Architecture and Protocol for Authorized Transient Control

Philip Robinson
TecO, Univeristät Karlsruhe & SAP Corporate Research
Vincenz-Priessnitz-Str. 1, 76227 Karlsruhe, Germany
Email: philip@teco.edu

Abstract. The range of Ubiquitous Computing environments envisioned are characterized as either constantly controlled by a permanent owner, i.e. a privatized environment, or support a general scope of controlling parties as in a public setting. This paper proposes a component architecture and operational protocol for an environment that and transiently changes from the first to the second, based on the validity of a so-called "controllable context".

1 Introduction

Security in Ubiquitous Computing is not comprehensively defined in general terms or in a single scenario. When security remains an ambiguous topic in the administration of an environment, the instinctive administrative decision is to secure as much as possible. This was the case when firewalls became part of enterprise infrastructures, in response to more and more organizations succumbing to malicious attacks, after being connected to the Internet [9]. Nevertheless, there are often exceptional cases that are not part of the everyday system execution, where the owner/ administrator of a networked environment will want to give up more control than typical. As the motivation for this paper stems from Ubiquitous Computing, it begins with a quotation from Mark Weiser in his 1991 paper [12], introducing its principles and goals, as well as his expectations for security and privacy within this domain of research and development.

"The most profound technologies are those that disappear…Fortunately, cryptographic techniques already exist to secure messages from one ubiquitous computer to another and to safeguard private information stored in networked systems. If designed into systems from the outset, these techniques can ensure that private data does not become public. A well-implemented version of ubiquitous computing could even afford better privacy protection than exists today. For example, schemes based on "digital pseudonyms" could eliminate the need to give out items of personal information that are routinely entrusted to the wires today, such as credit card number, social security number and address". {Quoted from Weiser [12]}

These *profound technologies* to which Weiser refers include electricity, heating, printing, and other such utilities that we easily take for granted in our everyday life and interactions, yet their sudden absence would cause total chaos. The argument is that computers have reached the stage where their sudden absence or malfunction could also cause societal disorder, yet we are still not at the stage where we can exist and not think twice about interacting with them [3]. The *ubiquitous computer* is therefore one that provides its services as a pure enhancement of everyday life and not a distraction [12]. Ubiquitous computers typically communicate over wireless networks, form spontaneous or ad hoc peer connections, and make decisions about service provision based on a localized sensing of the situation (conditions) of the users and their environment [13]. They are therefore embedded in the environment, serve a particular function and rely on distributed computation of data in the *network systems* to derive the situation.

Weiser claims that ubiquitous computing could even act as an enhancement to the way security is realized today. Therefore it can be understood that the decision to *secure messages from one ubiquitous computer to another* is a service decision of the ubiquitous computers and, consistent with the objectives of ubiquitous computing, should also be made with respect to the current situation. The decision of what is private and what may be disclosed (but not necessarily public) is an example of a security decision that can change depending on the situation. That is, there are some situations (ranging from critical to leisure) that one will want to disclose some piece of information that in another situation is considered as an item of nondisclosure. However, as intuitive as this may seem, if this is not *designed into the system from the outset*, this situation-based loosening of access control to a particular subject could be hazardous. However, not facilitating it could also deny critical access to information that could save lives, money and perhaps pleasure.

2 The Concept of Authorized Transient Control

A party that is "**authorized**" has approval to access and use a particular *target resource* for a particular set of *tasks* under a particular set of *constraints* [15]. The word "**transient**" applies to what is actually short in its duration or stay {Miriam-Webster Dictionary online: www.m-w.com}, as opposed to having preconceived intentions and natural tendencies to be long term or permanent. "**Control**" is referred to as *managerial power* or the execution of authority granted by an owner or *appointed representative* of a *domain*. The concept of "Authorized Transient Control" is the idea of an *authorized party* taking control of a domain of devices, services and data for a period of time agreed to in an *agreement* between the authorized party and the appointed representative of the domain.

The investigation of transience in ubiquitous computing is not new, as Stajano presented the concept of "*Secure Transient Association*" in his resurrecting duckling protocol [4]. He claims this as the new challenge of ubiquitous computing, where a simple object and its information concede to transient ownership by (and allegiance to) a subject or set of subjects for a particular task. Consider his example given below:

"If a householder owns a device, say a universal remote control, that lets her control various other devices in her home (such as hi-fi and television components, the heating system, lights, curtains and even the locks and burglar alarm) then she will need to ensure that a new device she buys from the shop will obey her commands, and not her neighbour's. She will want to be assured that a burglar cannot take over the heat sensing floodlight in the garden, or unlock the back door, just by sending it a command from a remote control bought in the same shop." {Quoted from Stajano, [4]}

This paper however applies further reasoning about similar issues, by defining a management architecture, methodology and protocol for allowing a subject to enter a ubiquitous computing environment (a domain of ubiquitous computers), claim authorized control, and configure the environment's devices and resources in a way that is best suited to the task and situation. This can be viewed as supporting the exceptions when the householder actually desires that a subset of her devices also obey the commands of her neighbour (or some other *trusted subject role*) and submit to their administrative control, while a particular situation is active.

