predict
prioritise
prevent

TRESPASS

Technology-supported Risk Estimation by Predictive Assessment of Socio-technical Security

Deliverable D4.3.2

Visualisation to simplify complex information

Project:	TRESPASS
Project Number:	ICT-318003
Deliverable:	D4.3.2
Title:	Visualisation to simplify complex information
Version:	1.0
Confidentiality:	Public
Editor:	Axel Tanner
Cont. Authors:	A. Tanner, L. Coles-Kemp, P. Hall, J. Barendse, O. Gadyatskaya
Date:	2016-10-31

SEVENTH FRAMEWORK PROGRAMME

Part of the Seventh Framework Programme
Funded by the EC-DG CONNECT

Members of the TRESPASS Consortium

1.	University of Twente	UT	The Netherlands
2.	Technical University of Denmark	DTU	Denmark
3.	Cybernetica	CYB	Estonia
4.	GMV Portugal	GMVP	Portugal
5.	GMV Spain	GMVS	Spain
6.	Royal Holloway University of London	RHUL	United Kingdom
7.	itrust consulting	ITR	Luxembourg
8.	Goethe University Frankfurt	GUF	Germany
9.	IBM Research	IBM	Switzerland
10.	Delft University of Technology	TUD	The Netherlands
11.	Hamburg University of Technology	TUHH	Germany
12.	University of Luxembourg	UL	Luxembourg
13.	Aalborg University	AAU	Denmark
14.	Consult Hyperion	CHYP	United Kingdom
15.	BizzDesign	BD	The Netherlands
16.	Deloitte	DELO	The Netherlands
17.	Lust	LUST	The Netherlands

Document History

Authors		
IBM	Axel Tanner	all
RHUL	Lizzie Coles-Kemp, Peter Hall	all
LUST	Jeroen Barendse	1, 2
UL	Olga Gadyatskaya	2

Quality assurance		
Role	Name	Date
Editor	Axel Tanner	2016-09-30
Reviewer	Fola Ogunsola	2016-09-30
Reviewer	Sven Übelacker	2016-09-30
Task leader	Jeroen Barendse	2016-09-30
WP leader	Lizzie Coles-Kemp	2016-09-30
Coordinator	Pieter Hartel	2016-10-15

Circulation	
Recipient	Date of submission
Project Partners	2016-09-30
European Commission	2016-10-31

Acknowledgement: *The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 318003 (TRE$_S$PASS). This publication reflects only the authors' views and the Union is not liable for any use that may be made of the information contained herein.*

Contents

List of Figures

Management Summary

The description of this deliverable as set out in the Description of Work is: '"Visualisations to simplify complex information: This deliverable is a refinement of the internal deliverable I4.3.1 based on experiences with the prototype in visualising the case studies, especially the one on cloud infrastructures. The visualisation now will target more advanced properties of risk."

The task of developing visualisation to simplify complex information is one that is undertaken by all the deliverables in this work package. This deliverable looks on a specific aspect of advanced properties of risk found in the cloud administration scenario and in so doing demonstrates the value of the general approach to responding to complexity described in more detail in The TRE$_S$PASS Project, D4.2.2 (2016).

Through the work of WP4, the TRE$_S$PASS project has responded to the challenges of visualisation and complexity in its visualisation strategy. Deliverable 4.2.2 outlines the visualisation principles that have been incorporated into the Attack Navigator Map (ANM) design to treat aspects of complexity. These visualisation principles to respond to complexity fall into three categories:

- The deployment of HCI tools to reduce visual complexity.
- The provision of tools to encourage effective ways of seeing complex visualisations.
- The narrative that underpins each visualisation, conceptualised into three spheres (social, technological and physical) to reduce conceptual complexity.

The above three categories of visualisation principles provide a general framework for responding to issues of complexity. The contribution of this deliverable, however, responds to a particular complexity problem of showing changes to risk parameters over time. This is an advanced aspect of information security risk in the cyber realm, not addressed in the general complexity responses found in the visualisation approaches contained within the ANM. During our early work in cloud administration, we found that there is still a lack of visual capability for making changes over time understandable to administrators in the current state of the art.

In this deliverable, we consider the visualisation of changes to be an advanced aspect of information security risk assessment for cloud administration because cloud systems, be it private or public, are highly attractive targets for intruders and there is a high risk for attacks that might occur in smaller steps over a long period of time (e.g., in *Advanced Persistent Threats*). Making changes of a cloud system more easily understandable by visualisation therefore is a means to handle such advanced risks. For this reason we

have put our efforts into prototypes looking especially into visualisations to make system changes understandable.

The specific contributions in this area of visualisation are:

- Visualising time as an integrated dimension by presenting a direct spatial dimension of the visualisation. We show this technique in operation in terms of the spatial dimension in the Cloud case study by showing the location of virtual machines on physical hosts over time, enabling cloud administrators to identify smaller steps over time in an attack build-up while keeping an overall overview.
- Visualising the intersection in access control between cloud administrators/users and cloud infrastructure for a specific moment in time using the technique of the time slider to allow the ANM user to move through time and observe/summarise changes over the selected time interval, enabling the cloud administrator to identify where vulnerabilities in access control may have been inserted.

Key takeaways:

- Characteristics of complexity are taken from the literature on systems: high-dimensionality, non-linearity, sensitivity to initial conditions, adaptivity and emergent behaviour. The state of the art in visualising such systems is to use networks to represent system dynamics, wherein nodes are seen as states and links as transitions. Cloud environments present the added challenge of a large number of elements that change over time.
- Visualising environments and actors that change over time presents a gap in the state of the art in risk visualisation. In the current state of the art the focus of information security risk visualisation is the tendency toward presentation of static pictures of risk.
- In responding to this gap in the state of the art, WP4 has developed the following techniques: capability to visualise risk asset location change over time and a manouevrable time slider for visualising access control change over time.
- We apply these techniques to address the gap in the state of the art in the specific case of cloud risk visualisations.

1 Introduction

The term "complexity" in the context of TRE$_S$PASS refers to a family of visualisation problems. These problems are:

- Visualising elements from very different domains and showing how they interact,
- Visualising a large number of elements,
- Visualising actors with many and varied attributes, and
- Visualising change to the risk variables over time.

WP4 has responded to these complexity problems by producing the following responses:

- multiple "spheres": the explicit goal of TRE$_S$PASS to include social and technical aspects leads to the requirement to visualise elements from very different domains including their interactions (The TRE$_S$PASS Project, D4.2.2, 2016).
- high number of involved entities: often larger scale systems of interest, (e.g., a cloud environment) contain a very large number of entities, so that it is impossible to represent all individual entities explicitly (The TRE$_S$PASS Project, D4.2.2, 2016).
- complex individual entities: including the actors in the socio-technical model leads to the requirement to describe, to some extent of relevance, the actors (or roles) including their motivations, cultural norms etc (The TRE$_S$PASS Project, D4.3.3, 2016).
- importance of the time dimension: scenarios and systems to be described are hardly ever static, but change over time, requiring a way to describe and visualise changes. It is this response that is the focus of this deliverable.

As an overall approach, the TRE$_S$PASS platform provides a means of visualising complexity and achieving stability in a complex system through a combination of automated and analogue tools. While its automated tools for gathering and representing technical data facilitate analysis of a dynamic socio-technical system, analogue methods developed in TRE$_S$PASS, such as paper prototyping and LEGO, offer a means of situated and participatory mind-mapping to supplement the automated analysis provided by the TRE$_S$PASS platform. However, in addition to these general responses, this deliverable examines an advanced aspect of information security risk that increasingly occurs in the cyber realm. The time dimension is a specific lens through which to understand security vulnerabilities arising from configuration and system changes over time. In an environment such as the cloud where administration is dependent on the use of technology to explore, understand and protect the cloud, the use of technology to understand changes over time and the resultant changes in the risk landscape is essential.

1.1 Goals

The goals of this deliverable are:

- The development of visualisations to simplify a specific aspect complex information
- The application of visualisations to simplify a particular aspect complex information that applies to the cloud case study

1.2 Choices made

The selection of techniques to develop visualisations to simplify complex information as a general response within the ANM is based on standard user interaction techniques. These techniques are placed within a hermeneutic framework for visual evaluation.

The development of the prototypes addressing the vulnerabilities introduced by changes to the cloud environment shown here was influenced by our early work with cloud visualisations. We developed the specific approaches through discussions with cloud administrators and compliance officers.

1.3 Foreground and background

The following elements are foreground IP:

- the *Time-Containment Visualiser* (TiCoVis)
- the *Cloud Environment & Actor Visualiser* (CEAV)

SAVE is background IP from IBM.

Other libraries and frameworks that are distributed under open source or creative commons licenses are background to their respective developers and include D3[1], jQuery[2], marked[3], highlightjs[4], CherryPy[5] and Dulwich[6].

[1] see Data-Driven Documents http://d3js.org.
[2] see jQuery http://jquery.com
[3] see marked https://github.com/chjj/marked
[4] Hightlightjs https://highlightjs.org
[5] CherryPy http://cherrypy.org
[6] Dulwich https://www.dulwich.io

1.4 Document structure

Chapter 2 defines background and understanding of complexity in this context, Chapter 3 introduces means to handle complexity in visualisation. Chapter 4 describes the prototypes developed during the project (with instruction for accessing the prototype in Section 1.6), highlighting the measures to mitigate complexity for the cloud use case. Results of evaluations of these prototypes are given in Chapter 5. The deliverable closes with concluding remarks in Chapter 6.

