might be done directly in the tool’s environment. This workflow is part of section 4.4.1.

Section 4.4.2 instead completely focuses on the whole architecture and integration of the single tools and their data sources through the main SIMARGL toolkit and up to the SIMARGL UI environment itself. For the development of the frontend environment, an important decision had to be made on which visualization platform SIMARGL should be based on, since a fully self-developed solution would not reach the stability and performance-proved abilities an existing solution already achieves in the available development time throughout this project. Furthermore, a lot of code would have to be written for basic functions one already could base upon. Therefore, the SIMARGL UI component depends on a free open source solution and the choice had to be made between the two well-known environments Grafana and Kibana, which is topic of section 4.4.3.

4.4.1 Workflow of switching tools

The SIMARGL toolkit serves as global overview for the whole monitoring of the in-use network cluster, the single nodes as well as the single tools as used on the network nodes. It perfectly fits a summarised but also highly detailed view for a professional and deep investigation, still with the possibility to switch to the single web interface of the selected tool itself. This may be required, if specific actions, interactions or advanced operations, which depend on the core abilities of the tool, can only be performed on the respective environment.

Thus, an action to switch to a tool is not only present on the list of available tools itself in the overall UI and main dashboard, but also on the single storyboard investigation dashboards as stated out in section 4.3. This allows an operator to switch to the desired tool using the available information, notifications and marker the tool requires without the need to search twice for the same issue. Another position is the detailed source view, which can be accessed on the storyboard investigation dashboard. In the detail view, marker at specific positions made by the tool itself allow an operator to switch to the exact same result. All three options are depicted in Figure 33.

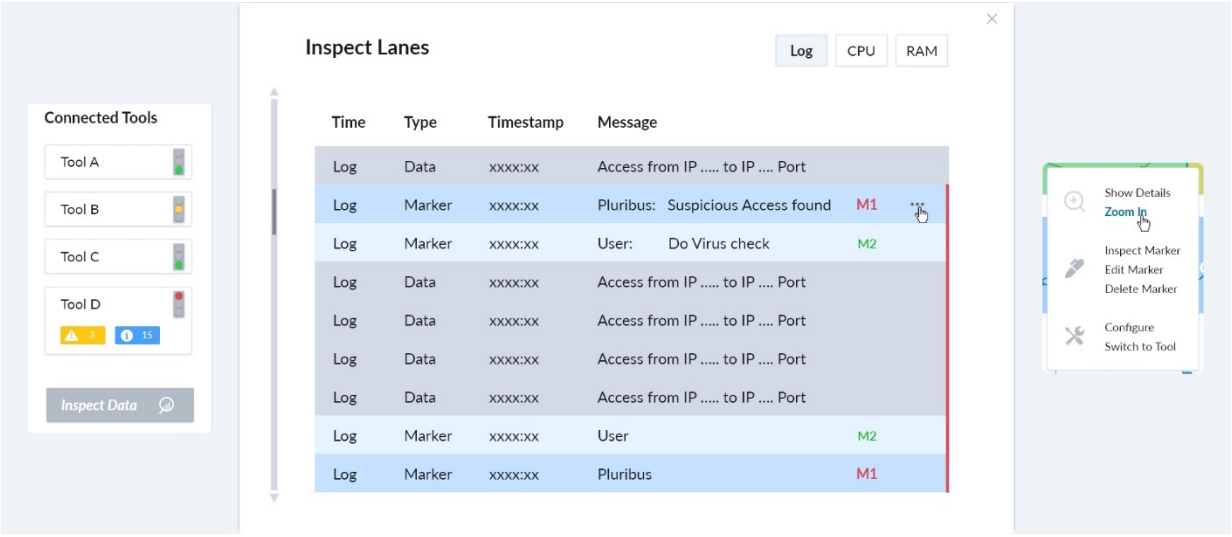


Figure 33: Three options to switch to specific tools

Such a tool redirection requires a standardized or configurable interface, which pass the required data from the SIMARGL user interface to the desired web application of the tool. The SIMARGL toolkit offers two possible ways to handle such a tool switching workflow. The first one, as described deeply in section 4.4.1.1, uses the URL of the tool's interface itself. This method might not require heavy changes on the external interface, unless the tool works completely without this well-known method of passing data. The second solution is more detailed on section 4.4.1.2, which pass the request data to a defined tool interface and expects an URL with a unique session ID as response, which is used to redirect the user.

4.4.1.1 Switching tool by using a URL request

The easiest way of passing information and required data to the web interface of the respective tool is by adding them to the query part of an URI. This method only requires a defined structure between the tool provider and the SIMARGL environment, which would also allow to be adaptive by offering a simple configuration page on the SIMARGL toolkit.

The URI scheme also allows an inline authentication using a username and a password, which also allows to secure the connection, without offering them to the public. An additional port and path to access the tool's interface on its desired location is possible as well. The only restriction is the maximum allowed and supported length itself, as well as its readability which does not really matter for the end-user.

The length of an URI is defined within the Hypertext Transfer Protocol Version 1.1. While the first version (RFC 2616) does not see any limit for this, the second one (RFC 7230) recommends but not exactly defines a maximum length of 8,000 characters. However, the length restriction itself depends more on the used browser and server software. While the last one can be configured, the browser cannot. This means, that by supporting the Internet Explorer the maximum available length will be restricted to exactly 2,048 characters. Modern browsers, including mobile ones, allow at least 8,192 characters up to over 64,000.

A SIMARGL operator will not be interrupted by this process on the frontend UI, except that a new tab opens on his browser pointing to the respective URL as described above.

4.4.1.2 Switching tool by sending a request package

The second supported method to send the required data and information to the respective tool's interface consists of a previous request that sends the data directly in the background and waits for a corresponding response. This response should contain either the respective URL containing a tool depending session ID, which is internally connected with the previously send package, or a respective error message which should be shown to the end user.

While the data type, using the JSON format, cannot be adapted by the tool provider he still is able to change the respective structure and data key values itself. However, this might require major changes and adaptations on the tool's core system, as well as a temporary or permanent storage of the passed dataset. The authentication to secure the request can also be defined and is standardized in the HTTP protocol.

Another advantage of this method is the insignificant limitation of the first request, since this get completely defined by the used server software of the tool itself. The tool developer can also log those requests for additional monitoring, too.

Unlike the URL method, this method might have an interrupting effect for a SIMARGL operator. This might occur, because an operator needs to wait until the first request has received a response from the server. In the meantime, a small loading circle will be shown followed by a new tab opened within his browser on success or an error message on a bad response.

4.4.1.3 Data source output collections

The SIMARGL environment intends to collect and store the outputs from all tools and their data sources, which are or may be relevant for the general graph or data visualization, the end-user information and notification as well as the general tool status examination and explanation. To be more precise, the SIMARGL UI distinguishes between the following three data collections.

Visualization collection

Generating the graphs and visual representation of a tool or it's depending data sources requires at least on collection of data on which it can build and base on. This continuous stream of data, which is more explicitly described in section 4.4.2.2, is an important point for the investigation of possible incidents, alerts and all other kind of issues. Therefore, the topicality as well as the long-time storage of this data are of the utmost importance.

Status collection

The SIMARGL environment does not only take care of the data visualization, but also of the tool status itself, which contains the performance, up- and downtime as well as its health status in general. This information can be used to explain specific situations, like breaks or gaps in the visual graphs, or assist by performing a special kind of investigation on the visualization mentioned above. In addition to the status report itself, SIMARGL also uses own techniques for recording status data and adding them, if necessary.

Notification collection

Since the analyzation of the data and its data source reports are performed by the single tools itself, they must notify the SIMARGL environment about all kind of abnormalities as well as all possible incidents, issues and alerts. Based on these notifications the SIMARGL UI takes care of the distribution and dissemination of the received alerts to the end user using the available ways. All passed notification will be stored for a long time to provide a respective history and permit maybe essential declining investigation.

4.4.2 Architecture integration

The logical architecture integration, as shown on Figure 34 below, starts with the interaction between the single partner tools with the Apache Kafka distributed streaming platform, which passes the data to the Apache Spark cluster computing framework. A more detailed description of this basic architecture using both Apache software stacks can be found on the deliverable document 2.5.