2.1. Applications of the Concept

The concept is more than just defining access controls. In ubiquitous computing once someone has entered your physical environment and they have tangible access to your ubiquitous devices, they have control unless some means of alerting or active resistance is in place. The question is when does one relax the resistance and invalidate the alerts. The application scenarios presented here were derived from discussion with colleagues and considering some of the proposed ubiquitous computing household and business application scenarios.

Emergency Response

Consider a person becoming ill while at home and a medical assistant (medic) rushing to their assistance. Once on the spot, the medic may require support from the environment in order to effectively treat the patient. For example, water may need to be boiled, the heating and lights may need to be lowered or raised dependent on the ideal conditions for treatment, the medic may need to contact the hospital to download treatment information and medical records, or the medic may need to issue instructions to family members and neighbours using a display device or intercom, as well as leave contact information and notes on further treatment on the family PC or telephone.

Ubiquitous Computing Offices for Lease

The ideal situation for a traveling executive is to have access to all their information, typical administrative services, and a comfortable, familiar, hence productive working environment wherever they go. However, they do not wish the inconvenience of a heavy laptop, struggling sometimes with different means of network connectivity, and not having any secretarial and administrative support on hand. Therefore, it

might be useful if an executive could turn up at an office area with full infrastructure and administrative staff, and have the utilities configured for their usages for an agreed-to period of time. This could also be a useful facility for small entrepreneurs who do not want the overhead of a fulltime administrative staff and infrastructure. There exists a company on the Internet that proposes such a model of providing office support [1]

External Audit Visits

IT and financial auditors often complain about the long pre-audit process of finding the right people and systems to provide audit information, being granted access to the information systems and logs, and then collecting the relevant audit data in accordance with the audit controls for the audit. From the perspective of the audited organization, they are sometimes reluctant to give up control of their systems or they just cannot be bothered with the hassle and just let the auditors have full access. A mechanism supporting authorized transient control over the necessary information would serve to benefit both parties.

2.2 The Problem Analysis

The first operational issue that comes to mind when looking at these scenarios is the establishment and maintenance of mutual trust between the subject acting as a *transient controller* and the resources of the *target environment* during the activation of the *controllable-context*. However, there are some earlier management issues that arise including, specification of the context under which the environment is rendered controllable, the evidence required for a subject to claim transient control within this context, and the agreement to terms and conditions that will be enforced throughout the activation of the context. Fig. 1 tries to capture the notional states of transient control that the administrator needs to prepare for.

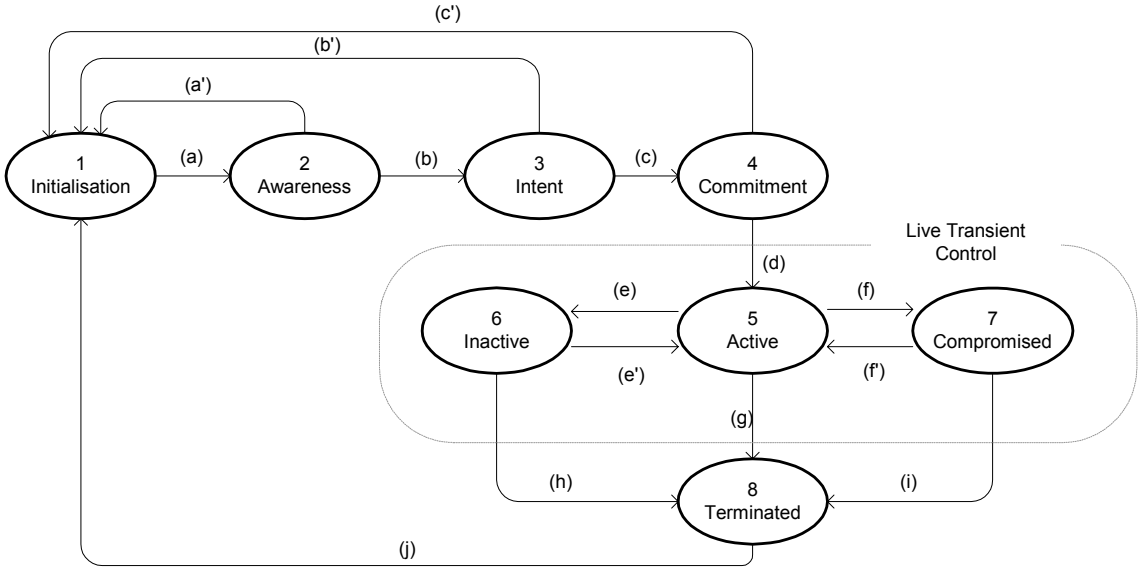

Fig. 1 State Model for Authorized Transient Control

The states labeled 1 to 8 are those where key decisions in authorized transient control are made. The transitions labeled (a) to (j) may also include other states of the target environment and transient controller, but these are treated as events that lead to the states of authorized transient control. These are explained below by identifying the administrative questions that arise within each phase.