1.5 Complexity in the Attack Navigator Map

The Attack Navigator Map (ANM) manages complexity in the form of a large array of elements by deploying standard interaction techniques and innovations in attack tree representations. In their traditional form, attack trees present a wide variety of relevant information, but are commonly shown as an arrangement of text in a directed graph. In usability terms, attack trees are problematic because the tree structure rapidly becomes very wide, with elements repeated to the extent that they become effectively unreadable. We have responded to this complexity problem by re-imagining the way the tree is laid out and labelled, arriving at more compact trees, and enabling the user to zoom, pan and collapse sub-trees at any level. Please refer to The TRE$_S$PASS Project, D4.2.2 (2016) for further details.

A summary of innovations in attack tree visualisations is as follows:

Attack tree linearisation It is widely agreed that visualisations with more simple elements are more readable than visualisations with fewer, complex elements (Tufte, 1990). Accordingly in TRE$_S$PASS we have turned trees into linear sequences of their required children: This results in more paths, but each path is easier to follow. The simplification or linearisation process is achieved by developing an algorithm that finds all conjunctive intermediate nodes and replacing them with a linearised form of its children. The siblings thus become sub-trees rather than individual nodes, and each individual path is extracted from the transformed tree.

Attack clouds As explained in The TRE$_S$PASS Project, D4.2.2 (2016), in an attempt to provide a better overview for very large attack trees (1.000—500.000 nodes) we developed what we refer to as the attack cloud. An attack cloud aims to represent all the steps possible in an attack tree while still understanding the full context. Steps that are a higher potential threat are closer to the root node at the centre, which creates a logical hierarchy of information. By removing duplicates, this approach could potentially also allow us to view entire attack trees as a threat landscape.

Stacking The ANM deploys a stacking technique when an element has many parameters and the same parameters are not always applicable to the instantiation of each element. For example, the number of parameters in attacker profiles changes depending on the situation. Intel provides a useful set of baseline attacker profiles in (Rosenquist, 2009). But there are cases where perhaps some parameters may not matter, so to allow for this, a unified legend, where thickness and colour represent threat level, makes it possible to represent an attacker profile as a set of stacked circles, in which each parameter is one of the circles. This technique allows extensibility if say, later on, a situation calls for an additional parameter by providing the ability to stack an additional circle. Parameters closer to the outside of a circle are weighted as visually more important.

1.6 How to access the prototypes

Prototypes are at the core of the three final deliverables from WP4, for a full list please see The TRESPASS Project, D4.2.2 (2016).

There are several ways to access the complexity prototypes discussed in this deliverable:

1. Via the visualisation prototypes, tools and methods showcase:
   https://visualisation.trespass-project.eu (no log-in required).
   This showcase includes related publications, methods for visualising security risks and information about the complexity prototypes:

   - Time-Containment Visualiser (TiCoVis):
     https://visualisation.trespass-project.eu/?p=55
   - Complexity prototype Cloud Environment Actor Visualiser (CEAV):
     https://visualisation.trespass-project.eu/?p=216

2. Via the TRESPASS portal at https://trespass.itrust.lu:
   First time log-in: Click on Sign-up, you will receive a confirmation email, you need to click on it to acknowledge the registration. The itrust ICT administrator will have to personally validate your account. Once you receive the validation email, you will be able to access all resources with the same credentials, especially the complexity prototypes:

   - Time-Containment Visualiser (TiCoVis): direct access via
     https://trespass.itrust.lu/tkb/tkb/TiCoVis
   - Cloud Environment Actor Visualiser (CEAV): direct access via
     https://trespass.itrust.lu/tkb/tkb/CEAV

2 System complexity as a visualisation problem

The overall goal of TRESPASS is to understand socio-technical systems in order to find risks, as well as to identify suitable changes to the system to lower these risks. In the context of systems thinking, although complexity bears to some extent an intuitive meaning, there is no single agreed definition, rather different areas of research highlight different characteristics making a system *complex* rather than merely complicated (Mitchell, 2009). In general, there is agreement that a complex system consists of many parts that interact in a way that the system cannot be understood with a purely reductionist approach ('the whole is more than the sum of its parts').

2.1 Characterising complex systems

Typically, characteristics such as high-dimensionality, non-linearity, sensitivity to initial conditions, adaptivity and emergent behaviour are associated with complex systems.

Weaver (Weaver, 1948) differentiates between *simple systems*, *disorganised* and *organised complexity*. *Simple systems* contain few parts and are governed by few variables (i.e., few-body-problems with gravitational interaction in physics). *Systems of disorganised complexity* contain a very high number of interacting parts, but allow the description of the overall system with statistical properties, e.g., a container filled with gas: although in this case the detailed dynamics of a single molecule is intractable due to the enormous number of particles and the strong dependence on initial conditions, macroscopic descriptions (like the definition of a temperature) can be used to successfully predict the overall behaviour. *Systems of organised complexity* deal with phenomena that escape descriptions with statistical properties (as for systems of disorganised complexity) and confront "dealing simultaneously with a sizeable number of factors which are interrelated into an organic whole" (Weaver, 1948, p. 539). As new behaviour can emerge even from very simple rules and interactions (see, e.g., Wolfram, 2002), it may be hard to understand such a system fully even in cases where all parts and interactions are completely understood. This means that systems of organised complexity can neither be adequately described with closed form solutions, nor with statistical approaches.

2.2 Socio-technical systems as Complex Adaptive Systems (CAS)

In cases where the parts or agents involved in a complex system are able to adapt or learn, the term Complex Adaptive System (CAS) is used to emphasise this additional aspect (Holland, 2006). Adaptive interaction between technology, society and human behaviour gives rise to intricate patterns of information flows and patterns of sharing and protection practices. The TRE$_S$PASS attack surface is composed of systems that fall into different socio-technical layers which means that adaptive everyday practice linking the different systems has to be understood as part of the risk assessment process in addition to the systems themselves. Consequently, the visualisation processes not only need to focus on the individual systems, but must also be capable of visualising the links between those systems thus giving rise to visual complexity.

One approach to visualising complex systems is to use networks to represent system dynamics, wherein nodes are seen as states and links as transitions (Gershenson & Niazi, 2013). Cloud environments present the added challenge of a large number of elements that change over time. Social factors in risk present both adaptive and difficult-to-predict challenges for visualisation. Even in cases where the technical environment might be merely complicated, the combination of socio-technical factors will lead to a system of organised complexity, as it introduces autonomous agents with hard-to-capture motivations interacting with the environment. This also makes the system evolving, as the system adapts over time and actors learn from experiences and changes. Visualisation within TRE$_S$PASS therefore needs to be able to visualise the possible behaviours of each sphere and the interaction between the spheres.

Due to the high number of interacting parts involved, it is usually not possible to gain a complete overview of every part of the system. Visualisation therefore has a very important role in this context to allow flexible representations of the system so that human experts can gain insights and understanding of its behaviour. This includes visualising proposed changes of the system (e.g., new policies) together with simulation of the changed system (as far as possible) to see whether the changes have the desired effects after the system has adapted. Changing complex adaptive systems can often lead to completely unexpected responses when the system adapts. As human cognition is limited in the amount of detail it can encompass and process, visualisations must be flexible enough to enable explorative investigations of the system, and furthermore, to refresh an overview based on the behaviour of the systems from point of view of different actors and contrasting interests.

The challenge is, as the quotation in Figure 2.1 emphasises, that visualisations must communicate something of value to the user's task-at-hand and must also be situated within the relevant contextual information, while stating the relevant requirements for success. However, any non-trivial system that attempts to show *all* elements and relationships quickly becomes an impossibly overloaded representation for the user. In order to reduce apparent visual complexity and to enable users to focus on the relevant information, a number of visualisation strategies need to be considered.

