Encyclopedia of Mobile Computing and Commerce

David Taniar
Monash University, Australia

Information Science REFERENCE

INFORMATION SCIENCE REFERENCE
Hershey · London · Melbourne · Singapore

Acquisitions Editor:	Kristin Klinger
Development Editor:	Kristin Roth
Senior Managing Editor:	Jennifer Neidig
Managing Editor:	Sara Reed
Assistant Managing Editor:	Diane Huskinson
Copy Editor:	Maria Boyer and Alana Bubnis
Typesetter:	Diane Huskinson
Support Staff:	Sharon Berger, Mike Brehm, Elizabeth Duke, and Jamie Snavely
Cover Design:	Lisa Tosheff
Printed at:	Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of Idea Group Inc.)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@idea-group.com
Web site: http://www.idea-group-ref.com

and in the United Kingdom by
Information Science Reference (an imprint of Idea Group Inc.)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: http://www.eurospanonline.com

Library of Congress Cataloging-in-Publication Data

Encyclopedia of mobile computing and commerce / David Taniar, editor.
p. cm.
Summary: "Nowadays, mobile communication, mobile devices, and mobile computing are widely available. The availability of mobile communication networks has made a huge impact to various applications, including commerce. Consequently, there is a strong relationship between mobile computing and commerce. This book brings to readers articles covering a wide range of mobile technologies and their applications"--Provided by publisher.
Includes bibliographical references and index.
ISBN 978-1-59904-002-8 (hardcover) -- ISBN 978-1-59904-003-5 (ebook)
1. Mobile computing--Encyclopedias. 2. Mobile communication systems--Encyclopedias. 3. Mobile commerce--Encyclopedias. I. Taniar, David.
QA76.59.E47 2007
004.16503--dc22
2006039745

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this encyclopedia set is new, previously-unpublished material. The views expressed in this encyclopedia set are those of the authors, but not necessarily of the publisher.

Distributed Heterogeneous Tracking for Augmented Reality

Mihran Tuceryan
Indiana University Purdue University Indianapolis, USA

Rajeev R. Raje
Indiana University Purdue University Indianapolis, USA

INTRODUCTION

Augmented reality (AR) is a technique in which a user's view of the real world is enhanced or augmented with additional information generated from a computer model (Azuma et al., 2001). The enhancement may consist of virtual artifacts to be fitted into the environment or a display of non-geometric information about existing real objects. Mobile AR (MAR) systems implement this interaction paradigm in an environment in which the user moves, possibly over wide areas (Feiner, MacIntyre, Hoellerer, & Webster, 1997). This is in contrast to non-mobile AR systems that are utilized in limited spaces such as a computer-aided surgery or by a technician's aid in a repair shop. There are a number of challenges to implementing successful AR systems. These include a proper calibration of the optical properties of cameras and display systems (Tuceryan et al., 1995; Tuceryan, Genc, & Navab, 2002), and an accurate registration of three-dimensional objects with their physical counterparts and environments (Breen, Whitaker, Rose, & Tuceryan, 1996; Whitaker, Crampton, Breen, Tuceryan, & Rose, 1995). In particular, as the observer (or an object of interest) moves over time, the 3D graphics need to be properly updated so that the realism of the resulting scene and/or alignment of necessary objects and graphics are maintained. Furthermore, this has to be done in real time and with high accuracy. The technology that allows this real-time update of the graphics as users and objects move is a *tracking system* that measures the position and orientation of the tracked objects (Koller et al., 1997). The ability to track objects, therefore, is one of the big challenges in MAR systems. This article describes a software framework for realizing such a distributed tracking environment by discovering independently deployed, possibly heterogeneous trackers and fusing the data from them while roaming over a wide area. In addition to the MAR domain, this kind of a tracking capability would also be useful in other domains such as robotics and location-aware applications. The novelty of this research lies in the amalgamation of the theoretical principles from the domains of AR/VR, data fusion, and the distributed software systems to create a sensor-based, wide-area tracking environment.