1. Initialisation*: the preparedness for transient control*

1.1. What is the ground work required to prepare a ubiquitous computing environment for transient control?
1.2. What is given up for transient control and what are the terms and conditions?
1.3. What are the instruments by which control may be authorized, handed over and monitored?
1.4. Consequently, who is (what are the roles) responsible for the initialization, deployment and maintenance of these instruments?
1.5. How can pre-meditated abuse of the system be avoided? For example, targeting a particular environment and forcing the context to request transient control.

State-transition **(a)** is the ubiquitous environment running in a "normal" mode until a context is detected where transient control can be permitted (this context is referred to as the *controllable-context*). There is therefore some mechanism for monitoring and reporting the context of the system.

2. Awareness*: the need for transient control is detected*

2.1. How is a situation that warrants transient control detected by the system?
2.2. Is this preferably done automatically or by some manual intervention?
2.3. What are the consequences for either decision and is it a decision that can be delegated to another system or person other than the target system or its *fulltime controller* (owner/ administrator)?
2.4. How is the information that this situation has occurred kept from being disclosed to improper, potentially malicious parties?
2.5. How is the system state changed in preparation for handing over transient control?

State-transition **(b)** is the issuing of alerts to relevant parties that they are eligible/ required to take transient control of the target environment. State-transition **(a')** proceeds if the controllable-context is invalidated or modified before the process of transient control can be advanced from awareness to intent.

3. Intent*: the transient controller presents himself*

3.1. How does the transient controller prove his identity and trustworthiness?
3.2. How does the environment prove its legitimacy to the transient controller? (e.g., with respect to the scenarios, the correct emergency, reputable office provider, or correct target audit systems)
3.3. Are other parties or roles required to complete and certify the authorization of the transient controller? (e.g. is a trusted third party required to act as *transient control authority*?)

State-transition **(c)** is therefore a process of forwarding presented evidence to the respective decision points of the transient controller and the target environment. If an agreement cannot be met or there is an intermediate update of the required evidence, the transient control process is not advanced and the state-transition **(b')** is invoked.

4. Commitment*: the control agreement is made between the transient controller and the target environment*

3.1. To whom is the initial agreement presented and who formulates it?
3.2. What are the general contents, attributes and structure of an agreement?
3.3. Who are the other parties involved in the creation, signing and verification of the agreement?

State-transition **(d)** is the configuration of the environment and monitoring to ensure the bootstrapping of the terms and conditions in the agreement. State-transition **(c')** occurs if the controllable-context is invalidated before the transient control configuration is completed, or if the agreement is prematurely voided.

5. Active*: the transient controller goes to work*

5.1. How is the agreement technically specified, enacted and enforced?

5.2. How is the transient control configured and monitored throughout the duration of the situation and the agreement? To whom is the role of *transient control monitor* appointed?
5.3. What is the process for responding to breaches in the agreement by any party?

State-transition **(e)** occurs if there is some reported or sensed lapse in the progress of the transient controller without breach of the terms and conditions of the agreement (technical difficulties). However, if there is a breach detected, then the state-transition **(f)** occurs.

6. Inactive*: the transient controller cannot progress for technical reasons other than breach of agreement or expiry - results from state-transition (e)*

6.1. Can the agreement be extended, renewed or even terminated while the control agreement and controllable-context are still valid?
6.2. What is the process for handling the agreement if the controllable-context is invalidated before the control-agreement expires (if time-based)?
6.3. Should we provide some form of manual override for the owner?
6.4. Consequently, is the fulltime controller always the legitimate party to have full power over a manual override?

7. Compromised: *the terms and conditions of the transient control have been breached during the validity of the controllable-context* - results from state-transition (f)

7.1. Should we in all cases proceed to termination of the agreement and controllable-context? If not, how is this justified?
7.2. How do we adjust the access controls of the system during this state?

8. Terminated: *the controllable-context is gracefully invalidated/ expired*

8.1. How do we completely relinquish transient control of the target domain and how is assurance of this presented?
8.2. Which data and secrets need to be wiped and how is this determined and consequently assured?
8.3. How is the integrity of the agreement validated and asserted and which party is entrusted with validating this?

These questions are followed up by the proposal of a system architecture and operational protocol that correspond with the actors and transient control states identified.