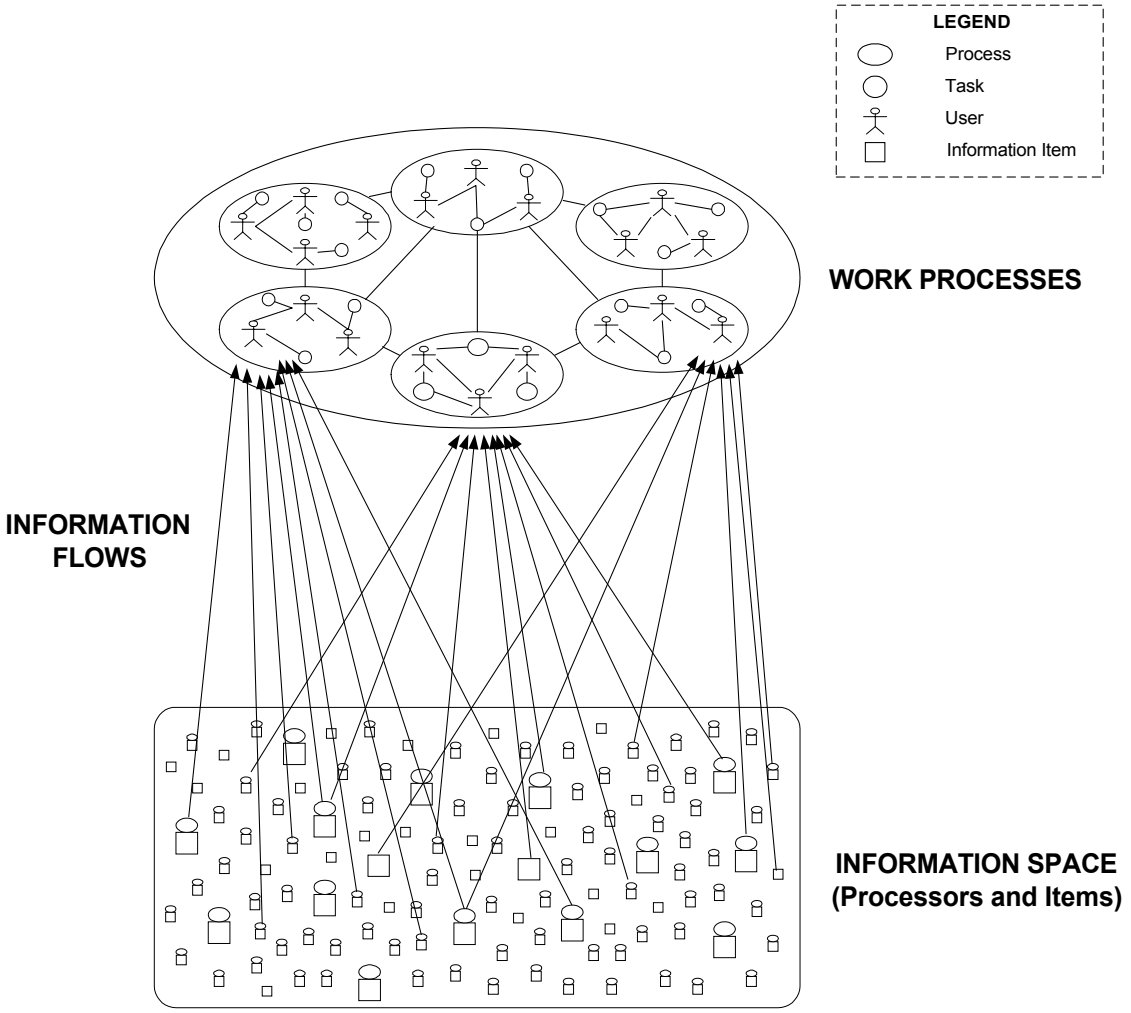

Figure 2.1: *Information space*: a generalised context for the system description problem: "The ellipses seen in the top of the figure represent a set of work processes relating to some generic system. Examples of such processes might be standard lifecycle processes such as acquisition, analysis, synthesis, configuration management, quality assurance, and evolution. Each of these processes requires people to perform a variety of tasks (indicated by circles). In order to perform these tasks, the people rely on information gathered about the system. Each individual user may have different information needs resulting in complex information flows from the information space." Reproduced from (Goodburn et al., 1999, p. 5).

2.3 Visualisation challenges: summary

These features make traditional risk assessment methods difficult to apply and require methods that bring together the social, technical, physical and organisation dimensions of risk in order to better understand the initial conditions, the emergent behaviour (both

technical and social) and the underlying environmental influences on adaptivity. Norman and Stappers (Norman & Stappers, 2015) note that "The non-independence of elements combined with non-linear causal relations and feedback reveals yet another component of these sociotechnical systems: the inter-relationships among the components can be more important than the components."

Consequently, the visualisation challenges for TRE$_S$PASS are:

- Identify the visualisation principles that enable the social, technical, physical and organisational dimensions of risk to be visualised both as an integrated whole and within their individual dimensions;
- Develop visualisation techniques based on the visualisation principles identified; and
- Develop general and specific techniques for visualising the inter-relationships between the social, technical, physical and organisational components.

One way of understanding such systems is to build computer simulations to model their emergent behaviours. This means that these approaches to understanding are fundamentally more explorative and have a 'game theoretic' aspect, i.e., need to take the reactions and interactions between different actors ('players') into account.

3 General principles

WP4's approach to responding to the challenges of visualisation complexity is based on the theory that visualisation and visual notation tools are needed that support varied, flexible, adaptive and situated responses:

- *Ashby's Law of Requisite Variety* states that, paradoxically, complexity is needed in order to tackle the challenges posed by complexity: "variety absorbs variety, defines the minimum number of states necessary for a controller to control a system of a given number of states." (Ashby et al., 1956).
- Complex systems have a high number of variables and a high level of interdependence, such that linear models do not account for transitions between states (Flach, 2012). To draw from classical organisational theory (Thompson, 1967), complex systems are characterised by *reciprocal interdependence*, where the functioning of some tasks will change depending on how other tasks are performed. (For example, the level of compliance with policy in an organisation will depend on how policy is communicated.) Whereas closed and simple linear and sequential systems can, in theory, be hierarchically controlled through goal-setting, monitoring and correcting, a complex system is more typically controlled by finding a stable balance among competing interests, achieved by mutual adjustment. As complexity increases, Ashby's Law suggests that the demand increases for distributed, flexible forms of control like this process of mutual adjustment (Flach 2011). Mutual adjustment means that all system elements adjust in response to changes in context. Ideally, security visualisation tools should support this distributed form of control.
- Despite the fact that a visualisation of a complex security problem usually represents the movement and distribution of information at a certain time and place, visualisations typically represent that activity in an abstracted or disembodied state. The disembodied, snapshot view tends to neglect the situatedness of a complex problem and tends to omit the fact that any activity represented in a visualisation will be in flux, subject to change. An analogy can be made with navigation, which, while it may be guided by a map, is never an entirely cognitive process, but is also embodied. We find our way "on the fly" (see Tim Ingold's account of navigation understood through psychologist James Gibson's argument that perception is the achievement not of a "mind in a body, but of the organism as a whole in its environment" (Ingold, 2000)).
- Another distinctive feature of complex and emergent systems such as socio-technical systems is the presence of feedback loops. As Norman and Stappers note, "Feedback changes the behavior of the system, making it impossible to understand the

whole through understanding each of its parts." (Norman & Stappers, 2015). This means that the system needs to be understood holistically rather than from an individual component perspective. Given Ashby's Law, security visualisation tools need to support feedback loops.

- "Physics of Notation" (Moody, 2009): Complexity management is one of the nine principles identified by Moody that are relevant for a good (i.e., cognitively effective) visual notation. Although created in the context of software engineering, this highlights the importance of visual notation features to explicitly cope with the complexity of the described system.

3.1 TRESPASS visualisation philosophy and complexity

The stated aim of reducing complexity for the user, is concerned with providing the user with various approaches to visualising sub-parts of the broader information set. This gives a user an improved and more concise understanding than a comparable case where the tools attempt to visualise all of the given complex data set at once. In terms of the general TRESPASS interface, the individual and targeted design of each of the tools is such that they perform relatively simple tasks by themselves, but together form a powerful platform that is able to collectively manage these complex data-sets as a whole. There remains scope to summarise the findings of groups of these tools, but this is intended to be qualified by the detailed insights or the conceptual model that emerges for the user through the use of the smaller tools.

This is a familiar route for the design of interfaces that deal with large data-sets. The pathway in these cases could be described as following the 'encapsulation' of hierarchically layered abstractions, where the user is able to access deeper levels of detail wherever and whenever this is needed. This is especially the case in the visualisation of different depths of analysis, sometimes referred to as 'drilling down' into the data, as long as it makes sense in view of the integrated data set. Abstraction relates to an analytical process where ideas gradually become distanced from their objects to different degrees. But equally, representations should not over-generalise and thus omit significant detail. The example of the attack tree simplification illustrates these points. The entire attack tree must be collated and visualised in order to be able to calculate upon all of the branches and their intermediate nodes. However a viewer is not helped by having access to all of this information at once. Instead, a user requires actionable and comprehensibly presented information, at differentiated levels.

As the growing public interest in journalistic scientific visualisation will testify, images capture a 'richness of relations in a way that a logical train of propositions never can' (Galison, 2006). Studies as diverse as Poincarre's provisional pictures, Minkowski's number theory and Margaret Geller's research into the clustering of galaxies substantiate the argument that the image often provides a means of understanding complexity that is inherently intuitive. Yet, as Peter Galison has noted in an account of visualisation's role in the key

breakthroughs in the sciences, images also deceive. 'Pictures create artifactual expectations, they incline us to reason on false premises' (Galison, 2006). Visualisations must always depict a selection of all available evidence, and as such, are abstractions of complex socio-technical environments. It is therefore important that the question of what is being *silenced* in a particular abstraction is a fundamental consideration in visualisation research.

Abstraction In general, but also especially in the context of TRESPASS, abstraction is important in relation to the goal of mitigating what can be an overriding complexity in some scenarios, that may confront the user of TRESPASS tools. It is also worth stating that risk assessment itself is a form of abstraction that enables a security practitioner to order contextual data and spotlight particular facets of the context before analysing it.

The ways in which abstraction can be used are:

- Initially in relation to formal modelling, for example, abstraction is needed to organise a view of the relevant infrastructure items, as well as to present a view of all of the actors and their behaviours. Simplifying these for the purposes of visualisation also tackles the more subtle features of these behaviours during iterative stages of modelling; secondly,
- In the visualisations abstraction can be used to mitigate complexity and information densities in a constrained visual space.

3.2 Using standard interaction techniques to respond to complexity within the ANM

The abstraction techniques are manifested in standard interaction techniques and these techniques have been deployed in the ANM in order to reduce the visual complexity of a particular aspect of a complex risk scenario.

Possible approaches to tackle the visualisation of complex systems include:

Filtering/highlighting/sorting filtering and/or highlighting and focusing can be used to select a subset of elements to reduce visual clutter; similarly, sorting of elements enables the focus to be confined to a subset, utilising a metric for the purposes of ranking

Exploiting visual form and representative functions utilise visual form and well-known representative functions to allow quick and high-level recognition, e.g., the hover function to foreground virtual machines involved in a specific flow of information (see Figure 4.4)

Using abstractions use abstractions in the set of elements to allow grouping 'similar' elements and combine into fewer elements in order to visualise effectively

Overview and drill-down give an overview of the total system, possibly starting with higher-level abstractions of subsystems, while allowing drill-down into individual subsystems to show more detail. This approach is explored for example with the "alluvial" view of relations between physical and virtual servers in TiCoVis (see Figure 4.3)

Multiple views show multiple views of the system from different viewpoints or 'gazes' to highlight different aspects of the system at the same time in a *coordinated-visualisation* (North & Shneiderman, 2000)

Some of these approaches can be combined for additional benefits, e.g., multiple views of the system are especially helpful when selections in one view are coordinated with all other visible views to see the selected entities in the different contexts and perspectives.

