User-Controlled Collaborations in the Context of Trust Extended Environments†

Jonathan Chan, Surya Nepal, David Moreland, Hon Hwang, Shiping Chen and John Zic
Networking Technologies Laboratory, CSIRO ICT Centre
Cnr Vimiera & Pembroke Rds, Marsfield, NSW 2122, Australia
{Firstname.Lastname}@csiro.au

Abstract
The area of virtual organisations researches mechanisms to enable entities from different organisations to collectively create a virtual enterprise for some mutual benefit. Our research is particularly interested in the type of collaboration where resources are normally ‘opaque’ and are only revealed ‘as required’ between a select group of collaborating partners. Working in this area has revealed a need to establish a trusted environment for all application transactions prior-to, during and after a collaboration e.g. a pre-active eContract negotiation phase, an active phase for all collaborating activities and a post-active phase for information retrieval. Central to our embodiment of the trusted environment is the development and use of a portable Trust Extension Device (TED) with applicability to each phase of the collaboration process.

1. Introduction

The emerging area of virtual organisations explores mechanisms that enable entities from different organisations to collectively create a virtual enterprise for some mutual benefit. This goal is typically achieved through service discovery, negotiation and execution based on service level agreements (SLAs) such as an eContract. An eContract is a specification that embodies the requirements for deploying, monitoring and enforcing the workflow of cross-organizational processes [1, 2]. As such, the eContract acts as a trusted intermediary between interacting cross-organisational processes, where services are ‘open’ to discovery by collaborating partners.

Another collaboration scenario is where resources (facilities and people) are normally ‘opaque’ to outsiders and are only revealed ‘as required’ between a select group of collaborating partners. In this case the eContract takes on a role at a higher level of abstraction whereby it is used to collectively capture resources contributed by each collaborating partner i.e. in this sense the collaboration is essentially user-controlled. One possible way of managing eContracts of this type is via a trusted third-party service provider (e.g. a Virtual Network Operator), as explained in a previous paper by the authors [4]. For example, in the movie industry post production houses prefer to form temporary collaborations with partners, clients, suppliers etc, say, to produce movie special effects. Due to the secrecy and confidentiality surrounding the movie making process, it is highly desirable to employ a trusted third-party service provider to manage the sharing of specialized resources. In this sense, the notion of trust has social connotations. A service provider acts as an impartial ‘honest broker’ among all collaborating partners; where the trustworthiness of a service provider can be corroborated by activities such as, activity logging, performance monitoring and audit trails. The eContract information may be used in a legal context to resolve disputes, trace malicious behaviour, etc, when used in conjunction with gathered evidence as mentioned above. User-controlled collaborations are also applicable to other market sectors such as the health industry and financial institutions. Within these areas we have also made exploratory contributions [5].

Apart from the social aspects of trust, where all collaborating partners consent to a third-party service provider acting as an honest-broker, there is also a need to establish a separate trusted operating environment for each collaborating partner with the third-party service provider. This can be achieved via a trust extended environment, to each user, prior-to, during and after a collaboration, e.g. a pre-active eContract negotiation phase, an active phase for all collaborating transactions and a post-active phase for information retrieval and sharing. One way of creating such a trust extended environment is the motivation of this paper.

The remainder of this paper is outlined as follows. In section 2, we describe an on-demand trust extended

†This work is supported by the Australian Government through the Advanced Networks Program of the Department of Communications, Information Technology and the Arts.
environment for user-controlled collaborations and also an emulated trust extension device within a simplified demonstration. Section 3 gives an exemplary scenario, namely a secure video conferencing service, which shows the use of this environment for eContract negotiation, collaborating transactions and information retrieval. Section 4 concludes with thoughts regarding further work.