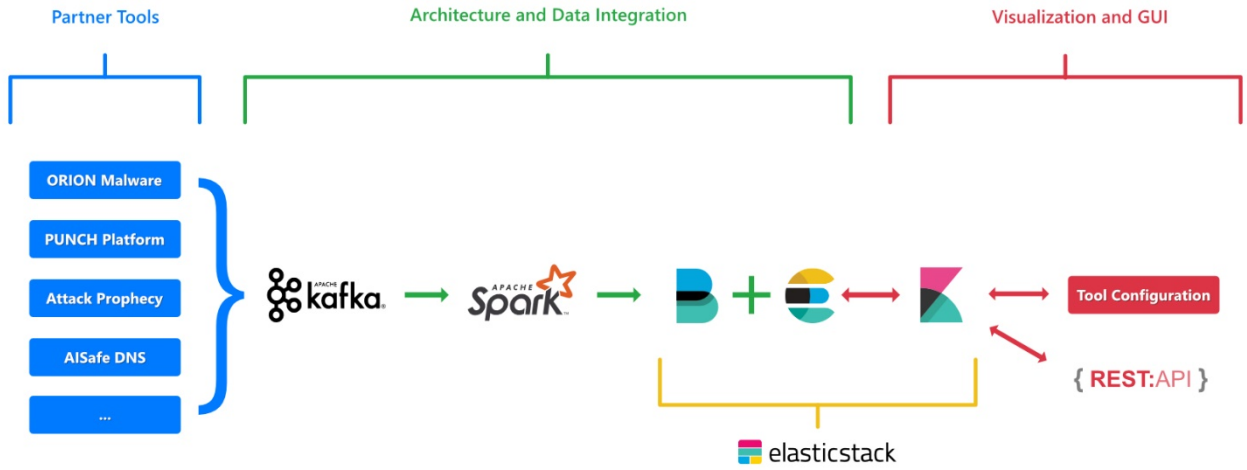


Figure 34: The final architecture starting with the partner tools until the final visualization and GUI.

This document focusses on the Elasticstack (formerly known as ELK), starting with the registration and integration of the single partner tools with the SIMARGL toolkit and UI environment, including the single tool configuration pages which are further discussed in section 4.4.2.1. After this establishment is made, the parsing of the received data by Apache Spark into the Beats family and similar tools, which are connected to the Elasticsearch, can be recognized and managed by the visualization platform.

Section 4.4.2.2 describes the normalization and preparation process for the Elasticsearch analysis and search engine. The following section, 4.4.2.3, is dedicated to the topic of embedding the prepared datasets and information into the visualization engine and its general surface, UI principles and UX behaviours.

Additional services besides the data visualization, provided by the SIMARGL toolkit for the single partner tools, can be directly accessed with the available REST API, which is topic of section 4.4.2.4. Furthermore, the tool developers also have the possibility to connect and offer visual interfaces for the REST API systems of their tools as well. This way the operator can execute or interact with an external tool without leaving the SIMARGL UI frontend.

4.4.2.1 Tool integration into the visualization framework

The tool provider, or at least the responsible tool integrator and further SIMARGL administrator, is fully responsible for the tool and its data source integration into the whole SIMARGL environment. The SIMARGL toolkit offers an easy to use configuration UI, as further described on section 4.5.2, helps to easily add, edit, delete, configure and manage all available tools and their data sources including additional checks to ensure a working integration. A single tool configuration, as stored on the SIMARGL environment, consists of the following important data amongst other ones.

Unique identifier

Using a unique identifier allows SIMARGL to separate each single tool and call them in a defined way without searching or parsing the basic information. This is a common and well-known way of storing data in a unique and identifiable structure. This unique identifier must also be passed by the tool into the data layer of the SIMARGL environment to ensure and guarantee an association between the data and the visualization layer.

Basic Information

Some basic information, consisting of a descriptive title, a small description and additional meta data are highly recommended for the end user. It replaces the handling and working with just unique identifiers and allows a human readable way to interact with the single tools.

List of node connections

The tools are mostly running on just one or a set of specific nodes or networks, within the configurable network cluster. This information is highly important for a few SIMARGL UI concepts, such as the ability to connect multiple data sources from different tools or only from a single tool into one view, when selecting a single node. A single node connection consists of a unique node identifier, as well as some additional descriptive information for the end-user.

List of source settings

Each single tool must at least provide one data source or general output connection, as shown on 4.4.1, which gets parsed, analysed and stored in the Elasticsearch, as described in 4.4.2.2. The connection between the tool and its data source(s) are required for the visualization and graph generation. Therefore, a single setting not only contains a unique identifier, which also must be used when sending output of data source to the Elasticstack, but also a description of how it should be displayed. However, the visualization configuration itself is further described in section 4.5.2.

All this information provided are stored as plain human readable JSON document on the SIMARGL system, which gets read and parsed on each reload. This allows to easily change or generate these files with additional tools, which may be an important aspect on huge network clusters or constantly changing tools and systems. It also allows to easily backup and restore the whole tool environment with minimum effort.

4.4.2.2 Source integration and preparation using the Elasticstack

The integration of the single data source and tool outputs are handled at first by the used Apache software family, so either by the Apache Kafka Engine directly or by Apache Spark which may also perform additional actions and analysis processes before the data get pushed to one of the installed data-shipper. A data-shipper prepares the information and creates indices for the used Elasticsearch indexing and analysing engine. The main applications used for this process are based on the libbeats library, namely the Beats family, since they are official supported and promoted by Elastic, the developer of the Elasticsearch engine. The main developers also already provide a set of official Beats products, next to many different community ones:

- Filebeat – used for whole log files and similar data structures
- Metricbeat – used for metrics
- Packetbeat – used for network data
- Winlogbeat – used for Windows event logs
- Auditbeat – used for Audit data
- Heartbeat – used for up- and downtime monitoring
- Functionbeat – used for cloud data

However, the Beats family may also just pre-process the past data packages for the additional Logstash engine, which additionally enhances and improves the output and passes the result into the Elasticsearch engine instead. Logstash was previously used as main and only data-shipper within the Elasticstack, but is used today just as enhancement and extender. But Logstash does not only provide the possibility to enhance and extend the provided information, it can also filter and adapt the source with additional and custom extensions as well. This way, the data source can be optimized further more.

Logstash also does not only support the Beats products as input source, but also many different products as well. This way the data-layer for the SIMARGL UI can also be filled with binary files, archives, database connections and further data types next to log and similar plain-text files. Using the MIME type or additional meta- data, such as the MAGIC number of a file, can be used to decide which data-shipper or indices mechanism should be employed or preferred.

The Beats products are generally designed to be very lightweight and performant, but one product usually only works for one type of log or data file. Logstash instead offers a low-latency and dynamic solution to parse but also enhance and filter data of every type, but with the cost of a higher workload on the server.

4.4.2.3 Source data and output visualization

The prepared data can now be used for visualization, using the configuration as defined and described in section 4.4.2.1 above within the desired engines. The most common and well-known systems are Grafana and Kibana, both supports the Elasticsearch system, output and interactions. While Kibana is more preferable to be used with Elasticsearch, since it is also a part of the Elasticstack called software stack and thus prepared for the specific environment. However, more information about the differences between both visualization frameworks can be found on section 4.4.3.

The configuration of the usage of Elasticsearch can be set in the core YAML file on Kibana or within the administrative dashboard on Grafana, by selecting them as data source. Both can be configured to use secure authentication methods, which is highly recommended since otherwise the analysis data can be accessed globally. It is also possible to connect both frameworks with the same Elasticsearch instance or cluster, without interrupting their mutual use, although this setup is unlikely to occur often.

However, after the tools are registered on the SIMARGL environment and Elasticsearch is filled with data and connected to the used UI framework, the visualization can now be generated. This process appears every time the SIMARGL dashboard is loaded, as well as in a defined time interval to always collect and show the latest data received. The live dashboard is also constantly monitored to prevent browser issues when data source connected graphs will be updated or errors occur during the reception of data.

In many cases, the graph generation and visualization process use the core abilities of the respective UI framework, in just a few cases additional libraries are used as well. While Kibana offers a mutable and highly configurable graph engine called VEGA, Grafana can only be extended using the d3.js library, which is also available on Kibana, too.

4.4.2.4 Additional services using the REST API

While most data, which should be shown, used or stored, are passed through Elasticsearch, some services, which are offered by the SIMARGL environment, may need a responding action as well. Therefore, the SIMARGL UI toolkit offers a REST API, which allows to easily interact with the SIMARGL environment without passing the action through the whole Apache based architecture. The API can be called easily via HTTP requests, using cURL or other similar utilities, the well-known JSON format in the request body and the HTTP standardized authentication header. Using the sockets or HTML5-WebSocket technology might be applied as well.

Furthermore, a tool provider can also implement a visual representation of the REST API for the tool to improve the usability and also the performance further. This way, an operator can call commands and interact with the tool's core abilities without switching or leaving the known UI environment and the tool instead does not need to validate or sanitize any switching processes from SIMARGL to the tool's UI interface.

4.4.3 Grafana vs. Kibana – A technical comparison

The differences between Kibana and Grafana are huge and impact the whole environment, which is surprising with the fact in mind, that Grafana was created as fork from Kibana itself, even if this happened a few years ago. Still both frameworks are using the node.js environment and passing through a language-rewrite process from plain JavaScript, using the Angular and React frameworks side by side, to Microsoft's TypeScript using only React for building the whole UI and interfaces.