3 System Architecture and Operational Protocol

The actors in authorized transient control have been subtly introduced in the last section. However, this section affirms their inclusion in the architecture and protocol, by first describing their views on the target environment and consequent roles in authorized transient control. There are two primary actors in the transient control of the target environment - the “fulltime controller” (FTC), who is typically the owner or administrator of the target system, and the “transient controller” (ATC). The operational protocol is specified in relation to these two primary roles. In addition, there are two secondary roles that support the initialization, operation and termination of the transient control; these are defined as the “transient control authority” (TCA) and the “transient control monitor” (TCM). The obligations of these actor roles are explained with the aid of fig. 2.

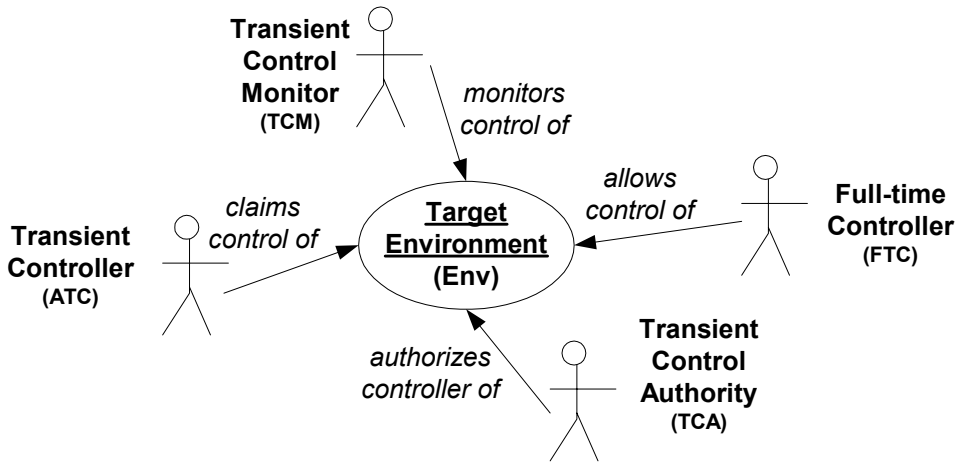

Fig. 2 Use Case Diagram for Authorized Transient Control

The Fulltime Controller (FTC) is relatively, permanently responsible for the target environment. He therefore suffers most loss if the environment is compromised. It is the task of the FTC to setup the permissions and controllable context of the environment.

The (Authorized) Transient Controller (ATC) is becomes an Authorized Transient Controller when he can present the evidence required to claim control of a set of resources in the target environment, during a controllable context.

The Transient Control Authority (TCA) is a role with which the FTC has a pre-established relationship and hence trusts the recommendations of ATCs made by this role. The TCA may also be the first in a chain of other TCA's, similar to a chain of Trusted Certification Authorities in PKI (Public Key Infrastrucure) [21]. The FTC will typically be affiliated with a TCA in that chain.

The Transient Control Monitor (TCM) is initially affiliated with the Target Environment and hence under the administration of the FTC, however, during transient control, it is also under obligation to monitor the agreement on behalf of the ATC. The TCM is responsible for issuing the initial alerts when a controllable context arises, as well as adjusting the relevant state of the target environment when it is under transient control (see fig. 1).

3.1 The Target Environment Management Architecture

To specify the relationships between these actors and the environment, as well as their interactions with each other, Fig. 3 presents the management components proposed to facilitate authorized transient control. There are five components for Resource, Permissions (Controllable Sectors), Context, Evidence and Agreement (Terms & Conditions) management respectively. These are defined below:

Resources Management: this is a service registry for all computational resources of an environment, which have subscribed their service interface to a central management utility. With reference to Fig. 3, R1, R2 and R3 could range from an embedded computer in a window to a telephone to a personal computer. The registry links the resource-specific service description to a standard meta-service interface [17, 19].

Controllable Sectors Management: this is the registry for permissions, where subsets of service interfaces are defined as permissible, in a similar fashion to capability-based access control system [20]. Note in Fig. 3 that the same resource R2 exposes a different interface in CS1 than in CS2. Additionally note that the resource R1 is not offered for transient control within any context.

Context Management: the context of an environment constantly changes, yet different applications or users place different interests in these changes. The Context Management component is capable of aggregating

sensor information from the physical environment, as well as application data, in order to reason about the character of the situation [13]. The TCM is therefore a subscriber to the Context Management as a consumer of the controllable context. Each monitored controllable context is associated with a controllable sector to define the permissions granted to an ATC within that context.

Evidence Management: in addition to the context-to-permissions association, each controllable context is associated with an Evidence-Set, which states the collective evidence (credentials, role-assertion, and trustworthiness measures) that the candidate controller must provide in the "intent" phase before gaining control i.e. becoming an ATC. Dependent on the evidence, the ATC is also assigned a role, which is considered as a context-relevant view of a controllable-sector. In addition, the link between the evidence management and context management allows the target environment to provide context-based proof of validity to the ATC if requested.

