

Bless, Roland; Hübsch, Christian; Mayer, Christoph P.; Mies, Sebastian; Waldhorst, Oliver P.; Zitterbart, Martina:

ariba: Rahmenwerk für Overlay-basierte Dienste

URN: urn:nbn:de:gbv:ilm1-2015210332

Published OpenAccess: January 2015

Original published in:

Praxis der Informationsverarbeitung und Kommunikation : PIK. - Berlin : de Gruyter (ISSN 1865-8342). - 34 (2011) 3, S. 151-155.

DOI: 10.1515/piko.2011.031

URL: <http://dx.doi.org/10.1515/piko.2011.031>

[Visited: 2015-01-20]

„Im Rahmen der hochschulweiten Open-Access-Strategie für die Zweitveröffentlichung identifiziert durch die Universitätsbibliothek Ilmenau.“

“Within the academic Open Access Strategy identified for deposition by Ilmenau University Library.”

„Dieser Beitrag ist mit Zustimmung des Rechteinhabers aufgrund einer (DFG-geförderten) Allianz- bzw. Nationallizenz frei zugänglich.“

„This publication is with permission of the rights owner freely accessible due to an Alliance licence and a national licence (funded by the DFG, German Research Foundation) respectively.“



ariba: Rahmenwerk für Overlay-basierte Dienste



Roland Bless

Institut für Telematik
Karlsruher Institut für Technologie
(KIT)
Zirkel 2
76131 Karlsruhe
Deutschland
bless@kit.edu

PD Dr. *Roland Bless* ist Privatdozent und Senior-Wissenschaftler am Institut für Telematik, Lehrstuhl Prof. Zitterbart. Er studierte Diplom-Informatik an der Universität Karlsruhe (TH) und promovierte bei Prof. G. Krüger im Bereich Dienstgütemanagement. 2009 habilitierte er an der Fakultät für Informatik des KIT. Seit 1998 ist er in der Internet-Standardisierung aktiv tätig und hält regelmäßig Vorlesungen am KIT zu den Themen „Next Generation Internet“ und „Multimediakommunikation“. Dr. Bless ist Mitglied in der GI, ACM SIGCOMM und IEEE ComSoc.



Christian Hübsch

Institut für Telematik
Karlsruher Institut für Technologie
(KIT)
Zirkel 2
76131 Karlsruhe
Deutschland
huebsch@kit.edu

Dipl.-Inform. *Christian Hübsch* schloss sein Informatikstudium mit dem Diplom an der Universität Karlsruhe (TH) 2006 ab. Seitdem arbeitet er als wissenschaftlicher Mitarbeiter am Institut für Telematik des Karlsruher Institut für Technologie (KIT), wo er in den Bereichen Anwendungsschicht-Multicast und Overlay-basierte Ansätze für Netze der Zukunft im Rahmen des BW-Fit Projekts SpoVNet forscht.



Christoph P. Mayer

Institut für Telematik
Karlsruher Institut für Technologie
(KIT)
Zirkel 2
76131 Karlsruhe
Deutschland
mayer@kit.edu

Dipl.-Inform. *Christoph P. Mayer* erhielt 2007 sein Informatikdiplom von der Universität Karlsruhe (TH). Seit 2008 arbeitet er als

wissenschaftlicher Mitarbeiter am Institut für Telematik des Karlsruher Institut für Technologie (KIT). Er beschäftigt sich hauptsächlich mit Overlay-basierten und DTN-basierten Architekturen für Netze der Zukunft im Rahmen des BW-Fit Projekts SpoVNet.



Sebastian Mies

Institut für Telematik
Karlsruher Institut für Technologie
(KIT)
Zirkel 2
76131 Karlsruhe
Deutschland
mies@kit.edu

Dipl.-Inform. *Sebastian Mies* erhielt 2006 sein Informatikdiplom von der Universität Karlsruhe (TH). Seitdem arbeitet er als wissenschaftlicher Mitarbeiter am Institut für Telematik des Karlsruher Institut für Technologie (KIT). Er beschäftigt sich dort hauptsächlich mit der Kopplung heterogener Netze sowie dem Routing in spontanen/ad-hoc Netzen im Rahmen des BW-FIT Projekts SpoV-Net.