Both visualization frameworks are developed on GitHub and therefore their creators offer a FOSS (Free and Open Source Software) version of each system. The main differences between both include, but are not limited to:

- The connection and processing of the source data interfaces
- The structure and generation of the graphical user interface
- The scope and creation of (custom) graphs and dashboards
- The complexity and structure of the user and programming interfaces
- The role and user management abilities
- The security and authentication abilities
- The core functionalities and features in general

The following comparison focuses on four main aspects:

1. Connection and processing of source data interfaces
2. Structure and creation of the GUI
3. Complexity and programming interfaces

4. Security and role / user management

Connection and processing of source data interfaces
The Kibana framework mostly is offered within the “Elasticstack” called software stack, formerly named “ELK” which stands for “Elasticsearch Logstash Kibana”. This includes the powerful Elasticsearch engine, which is the main data-connector to the Kibana environment. The Elasticsearch system allows different kind of inputs using the Beats products or the Logstash application, which both is further described in section 4.4.2.2. Grafana instead allows, next to using the Elasticsearch system, also various other data sources. It is even possible to directly connect a MySQL or PostgreSQL database system to it. This allows Grafana to be pretty more flexible regarding the desired data sources and information without relying on a single software.
Kibana allows to access and manage plain data directly, also supporting an own query language, KQL (Kibana Query Language, not to be confused with Microsoft’s Keyword Query Language) on top of the previously used and still available Lucene query syntax. A simple data structure view is available as well, which is the only way to view the available data streams on Grafana.

Structure and creation of the GUI
Both, Kibana as well as Grafana, are currently in a refactoring process where they are switching the main language from JavaScript to TypeScript and the used component framework from Angular to React. In the meanwhile, they are also currently changing the way how a custom application or plugin gets processed and viewed on the dashboard and on custom administration pages.

While Kibana strictly separates the frontend of a plugin from its backend, which is responsible for the data binding and preparation, Grafana instead keeps everything in one area. This allows Kibana to be more lightweight and performant on the final view but requires a longer loading time on the first step, while Grafana acts the other way around. This means, that Grafana may consume more performance and resources and may get slow, depending on how many data sets and graphs are finally shown.

Depending on the visualizations Kibana comes with a big set of core graphs and still offers an own graph-creating system called VEGA, next to a CANVAS environment, which allows to design the related data, too. Grafana instead just offers a set of well-known and most common visualizations, which are still highly configurable. However, both are supporting also the d3.js graph library, even if the connection to the desired data sources as well as its output and preparation must be done without help by the framework.

Complexity and programming interfaces
Kibana offers substantially more advanced possibilities directly on the frontend environment and dashboard, tailored for simple but also professional users. However, all this additional functions and features are very confusing, at least for technical not really affine people. Showing or doing simple tasks may get quite more complex compared to Grafana. Grafana instead offers a really basic environment, still allowing advanced interactions, but they are rather limited. This allows Grafana to be more cleaned up, self-descriptive and understandable.

The interfaces and dashboards of both frameworks are very well documented on the official websites. More complex features and interactions are well described and partly also shown on additional videos and tutorials, which both frameworks offer for free. In general, the end-user should not become a problem to learn and understand the user interfaces, even if it might require some time to get used it. Instead, the documentation of the available application programming interfaces for extending the existing environment are taciturn, partly meaningless or completely absent. While Grafana at least tries to offer a list of available methods and interfaces, Kibana leaves the most important parts undocumented for developers. However, this can also be owed to the current re-writing process.

At least, Grafana offers a high backward compatibility, which allows to investigate other plugins and applications to get a deeper in-view knowledge. Kibana instead is changing itself on almost each minor

release, which makes it very difficult to learn and build a solid understanding for the internal mechanisms, especially because the developers do not document the source code itself either. This might lead to a high maintenance effort for keeping the SIMARGL toolkit secure and updated.

Security and role / user management
Kibana is using Elasticsearch as backend engine, which completely takes over the user and role management, at least, if this option is enabled on Kibana's core configuration file. The open source version of Kibana already allows basic authentication, using a username and a password, as well as a complete user and role management, which should fulfil all requirements. Using the free and open source extension OpenDistro will also add additional services, such as a 2-Factor authentication and more, if really necessary.

Grafana instead just offers a really basic user and password authentication in the free open source version, which is already enabled per default. The role management does not go beyond 3 non-configurable roles, which may not be sufficient according to the requirements of a professional SOC. Currently, there is also no real possibility to change or extend this behaviour without rewriting core components.

In summary, Kibana offers a more extended environment and is capable of more advanced requirements, as they might appear on professional and big SOCs. The system scores with offering many core functionalities, the VEGA graph editor and a highly configurable security, role and user management. The graphical user interface is more complex, but offers also far more access to the core abilities and can be easily learned, even for non-technical people. However, the strict release history, the undocumented code and the ongoing major changes to the core system increase the maintenance effort immensely. Since all used plugins and external tools must be tested and updated frequently on each single patch release.

Grafana, instead, offers a really basic environment, which might still be capable of an advanced usage but misses really deep and partially necessary settings. The user and role management, as well as the security environment, are really basic and cannot be extended easily. Therefore, the graphical user interface is really easy to learn and to understand and the core API provides high backward compatibility, which allows for an easier maintenance with less effort as well.

Hence, Kibana is the better choice for the SIMARGL toolkit and environment, since it already offers important features and functions, such as the security and role management, even if the amount of maintenance is quite increased due to the undocumented API and the ongoing major changes. The comparison matrix below summarises all factors which led to the decision for using Kibana as the foundation in the SIMARGL UI environment.

Table 2: Comparison matrix between Grafana and Kibana

Criteria	Weight	Justification	Grafana	Value	Kibana	Value
Security / Authentication	8	Allows access to several security tools	4	**32**	8	**64**
Role / User Management	5	According to the use and employees	4	**20**	9	**45**
Access to various sources	2	Data will be passed by ES anyway	9	**18**	1	**2**
Easy-to-Use GUI	3	An induction is usually happened anyway	7	**21**	4	**12**
API Stability	7	Stable API means less maintenance	8	**56**	3	**21**
Documentation	5	Complexity and costs are impacted	6	**30**	3	**15**
Community Activity Level	3	Active community may be helpful	7	**21**	6	**18**
Native Full-Text Search	5	Additional help for creating UX tools	0	**0**	9	**45**

Partner Valuation	5	Valuable input from our partners	1	**5**	7	**35**
Total Score			47	**203**	46	**257**

4.5 Configuration

The SIMARGL toolkit focus on many different important aspects. One of them is the general monitoring of the whole network cluster as well as all available nodes, such as servers, which are or should be available, depending on the used environment. Section 4.5.1 examines this topic starting from the technical backend as well as the administrative configuration UI. The connection and configuration of the single tools, which are globally available or just connected to a single node or to multiple nodes, as well as the configuration and usage of the single data sources, which each tool provides, are part of section 4.5.2.

Furthermore section 4.5.3 shows and describes how the tools and their outputs can be connected to the custom dashboards and the unique UI/UX principles of the SIMARGL toolkit, using the core graphs or creating completely own ones with the VEGA Graph editor, the CANVAS integration or other frameworks and libraries such as d3.js.

Another important aspect, as described in Figure 6 in section 2.2.1, is the use in different levels of investigation, which includes but is not restricted to the detection to anomalies and responding as well as analyzing incidents. In general, this means that issues and more important or complex incidents can be pushed to a higher level for the further investigation. Section 4.5.4 is dedicated to this subject and describes the configuration, look and behaviour allowing different roles and levels of users to work on the same platform. The main UI interface and available dashboards are also highly configurable, as described in 4.5.5, to fit the needs of each single user independently, which allows to improve the workflow by supporting their individual preferences. At least, the adaptive UI and context sensitive behaviour of the SIMARGL toolkit as well as the configurable UI/UX principles are finally carried out in section 4.5.6.

4.5.1 Configuration of the network cluster

The network cluster contains and describes all available nodes of the internal and external observed networks. In this context, a node always describes a functional part or component, which can be either a server, a connector or also just a single container running on a server itself. However, the configuration, as depicted in Figure 35, also allows to group multiple nodes into a sub-network, allowing to organize and present the whole cluster much more efficient and clearer.