Terms & Conditions Management): the final aspect of the environment management is an assignment of terms and conditions that specify the boundaries of the control surrounding the accessible service interfaces. These are defined for both the active and terminated states and referred to as the "control agreement". If the controller breaches the terms and conditions, then the transient control state moves to "compromised". The controller may also present his terms and conditions in the "intent" phase, in which case these are incorporated in the control agreement.

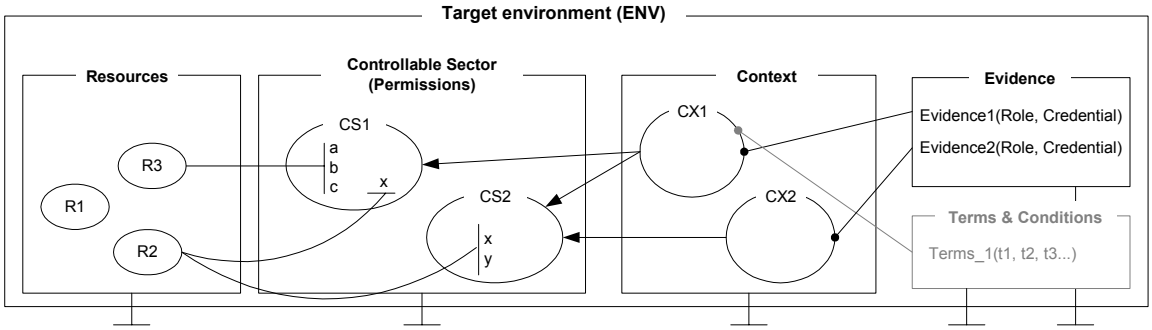

Fig. 3: Environment Management Components of Authorized Transient Control Architecture

3.2. The Operational Protocol

The operational protocol suggests how these management components and actors interact throughout the phases of authorized transient control. It also considers the security mechanisms that are required to facilitate the integrity of the protocol and assertions made by the actors, presented with the aid of an activity diagram. The "Emergency Response" scenario is used as the illustration of the protocol because of the clarity it affords and popularity in pervasive computing circles; nevertheless the other scenarios are also applicable but would require more creative naming and identification of application-specific actors assuming the roles in the protocol. The management components in Fig. 3 are collectively referred to as "the target environment" (Env) in the description of the protocol.