Oliver P. Waldhorst

Institut für Telematik
Karlsruher Institut für Technologie
(KIT)
Zirkel 2
76131 Karlsruhe
Deutschland
waldhorst@kit.edu

PD Dr. rer. nat. habil. *Oliver P. Waldhorst* erhielt die Abschlüsse Diplom-Informatiker im Jahr 2000 und Dr. rer. nat. im Jahr 2005, beide von der Universität Dortmund. Die Fakultät für Informatik des Karlsruher Institut für Technologie (KIT) habilitierte ihn im Februar 2011. Er leitet seit August 2007 eine Nachwuchsgruppe am KIT, die aus Mitteln der Exzellenzinitiative sowie aus Drittmitteln finanziert wird. Während seiner Promotion arbeitete er als Wissenschaftlicher Mitarbeiter an der Universität Dortmund (April 2000–Dezember 2005) und nach erfolgreichem Abschluss als Wissenschaftlicher Mitarbeiter an der Universität Leipzig (Januar 2006–August 2006). Von September 2009 bis Februar 2010 besuchte er die University of Toronto als Visiting Researcher in der Gruppe von Prof. Jörg Liebeherr. Von April 2011 bis März 2012 vertritt er Prof. Andreas Mitschele-Thiel an der TU Ilmenau.



Martina Zitterbart

Institut für Telematik
 Karlsruher Institut für Technologie
 (KIT)
 Zirkel 2
 76131 Karlsruhe
 Deutschland
 zit@tm.uka.de

Prof. Dr. *Martina Zitterbart* ist Professorin für Informatik am Karlsruher Institut für Technologie (KIT), wo sie das Institut für Informatik leitet. Sie erhielt 1990 ihren Dokortitel von der Universität Karlsruhe (TH) und arbeitete von 1991 bis 1992 als Visiting Researcher im IBM T.J. Watson Research Center in NY, USA. Von 1995 bis 2001 war sie Professorin an der TU Braunschweig. Ihre aktuellen Forschungsinteressen umfassen das Internet der Zukunft, Sensornetze, Netze für Energie sowie Architekturen, Algorithmen und Protokolle für Kommunikationssysteme im Allgemeinen. 2002 erhielt Martina Zitterbart den Alcatel SEL Forschungspreis für Technische Kommunikation.

1 Einleitung

Das *ariba*-Rahmenwerk ermöglicht, was sich Dienst- und Anwendungsentwickler von einem Netz wünschen: Nahtlose Kommunikation, die „einfach funktioniert“. Selbige ist heute durch wachsende Herausforderungen erschwert: Drahtlose Netzzugänge, Mobilität der Teilnehmer, erhöhte Heterogenität von Kommunikationsgeräten wie auch des Netzes, sowie der Einsatz von Zwischensystemen (z. B. Firewalls oder NAT-Gateways) erschweren die Ende-zu-Ende-Kommunikation und somit die Entwicklung neuartiger Dienste und Anwendungen. Für jede dieser Herausforderungen existieren maßgeschneiderte Einzellösungen (z. B. MobileIP, STUN, TURN, ICE, HIP usw.). Diese benötigen jedoch meist eine zusätzliche Unterstützung durch dedizierte Infrastruktur (Herausforderung ①), müssen aufwändig konfiguriert werden und passen sich ändernden Netzbedingungen nicht dynamisch an ②. Zudem lassen sie sich zumeist nicht über heterogene Netze (z. B. reine IPv4- oder IPv6-Netze bzw. verschiedene Netze hinter NAT-Gateways) hinweg betreiben ③.

Um neuartige Dienste und Anwendungen einfach entwickeln und spontan nutzen zu können, wurde im Rahmen des Projekts *Spontane Virtuelle Netze* (SpoVNet) [1, 5] das *ariba*-Rahmenwerk, eine frei verfügbare Software-Bibliothek, entwickelt. *ariba* bietet Dienst- und Anwendungsentwicklern eine einfach zu verwendende und dennoch umfangreiche Schnittstelle, welche den direkten Umgang mit den Herausforderungen ①–③ abnimmt und somit die Entwicklung von Diensten und Anwendungen wesentlich vereinfacht.