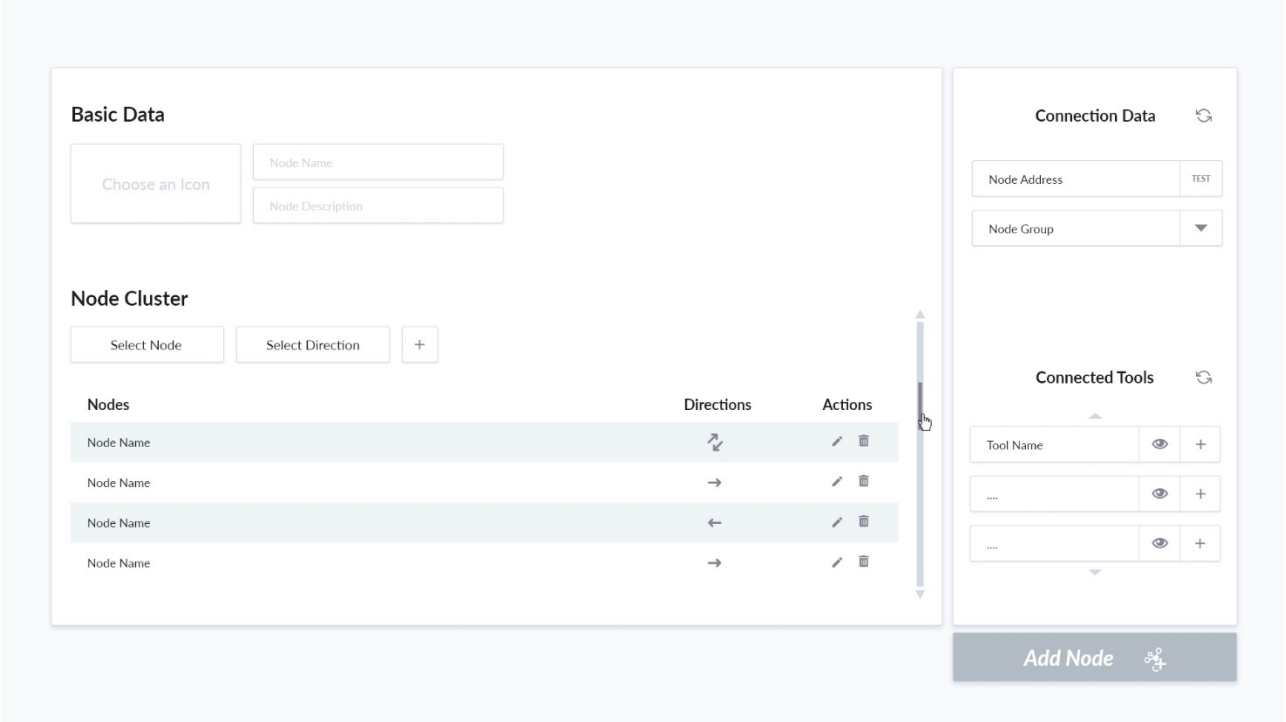


Figure 35: Configuration concept screen for adding a new node

The SIMARGL toolkit has to be configured especially for the network cluster to be used with the toolkit. The simple configuration assistant as shown above allows to add, edit, change and connect single nodes and their assigned tools for building up the monitoring of the whole network cluster depending on this information. Internally the data get stored as plain JSON documents, which allows to easily export and import the network cluster and to create backups.

The creation process guides an administrative user with a few questions, collecting data such as a unique identifier, a descriptive title and a short description for the end-user. The connection details, such as the IPv4 or IPv6 address and other protocols, which are used on the specific node, can be passed as well.

All this data affect the graphical overview and the tabled list view, where the single nodes build the whole network cluster. Both are further described in section 4.2.1. However, this is not only a simplified view about important connections but also a very interactive way of viewing and handling occurred issues and incidents. It helps to keep the important details in one single surface without being inundated with information. However, the possibility to dig deeper into any kind of data structure and issue is still possible and reachable with just a few clicks.

4.5.2 Configuration of the single tools and data sources

Next to the network cluster data, the SIMARGL toolkit also requires the information and connection data of each single tool and their data sources. The basic information is:

1. A unique identifier
2. A descriptive title
3. An optional description for the end-user
4. A list of data sources which the tool provides (including its results) containing:
    a. The name and address of the source
    b. The time interval when the source gets updated

c. The visual representation of the source within the SIMARGL dashboard

These attributes are sufficient to configure and set up a whole tool, including their data sources, as depicted in Figure 36. The information provided by the data sources get automatically processed, parsed and indexed by the internal mechanics, namely the Elasticsearch engine including the Beats family. A more detailed view of this technical background has been described in section 4.4.2.1.

However, using additional information allows the SIMARGL toolkit to handle them more precisely, including an extended visual representation and integration within the UI. This additional data may consist of:

a. A list of nodes where the tool is used
b. A list of services the tool is using from the SIMARGL toolkit
c. A list of custom visualizations and settings for the desired representation

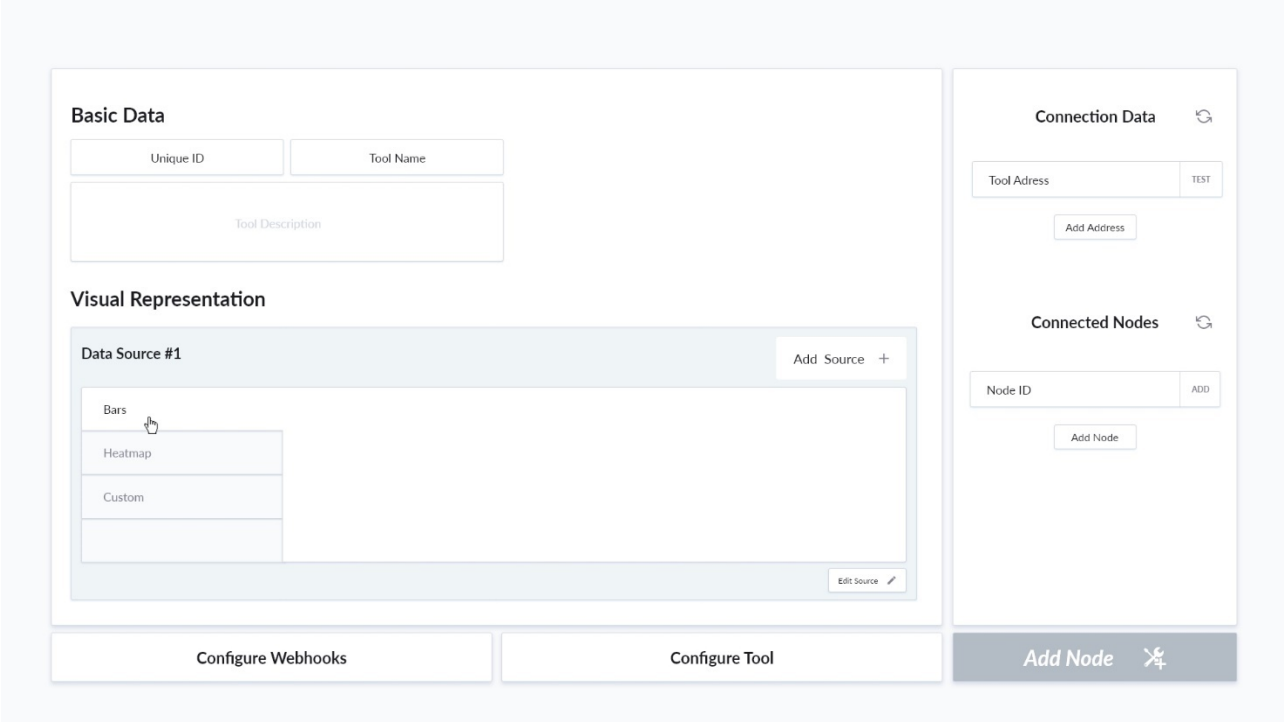


Figure 36: Configuration concept screen for adding a new tool

The first point (a) is used for the graphical network cluster overview, which allows to list all available tools and data sources depending on the selected node. This also allows to directly jump and interact with all kind of issues and potential incidents.

The SIMARGL toolkit also offers a lot of additional services, which can optionally be used by each tool itself (b). One of them shows and observes the health status, performance and downtime of a tool. While the downtime gets automatically managed by SIMARGL, the performance and health status itself must be provided by the server or the respective tool directly. This information can be important for a later investigation.

The whole configuration is also stored within a plain JSON file, allowing to export and import them easily. An automatic generation of the respective configuration file, for example by an additional utility, is also possible, since the sources and data will be read and parsed on each visit, instead of reloading the whole server or restarting the SIMARGL toolkit container.

A more detailed and depending configuration of the tool itself must be performed in each tool, unless the tool provider offers a respective interface for the SIMARGL toolkit to read, apply and change the available settings. However, the configuration UI itself provides an easy way to directly switch to the tool configuration page and furthermore presents a summarised description of each tool, at least if it is provided by the tool provider itself.

To further configure tools and data source, the SIMARGL UI provides three options:

1. Configuration for webhooks
2. Configuration for webhooks
3. Configuration of available data sources

Configuration for webhooks
The SIMARGL configuration UI also allows to set up and configure different kinds and types of webhooks. These settings do not only allow the SIMARGL toolkit to connect and interact with the tool, but also the other way around. The main configuration consists of the webhook URL of the tool, optional authentication headers, the data format (using JSON) as well as a configurable body. Especially the last aspect allows tool developers to configure the requested content to their needs.

As stated out above, the tool can also create a webhook to communicate with the SIMARGL toolkit, for example for requesting datasets or notifications, checking current incidents and issues or externally setting up some configurations or temporary options.

Configuration for switching tools
The SIMARGL UI environment offers the possibility to jump to the tool's web interface as well, either for additional configurations or for a deeper investigation or interaction with a created incident as described in section 4.4.1. This redirection and forwarding process must be configured as well, since each single tool offers an own kind of request handling.

To ensure a flowless interaction between the different consortium tools and the SIMARGL UI environment, the tool creator needs to define which information are required and how they should be forwarded to the respective environment. This configuration can be directly set up and tested on the respective tool pages.

Configuration of available data sources
Each tool needs to provide at least one data source, this can either be some input information, such as the node-based NetFlow, or any kind of results, such as marker on an existing input source. To ensure a clear overview and a flowless combination of the single tools and their data sources, the connection must be configured and managed manually. Additional options, such as a ping address or similar settings can ensure the functionality of the respective source as well, but must be supported by the tool or the respective data source connector. More information about the data source connection between the single tools and the SIMARGL UI environment can be found on section 4.3.1.