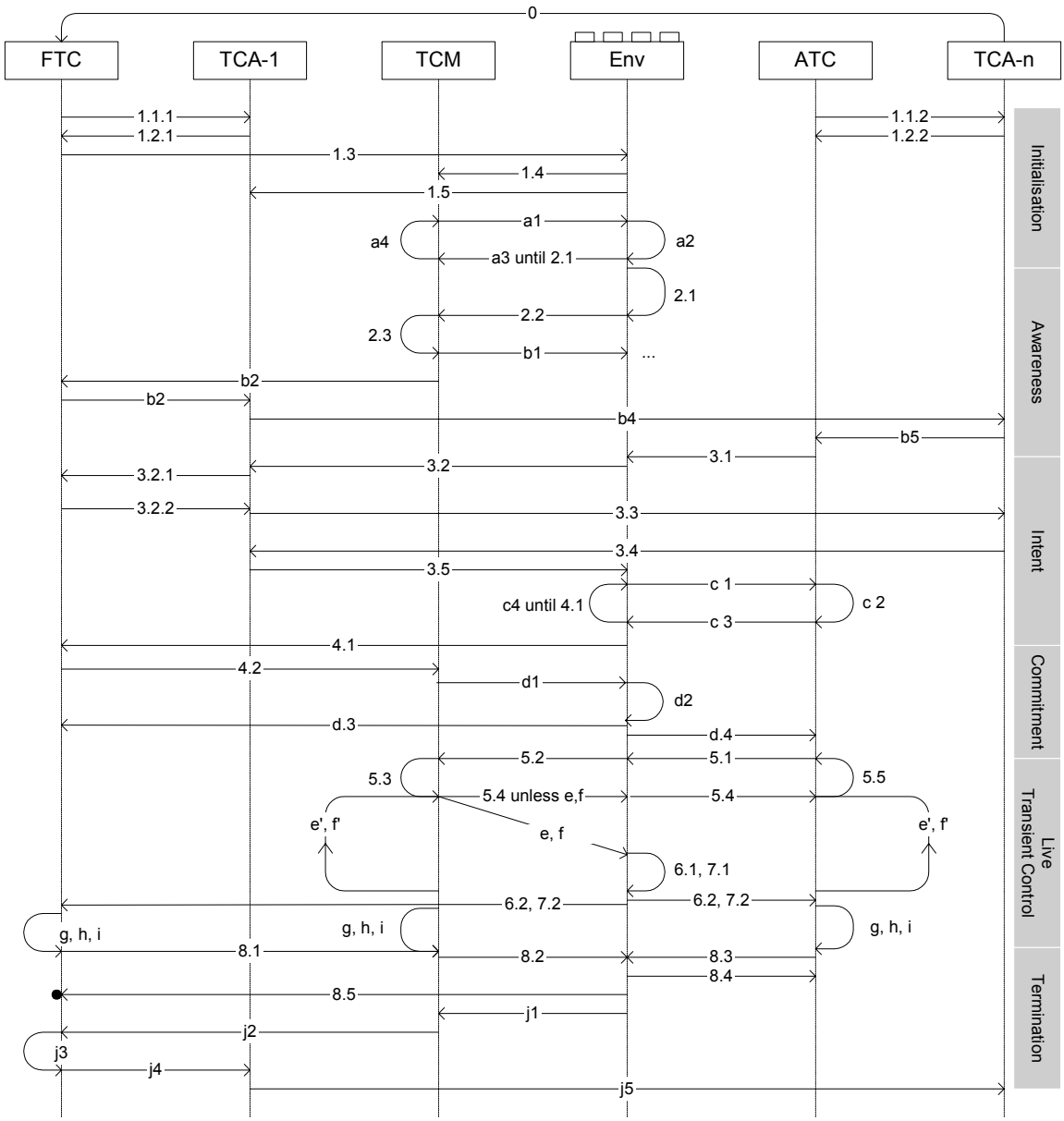

Fig. 4: Activity Diagram of Authorized Transient Control Operational Protocol. The activities correspond with the state-model in fig. 1.

Initialisation: **(0)** there may exist a direct or indirect relationship between the local TCA-1 and the external TCA-n with respect to the domain of the target environment. TCA-n represents a certifiable authority at position n in a chain of trust rooted by TCA-1 from the perspective of the FTC (full time controller) [21]. With reference to the "emergency response" scenario, TCA-n could be a hospital, while TCA-1 would be the FTC's health insurance company. From a technology perspective, TCA-1 could be an application running on the FTC's membership smartcard, bearing a valid certificate and public/private key pair on behalf of the insurance company. **(1.1.1)** The FTC subscribes to a TCA as its local transient control authority (i.e. TCA-1). **(1.1.2)** Asynchronously, a similar subscription process occurs between potential ATCs (authorized transient controllers) and TCAs, possibly in the trust chain of TCA-1. **(1.2.1)** The FTC receives a key pair and/ or a membership agreement from the TCA. **(1.2.2)** ATCs also receive key pairs and agreements from their TCA-n; consider a medical assistant being issued with these articles embedded in his ID card. **(1.3)** Returning focus to the target environment (Env) initialization, the FTC specifies the "control

sectors", "controllable context", "evidence" and "terms and conditions" of Env based on the agreement he received from his TCA as well as personal preferences (see fig. 3). **(1.4)** The context management component of the Env updates the TCM (trusted control monitor) with the controllable context for which the environment is to be monitored. **(1.5)** The Env forwards the evidence and terms and conditions to TCA-1 that will be used to issue yet constrain control, when the controllable context is active and a candidate ATC presents himself. After initialization, there is a constant loop of **(a1)** monitoring by the TCM, **(a2)** the Env updating its "current context" variables, **(a3)** Env sending context information of interest (subscribed) to the TCM, and **(a4)** the TCM making the decision as to if the context is meant to be transiently controllable – this leads to the state of "Awareness".

Awareness: **(2.1)** The Env detects that it is experiencing a context that is labeled as "controllable", marks it as "pending", and therefore **(2.2)** issues this as a priority context to the TCM. **(2.3)** The TCM verifies that this is indeed a "controllable context" and prepares the list of alerts and environment configuration data. **(b1)** The TCM informs the Env to prepare for transient control by updating its configuration data and also **(b2)** sends the first alert to the FTC. Note that the FTC refers simultaneously to the human user as well as the computational device that provides the interface to the management components and stores the private key of the user – consider that in a medical emergency the "controllable context" may include that the user collapsed and is therefore unable to actively participate in sending alerts. **(b3)** The FTC then delegates the task of issuing relevant alerts to the TCA-1, **(b4)** which contacts the range of relevant authorities in its trust chain TCA-n. **(b5)** TCA-n then approves an ATC to respond to the controllable context at Env – such as the hospital dispatching an ambulance crew. Env waits until ATC arrives, presents his credentials and asserts his intent to control.

Intent: **(3.1)** The ATC presents his credentials to Env (as Env provides the only visible interface with which the ATC can directly interact), **(3.2)** and Env forwards these to TCA-1. **(3.2.1, 3.2.2)** Depending on the authentication requirements, TCA-1 may forward the credentials to the FTC for first verification of acceptance. **(3.3)** TCA-1 also verifies the origin and authenticity of the ATC by requesting verification from TCA-n (consequent chained-verification may be incurred). **(3.4)** The TCA-n responds with either a positive verification, a revocation or that the ATC is unknown. **(3.5)** In the case of a positive verification the Env is forwarded the proposed terms and conditions of the ATC, otherwise, the Env is advised to either re-issue the alerts or negotiate with the ATC for further credentials – bear in mind that the controllable context may be expired within this transition. **(c1)** Env then issues an agreement to the ATC, who **(c2)** accepts, rejects or postpones acceptance, but in any event **(c3)** forwards a signed decision to Env. **(c4)** Env carries out a similar decision process until a settlement can be reached and the process of committing to the agreement afforded.