In the remainder of this deliverable we report on two prototypes that deploy these techniques to visualise two particular aspects of advanced information security risk assessment in the cyber realm.

4 Visualising complexity in the Cloud scenario

The *Cloud use case* in TRE$_S$PASS task T7.2 enables us to use our general approach to visualising complexity and to specifically develop some new tools.

The cloud represents the three spheres that we use within TRE$_S$PASS visualisations, namely social, technological and physical:

- a physical setting with rooms, doors and windows where, e.g., physical infrastructure pieces of a cloud environment are situated and where the different actors have access and can move,
- *software-defined* virtual parts, like virtual machines, virtual network and storage, situated in an abstract and completely separate space, and
- the *social* space where a distance between actors defines weak or strong relationships.

At the same time, the physical elements (e.g., servers, network) can range in the tens of thousands, the virtual, software-defined components (e.g., virtual machines) can range in hundreds of thousands. In addition, there is typically a very large number of users of the cloud infrastructure and rather few, but very powerful, administrators.

All these elements will interact (cooperating or interacting maliciously) leading to a complex behaviour over time, which leads in effect to a *Complex Adaptive System*—compare the discussion in The TRE$_S$PASS Project, D4.2.1 (2014).

In the following, we show our work to represent a cloud environment as an example for a complex environment in a visually understandable way. Section 4.1 makes the start by showing work earlier during the project depicting a live cloud environment in real-time as a general graph. During this work we found that there is still a lack of visualisation making changes over time understandable in the current state of the art. Current software for managing cloud environments, like VMware vCenter or the Horizon dashboard of OpenStack, focus on the current state of the infrastructure and the ways to configure and manage the system. Corresponding monitoring tools, although showing the time aspect, are focused on technical details like memory or CPU utilisation, rather than changes of the structure or access control role. These changes are hardly visualised at all and mainly contained in text-based log files.

Identifying changes in a complex and highly dynamic system is difficult but is a necessary aspect of cloud risk assessment. Missing relevant changes may lead to failure identifying violations of required policies, or missing steps indicating an intrusion (from the outside or

by an insider). We can therefore see that failure to identify the changes that have taken place over time is a significant risk vulnerability in cloud administration.

If we refer back to the visualisation challenges for TRESPASS that we stated earlier in this deliverable, we can see that our work in this deliverable responds to these challenges in a particular way:

- The visualisation prototypes presented here identify the visualisation principles that enable the social (in this case cloud actors), technical, physical and organisational (particularly the administrative) changes to the cloud environment to be visualised both as an integrated whole and within their individual dimensions;
- Develop visualisation techniques to respond to the challenges of change over time based on the visualisation principles identified; and
- Develop general and specific techniques for visualising the inter-relationships between the social, technical, physical and organisational components and thereby enabling cloud administrators to identify potential vulnerabilities in the cloud environment resulting from system and configuration changes.

As cloud systems, be it private or public, are highly attractive targets for intruders there is a high risk for attacks that might occur in smaller steps over a long period of time (e.g., in *Advanced Persistent Threats* (see for example (Fernandes, Soares, Gomes, Freire, & Inácio, 2014) and (Five, 2011)). Making changes of a cloud system more easily understandable by visualisation therefore is a means to handle such advanced risks.

For this reason we have put our efforts into prototypes looking especially into visualisations to make changes of the system understandable. The two following prototypes, TiCoVis described in Section 4.2 and CEAV in Section 4.3, aim to give on the one hand a direct representation of change over time for a specific type of relation (in TiCoVis) and on the other hand a more complex structural representation of the cloud environment for selected time intervals, both focusing on change occuring over time.

4.1 Live Visualisation of a cloud environment – SAVE

SAVE is a data extraction and policy analysis tool that was developed by IBM as part of the EU FP7 TClouds project (TClouds, 2013). The policy analysis required a simple network topology that the SAVE data collection engine built from information extracted from a number of cloud operating systems. During the TRESPASS project the data collection engine was extended to capture a richer set of information better suited to the requirements of the TRESPASS modelling language developed in WP1 (see The TRESPASS Project, D2.2.2 (2015)). The extended data extraction capabilities, in particular the ability to capture a consistent snapshot of a virtualised infrastructure were later transferred to an IBM product.

As part of these extensions, work in the TRESPASS project on the visualisation of the status of the cloud environment focused on visualising the detailed system state in a live

graph presentation (see Figure 4.1) for exploration and real-time highlighting of policy violations (see Figure 4.2). Detailed description of the work on security analysis and policy checking can be found in Bleikertz, Vogel, and Groß (2014) and Bleikertz, Vogel, Groß, and Mödersheim (2015).

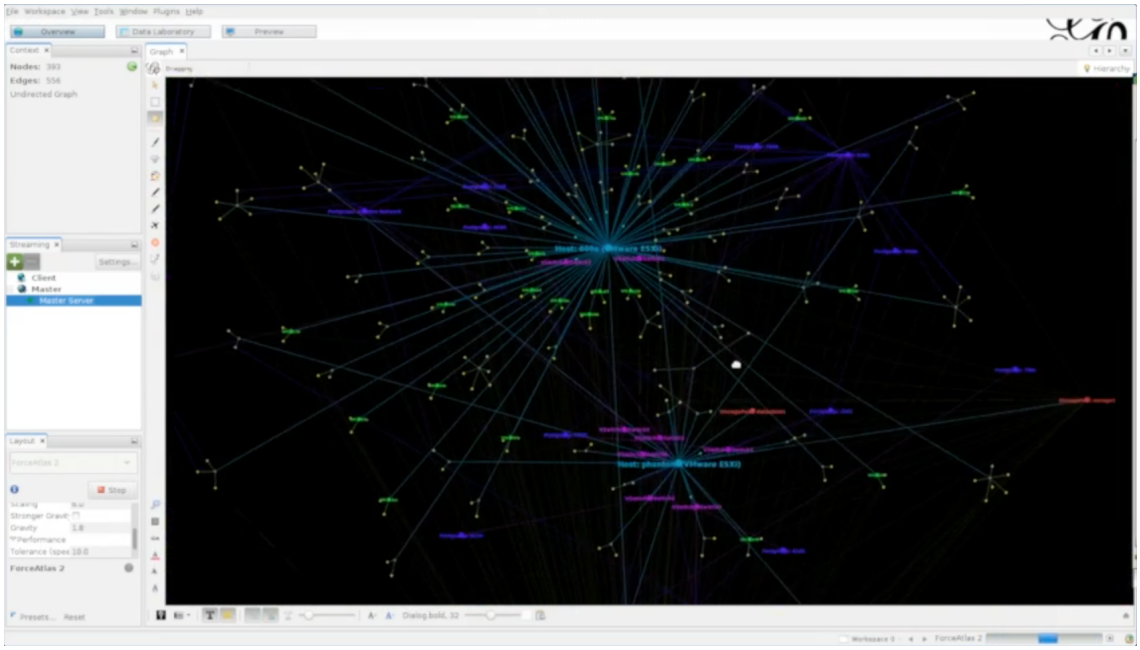

Figure 4.1: Graph visualisation of the live cloud environment state (using Gephi[1]). Different colors indicate different component types like physical servers, virtual machines, storage, network. The interface allows zooming and selecting components for a more detailed exploration.

Following on from this work, we investigated how the changes in a complex cloud environment could be visually represented, to allow understanding in a visual way of the history of the system and some of the associated risks in that history. This lead to the prototypes described in the next two sections.

[1] Gephi - The Open Graph Viz Platform at https://gephi.org

Figure 4.2: SAVE Visualisation showing a policy violation occurring because of a disallowed network connection between test and production machines.

4.2 Representing selected containment-relation over time – the Time-Containment Visualiser (TiCoVis)

The Time-Containment Visualiser (TiCoVis) creates an "alluvial" view of a selected "container-content" relation, e.g., between physical servers and virtual machines, over time. In an alluvial diagram, time is an integral part of the visualisation and the "flow" of contained elements between containers over time is directly visible as it is laid out spatially. Zooming and panning functionality allows the big picture to be viewed over time as well as details for specific time intervals.

As in this prototype the focus is just on one specific containment-like relation between two types of instances, here the placement of virtual machines on physical hosts, it is possible to explicitly show time as one dimension of the representation. However, the large number of elements and change events require steps to visually summarise and focus on changes rather than unchanging elements.

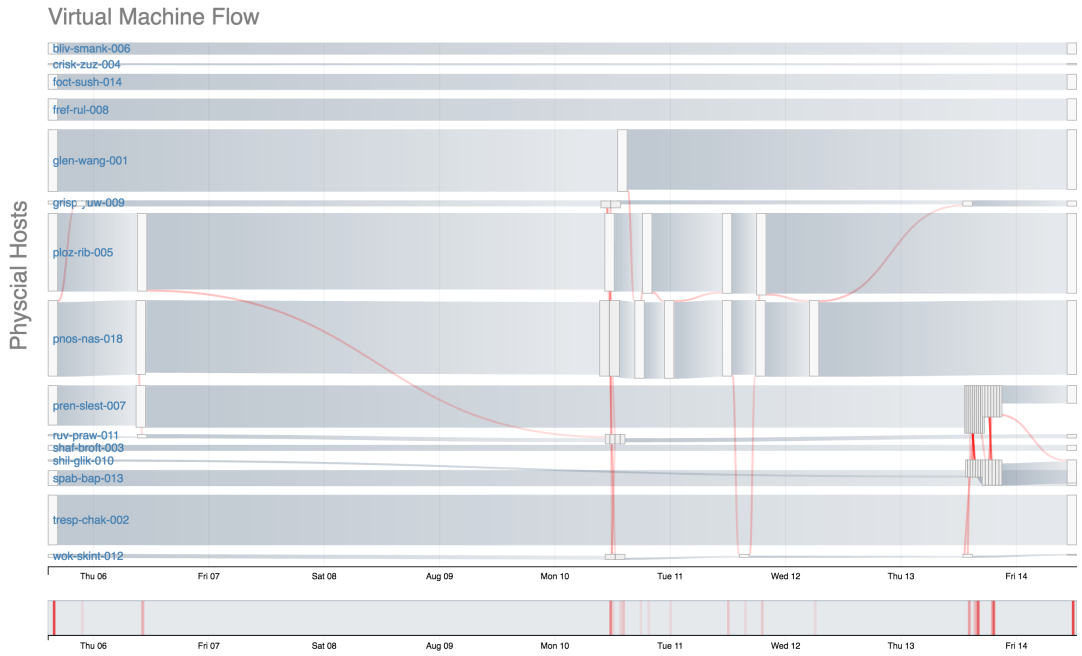

Figure 4.3: The alluvial flow of virtual machines contained by physical hosts over time.