BACKGROUND

Although a few approaches for tracking have been proposed (e.g., Hightower & Borriello, 2001; Koller et al., 1997; Neumann & Cho, 1996; State, Hirota, Chen, Garrett, & Livingston, 1996), the ability to track objects accurately and in real time over a *wide area* does not yet have a satisfactory solution. Moreover, many of these approaches require a highly engineered environment with a uniform set of trackers whose architecture is known in advance (Welch, & Foxlin, 2002; Ubisense, 2006). Assuming that trackers have been deployed and are operating and exist in the environment, this research deals with questions of how to discover what trackers exist in a local area, what quality-of-service (QoS) properties they have, and how to make the best use of their measurements in a mobile and dynamic environment.

The wide-area, ubiquitous tracking that is the focus of this article has been addressed mainly in the pervasive/ubiquitous computing community. An early tracking system was HiBall that utilized a ceiling instrumented by LED lights (Welch, 1999). The HiBall tracker covered a wider area than a typical magnetic tracking system, and in the implementation its range covered a room or a lab. The scalability of such a system was limited because of the increased cost of extending beyond the size of a lab. The BAT system, which used ultrasound as the core technology (Harter, Hopper, Steggles, Ward, & Webster, 2002; Newman, Ingram, & Hopper, 2001), had a limited resolution. The location sensing system, by Ubisense (2006) uses the ultra-wide-band technology and has a better resolution (6 inches positional accuracy, according to company Web sites).

Researchers at Intel Research studied the use of existing wireless hotspots and cell phone towers to compute location information over wide-areas (Schilit et al., 2003; Borriello, Chalmers, LaMarca, & Nixon, 2005). Bahl and others studied localization techniques using existing Wi-Fi wireless hubs (Bahl & Padmanabhan, 2000; Balachandran, Voelker, & Bahl, 2003). Their methods assume a ubiquitous infrastructure that exists for other purposes (networking) that can be tapped into for localization of users. Typically, their resolution tends to be low and is not sufficient for typical AR applications.

Ubiquitous tracking systems specifically for AR systems have been also studied by Bauer et al. (2002). Newman et al. (2004), and Reitmayr (2001), and have resulted in prototypical systems, some of which are component based (e.g., DWARF by [Bauer, Bruegge, Klinker, MacWilliams, Reicher, Riss, et al., 2001]).

THE UNIFRAME-BASED MOBILE TRACKING SERVICE FOR AR

As indicated earlier, the distributed tracking system is an example of a heterogeneous, distributed computing system. The overall architecture and various components of the distributed tracker subsystem that is the focus of this article are shown in Figure 1.

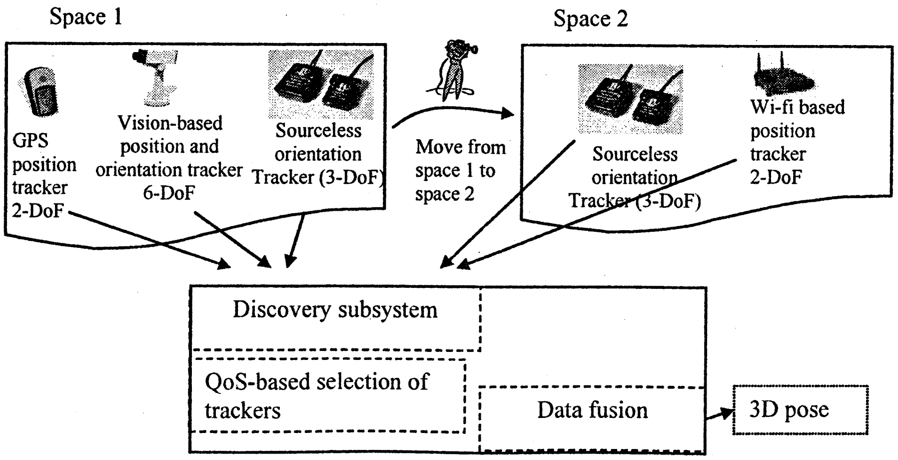
The software realization of this tracking system is based on the principles of uniframe (Olson, Raje, Bryant, Burt, & Auguston, 2005). Uniframe provides an environment for an interoperation of heterogeneous and distributed software components, and uses the principles a meta-component model, service-oriented architectures, generative programming, and two-level (TLG) and event grammars (EG). The realization of the distributed tracking system, using the UniFrame, begins with a generative domain model (GDM) (Czarnecki & Eisenecker, 2000) created by experts from the tracking domain. This GDM contains various details, such as the software architecture of the tracking system expressed in terms of underlying components, their interactions, the rules for generating middleware, and the rules for the prediction and monitoring of the quality of the integrated system. Each component is defined by a Unified Meta-component Model (UMM) specification (Raje, 2000). The UMM has three parts: (a) components; (b) services, offered by components, and associated guarantees; and (c) infrastructure for deploying and discovering components. A developer who wishes to create specific components for the tracking system consults the GDM and creates implementations using the UMM specifications encoded there. After a component is developed, it is validated against the quality requirements, both functional and QoS. The developer also creates an associated UMM specification for that component. This specification and the component are deployed on the network. These components are also registered with the uniframe resource discovery system (URDS) (Siram, 2002).