2. Creation of On-demand Trusted Environments

2.1 Motivation

In a previous paper [4] we focused on the creation of managed virtual enterprises, which used an eContract service to capture the specifications that constituted what we termed a ‘collaborative context’ (e.g. virtual enterprise). However this work requires strengthening from a trust and security perspective, namely, eContract negotiations were not subject to trust attestation procedures and all negotiated resources were assumed to be trustworthy not only within their own company but also between all collaborating companies. The above issues motivated the requirement of an extended trusted environment for all phases of a collaboration. This trust extended environment addresses the following problem-space. Firstly, collaborating partners together with third-party service providers need to ensure that eContract negotiations are legitimate. And secondly, that collaborating activities operate within the boundary specified by the eContract.

2.2 Related Work

Recently there has been a surge in applications requiring the integration of trusted operations within an untrusted environment. These applications are driven from the daily demands of commonplace activities such as on-line shopping and remote access to personal computers. For the on-line shopping application [6] a trusted device (PDA or mobile phone) is used in the context of managing credit card disputes, for example, the non-repudiation of credit card transactions. Similarly, [7] utilizes a PDA as a trusted device connected to an untrusted host to emulate mouse-clicks for the purpose of displaying the screen of a remote personal computer. These solutions authenticate operations and credentials between local-remote entities via portable devices. Alternatively, research groups from Stanford [8] and Princeton [9] integrate a trusted virtual machine within an untrusted host. As such the trusted VM is a more versatile tool for accommodating many application scenarios that require a trusted environment.

In this paper we propose a solution founded on the notion of trust portability [10], where the embodiment of trust is not integrated with a particular computing platform i.e. the trusted computing environment is encapsulated into a portable device that may be plugged into any untrusted host. As such an on-demand trusted environment is created that collaborating partners can work within to collectively achieve a common goal.

2.3 Trust Extension Device (TED) Attestation

We explore the concept of a Trust Extension Device (TED), which is envisaged to be a small portable device (e.g. Gumstix-like portable [11]) that a user may plug into any host machine to create, on-demand, the trusted environment required for collaborations to take place. In this sense trust is extended to wherever the TED is plugged in. How this is achieved is described in the following section.

Physically a TED is a ‘closed-box’ system (i.e. tamper-resistant) that can be conveniently carried in the pocket. As a closed-box, a TED can run its own operating system and a combination of preconfigured applications and other service modules that are loaded by service providers in accordance with the attestation procedures shown in Figure 1. The central component of the attestation procedures is a TED manager. The TED manager issues TEDs offline and verifies their integrity at run time. Furthermore, the TED manager acts as a centralised certifying authority on behalf of multiple third party service operators. In this set up it is possible for a TED to host service modules from multiple service providers while the burden of TED attestation, by each service provider, is eliminated. The interactions shown in Figure 1, between a TED, TED manager and third party service providers, are briefly described as follows:

- For the TED manager to act as the certifying authority and trust attester, all service providers must register themselves and update their service module signatures with the TED manager (points 1 and 2).
- When a user subscribes to a particular service, the TED sends a request (point 3) to the service provider to acquire the appropriate service module.
- On receiving the request, the service provider sends a challenge to the TED (point 4).
- On receiving the challenge, the TED’s Trusted Platform Module (TPM) [12] generates a session key (point 5) which in turn is used to generate the ‘integrity measurement’ of the TED (point 6).
(PCR) [12] and all existing service modules (i.e. sm0, …, smi), is sent to the TED manager (point 7) for verification.

- On receipt of the integrity measurement, the TED manager attests its trustworthiness against the TPM credentials and the service module signatures (point 8). Note that these signatures were previously uploaded by the service providers in point 2 of Figure 1.

- After successfully verifying the integrity measurement, the TED manager issues a signed service credential which is sent back to the TED (point 9).

- The TED, on receiving and verifying the service credential, sends it in a challenge response to the service provider (point 10).

- Upon receipt of the challenge response and verification of the service credential (signed by the TED manager), the service provider issues the requested service module to the TED (points 11 and 12).