Figure 4.3 shows the entry screen of TiCoVis with the representation of data from a live medium-level private cloud (for protection, the data is suitable anonymised – a cloud administrator would of course see instead the host names with which they are familiar).

The different horizontal bands represent physical host machines over time, the width of the bands indicates the number of virtual machines deployed on that host. Rectangles represent a host at a specific point in time, when a change occurred for this host. Flows are coloured with a gradient in order to further clarify visually where changes are occurring.

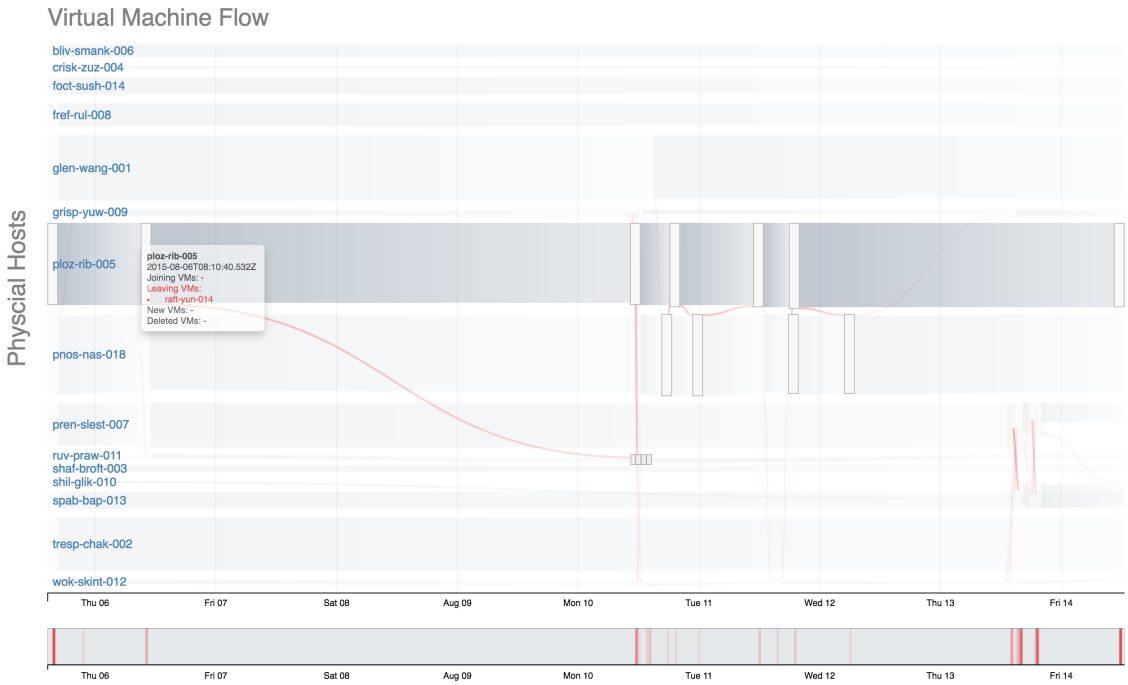

Figure 4.4: Same screen as above, showing that hovering over connections or host rectangles gives details of virtual machines remaining respectively moving between hosts (for connections) as well as joining/leaving/new/deleted virtual machines (for host rectangles).

Time is represented in the horizontal axis. The upper part shows the main information for the time interval selected in the timeline at the bottom. The timeline summarises the available data as a fixed full interval by placing red markings for all change events, giving a direct indication where, respectively when, changes occurred.

Hovering over flows or host rectangles will dim all flows except the ones connected to this flow/host and show information of the involved virtual machines and related changes in a tooltip (see Figure 4.4) – animations of the features described here can be found at our visualisation showcase at https://visualisation.trespass-project.eu/?p=55.

Zooming and panning is enabled on both the data area and the timeline for easy selection of arbitrary time intervals, enabling to identify time intervals dense with changes (see Figure 4.5 for a detail).

However, at some time intervals so many changes were done to the system that marking each change by a rectangle would lead to complete overload. For these time intervals, the entries have been summarised into special summarisation nodes (double the width of normal nodes with slightly darker color and a pattern indicating how many events are summarised within). This can for example be seen in Figure 4.3 in the lower right hand side. Zooming into this time regime will gradually unfold the contained nodes (shown in Figure 4.5).

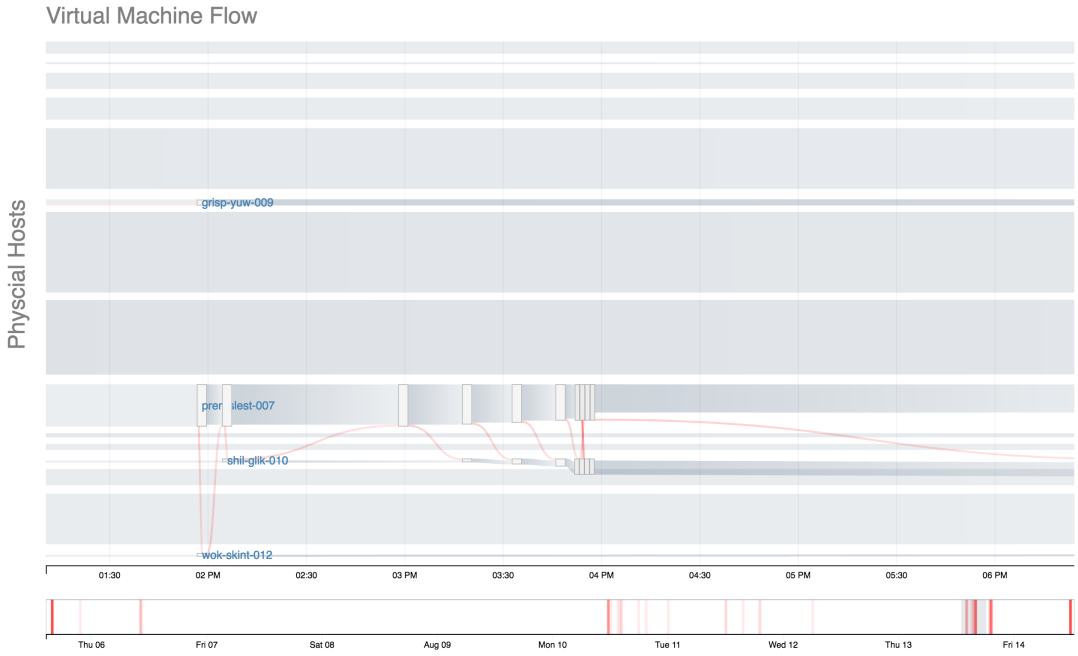

Figure 4.5: The alluvial flow of virtual machines contained by physical hosts over time - zoomed to a smaller time-interval, resolving previous summarisation steps. Hovering gives details of virtual machines joining/leaving as well as new/deleted for the specific host.

To clarify the specific steps used to simplify the large set of data, we show intermediate stages of the visualisation work in Figure 4.6–4.8:

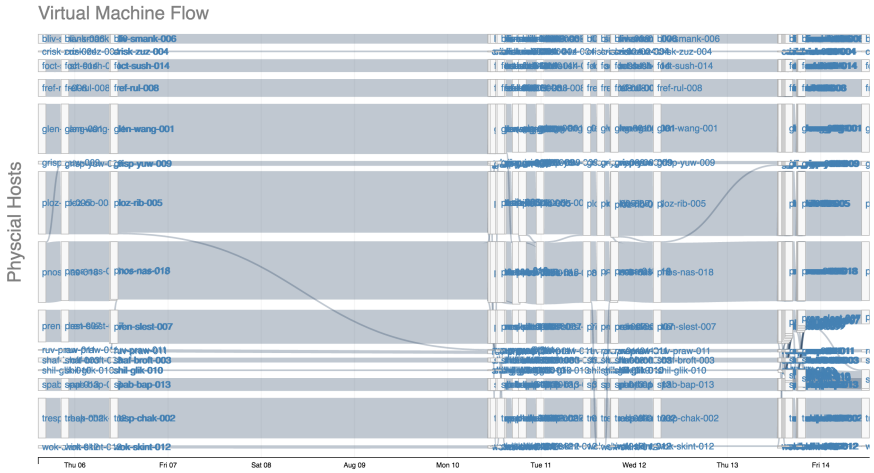

Figure 4.6: Step 1 of visual data representation - original data set (630 nodes, 655 links).