A system integrator planning to create the tracking system, from independently developed and deployed components, issues a query consisting of the requirements the tracking system must meet. The query processing consults the GDM, divides the query into many sub-queries, each corresponding to a single component UMM specification. These sub-queries are passed to the URDS, which searches for appropriate matching components. If such components are found, these are returned to the system integrator, who then selects a subset of these results, provides any proprietary components, and requests the process to assemble the integrated system conforming to the design. The uniframe system generator (Huang, 2003) carries out the generation of the integrated system. The key challenges in creating the tracking system are: (a) designing the GDM, (b) the discovery of components, and (c) the generation of a prototypical tracking system. These are briefly discussed below.

Designing the GDM

The GDM is developed by the domain experts and contains the software architecture of the family of systems, along with many associated details. For a tracking system, it can be either handcrafted or generated via the uniframe system generator (Huang, 2003). One important piece of the GDM relates to the specification of components that make the software architecture of the tracking system. The specification provides

Figure 1. Architecture and components of the distributed tracking subsystem for mobile AR applications



the necessary details during the discovery process and the system generation process. The approach for specifying the components is indicated as follows.

UMM-Specifications

Each sensing device used in the tracking system is represented by a corresponding software component that encapsulates its behavior. For example, a GPS sensor in the tracking system is encapsulated as a component offering a service that provides 3DOF position information with certain accuracy. As indicated earlier, each component in uniframe has an associated UMM specification. This specification contains many attributes (Raje, 2000) that reflect various details related to that component. In the context of the tracking system, the functional and the QoS attributes of a component are the most important ones. For example, a partial specification (for the sake of brevity) for an Inertia Cube Tracker component is:

Component Name: InertiaCubeTracker
Domain Name: Distributed Tracking
Informal Description: Provides the orientation information.
Computational Attributes
Inherent Attributes:
Id: cs.iupui.edu/InertiaCubeTracker;
...
Validity: 12/1/07 Registration: pegasus.cs.iupui.edu/HH1
Functional Attributes:
Functional Description: Provides the orientation of a tracked object.
Algorithm: Kalman Filter;
Complexity: O (n^3)
Syntactical Contract:
Vector getOrientation();
Semantic Contract:
Pre-condition: {calibrated (InertiaCubeTracker)== true}
Post-condition: {sizeof (orientationVector) == 3)}
Synchronization Contract:
Nature: Multi-threaded Synchronization Policy
Implementation Mechanism: semaphore
Technology: CORBA
....
Quality of Service Attributes
QoS Metrics: resolution, drift, lag-time,
resolution: 0.1 degrees; drift: 0.01 degrees/sec; lag-time: 1 ms
....

The preceding partial specification shows many important characteristics of the UMM-specification. These are: (a) the specification is comprehensive and highlights many aspects of a component; (b) the specification is an enhanced realization of the concept of multi-level contracts (Beugnard, Jezequel, Plouzeau, & Watkins, 1999), thus the specification not only describes the functional aspects, but also emphasizes the QoS attributes of a component; and (c) the specification is consistent with the concepts of a service-oriented view of software components. The detailed nature of the specification provides sufficient information for the discovery of components to create the tracking system.