Referring to Figure 2, when a TED is used in an untrusted host environment a TED service module (SM) is invoked and its executable is processed using only the TED’s CPU and memory. However, there are some security aspects regarding utilisation of the host computer’s I/O and peripherals by the TED that need addressing. For example, to communicate with the outside world (e.g. service providers) through the host’s peripherals a TED ‘driver’ (possibly supplied by the TED itself) needs to be installed in the untrusted host. But before installing the driver it would be advantageous for the TED to attest that the host computer platform is equipped with Secured I/O [13]. This would ensure that a trusted path can be established between the TED and the computer’s peripherals to mitigate attacks by keystroke loggers, etc; but how this trusted path is actually realised is beyond the scope of this paper.

**Figure 2** – TED’s hosted environment

### 2.4 TED Emulation

To demonstrate aspects of trust portability, on a simplified testbed, we developed a software-emulated TED with associated trust attestation protocols. The emulated TED fulfils the following implementation criteria:

- The compelling physical attributes of TED, namely, self-contained, powered-by-host and portability, shall be preserved.
- No special software is required in the host computer to support TED operations.
- The accommodation of a guest operating system to resemble the self-contained environment of an envisaged hardware implementation.
- The primary functionality of TED, i.e. the TPM, shall be trust attestable.

Our first realisation of the emulated TED is a USB memory stick preloaded with a number of software packages. Referring to Figure 3, the guest operating system (i.e. the Ubuntu v6.06 [16]) is supported by the virtualisation software (i.e. the QEMU v0.8.2 [14]) to achieve isolation between the guest and host operating systems. The QEMU product was selected because its...
operation does not require the installation of special drivers in the host. To emulate the TPM function we used the Software-based TPM Emulator v0.4 [15] which runs on top of the Ubuntu guest OS. We acknowledge that there are security loopholes in our emulated design e.g. the use of host memory by the emulated TED is not isolated from malicious activities by the host machine. However adding robustness to the host security is not the aim of the emulated demonstration.

To validate all transactions between the emulated TED and the TED manager, we implemented a simplified version of the attestation protocol described in Section 2.2, where the TED manager and the third party service provider are combined in a single entity. We have a simple demonstration of TED portability, as shown in Figure 4, using an online banking application.

The TED concept and the attestation mechanism, described in section 2, are applicable to various service scenarios. We now show how the TED is used to create a trusted environment for specifying collaborative resources with an uploaded eContract service module. As a specific instance, an eContract may be used to capture the resource requirements for a video conferencing service, where the requirements are:

- List of nominated conferees.
- Selected network providers to provide secure networking functionality, e.g. L3 MPLS VPN.
- Selected third party provider hosting a conference bridge, storage and retrieval facilities.
- Conference duration.

To capture the demonstration, Figure 5 shows a screen snapshot of the user interface, i.e. the emulated TED operating within the host environment. Within the QEMU virtualisation window, the Ubuntu guest OS runs the online banking application to retrieve a customer’s record. For debugging purposes, the logging window shows correct operation of the trust attestation procedures, i.e. a session key is successfully attested by the TED Manager of the service provider.
3.1 Pre-active Phase of Collaboration

The trust attestation framework outlined in section 2.2 is applicable to eContract negotiation transactions between collaborating partners and a service provider. Consider a user (Negotiator N1, in Figure 6) who wants to set up a new collaboration with other partners (Negotiators N2, N3 and N4, in Figure 6). We assume that all TEDs have been loaded with an eContract service module. Referring to Figure 6, on plugging a TED into a host machine N1 the eContract service module is invoked which requests the creation of a new collaboration. On receipt of this request, the service provider composes an eContract template in accordance with the resource options held against N1. On receipt of the eContract template from the service provider, N1 populates it and submits it to the service provider. The service provider essentially orchestrates the formation of the collaboration and as such sends a notification (e.g. by email, SMS, …) to N2 to inform about the existence of the collaboration. On plugging the TED into a host machine, N2 requests to update the eContract initiated by N1. The service provider composes an amended eContract template consisting of N1’s contributions and the resource options available to N2. This template is sent to N2 to choose his own resource contributions. The eContract is then returned to the service provider and the eContract is modified in a similar way by N3 and N4.