ariba verwendet Peer-to-Peer-Techniken (P2P), um nicht auf eine Unterstützung durch dedizierte Infrastruktur angewiesen zu sein. Dabei werden ausschließlich Ressourcen der Teilnehmer genutzt ①. Für jede Instanz einer Anwendung stellt *ariba* ein dediziertes P2P-Overlay bereit, also

ein Netz aus logischen Verbindungen zwischen Endgeräten, dass unabhängig von den darunterliegenden Verbindungen in den physikalischen Netzen (dem Underlay) ist. Dieses P2P-Overlay ermöglicht den Zusammenschluss und Zusammenhalt der teilnehmenden Instanzen zu einem spontanen virtuellen Netz, d. h. die Teilnehmer können sich durch das P2P-Overlay finden, um dann später in direkten Kontakt zu treten. Durch Einsatz von Selbstorganisationsmechanismen wird der Konfigurationsaufwand minimiert ②. Die Struktur des P2P-Overlays passt sich dabei Veränderungen im Netz dynamisch an. Anwendungen müssen daher z. B. Unterbrechungen der Kommunikation und Mobilität der Teilnehmer nicht selbst behandeln. Möchten zwei Anwendungsinstanzen Daten austauschen, erfolgt eine initiale Kontaktaufnahme über das P2P-Overlay, die weitere Kommunikation findet anschließend auf möglichst direktem Wege im Underlay statt. Falls ein unmittelbar direkter Kontakt jedoch nicht möglich ist, stellt *ariba* mittels als „Relay“ fungierender Teilnehmer eine effiziente Erreichbarkeit aller Teilnehmer untereinander her ③, damit Anwendungen über heterogene Netze hinweg durchgängig funktionieren. In Abschnitt 2 wird auf den Aufbau des *ariba*-Rahmenwerks eingegangen. Abschnitt 3 stellt die Schnittstellen zu *ariba* und deren Verwendung vor, bevor abschließend in Abschnitt 4 praxisrelevante Aspekte der Architektur behandelt werden.

2 Rahmenwerk

Abbildung 1 veranschaulicht die Architektur des *ariba*-Rahmenwerks. Die Hauptkomponenten sind die *Base Communication* und das *Base Overlay*, deren Funktionen im Folgenden beschrieben werden.

Base Communication. Die „direkte“ Kommunikation zwischen Teilnehmern wird in *ariba* durch die Base Communication realisiert. „Direkt“ bedeutet hierbei, dass zwei Knoten durch die Verwendung existierender Protokolle und Adressen des Underlays Nachrichten austauschen können. Falls eine direkte Erreichbarkeit zwischen zwei Teilnehmern nicht mög-

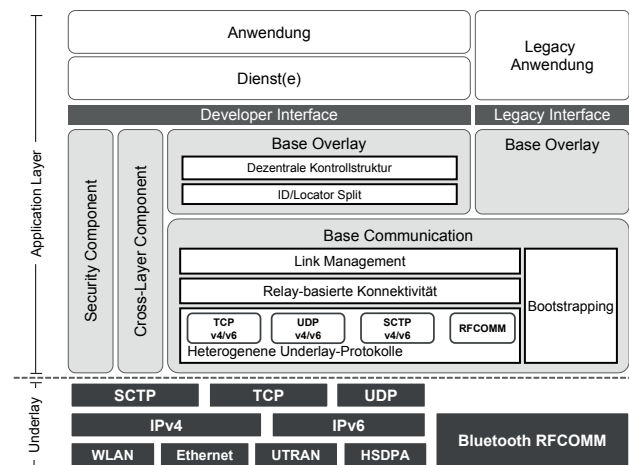


Abbildung 1 Architektur des *ariba*-Rahmenwerks.

lich ist (z. B. durch Protokollheterogenität oder unterschiedliche NAT-Domänen), so findet die Base Communication sogenannte Relay-Pfade, d. h. indirekte Pfade unter Zuhilfenahme anderer Teilnehmer (Relays), über welche die Erreichbarkeit der beiden Teilnehmer hergestellt werden kann. Die Base Communication behandelt so die Problematik der Protokollheterogenität und ermöglicht eine grundlegende Kommunikation zwischen allen Teilnehmern, vorausgesetzt, ein Relaypfad existiert.