Discovery Process

In a distributed tracking environment, it is conceivable that there would be many different software instances offering similar types of services. For example, there may be multiple trackers (each encapsulating a different inertia tracker) implemented in a variety of ways and possibly offering different qualities of the tracking results. Hence, to realize such a tracking system, it is necessary to discover various alternatives available over the network. That is the role of the discovery service. The discovery process is realized using the principles of the URDS (Siram, 2002). The URDS is a hierarchical, proactive, and interoperable discovery service. The components are selected based on their type and the QoS (e.g., resolution) values for a specific tracking system.

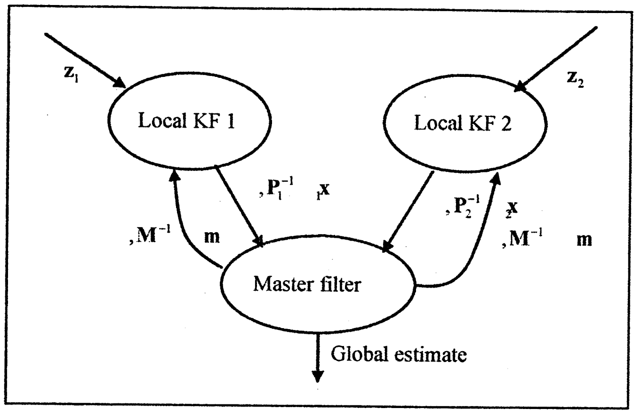
Generation of the Prototypical System

Uniframe uses the principles of glues of wrappers that are generated using the concepts of TLG and EG. The purposes of the glue and wrapper code are to allow an interoperation between heterogeneous components and also to insert any instrumentation code that can collect event traces to observe the actual QoS values during the execution of the system. A system generator (Huang, 2003) accepts the selected components as the input and uses the information in the GDM to semi-automatically create the distributed system. The rules for the generation are a part of the GDM and are developed during the formalization of the GDM. Another important facet of the generation process is the ability to make static predictions (based on the QoS properties of the selected components) and compare them against the actual execution results. The process of prediction is based on the principles of sensor fusion, as described in the following section.

DATA FUSION

Once the individual tracker components are selected through the discovery process described earlier, their results need to be combined (fused) in order to get the best estimate of the position and orientation of the tracked object. The usual

Figure 2. Federated Kalman filter architecture (Carlson, 1990)



Note: Here the $\hat{x}_i$ and M_i are the state vector and covariance matrices for the local filters. The z_is are the measurements for each sensor i. The quantities ***m*** *and* ***M*** *are the global state vector and covariance matrix estimated by the master filter, respectively. In this figure, the global estimates are fed back to the local filters.*

framework for fusing multi-modal sensor data for tracking is the various modifications and extensions of the Kalman filter (Brown & Hwang, 1997; Kalman, 1960; Welch & Bishop, 2001). The Kalman filter is a recursive estimation method that tries to estimate the state of a discrete-time controlled process (i.e., the pose of a tracked object, possibly with velocity and acceleration information) using observable measurements (i.e., data from tracker sensors). The state is estimated in each time step by a set of update equations in the form of "predict" and "correct" cycle. The typical Kalman filter formulation requires that all the state variables for the available devices and the relevant equations be set up globally at the beginning. Given the dynamic and distributed nature of the framework described in this article, this approach is not practical. Instead, the federated Kalman filter first described by Carlson (1990) is to be used. In this framework, the sensors have their own local Kalman filter running that estimates the local state and covariance. Then for each object that is tracked, there is a master Kalman filter that uses the estimates coming from the local Kalman filters and computes a global estimate of the state and covariance. The results of the master filter can be fed back to the local filters to improve their local state estimates also. The rough architecture of such a federated Kalman filter is shown in Figure 2.

The federated Kalman filter allows for the assembling of a master filter depending on the new set of sensors found through the discovery process. The covariance matrix is a measure of the error in the state information and can be used as part of the QoS information.