4.5.3 Custom graphs and dashboards

The SIMARGL toolkit, as well as the Kibana backend itself, already cover the most common and used graphs and visualizations. However, custom graphs are still an important part, especially when it comes to abstract data sources or the generation of complex graphs. Therefore, the SIMARGL toolkit offers the possibility to define and inject own visualizations directly when providing or adding the tool itself or afterwards when customizing the default or custom dashboards.

Creating a custom graph can either be performed by using the VEGA graph editor or the CANVAS environment, both are functionalities offered by the Kibana framework and are described in section 4.4.2.3. In order to be more flexible, we decided to add a third possibility by integrating a programmable interface based on the d3.js library, which is well documented, feature- and example-rich and established in the development of web interfaces.

The creation of custom dashboards in addition to the graphical overview and the investigation dashboards provided by the SIMARGL toolkit is still possible using Kibana's own abilities. Nevertheless, this requires advanced knowledge of the Kibana environment. It is important to emphasize, that these custom dashboards are not able to directly interact with the SIMARGL toolkit and are therefore more for embedding or providing own pages. However, this feature might still be relevant, when it comes to the utilization of the different tools provided by consortium partners. Instead of switching to another environment or web application, which might be not very convenient for an end user, the tool provider is able to integrate their own interfaces and systems directly into the Kibana environment, allowing a more seamless interaction with the SIMARGL toolkit.

4.5.4 Level and role-based interface

The level and role-based management, as used on the SIMARGL toolkit, refers to the common standards of a Security Operation Centre (SOC), as summarised in Figure 6 in section 2.2.1. This results into different views on the main graphical overview, as well as on the investigation dashboards and views.

The configuration is only allowed to be done by an administrator or another user with similar administrative rights. The configuration is performed by directly using Kibana's core abilities, but the SIMARGL toolkit still offers custom pages to prevent the usage of the integrated command line interface. This allows an administrator to easily create new users and groups, connect them using a button, a dropdown field or a simple drag and drop interaction. Furthermore, a third grouping layer, called "Level", extends the core abilities by another system, which is an important function in the main features of the SIMARGL UI environment.

4.5.5 User preferences

While the main views and dashboards are controlled by the user's role, the user should still be able to change, design and define some custom UI elements and UX behaviours. That is a really important part, since there is no strict nor defined way, how a user interacts with a web application, of course even if there are a few known standards and recommendations as stated out in section 2.1.2.

The configuration does not only contain structure, layout and behaviour settings, but also allows the user to use and customize the used colour scheme. This way, the SIMARGL UI toolkit takes also care for disabled people by offering a high-contrast mode, showing the whole web application in a black and white styled design, using bright colours to signalize interactions and with as few gradations as possible.

Another important point is to change the level of details on the first view. While the user will still be able to zoom into single graphs and the storyboard itself, it can be confusion to see everything directly. This way, the user is able to start with a simple overview, which just contains the most common and important data and information. The investigation possibilities, offered by the SIMARGL toolkit, will help the user to see and view the context and data he really needs to.

4.5.6 Adaptive UI and context sensitive behaviour

Next to the user preferences and their configuration, the user is also able to adapt and change the UI and dashboard views directly. Next to an inline filter and search ability, the visualizations as well as the storyboard can be directly configured.

To also achieve a seamless workflow for professional users, the SIMARGL toolkit supports – next to a right-click interaction to show all possibilities on the respective view or element – configurable keyboard shortcuts and a configurable context-sensitive environment. Allowing to create a custom set of tools, context menus and interaction toolbars will improve the abilities and the speed for all kind of end users.

Smart devices, such as tablets and smartphones, will also benefit from the responsive abilities of the SIMARGL toolkit UI, which includes general and important information within a really simple and basic view. Furthermore, the UI offers the typically and already user-approved swipe and touch gestures and interactions to directly support a native workflow and behaviour. While smartphones and tablets may be able to display and use a minimalistic SIMARGL UI, very small displays such as wearables and smart watches are definitely not capable. Therefore, an additional application may be able to show at least such information as the current node status, alerts, the status of ongoing tasks or incidents or other kind of basic information in a very simplified and minimalistic view. The connection to the SIMARGL environment at least would support applications for such devices.

5. Evaluation

The evaluation process for the SIMARGL UI should be performed with users, that are familiar with cyber security tools and environments such as operators within a SOC or a similar team and should incorporate their typical use cases. Using real input data and results should be preferred, whenever possible, instead of using random generated or maybe even slightly obsolete datasets. Both aspects are considered when performing the UI evaluation with cyber security experts from the SIMARGL consortium partners.

However, the main topics for the evaluation are:

- Usability and user experience
- Featured functions, utilities and visualizations
- Collaboration workflow and performance
- Cognitive workload and behaviour

The evaluation will be conducted in two steps: pre-evaluation and final evaluation. The pre-evaluation is topic of section 5.1 and uses a system skeleton with a first clickable prototype, which mainly works with randomized information. The main goal of this pre-evaluation step is to validate the developed UI concept with typical users and to collect further insights for possible improvements which have not been considered yet. The results ensure that the UI incorporates all relevant aspects and addresses the main requirements before the final implementation phase. The second step, as described in section 5.2, is aimed to evaluate the developed comfortable, performant and interactive user interface, that works seamlessly with real datasets provided by the consortium partners and analysed with the available and developed tools in SIMARGL. It is also aimed to collect suggestions for the further development of the whole SIMARGL toolkit and environment as well.

5.1 Pre-evaluation

The pre-evaluation phase aims to validate the usability of the developed UI concept and to collect valuable feedback for small adaptations and further improvements before the UI components will be completely implemented. The best solution to achieve this goal is to test the UI concept with real operators and cyber security experts from the SIMARGL consortium. Therefore, a clickable UI prototype with a working skeleton environment will be provided, using random generated datasets and including most of the important features and behaviours. The UI prototype will be tested within an agreed framework with the consortium partners in the next months. This process does not only prove and test the UI concept with its solutions but also enhance the developed well-known and familiar features and functionalities. The feedback of cyber security experts and end users as the result of the pre-evaluation is important for completing the implementation of the UI. The implemented UI components will then be evaluated in the final evaluation phase, as described in section 5.2.

5.2 Evaluation

The evaluation of the developed UI components aims at the general usage of the whole SIMARGL toolkit within the UI, using a real environment with typical outputs generated from the different available data sources, the various consortium partner tools and the developed algorithms. The performed tests are aimed to ensure the usability of the system and focuses also on performance, stability and workflow in general. This does not only apply to the JavaScript-written and node.js based UI backend, but also to its completely browser-based frontend environment. Included is the verification of the stability of the UI components across multiple operating systems and browser kits as well. During the evaluation phase the consortium partners, especially the end users, will test the UI and the whole SIMARGL toolkit and provide valuable feedback.

For the evaluation phase, an evaluation plan has been developed. RoEduNet, for example, will use the SIMARGL toolkit to analyse the traffic that flows through one of its clients, managed by the Network Operational Centre (NOC) from Bucharest. The traffic that flows through the network will contain data from office, research, and educational domains. The SIMARGL toolkit will be used to analyse data from the endpoints and firewall alerts.

In the deployment of the SIMARGL toolkit, following UI elements will be tested:

- Login – the user can use the company specific credentials to access the tool.
- Responsiveness – the UI is responsive to user actions.
- Design – the fonts, colours, animations, and transitions between pages/components are appropriate for extended use by security experts. Continuous use of computer applications can cause eye strain or frustration if the application design is overcomplicated or attempts to provide too many pleasing elements to non-regular users (eye candy).
- Appropriate data access – the security experts can access all the information points and tools that are available for them. Additionally, they should not be able to access information that they do not have clearance for.
- Intuitiveness – the displayed data is categorized in an intuitive manner. Additionally, the security experts should easily identify the location of menus and find component configuration straightforward.
- Data aggregation – the application should display aggregated information provided by firewalls, endpoints, and other data sources.
- Detailed information and logs – the information about the analysed traffic should be logged and accessible for further investigation. Filters should be available for the logged data in order to allow advanced analysis of the data. The user should be able to access the logs either going from the data aggregation page or from the main menu.
- Component selection – test if each component from the SIMARGL toolkit can be added, configured, and removed from the system.

The whole evaluation process will ensure that the developed SIMARGL UI with all of its features and functions works as expected in real environments and within real usage scenarios. In addition, it will provide suggestions for further improvements and for the future development of the whole toolkit.