Commitment: **(4.1)** The pending agreement to terms and conditions is forwarded to the FTC to receive a final signature. **(4.2)** The FTC then notifies the TCM that the controllable context has been authorized to become active and **(d1)** that the Env should be monitored for correct configuration according to the agreement – the TCM is therefore bound to act on behalf of the ATC. **(d2)** Env then activates the valid controllable sectors matching the context, and **(d3)** sends a confirmation to the FTC that this has been done, as well as a **(d4)** signed claim of correct configuration to the ATC, allowing him to begin his tasks. Attached to the claim are the service interfaces and descriptions that he can use during the live transient control.

Live Transient Control (Active/ Inactive/ Compromised): **(5.1)** The ATC begins his workflow or task list through the interfaces of the Env presented to him. **(5.2)** Each service request and performance outcome is copied to the TCM and **(5.3)** are certified that they meet the terms and conditions of the active agreement. **(5.4)** The service response is returned to the ATC and status of the agreement is asynchronously forwarded as well. There are four agreement states: "ACTIVE" if the agreement and controllable context remain valid, "INACTIVE" if there is some technical difficulty that does not breach the agreement but hinders the usability of the services, "COMPROMISED" if the agreement its breached, and "EXPIRING" if the absolute values of the controllable context variables (time being the most intuitive variable in this case) gracefully decay but remain within the relative boundaries of the context. For the cases of "INACTIVE" and "COMPROMISED", the TCM issues a high-priority notification **(e, f)** to the Env to attempt resolution of these non-productive states. **(6.1, 7.1)** Env first checks if there exists rules for automatic reconfiguration

based on the current-state of the variables of the controllable context. **(6.2, 7.2)** It also issues notifications to the FTC and ATC to react to the non-productive state. **(e', f')** If these states are resolvable the agreement state is returned to "ACTIVE", otherwise, **(g, h, i)** the agreement state goes to "EXPIRING" and proceeds to terminate the transient control.

Terminated: **(8.1)** The FTC confirms the termination of transient control, **(8.2)** which results in the control sectors, providing the ATC with access to Env, being closed. **(8.3)** The ATC acknowledges the termination of the agreement by requesting a certification that his tasks were completed. **(8.4)** The Env provides this certification for the ATC, **(8.5)** as well as for the FTC. **(j1)** The TCM then records the termination context of the Env and **(j2)** sends this appended to the controllable context logs to the FTC. **(j3)** The FTC creates a recommendation for the ATC based on the log, signs it and **(j4)** sends it to the TCA-1. **(j5)** TCA-1 then distributes the recommendation along the trusted authority chain to TCA-n, leading to a re-initialization of the Env.

4 Related Work and Conclusion

The concept of secure transient associations in was first expressed in Stajano's Resurrecting Duckling [4] protocol for ad hoc computing. It was through reading this reference that the idea for investigating transience a bit further in the domain of ubiquitous computing arose. The pragmatic idea of transient control is not entirely new if we consider systems that manage time-sharing networked computers (from the time of the mainframe) [8] and time-based access control, such as the systems used in Internet cafes and other public terminals [2]. Nevertheless, although the principles are somewhat similar, in that a subject is granted privileged access to a resource for a specified time, transience as explored in this paper is not solely based on time, rather it is based on the expiry of a more general context or situation. Corner and Noble also discuss "transient authentication" where there argument is that permanent authentication is not practical for mobile and ubiquitous computing [14] – transient authentication is facilitated by a hardware token constantly authenticating the user's presence or proximity. Proximity is a candidate context to be monitored by the TCM.

This leads to a review of the work in "Context-Aware Access Control". Perhaps the first person to complete a full thesis in this area has been Michael Covington at Georgia Tech, with his work on Environment Roles [11]. He suggested an extension to the accepted RBAC (Role Based Access Control) model [15] for managing permissions, by also including a dynamic context-derived role that regulates the static permission granted to a requesting subject. Most other work in this area have also presented architectures along the same lines, where they endeavour to make security a more adaptive part of a computational system, in the spirit of real-world protection goals [5]. What this paper seeks to contribute is more emphasis on the operation and management of systems using context in security management, as opposed to the architectural aspects.