FUTURE TRENDS

The future plans include testing the methods by an integrated system that consists of multiple trackers distributed over a sufficiently large area. A variety of trackers such as vision-based trackers (e.g., AR Toolkit), Wi-Fi based trackers, magnetic trackers, and inertial trackers will be utilized in testing this prototype system. Application prototypes will be created to show the effectiveness of the proposed research. These prototypes will act as the test-beds and will provide feedback to the principles of the proposed discovery service.

CONCLUSION

The software framework outlined in this article is a promising approach to developing practical wide-area tracking systems that can utilize existing tracker infrastructures. The limitations of the framework are mainly due to the hardware. For example, one cannot foresee all the possible trackers and thus equip the tracked object accordingly ahead of time. In order to accommodate this, the framework assumes multiple, heterogeneous *classes* of tracking systems rather than instances of trackers. An example of tracker class might be a vision-based tracker. Thus, there may be many instances of such trackers, but one needs to equip the object with a standard fiducial marker.

REFERENCES

Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., & MacIntyre, B. (2001). Recent advances in augmented reality. *IEEE Computer Graphics & Applications, 21*(6), 34-47.

Bahl, P., & Padmanabhan, V. (2000). RADAR: An in-building RF-based user location and tracking system. *Proceedings of the IEEE Infocom 2000* (pp. 775-784). IEEE CS Press.

Balachandran, A., Voelker, G., & Bahl, P. (2003). Wireless hotspots: Current challenges and future directions. *Proceedings of the 1st ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots* (pp. 1-9). ACM Press.

Bauer, M., Bruegge, B., Klinker, G., MacWilliams, A., Reicher, T., Riss, S., Sandor, C., & Wagner, M. (2001) Design of a component-based augmented reality framework, In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR 2001)* (pp.45-54).

Bauer, M., Bruegge, B., Klinker, G., MacWilliams, A., Reicher, T., Sandor, C., & Wagner, M. (2002). An architecture concept for ubiquitous computing aware wearable computers. *Proceedings of the International Workshop on Smart Appliances and Wearable Computing.*

Beugnard, A., Jezequel, J., Plouzeau, N., & Watkins, D. (1999). Making components contract aware. *IEEE Computer, 32*(7), 38-45.

Borriello, G., Chalmers, M., LaMarca, A., & Nixon, P. (2005). Delivering real-world ubiquitous location systems. *Communications of the ACM, 48,* 36-41.

Breen, D., Whitaker, R., Rose, E., & Tuceryan, M. (1996). Interactive occlusion and automatic object placement for augmented reality. *Proceedings of the Eurographics '96 Conference* (pp. 11-22).

Brown, R., & Hwang, P. (1997). *Introduction to random signals and applied Kalman filtering* (3rd ed.). New York: John Wiley & Sons.

Carlson, N. A. (1990). Federated square root filter for decentralized parallel processes. *IEEE Transactions on Aerospace and Electronic Systems, 26*(3), 517-525.

Czarnecki, K., & Eisenecker, U. (2000). *Generative programming: Methods, tools, and applications.* Boston: Addison-Wesley.

Feiner, S., MacIntyre, B., Hoellerer, T., & Webster, T. (1997). A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. *Proceedings of the International Symposium on Wearable Computers (ISWC'97)* (pp. 74-81).

Foxlin, E., & Durlach. N. (1994). An inertial head-orientation tracker with automatic drift compensation for use with HMDs. *Proceedings of the VRST '94 Conference* (pp. 159-173), Singapore.

Harter, A., Hopper, A., Steggles, P., Ward, A., & Webster, P. (2002). The anatomy of a context-aware application. *Wireless Networks, 8*(2-3), 187-197.

Hightower, J., & Borriello, G. (2001). Location systems for ubiquitous computing. *Computer, 34*(8), 57-66.

Huang, Z. (2003). *The uniframe system-level generative programming framework.* Unpublished MS thesis, Department of Computer and Information Science, Indiana University Purdue University Indianapolis, USA. Retrieved May 11, 2006, from http://www.cs.iupui.edu/uniFrame/

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering,* 35-45.