6. Conclusions

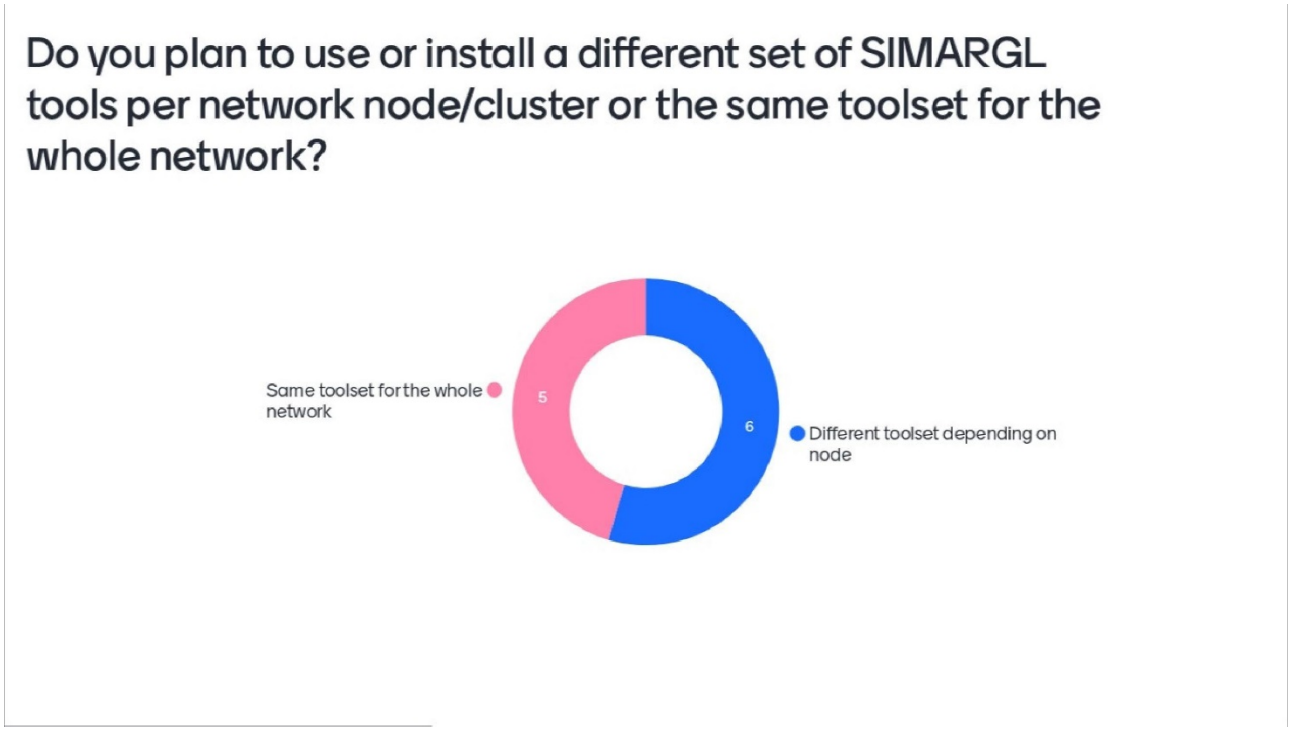
In this document the developed UI/UX concept for the SIMARGL toolkit with its design principles, features and functions are presented. The developed concept bases upon the results of preparatory work identified during a state-of-the-art research, the corporation with cybersecurity experts from the consortium partners along with a SOC team of the German ISP Telekom and the incorporation of related work performed in previous steps of the SIMARGL project, especially on requirements, usage scenarios, architecture and data production.

The developed UI focuses on an optimal assistance for cyber security analysts and provides major views for an overview over the overall state of the observed network including all integrated tools and corresponding notifications and for the investigation of incidents. The UI offers a flexible solution for the connection of different tools from the consortium partners, integrates their results and data outputs and allows a switch to specific tools at appropriate positions. In addition, collaboration between different users is well supported with a lightweight ticket system and a history where all performed steps can be followed and identified aspects and results can be shown directly in the investigation view. To further support the requirements of a SOC with many different users, the UI also allows the definition of specific user roles and rights. In general, the UI component of the SIMARGL toolkit follows a flexible design and is very adaptive via configuration to support various deployment and usage scenarios, especially in the structure of the observed networks and the integrated tools. This flexibility also allows individual user preferences.

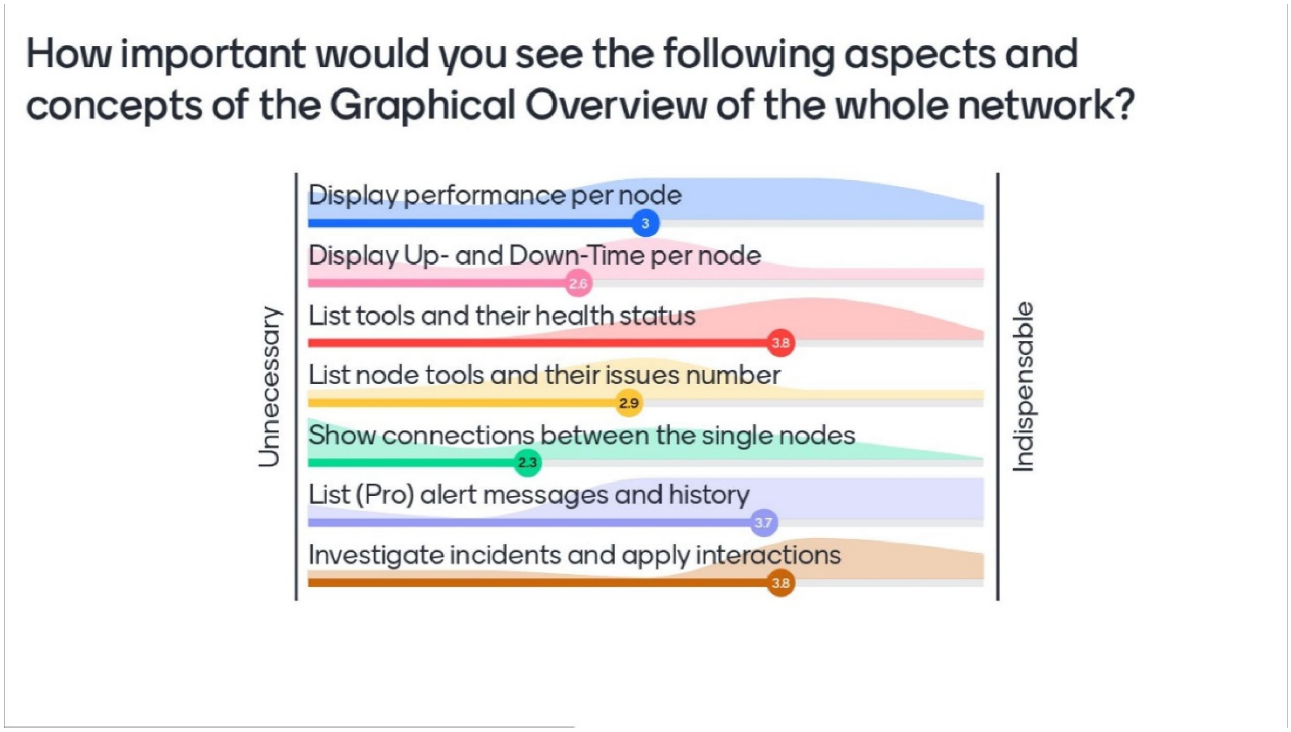
The developed UI concept presented in this document builds the foundation for the further implementation of the UI components for the SIMARGL toolkit in the following months of the project. In order to validate the usability of the UI and to collect valuable feedback for further improvements, an evaluation with end-users is planned at a later stage in the project.

Annex A: Questionnaire for UI concept

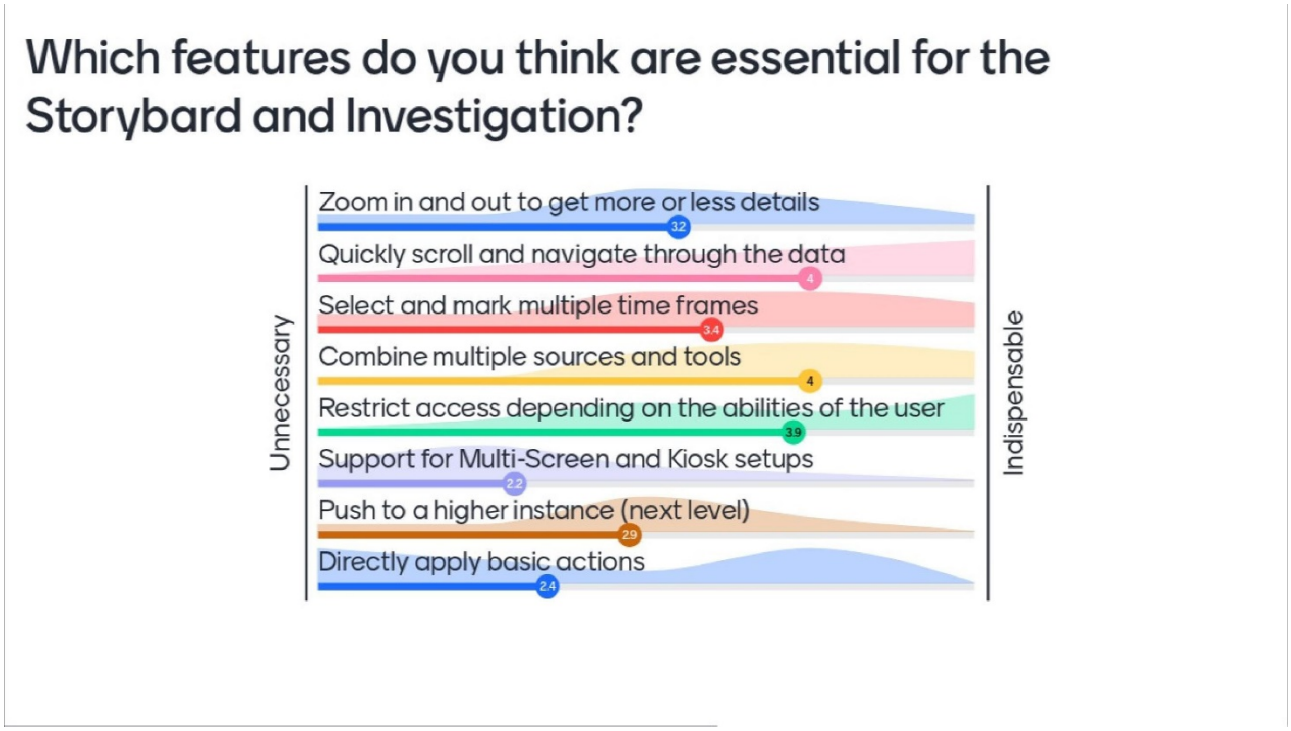
Question 1:



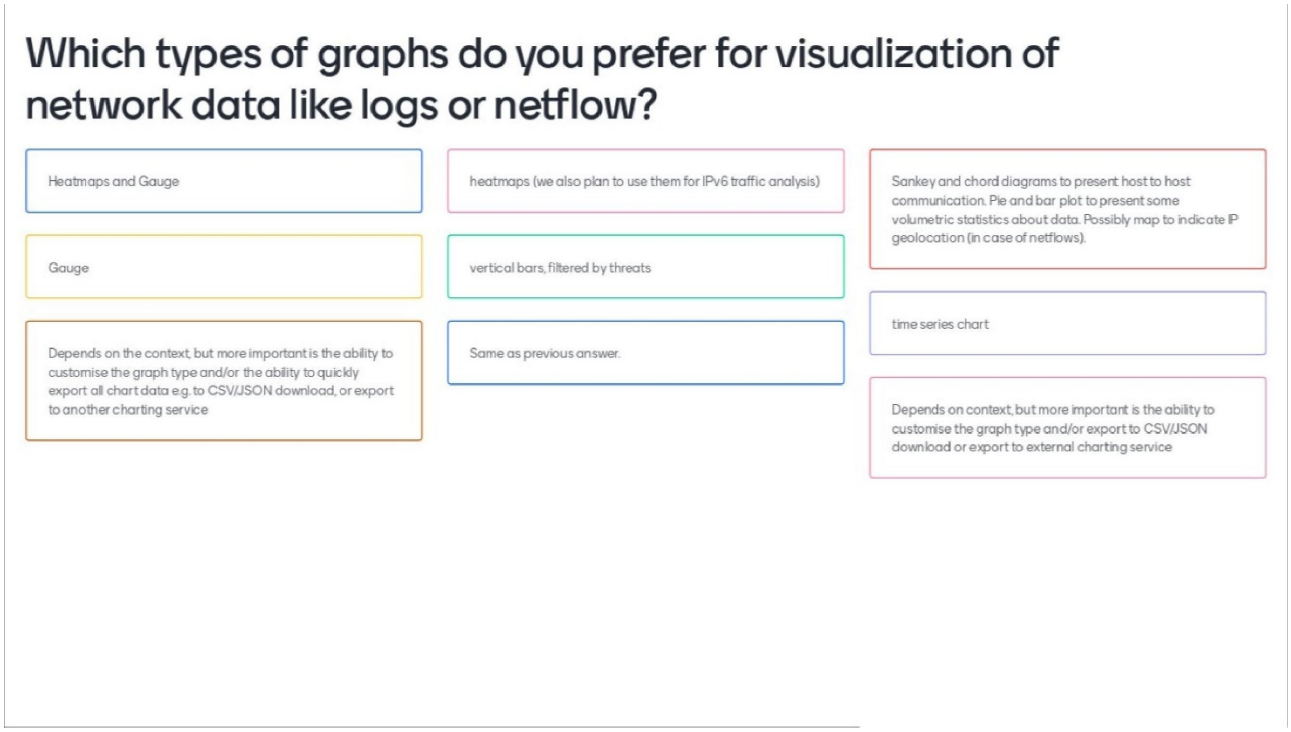
Question 2:



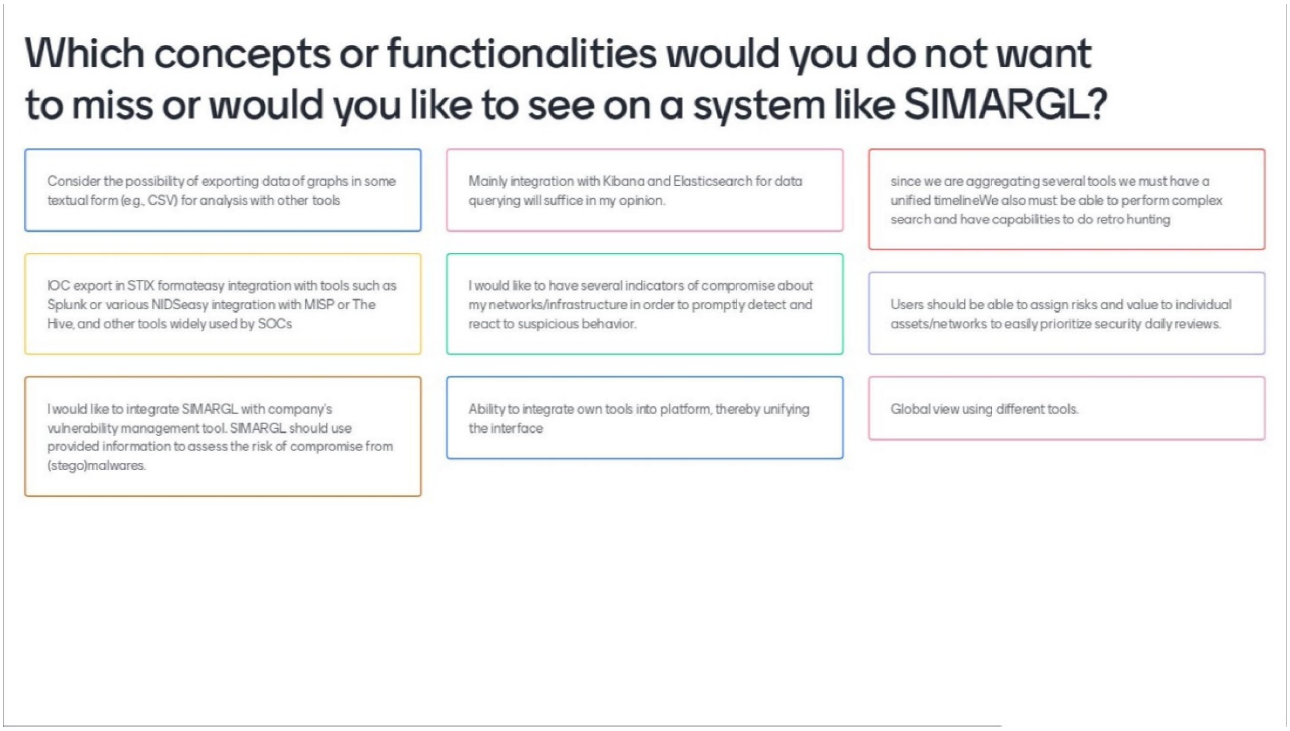
Question 3



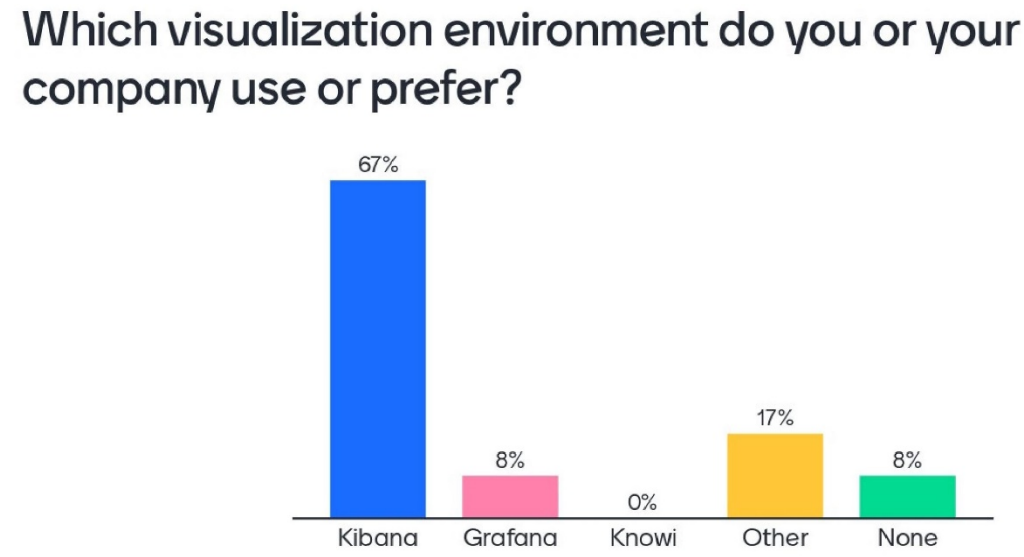
Question 4:



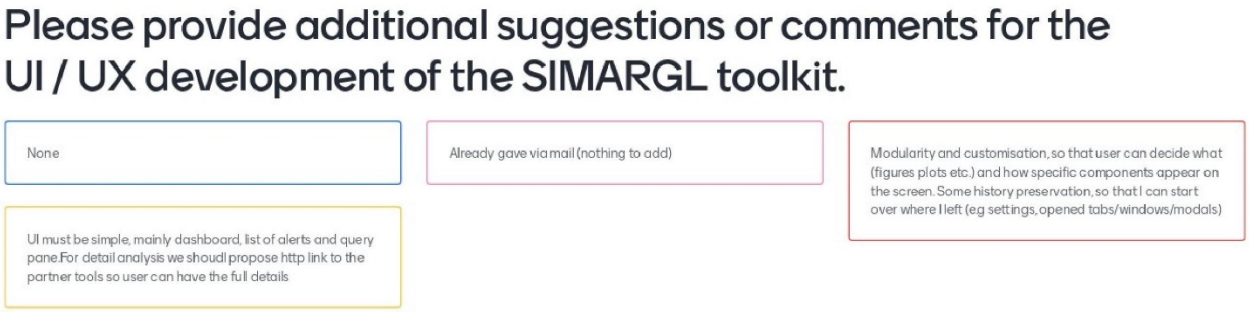
Question 5:



Question 6:



Question 7:



References

[1] Nielsen, J. (1993): *Usability engineering*. Morgan Kaufman, Academic Press, USA.

[2] Krug, S. (2014): *Don't make the think. Revisited. A common-sense approach to Web and mobility users.* 3rd edition. Pearson education, USA.

[3] Krug, S. (2010): *Rocket surgery made easy – Web usability*. Addison-Wesley, USA.

[4] Lieberman, H. and Paternò, F. and Wulf, V. (2006): *End-user development*. 2nd edition. Springer, Germany.

[5] Ash, T. and Page, R. and Ginty, M. (2012): *The definitive guide to testing and tuning for conversions*. John Wiley and Sons, USA.

[6] A/B-Testing (2020): https://www.optimizely.com/de/optimization-glossary/ab-testing/ visited on January 22.

[7] Stern, J. (2010): *Social Media Metrics: How to Measure and Optimize Your Marketing Investment*. Wiley and Sons, USA.

[8] Gonçalves, B. and Ramasco, J. (2008): Human dynamics revealed through Web analytics. Physical Review E 78(2), pp. 26-123.

[9] Brown, F. and Reilly, M. (2009): The Myers-Briggs type indicator and transformational leadership. Journal of Management Development 28(10), pp. 916-932.

[10] Hassenzahl, M. (2006): Hedonic, emotional and experiential perspectives on product quality. In C. Ghaoui (Ed.), Encyclopedia of Human Computer Interaction, pp. 266–272.

[11] Hassenzahl, M. and Tractinsky, N. (2006): User experience – a research agenda. In: Behavior and Information Technology, 25(2), pp. 91–97.

[12] Tractinsky, N. and Hassenzahl, M. (2005): Arguing for Aesthetics in Human-Computer Interaction. In i-com - Zeitschrift für interaktive und kooperative Medien, 3, pp. 66–68.

[13] CUE-Model (2020): http://mecue.de/english/index.html visited on January 22.

[14] Grafana (2020): https://grafana.com/ visited on January 22.

[15] Kibana (2020): https://www.elastic.co/de/products/kibana visited on January 22.

[16] Briggs, K. and Myers, I. (1995): *Gifts Differing: Understanding Personality Type*. Davies-Black Publishing, USA.

[17] Keirsey, D. (1998). *Please understand me II: Temperament, character, intelligence*. Prometheus Nemesis Book Company.

[18] AttrakDiff (2020): http://www.attrakdiff.de/index-en.html visited on January 24.

[19] Thüring, M., and Mahlke, S. (2007): Usability, aesthetics and emotions in human–technology interaction. International journal of psychology, 42(4), 253-264.

[20] Minge, M., and Thüring, M. (2009): Dynamics of User Experience. Judgments of Attractiveness, Usability, and Emotions Over Time. Technical Report 10-2009, Berlin: TU Berlin.

[21] SOC (2020): https://www.telekom.com/en/blog/group/article/one-soc-fits-all-523112 visited on January 27.

[22] Bhatt, S., Manadhata, K., and Zomlot, L. (2014): The operational role of security information and event management systems. *IEEE security & Privacy*, *12*(5), pp. 35-41.

[23] Reed, T., Abbott, G., Anderson, B., Nauer, K., and Forsythe, C. (2014): Simulation of workflow and threat characteristics for cyber security incident response teams. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 58(1), pp. 427-431. Sage CA: Los Angeles, CA: SAGE Publications.

[24] Muschiol, J. (2014): Universal communication model and evaluation with the elderly society. Ph.D. thesis. Universidad católica de Murcia. Spain.

[25] Betke, E., and Kunkel, J. (2017): Real-time I/O-monitoring of HPC applications with SIOX, elasticsearch, Grafana and FUSE. In *International Conference on High Performance Computing*, pp. 174-186. Springer.

[26] Grafana visualization (2020): https://grafana.com/grafana/#visualize visited on January 27.

[27] Gupta, Y. (2015): Kibana essentials. Packt Publishing Ltd.

[28] Bajer, M. (2017): Building an IoT data hub with Elasticsearch, Logstash and Kibana. In *5th IEEE International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, p. 63-68.

[30] Shah, N., Willick, D., and Mago, V. (2018): A framework for social media data analytics using Elasticsearch and Kibana. *Wireless networks*, pp. 1-9.

[31] Kibana visualization (2020): https://www.elastic.co/de/products/kibana/features visited on January 27.

[32] Basic types of data (2020): https://safecomputing.umich.edu/dataguide/?q=all-data visited on January 27.

[33] Data categories (2020): https://security.gatech.edu/DataCategorization visited on January 27.

[34] Network data types (2020): https://techbeacon.com/enterprise-it/5-network-data-types-every-security-team-should-monitor visited on January 27.

[35] Staheli, D., Yu, T., Crouser, R. J., Damodaran, S., Nam, K., O'Gwynn, D., and Harrison, L. (2014): Visualization evaluation for cyber security: Trends and future directions. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, pp. 49-56.

[36] Ma, L. (2006): Cyber security through visualization. In *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation, vol. 60*, pp. 3-7.

[37] Gates, C., and Engle, S. (2013): Reflecting on visualization for cyber security. In *IEEE International Conference on Intelligence and Security Informatics*, pp. 275-277.

[38] Fink, A., North, L., Endert, A., and Rose, S. (2009): Visualizing cyber security: Usable workspaces. In *6th IEEE international workshop on visualization for cyber security* (pp. 45-56).

[39] Tamassia, R., Palazzi, B., and Papamanthou, C. (2008): Graph drawing for security visualization. In *International Symposium on Graph Drawing*, pp. 2-13. Springer, Berlin, Heidelberg.

[40] Nataraj, L., Karthikeyan, S., Jacob, G., and Manjunath, S. (2011): Malware images: visualization and automatic classification. In *Proceedings of the 8th international symposium on visualization for cyber security*, pp. 1-7.

[41] Ferebee, D., Dasgupta, D., Schmidt, M., and Wu, Q. (2011): Security visualization: Cyber security storm map and event correlation. In *2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, pp. 171-178.

[42] Englert, R., and Glass, G. (2006): An architecture for multimodal mobile applications. In *20th Symp. on Human Factors in Telecommunication (HFT 2006), Sophia Antipolis, France, ETSI.*

[43] Englert, R., and Gesche, J. (2007): Design and usability for personalized user interfaces of telecommunication services. Guidelines for a Decision Support Method Adapted to NPD Processes. Int. conference on engineering design, ICED'07, Cite des sciences et de l'industrie, Paris, France.

[44] McKenna, S., Staheli, D., and Meyer, M. (2015): Unlocking user-centered design methods for building cyber security visualizations. In *IEEE Symposium on Visualization for Cyber Security (VizSec)*, pp. 1-8.

[45] McKenna, S., Mazur, D., Agutter, J., and Meyer, M. (2014): Design activity framework for visualization design. *IEEE Transactions on Visualization and Computer Graphics*, *20*(12), pp. 2191-2200.

[46] Jepsen, K., Glass, G., and Englert, R. (2009): When 'one fits all' does not fit–study of visualization types for mobile help systems. People and Computers XXIII Celebrating People and Technology, HCI, British Computer Society, pp. 398-404.

[47] Mandiant (2020): http://www.mandiant.com visited on January 29.

[48] Fink, G. (2006): Visual Correlation of Network Traffic and Host Processes for Computer Security. (Ph.D. dissertation, Virginia Polytechnic Institute and State University), pp. 105-112.