Um möglichst viele Kommunikationsprotokolle unterstützen zu können und damit die Anbindung von Teilnehmern an möglichst viele unterschiedliche Netze zu ermöglichen, kann *ariba* durch Transportmodule erweitert werden. In der aktuellen Version bietet es die Unterstützung der Schicht-4-Protokolle TCP und UDP (SCTP befindet sich in Planung), welche sowohl auf den Schicht-3-Protokollen IPv4 und IPv6 aufsetzen können. Weiterhin wird das Schicht-2-Protokoll Bluetooth RFCOMM unterstützt. Ob ein Teilnehmer als Relay genutzt werden kann, bestimmt die Base Communication autonom, d. h. ohne manuelle Konfiguration.

Um Mobilität, Multihoming, Umgehung von NATs, oder Heterogenität zu verbergen, identifiziert die Base Communication eine direkte Kommunikationsbeziehung („Link“) zwischen zwei Teilnehmern mit einer *LinkID*. Diese LinkID bezeichnet einen Link, welcher aus mehreren stückweisen direkten Transportverbindungen im Underlay zusammengesetzt sein kann. Falls sich Netzadressen, z. B. durch Mobilität, ändern, so bleibt diese LinkID identisch.

ariba erhält Links selbst-organisierend aufrecht, führt falls nötig eine Zwischenspeicherung von Nachrichten durch und befreit somit den Anwendungsentwickler von einer komplexen Behandlung zahlreicher Netzwerkprobleme.

Der Base Communication wird zum Aufbau eines Links eine Menge von Underlayadressen eines anderen Teilnehmers (im sogenannten Endpoint Descriptor) übergeben. Die Base Communication liefert daraufhin eine LinkID zurück, welche zum Nachrichtenaustausch genutzt werden kann. Durch Angabe bestimmter Anforderungen an den Link, z. B. ob eine zuverlässige Kommunikation gewünscht ist, kann die Selektion der Protokolle im Underlay beeinflusst werden. Tabelle 1 zeigt einen beispielhaften Endpoint Descriptor mit TCP, UDP, IPv4, IPv6, Bluetooth RFCOMM und Bluetooth MAC Adressen. Zur Identifizierung des lokalen Geräts nutzt die Base Communication eine eindeutige *NodeID*. Diese stellt sicher, dass (z. B. während der Wiederherstellung eines Links) immer das selbe Gerät adressiert wird. Wird der Endpoint Descriptor an die Base Communication zum Aufbau eines Links übergeben, so prüfen die Transportmodule jeweils, mit wel-

NodeID	
c07e00c4535f64a549fd15cf41439127b785710cfa029384	
Endpoint Descriptor	
layer 4	tcp{31016};udp{32020};
layer 3	ip{192.168.178.23 129.13.182.17 2800:1450:8006::68};
layer 2	rfcomm{10};bluetooth{00:26:5e:ab:f9:e7};

Tabelle 1 Beispielhafter Endpoint Descriptor mit NodeID.

chen Adresskombinationen Verbindungen mit den gegebenen Anforderungen hergestellt werden können. Verschiedene Metriken können implementiert werden, aufgrund derer ein Verbindung als dauerhaft gegenüber anderen akzeptiert wird. Aktuell wird eine Metrik verwendet, die denjenigen Link akzeptiert, welcher am schnellsten hergestellt werden konnte.

Ein generelles Problem dezentraler P2P-Overlays ist der initiale Beitritt, das so genannte *Bootstrapping*. Dies erfordert die Auffindung von Teilnehmern welche sich bereits in diesem Overlay befinden. *ariba* bietet hierfür Bootstrap-Module, welche den Einstieg als auch erneutes Verbinden mit dem Overlay nach Ausfall der Netzanbindung eines Teilnehmers unterstützen. Derzeit existieren in *ariba* Bootstrap-Module zum Auffinden über IPv4/v6-UDP-Broadcast, Multicast-basiertes DNS (mDNS) und Bluetooth Service Discovery Protocol. Jedes dieser Module verbreitet den lokalen Endpoint Descriptor des Teilnehmers, welcher von beitretenden Teilnehmern genutzt werden kann, um dem Overlay beizutreten. Weiterhin ist die manuelle Angabe von Bootstrap-Informationen möglich.