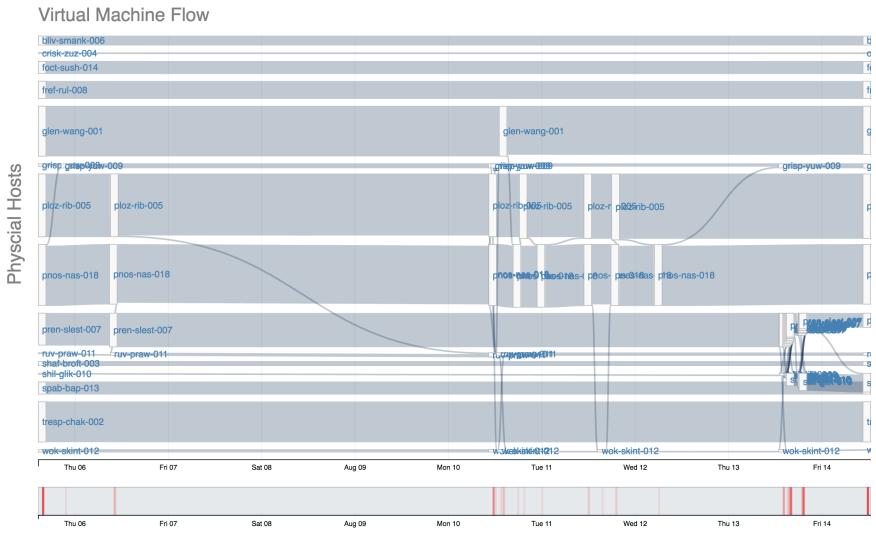

Figure 4.7: Step 2 of visual data abstraction: eliminating unchanging host information reduces the data to 89 nodes & 114 links. A timeline is added to highlight in summarised form at which times changes occurred (red markings in timeline), as well as to select a time-interval to focus and zoom in.

1. Figure 4.6 shows the original data consisting of 630 nodes and 655 connections.
2. To reach Figure 4.7 unchanging elements of the data have been reduced (leaving 89 nodes with 114 connections), additionally introducing an overview and selection timeline marking changes in the data.

Figure 4.8: Step 3 of visual data abstraction: highlighting changing elements as opposed to unchanging, as well as cleaning up unrequired labels. In the final step 4 of abstraction, leading to Figure 4.3, special summarisation nodes are introduced where existing nodes are too close and overlap. This reduces the data visually to 56 nodes & 81 links.

3. In Figure 4.8 focus has been put on changes by highlighting virtual machine movements between different hosts as compared to virtual machines staying on the same host. Gradients in the connections further strengthen the visual appearance of change times; overlapping and unnecessary labels have been removed.
4. As a final step, summarisation nodes replace overlapping nodes, indicating by the density of the vertical stripe pattern how many nodes are summarised.

In summary, this visualisation uses a novel combination of an alluvial flow representation with an explicit time scale with fluid summarisation/unfolding during zooming in the time scale.

4.3 Structural representation of cloud/actor system – the Cloud Environment & Actor Visualiser (CEAV)

The Cloud Environment & Actor Visualiser (CEAV) visualises a cloud environment, including infrastructure such as physical servers and virtual machines as well as cloud actors. The environment is depicted over time with a focus on the roles the administrators have on parts of the infrastructure. As cloud environments typically have a large number of components, the view abstracts/summarises unchanging parts visually, allowing the user to focus on the changing elements over a given time interval. The time interval of interest can be selected from a timeline that indicates changes in an overview of the available date range.

Representing the overall cloud environment including its actors does not leave room for the explicit representation of time as a spatial dimension (as in TiCoVis). Therefore in this prototype snapshots of the system state, and respectively highlighted changes during a selected time interval, are shown together with a timeline to summarise times of change as well as to select the time interval for which to show changes.

Figure 4.9 shows the initial view of CEAV, again for data from a real medium-level private cloud (for protection, the data is anonymised – a cloud administrator would see instead the user and infrastructure names with which he is familiar). This more complicated view shows on the left hand side the various cloud actors (represented by their user ids in the cloud), and on the right hand side a depiction of the parts making up the cloud infrastructure. Both parts are connected by role links that show what level of access control the actors on the left have over which part of the cloud infrastructure on the right. The infrastructure parts form a hierarchy through parent-child relationships (as given by the cloud management backend, here VMware vCenter) that is essentially used for grouping of similar types of the infrastructure. Additionally there are many other types of relations between the elements, e.g., the containment relationship between virtual machines and physical hosts, as exploited in TiCoVis above. These relationships can be highlighted and named when selecting individual elements as is shown in Figure 4.10 – animations of the features described here can be found at our visualisation showcase at https://visualisation.trespass-project.eu/?p=216.

This structural representation of the cloud environment is accompanied by a timeline below that shows the full range of observation available, marking again where changes occurred (where red marks changes in the infrastructure, blue a change in actors or roles). In this way, administrators are able to identify social, organisational, technological and physical changes to the cloud environment and consider these changes as vulnerabilities both within their individual domains and as a composition. This is in-line with the visualisation challenges identified and with WP4 responses to such challenges.

As the structure here is more complex and there is typically a large number of cloud infrastructure elements, the representation focuses on changes occurring in the time interval selected in the timeline. Hereby a red colour signifies vanishing, a green colour newly introduced relationships in the graph. To keep the representation visually readable despite the large number elements, abstraction is employed here to summarise similar elements

into nodes marked 'Unchanging' together with a counter of summarised nodes (on this particular hierarchical level, each of which potentially represents a much larger number of lower-level elements).

Figure 4.11 shows the selection of a smaller time interval around the time where the role changes occurred. Hovering over the role connection shows details of the role type and the explicit user respectively infrastructure element.

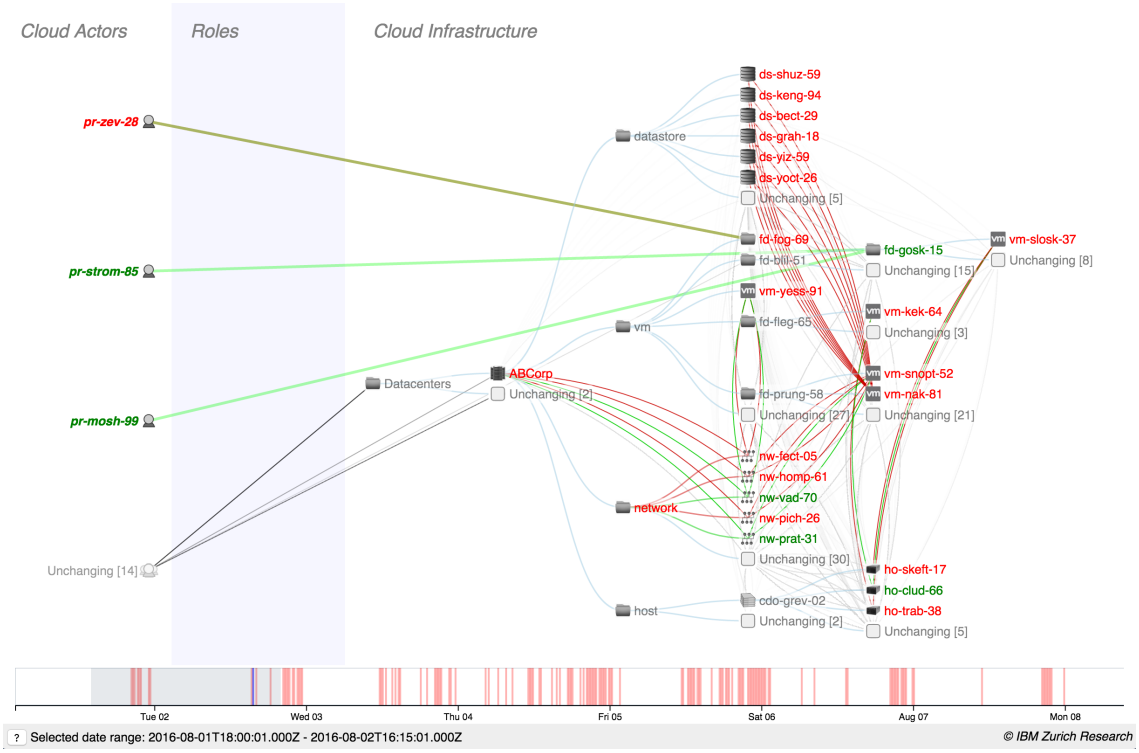

Figure 4.9: Changes of the cloud environment over time. The upper part shows the cloud actors to the left, the cloud infrastructure parts to the right, while connecting both parts by showing the access roles the actors have on the infrastructure. A timeline below shows where changes occur (red for changes in the infrastructure, blue for access role changes), allowing the selection of a time interval for which the changes are summarised and highlighted above.

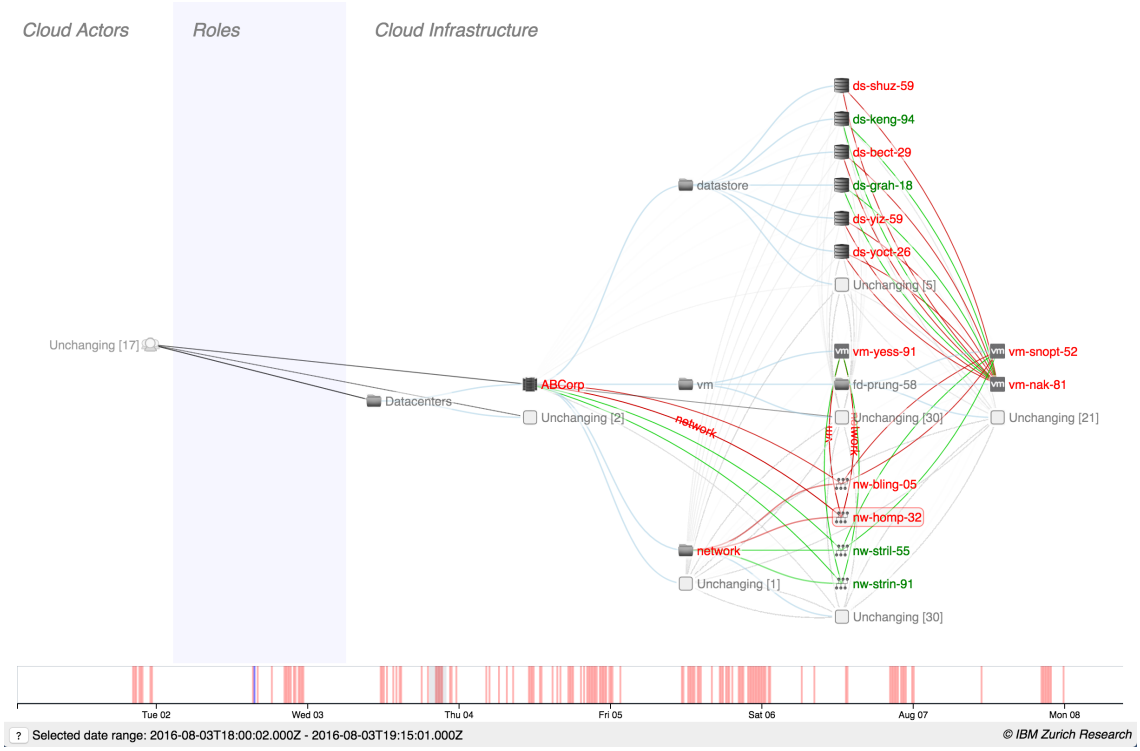

Figure 4.10: Changes of cloud environment over time: selection of one element makes it possible to see the relationship with other elements of the infrastructure by highlighting the corresponding connections.