Note that the resources contributed by a negotiator will very likely be influenced by what others have contributed. But there is a difficulty here in that a negotiator has visibility of prior contributions but not later contributions (e.g. N3 can see the contributions made by N2 and N1 but not those of N4). As such after N3 has made his contributions he must be given the opportunity to modify them subject to what N4 contributes; and similarly for N2 and N1. A mechanism is therefore required to handle this potentially time-consuming process as efficiently as possible to reduce the number of re-negotiation iterations when collectively agreeing an eContract. To achieve this aim, as part of the eContract service module functionality, a service provider may propose a specific negotiation protocol such as the following dual-stack (contribution/agreement) mechanism. For example, depending on what others have contributed N4 will choose from his own resource options and agree them (see Figure 6 step (a)), after which only N3 will be sent the eContract to agree/amend it as influenced by what N4 has committed to. Note that at this stage only N3 and N4 need to be involved in the negotiation process until they reach agreement (see Figure 6 step (b)), after which the eContract is sent only to N2. At this point N2 has visibility of the agreed contributions made by N3 and N4. Figure 6 step (c) shows that N2 chooses to amend his contributions as influenced by what N3 and N4 contributed. N3 and N4 therefore need to re-enter into negotiations (i.e. popped from the agreement stack back into the contribution stack as shown in Figure 6 step (d)) with N2 as they may choose to amend their previous contributions subject to what N2 re-negotiates. Now N2, N3 and N4 need to agree the eContract among themselves. Figure 6 step (e) shows N3 timing out, therefore N1, N2 and N4 re-enter the negotiation phase and proceed as before (see Figure 6 steps (f), (g) and (h)) until the eContract is agreed by all. Although this procedure appears time consuming we envisage that in reality negotiators will have prior knowledge of each others resources and what is expected of each other to contribute. The eContract simply formalises this prior understanding and the mechanism for re-negotiating is available to make minor tweaks.

Figure 6 – An efficient eContract negotiation mechanism

3.2 Active Phase of Collaboration

After the eContract has been collectively approved and the nominated conference time has arrived, each
conferee plugs their TED into their respective host machine. The video conferencing service procedures are outlined as follows:

- When the conferees are ready to start the meeting the network is instantiated [4] which connects the host machines and the conference facilities (storage, bridge, etc).
- Attestation procedures similar to those described for Figure 1 are invoked between the TEDs, TED manager and video conferencing service provider; whereupon each TED is uploaded with the video conferencing service module. A trusted environment is now established between the TEDs and the service provider. (Note: Ideally each TED will also attest the existence of trusted paths for the host’s peripherals).
- Once the conference starts the service provider archives all transactions in secure storage. The archived transactions may be retrieved either during or after the conference has terminated.

3.3 Post-active Phase of Collaboration

After a video-conference has finished, archived transactions may only be retrieved by a nominated user by plugging their TED into a host machine. On attestation of the TED the archived conference transactions may be retrieved into the video conferencing service module for immediate playback. Since the TED is a closed box to the user the replay contents cannot be stored in the host machine or the TED for playback at a later time. Once the retrieved transcripts have been played back they are deleted by the service module; but how they are deleted is an implementation decision. Although outside the scope of this work, as part of the retrieval service contractual agreement, a legally binding disclaimer may be issued to conferees stating that retrieved information must not be disseminated to non-conferees by any means. These procedures ensure that the video conferencing session and retrieval service operates within the trusted environment created by the TED.

4. Conclusions and Future Work

This paper described the concept of extending trust via a portable device, namely the TED, to users who wish to operate within a secure environment. Within this trusted secure environment collaborations are generally conducted within three stages;

- Pre-active: eContract negotiation for specifying specialised resources.
- Active: Instantiation of negotiated resources and interaction between them, e.g. a video-conferencing application.
- Post-activate: Information retrieval of stored collaboration transactions, e.g. a common record composed by a video-conferencing service provider.

For future work we are developing a hardware version of TED, within a Gumstix-like device, which embodies the functionality described in this paper, to realise scenarios of the type represented by our exemplary case in section 3.

5. References