Base Overlay. Das *Base Overlay* realisiert ein strukturiertes Overlay, welches *ariba* als grundlegende Kommunikationsstruktur dient. Alle Teilnehmer können mittels des Overlays Kontakt aufnehmen, um anschließend eine direkte Kommunikationsbeziehung in Form eines Links mittels der Base Communication zu etablieren. Das Routing im Overlay wird mittels Key-Based-Routing-Mechanismen (KBR) realisiert. Pro Anwendungskontext wird ein solches Base Overlay aufgebaut. Hiermit wird eine natürliche Trennung verschiedener Anwendungsinstanzen durch jeweils abgeschlossene Kontexte erreicht. Das Base Overlay bietet Anwendungen zwei Konzepte, welche die Entwicklung verteilter Anwendungen vereinfachen: Die bereits in der Base Communication vorgestellten NodeIDs und LinkIDs. Teilnehmer im jeweiligen Overlay-Netz sind über NodeIDs adressierbar, welche (wie auch LinkIDs) invariant bei Mobilität und Multihoming sind ②,③.

Die eingesetzten KBR-Mechanismen im Base Overlay ermöglichen eine einfache Umsetzung zur NodeID-basierten Adressierung und bieten insbesondere die Merkmale Skalierbarkeit, Selbstkonfiguration und Selbstwartung. *ariba* ist auf kein festes KBR-Protokoll im Base Overlay festgelegt. Es kann, abhängig von den Anforderungen der Anwendung, jedes strukturierte P2P-Overlay mit KBR-Funktionalität integriert werden (z. B. Pastry, S/Kademlia). Die aktuelle Implementierung verwendet ein modifiziertes Chord-Protokoll, welches bidirektionale Verbindungen unterhält sowie mit Overlay-Partitionierung umgehen kann.

Einordnung im Protokollstapel. *ariba* als Ganzes setzt auf der Transportschicht des ISO/OSI-Schichtenmodells auf, kann aber auch Protokolle anderer Schichten verwenden (vgl. Abbildung 1). Hierzu zählen verschiedene Transport-, Netzwerk- sowie Schicht-2-Protokolle. Durch die Unterstützung verschiedener Protokolle kann implizit zwischen ihnen vermittelt werden: Wird ein Relay-Pfad über unterschiedliche Protokolle hinweg etabliert, verwendet *ariba* in jedem Pfadabschnitt das hier vorhandene Protokoll und schafft somit

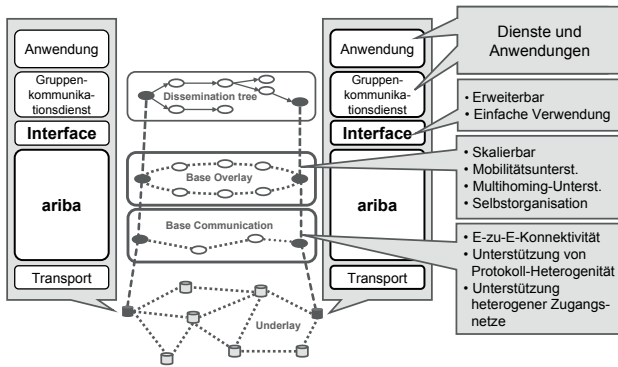


Abbildung 2 Schichten und Komponenten in ariba.

die Ende-zu-Ende-Erreichbarkeit. Die Unterstützung weiterer Protokolle ist flexibel durch Module möglich und benötigt keine weiteren Anpassungen des Rahmenwerks ③.

Abbildung 2 verdeutlicht beispielhaft die Schichtung und Funktionen der vorgestellten Komponenten: In diesem Beispiel verwendet eine Anwendung einen Gruppenkommunikationsdienst, welcher eine eigene, Overlay-basierte Verteilstruktur über *ariba* etabliert. Der Gruppenkommunikationsdienst verwendet für die Kommunikation zwischen Knoten dabei lediglich die angebotenen NodeIDs und LinkIDs des Base Overlay. Letzteres wiederum stellt Verbindungen über die Base Communication her. Den Zugangspunkt zu *ariba* bildet die Schnittstelle (*Interface* in der Abbildung). Ihr Aufbau und ihre Funktion sind Gegenstand des folgenden Abschnitts.