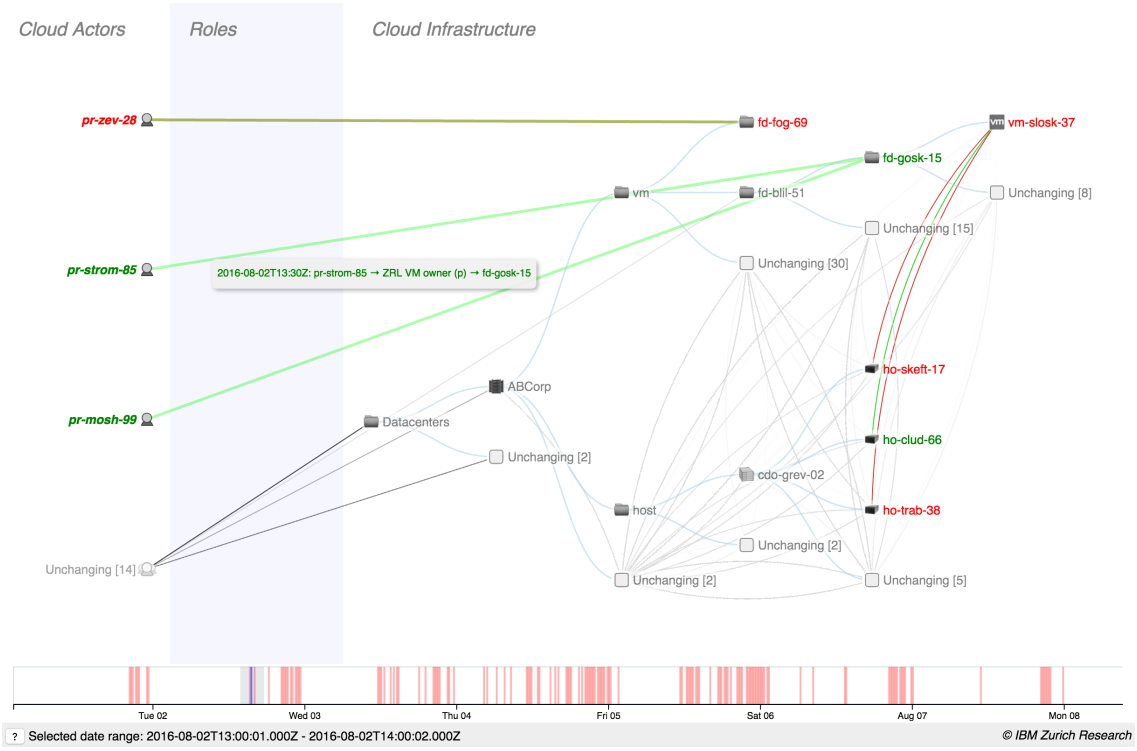

Figure 4.11: Changes of cloud environment over time: selection of a smaller time interval around the role change shows the correspondingly different set of environment changes.

In the following figures we again show the major steps leading from the original dataset to the final prototype:

1. Figure 4.12 shows the original complete data without any time indication or notion of change.
2. In Figure 4.13 a timeline is added showing the change events, allowing the selection of a time interval for which changes are highlighted in the structural representation. As can be seen, this representation is still hardly readable due to the large number of elements.
3. Introduction of the summarisation of unchanging elements leads to the prototype as it is shown in Figure 4.9.

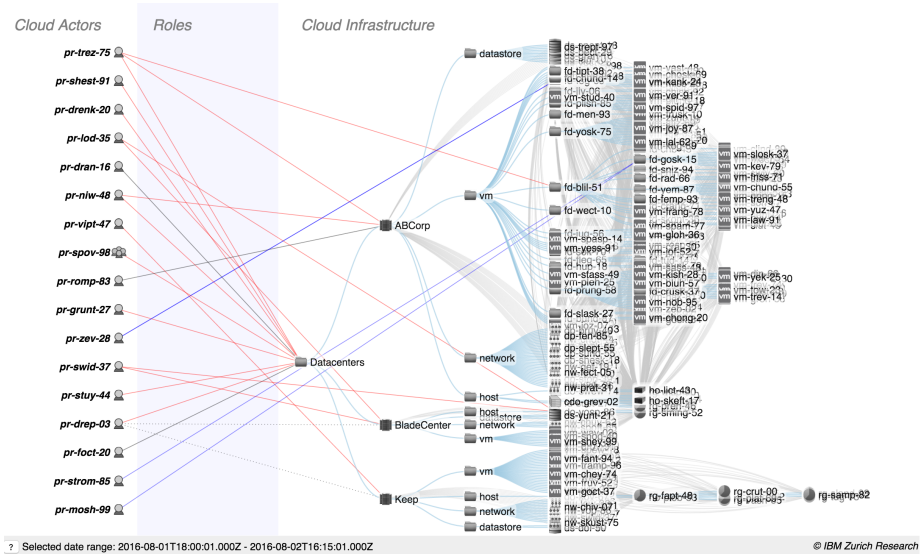

Figure 4.12: Structure of a cloud environment: original data without time selection.

In summary, this prototype is able to show in a visually understandable way changes occurring in a cloud environment, be it on the infrastructural side or the cloud actors. A special focus is given to the access roles the actors have over the cloud infrastructure. This is done by combining a timeline showing changes over the time of data collecion with a structural representation focusing on change during a selected time interval. Due to the large number of elements, this requires a strict summarisation and abstraction of unchanging elements to generate a visual representation that is still readable. Using this technology, cloud administrators are able to identify where vulnerabilities are introduced by both individual and compositions of change to the cloud environment.

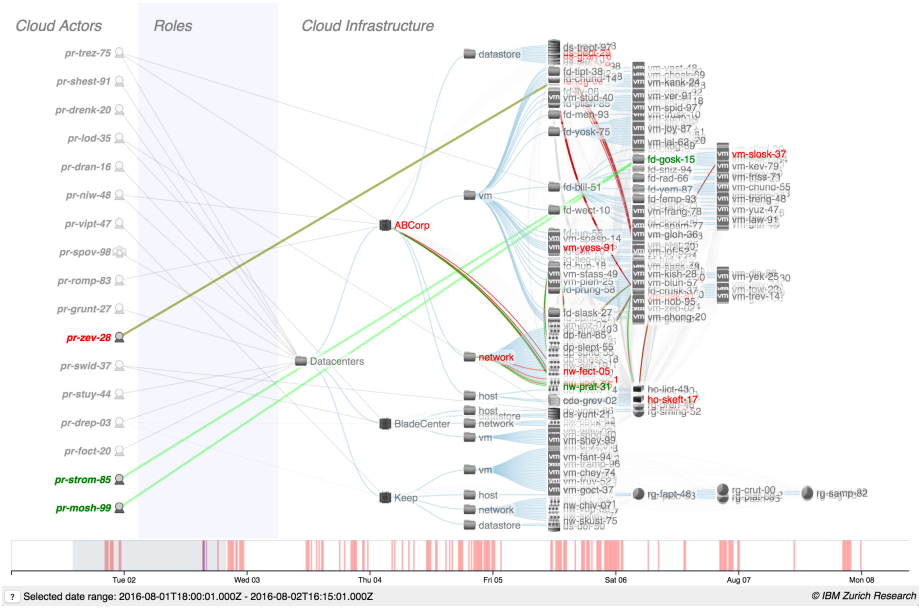

Figure 4.13: Structure of a cloud environment: addition of timeline and highlighting changes occurring during the selected time interval.

5 Evaluation

In order to better understand how our prototypes might be used, TiCoVis and CEAV were shown to representatives of our two target user communities: a cloud administrator and a compliance officer working at IBM. The goal of the evaluation session was to identify whether the philosophy of the complexity prototypes aligns with the real-world goals and tasks of these user communities.

We used the following process to evaluate the prototypes with our two target user communities:

- Short introduction to the background of the TRE$_S$PASS project in general;
- Short overview of the goals of the complexity visualisations; and
- Walk-through of both prototypes.

5.1 Feedback for TiCoVis

Cloud administrator The cloud administrator had the following feedback:

- For this relationship, i.e., virtual machines to hosts, the interviewee doubted that the visualisation can scale up to large cloud environments. However, a different relationship or a different definition of the flows could overcome this, e.g., when looking at classes of hosts like production/test/development or migration of data from encrypted to unencrypted storage the representation could work and be useful. In that case, any migration between the flows would clearly indicate a problem that has to be investigated.
- The visualisation could also be useful as an “ambient” view in a cloud operation centre: with such a view, when updated in realtime, one could see if the system fundamentally changes away from a “normal” behaviour, e.g., when rapid oscillations between hosts would start to occur, and investigate accordingly.