Koller, D., Klinker, G., Rose, E., Breen, D., Whitaker, R., & Tuceryan, M. (1997). Real-time vision-based camera tracking for augmented reality applications. *Proceedings of the Symposium on Virtual Reality Software and Technology* (pp. 87-94).

Neumann, U., & Cho, Y. (1996). A self-tracking augmented reality system. *Proceedings of the ACM Symposium on Virtual Reality and Applications* (pp. 109-115).

Newman, J., Ingram, D., & Hopper, A. (2001). Augmented reality in a wide area sentient environment. *Proceedings of the International Symposium on Augmented Reality* (pp. 77-86), New York. IEEE Press.

Newman, J., Wagner, M., Bauer, M., MacWilliams, A., Pintaric, T., Beyer, D., et al. (2004). Ubiquitous tracking for augmented reality. *Proceedings of the 3rd IEEE and ACM International Symposium on Mixed and Augmented Reality* (pp. 192-201).

Olson, A., Raje, R., Bryant, B., Burt, C., & Auguston, M. (2005). UniFrame: A unified framework for developing service-oriented, component-based, distributed software systems. In Z. Stojanovic & A. Dahanayake (Eds.), *Service oriented software system engineering: Challenges and practices* (Chapter IV, pp. 68-87). Hershey, PA: Idea Group Inc.

Raje, R. (2000). UMM: Unified Meta-object Model for open distributed systems. *Proceedings of the 4th IEEE International Conference on Algorithms and Architecture for Parallel Processing* (pp. 454-465). IEEE Press.

Reitmayr, G., & Schmalstieg, D. (2001). An open software architecture for virtual reality interaction. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (pp. 47-54). ACM Press.

Schilit, B.N., LaMarca, A., Borriello, G., Griswold, W.G., McDonald, D., Lazowska, E., et al. (2003). Challenge: Ubiquitous location-aware computing and the "Place Lab" initiative. *Proceedings of the 1st ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots* (pp. 29-35). ACM Press.

Siram, N. (2002). *An architecture for discovery of heterogeneous software components.* Unpublished MS thesis, Department of Computer and Information Science, Indiana University Purdue University Indianapolis, USA. Retrieved May 11, 2006, from http://www.cs.iupui.edu/uniFrame/

State, A., Hirota, G., Chen, D., Garrett, W., & Livingston, M. (1996). Superior augmented reality registration by integrating landmark tracking and magnetic tracking. *Proceedings of SIGGRAPH '96* (pp. 429-438).

Tuceryan, M., Greer, D., Whitaker, R., Breen, D., Crampton, C., Rose, E., & Ahlers, K. (1995). Calibration requirements and procedures for a monitor-based augmented reality system. *IEEE Transactions on Visualization and Computer Graphics, 1*(3), 255-273.

Tuceryan, M., Genc, Y., & Navab, N. (2002). Single Point Active Alignment Method (SPAAM) for optical see-through (HMD) calibration for augmented reality. *Presence: Teleoperators and Virtual Environments, 11*(3), 259-276.

Ubisense. (2006). Retrieved May 3, 2006, from http://www.ubisense.net/

Welch, G., & Bishop, G. (2001). An introduction to the Kalman filter. *Proceedings of the Siggraph Course,* Los Angeles.

Welch, G., & Foxlin, E. (2002). Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Computer Graphics and Applications, 22*(6), 24-38.

Whitaker, R., Crampton, C., Breen, D., Tuceryan, M., & Rose, E. (1995). Object calibration for augmented reality. *Proceedings of Eurographics'95* (pp. 15-27).

KEY TERMS

Augmented Reality: Superimposing information on the view of the physical world for the purposes of providing information.

Degrees-of-Freedom (DOF): For a tracker of a particular type, the number of independent dimensions of information obtained from the sensor hardware.

Kalman Filter: A linear estimation technique first proposed by Rudolph Kalman in 1960 that is extensively used in tracking and navigation applications.

Tracking: A hardware/software system that can provide the position and/or orientation of an object being tracked in real time.

UniFrame: A unifying framework that supports a seamless integration of distributed and heterogeneous components.

UniFrame Resource Discovery System (URDS): A system that provides an infrastructure for proactively discovering components deployed over a network.