3 Schnittstellen und Verwendung

ariba bietet zwei Schnittstellen für seine direkte Verwendung: Das *Developer Interface* und das *Legacy Interface*. Das *Legacy Interface* ermöglicht den Betrieb existierender, unmodifizierter Anwendungen und Dienste. Mit seiner Hilfe können bestehende IPv4-basierte Anwendungen ohne Änderungen von den Vorteilen des *ariba*-Rahmenwerks profitieren. Hierbei kommuniziert die Anwendung über eine virtuelle Netzwerkschnittstelle, welche *ariba* transparent nutzbar macht. Das *Developer Interface* hingegen bietet den Zugang zu *ariba* für die objektorientierte Entwicklung von Anwendungen und Diensten (vgl. Tabelle 2). Es besteht aus zwei Teilen, einer knotenspezifischen Schnittstelle und einer dienstspezifischen Schnittstelle. Dienste und Anwendungen operieren ausschließlich auf virtuellen Transportverbindungen (den in Abschnitt 2 erwähnten Links), die über die dienstspezifische Schnittstelle verwaltet werden können. Der knotenspezifische Teil definiert Funktionalität für die Erzeugung und den Beitritt zu einer *ariba*-Instanz, während der dienstspezifische Teil Mechanismen für die Erzeugung virtueller Links und das Versenden von Nachrichten über *ariba* anbietet. Durch die Verwendung spezieller Dienst-IDs lassen sich mehrere Dienste gleichzeitig über den dienstspezifischen Teil an *ariba* binden; die IDs werden dann für das Multiplexen der Nachrichten in einer einzelnen *ariba*-Instanz verwendet. Die Verwendung von *ariba* erfordert keine auf-

	Funktion	Beschreibung
Knoten	<code>initiate</code>	Erzeuge <i>ariba</i> Instanz
	<code>join</code>	Tritt <i>ariba</i> Instanz bei
	<code>leave</code>	Verlasse <i>ariba</i> Instanz
	<code>onJoinCompleted</code>	Zeige Beitrittserfolg an
	<code>onJoinFailed</code>	Zeige Beitrittsmisserfolg an
	<code>onLeaveCompleted</code>	Zeige Erfolg für Verlassen an
	<code>onLeaveFailed</code>	Zeige Misserfolg für Verlassen an
Dienst	<code>bind</code>	Melde Dienst mit eindeutiger ID an
	<code>unbind</code>	Melde Dienst ab
	<code>establishLink</code>	Erzeuge virtuellen Link zu NodeID
	<code>dropLink</code>	Baue virtuellen Link ab
	<code>sendMessage</code>	Sende Nachricht über virtuellen Link
	<code>onLinkUp</code>	Zeige erfolgreiche Linkerzeugung an
	<code>onLinkDown</code>	Zeige erfolgreichen Linkabbau an
	<code>onLinkChanged</code>	Zeige Änderung bei Link an
	<code>onLinkFailed</code>	Zeige Linkbruch an
	<code>onLinkRequest</code>	Zeige eingehenden Linkaufbau an
	<code>onMessage</code>	Eingehende Nachricht auf virtuellem Link

Tabelle 2 Überblick der *ariba* Schnittstelle für Dienst- und Anwendungsentwickler (*Developer Interface*).

wändige Konfiguration. Vom Entwickler müssen lediglich die angebotenen Schnittstellenfunktionen implementiert werden, was eine vereinfachte Benutzbarkeit und eine Erweiterbarkeit durch Abstraktion schafft. Durch das Verbergen von Netzwerkadressen und Transportprotokollen mit Hilfe von NodeIDs und LinkIDs und der Umsetzung von Anwendungsanforderungen durch automatische Selektion der Protokolle im Underlay wird das schnelle Entwickeln von Anwendungen erleichtert (*Rapid Prototyping*). *ariba* wurde ereignisorientiert konzipiert, was den Anwendungsentwickler vom Umgang mit komplexen Nebenläufigkeiten entbindet. Durch die bereitgestellten Ereignisfunktionen werden Dienste und Anwendungen über dienst- und knotenspezifische Ereignisse benachrichtigt und können entsprechend reagieren. Dieses Verhalten erleichtert die Anwendungsentwicklung stark und erlaubt ein Fokussieren auf die eigentliche Dienstfunktionalität.