The cloud administrator also identified a number of areas where the prototype might be re-focused:

- In cases where policies could be directly tested, he would prefer the system to raise an alarm and in urgent cases push the alarm by SMS to the cloud administrator rather than to rely on the attention to the visualisation.

- Fundamentally, from an operational perspective, he trusts the correct workings of the automation to follow the given policies, therefore does not look for visual confirmation. As a result, he suggested that we place less development effort in that direction.
- Specifically in VMware-based clouds, migration of virtual machines between physical hosts is mostly happening automatically, so the selection of this relationship for the TiCoVis prototype is not optimal.

Compliance officer The Compliance officer had the following feedback:

- Interesting representation of the data, how could one understand the reasons for the behaviour? As this is not directly in the data, this would require linking to further data sources.
- Discussing the focus on change: better focus on the times where events occur rather the periods without change. However, in the current visualisation it is positive that one keeps the intuitive feel of the time scale, of when and how often events occur.
- To strengthen the aspect of time scale, a clearer structure in the lower time-line could be helpful, showing day and week structure more clearly (although this would need to take into account a potential span across time zones and cultural preferences, i.e., weekend days in different cultures). This could allow to quickly find possible regular changes ('every day at 1am').
- A visualisation like this can be helpful in explanations to auditors to proof certain events over time (rather than looking through log files in text form).

5.2 Feedback for CEAV

Cloud administrator The cloud administrator had the following feedback:

- More interesting as more details are shown, filter changes (i.e., changes that are driven by automatic mechanisms).
- Focus on access roles – could be a real-time display in an operation centre showing the time interval to the last change – as for roles no change should be the norm. Possibly switch to black background in normal operations, with coloured background in case of changes as an alert mechanism.
- In a different setting, such a visualisation could be interesting for showing application- or project-level roles to highlight last changes.

Compliance officer The Compliance officer had the following feedback:

- Good different view showing a familiar structure (the VMware cloud environment) in a larger context. Clear structure with actors / roles / infrastructure.
- Could be used on an operational level ('what happened tonight?'), but also in forensic investigations for a quick understanding of what happened during a longer time before an incident occurred.
- If the data used would come from a different source than the normal operations data, it could be used as a quick independent secondary assessment of the correct behaviour of the system.

5.3 Conclusions from evaluations

Both the cloud administrator and compliance officer agreed that for operational purposes they would rather get alarms by the system in case of serious events than having to rely on operators to be attentive to a visualisation.

From their feedback it can be seen that TiCoVis and CEAV could be useful in the following contexts of use:

- as an "ambient" visualisation of the cloud environment in an operation centre to detect a behaviour change of the environment, or get a quick overview about "what happened tonight"
- as an exploratory tool for forensic investigations that might span a longer time period to quickly find time intervals of interest.

As we can see from the feedback presented in this chapter, the two prototypes address an advanced dimension of information security risk assessment not addressed by other tools, namely the security vulnerabilities introduced by system and configuration changes over time. In particular, we have developed technologies that enable a cloud administrator to detect and examine cloud behaviour change and to identify time intervals of interest. Such a technology has the potential for real impact because, for example, it enhances a cloud administrator's ability to identify long-running stealth attacks, e.g., in the form of advanced persistent threats, that can be very hard to identify as the constituting events and circumstantial evidence will be spread out over time in an otherwise already highly dynamic system.

In the case of both prototypes, the visualisations have the potential to augment the strong human ability for visual pattern recognition necessary in this form of risk assessment.

6 Conclusions

This deliverable presents the work that WP4 has undertaken to examine the problem of visualising changes over time to complex technological environments. The particular environment that we have focused on is the cloud environment. We initially developed a cloud visualisation prototype from the work IBM had conducted on the TClouds project (TClouds, 2013). This visualisation displayed the real-time status of the cloud environment as a general graph. From this visualisation work, we identified the need to visually distinguish between the static and changing parts of the cloud environment in order to highlight changes over time. In particular, we identified that change over time is an advanced property of risk in the cloud environment that needs to be visualised contrasted to the elements that do not sustain change.

We have developed two prototypes to present two views of environment change that are important to the cloud administrator community. These views are:

- Changes to content-containment-type relationships (for example the containment of virtual machines on physical servers in a cloud environment) over time using time as a spatial dimension.
- Visualisations of the structural changes in the relationship between actors and infrastructure parts of the cloud environment.

As these prototypes demonstrate, the visualisation responses to advanced aspects of risk represent innovation, but also rely on the fundamental visualisation techniques for responding to complexity that are inherent in the core Attack Navigator Map (ANM). We therefore also, in this deliverable, re-cap on the complexity responses that the ANM offers and demonstrate how we use those visualisation techniques as part of our response to the visualisation challenges identified above.

The feedback by a cloud administrator and a compliance officer at IBM reflect the usefulness of this direction of inquiry and articulate the potential value of these tools.

References

Ashby, W. R., et al. (1956). An introduction to cybernetics. *An introduction to cybernetics.*.

Bleikertz, S., Vogel, C., & Groß, T. (2014). Cloud radar: Near real-time detection of security failures in dynamic virtualized infrastructures. In *Proceedings of the 30th annual computer security applications conference* (pp. 26–35). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2664243.2664274 doi: 10.1145/2664243.2664274

Bleikertz, S., Vogel, C., Groß, T., & Mödersheim, S. (2015). Proactive security analysis of changes in virtualized infrastructures. In *Proceedings of the 31st annual computer security applications conference* (pp. 51–60). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2818000.2818034 doi: 10.1145/2818000.2818034

Fernandes, D. A., Soares, L. F., Gomes, J. a. V., Freire, M. M., & Inácio, P. R. (2014, April). Security issues in cloud environments: A survey. *Int. J. Inf. Secur.*, *13*(2), 113–170. Retrieved from http://dx.doi.org/10.1007/s10207-013-0208-7 doi: 10.1007/s10207-013-0208-7

Five, C. (2011). Advanced Persistent Threats : A Decade in Review. (June), 1–13. Retrieved from http://www.commandfive.com/papers/C5{_}APT{_}ADecadeInReview.pdf

Flach, J. (2012, September). Complexity: learning to muddle through. *Cognition, Technology & Work*, *14*(3), 187-197. doi: 10.1007/s10111-011-0201-8

Galison, P. (2006). Images scatter into data, data gather into images. *Images: A Reader*, 236.

Gershenson, C., & Niazi, M. A. (2013). Multidisciplinary applications of complex networks modeling, simulation, visualization, and analysis. *Complex Adaptive Systems Modeling*, *1*(1), 1–4.

Goodburn, D., Vernik, R., Phillips, M. P., & Sabine, J. (1999). Integrated visualisation and description of complex systems. Retrieved from http://www.dsto.defence.gov.au/corporate/reports/DSTO-RR-0154.pdf

Holland, J. H. (2006). Studying complex adaptive systems. *Journal of Systems Science and Complexity*, *19*(1), 1-8. Retrieved from http://dx.doi.org/10.1007/s11424-006-0001-z doi: 10.1007/s11424-006-0001-z

Ingold, T. (2000). *The perception of the environment: essays on livelihood, dwelling and skill*. Psychology Press.

Mitchell, M. (2009). *Complexity: A guided tour*. Oxford University Press, USA.

Moody, D. (2009, November). The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Trans. Softw. Eng.*, *35*(6), 756–779. Retrieved from http://dx.doi.org/10.1109/TSE.2009.67 doi: 10.1109/TSE.2009.67

Norman, D. A., & Stappers, P. J. (2015). Designx: Complex sociotechnical systems. *She Ji: The Journal of Design, Economics, and Innovation*, *1*(2), 83 - 106. Retrieved from http://www.sciencedirect.com/science/article/pii/S240587261530037X doi: http://dx.doi.org/10.1016/j.sheji.2016.01.002

North, C., & Shneiderman, B. (2000, November). Snap-together visualization: Can users construct and operate coordinated visualizations? *Int. J. Hum.-Comput. Stud.*, *53*(5), 715–739. Retrieved from http://dx.doi.org/10.1006/ijhc.2000.0418 doi: 10.1006/ijhc.2000.0418

Rosenquist, M. (2009, December). *Prioritizing information security risks with threat agent risk assessment.* "http://www.communities.intel.com/servlet/JiveServlet/download/4693-1-3205/Prioritizing_Info_Security_Risks_with_TARA.pdf". Intel Corporation.

TClouds. (2013). *Trustworthy Clouds - Privacy and Resilience for Internet-scale Critical Infrastructure.* http://www.tclouds-project.eu. (Online; accessed 2016-09-20)

The TRE$_S$PASS Project, D2.2.2. (2015). *Data extraction from virtualized infrastructures.* (Deliverable D2.2.2)

The TRE$_S$PASS Project, D4.2.1. (2014). *Initial report on visualizations of information security risks.* (Deliverable D4.2.1)

The TRE$_S$PASS Project, D4.2.2. (2016). *Methods for visualization of information security risks.* (Deliverable D4.2.2)

The TRE$_S$PASS Project, D4.3.3. (2016). *Visualizations of socio-technical dimensions of information-security risks.* (Deliverable D4.3.3)

Thompson, J. D. (1967). *Organizations in action: Social science bases of administrative theory*. Transaction publishers.

Tufte, E. (1990). *Envisioning information*. Graphics Press.

Weaver, W. (1948). Science and Complexity. *American Scientist*, *36*(536).

Wolfram, S. (2002). *A new kind of science*. Wolfram Media.