Giving up control to a stranger is always an extreme case of trust. The state-of-the-art in trust management is typically based on PKI methodologies, where either an established certification authority (CA) acts as a hierarchical authority on the trustworthiness of a principal, or this is facilitated by more lateral networks i.e. webs of trust [7]. As the scenarios used in this paper suggest pre-existing relationships between TCAs, either of these trust models would be applicable in authorizing transient control. At another system level, the TCPA (Trusted Computing Platform Alliance) supports a standard fro measuring the trustworthiness of a computer system and its software, also using public key technology [16]. The system proposed in this paper must also support a lower level measuring of trustworthiness, as the TCM certifies that the system has been correctly configured to the FTC and ATC.

The term "transients", "transient control" and "transient controller" have established meaning in the field of adaptive industrial control systems, such as nuclear reactors, where the system-state is reconfigured in response to a system fail-state in spite of the fail-state being resolved before the reconfiguration procedure is completed. This is a result of only the steady-state being considered and the system state variables being changed after adaptive reconfiguration has already occurred [6, 10]. In [6] the "transient controller" is introduced as a component injects an "anti-transience" signal to minimize the effects of the reconfiguration transience that may occur. Although a slightly different interpretation of "transience" is applied in this

paper – considering that a state model is the basis of the protocol design - the issues of transients occurring in state-switching and reconfiguration will still need to be investigated.

Acknowledgements. My colleagues at TecO and SAP Corporate Research for their useful feedback during an internal workshop presentation. The WiTness project has also been a great means of generating ideas and reviewing the state-of-the-art in pervasive applications (IST 2001 32275, www.wireless-trust.org).

References

1. Advantage Business Services, “Advantage Business Services provides a virtual office environment with complete administrative services”. URL: http://www.advantagebusinessservices.com/services.html
2. CyberCaféPro. "monitors the actions of up to 200 Client Computers while controlling important aspects of any cafe." URL: www.cybercafepro.com
3. D. A. Norman. "The Invisible Computer". MIT Press, Cambridge, MA, 1998.
4. F. Stajano. The Resurrecting Duckling -- what next? In Proceedings of the 8th International Workshop on Security Protocols, April 2000.
5. Ghita Kouadri Most´efaoui1 and Patrick Br´ezillon2, A Generic Framework for Context-Based Distributed Authorizations, Springer, LNAI2680, pp. 204–217, 2003
6. Gyula Simon, Tamás Kovácsházy, Gábor Péceli, "Transient Management in Reconfigurable Control Systems," Technical Report, Budapest, Hungary, 2002
7. Herzberg, Y. Mass, J. Mihaeli, D. Naor, and Y. Ravid. "Access control meets public key infrastructure, or: Assigning roles to strangers." In IEEE Symposium on Security and Privacy, pages 2--14, 2000.
8. John McCarthy. "Reminiscences on the History of Time Sharing". Stanford University 1983 Winter or Spring URL: http://www-formal.stanford.edu/jmc/history/timesharing/timesharing.html
9. Kurt Seifried. "Firewall overview" URL: http://www.seifried.org/security/network/firewall/20011025-firewall-overview.html. Last updated 25-10-2001
10. Liberzon A. S. and Morse A. S. "Basic problems in stability and design of switched systems"[J], IEEE Contr. Syst. Mag., 1999: 19(5): pp.59-70
11. M. J. Covington, P. Fogla, Z. Zhan and M. Ahamad, "A Context-aware Security Architecture for Emerging Applications", Annual Computer Security Applications Conference (ACSAC), December 2002.
12. M. Weiser. "The computer for the 21st Century". Scientific American 265(3): 66-75, 1991.
13. Michael Beigl, Tobias Zimmer, Albert Krohn, Christian Decker and Philip Robinson, "Smart-Its - Communication and Sensing Technology for UbiComp Environments" Technical Report ISSN 1432-7864 2003/2.
14. Noble, Corner. "The case for transient authentication". Presented at the 10th ACM SIGOPS European Workshop, September 2002
15. Sandhu, R., Coyne, E.J., Feinstein, H.L. and Youman, C.E. (1996). Role - based access control models. IEEE Computer, 29(2), pp. 38-47.
16. TCPA (Trusted Computing Platfor Alliance). "HW and OS based trusted computing platform that implements trust into client, server, networking, and communication platforms." URL: http://www.trustedcomputing.org/home
17. The UPnP Forum. "an industry initiative designed to enable simple and robust connectivity among stand-alone devices and PCs from many different vendors". URL: www.upnp.org
18. Tim Kindberg, Kan Zhang & Narendar Shankar, Context authentication using constrained channels, In proceedings WMCSA 2002
19. Web Services Description Language (WSDL). "an XML format for describing network services". URL: http://www.w3.org/TR/wsdl
20. J. S. Shapiro, J. M. Smith, and D. J. Farber. "EROS: a fast capability system." ACM Symposium on Operating Systems Principles (SOSP'99), pages 170--185. 1999
21. R. Perlman. An Overview of PKI Trust Models. IEEE Network, 13(6):38--43, 1999