4 Praxis und Anwendung

Neben den Schnittstellen und Hauptkomponenten stellt *ariba* eine komplette Programmierumgebung mit einer Vielzahl an Hilfsmechanismen bereit, welche eine einfache Entwicklung von neuartigen Anwendungen und Diensten ermöglicht. Zu nennen sind hier die einfache Verwendung von Timern, Threads und Serialisierung von komplexen Datentypen zum Versand über das Netzwerk.

ariba wurde in C++ unter Linux entwickelt und nutzt zur Erbringung seiner Dienste bewährte Bibliotheken (z. B. STL, Boost). Dies erleichtert die Portierung auf andere Systeme.

ariba ist Open-Source und unter der FreeBSD-Lizenz verfügbar. Es kann demnach erweitert, modifiziert und in kommerziellen Projekten verwendet werden. Die Schnittstellen sind mit Doxygen dokumentiert, und es existieren eine Vielzahl von Tutorials, wie diese zu verwenden sind. Eine direkte Anbindung von *ariba* an das Simulationsrahmenwerk *OverSim* befindet sich in Entwicklung.

Die *ariba*-Software wird seit mehreren Jahren kontinuierlich weiterentwickelt und gepflegt. Ihr Reifegrad wurde bereits in mehreren Demonstratoren unter Beweis gestellt. Dies geschah unter anderem auf der ACM SIGCOMM 2009 [4] in Barcelona, auf der besonders die Ausbringung von Diensten auf heterogenen Netzen demonstriert wurde. Dieser Demonstrator erhielt dem 2. Platz und ein „Honorable Mention“ von den Teilnehmern der Konferenz. Des Weiteren wurde auf der IEEE INFOCOM 2010 [2] in San Diego die Unterstützung von Legacy-Applikationen durch *ariba* demonstriert. Der Fokus der Demonstration auf der KiVS 2011 [3] lag auf der einfachen Benutzbarkeit von *ariba* aus Anwender- und Entwicklersicht.

ariba bildet die Grundlage für mehrere Anwendungen und Bibliotheken, die im Rahmen des SpoVNet-Projekts entwickelt wurden, nämlich für das an der Universität Mannheim entwickelte dezentrale Echtzeitspiel Planet $\pi 4$, sowie für den Ereignisbenachrichtigungsdienste EONSON und den Ereigniskorrelationsdienst CORDIES, die beide an der Universität Stuttgart entwickelt wurden. Weiterhin wurde eine Schnittstelle für den VLC Mediaplayer entwickelt, welche die Verwendung von *ariba* als Kommunikationsschicht für Videostreaming mit VLC ermöglicht. In den zuvor erwähnten Demonstratoren war beispielhaft eine Chat-Anwendung realisiert. Diese Beispielanwendungen lassen die Breite der durch *ariba* unterstützten Anwendungen erahnen.

Eine kontinuierliche Anstrengung ist die Portierung der Software auf verschiedene Plattformen um Anwendungsentwicklern neben der Abstraktion vom Netzwerk eine Abstraktion der Plattform bereitzustellen. Neben der Standardplattform Linux wurden erfolgreiche Portierungen geschaffen für OpenWRT-basierte Router, für das auf dem iPhone und iPad verwendete iOS, für die Smartphone Betriebssysteme OpenMoko und Maemo, sowie für das Android-Betriebssystem.

Das *ariba*-Rahmenwerk ist verfügbar unter <http://www.ariba-underlay.org>.

Danksagung. Diese Arbeit wird gefördert im Rahmen des Projekts *Spontane Virtuelle Netze (SpoVNet)* von der Baden-Württemberg Stiftung als Teil des BW-FIT Programms sowie teilweise im Rahmen der Nachwuchsforschergruppe *Controlling Heterogeneous and Dynamic Mobile Grid and Peer-to-Peer Systems (CoMoGriP)* durch das Zukunftskonzept des Karlsruher Instituts für Technologie (KIT) im Rahmenwerk der deutschen Exzellenzinitiative.

Literatur

- [1] R. Bless, C. Hübsch, C. P. Mayer, and O. P. Waldhorst. *Future Internet Services and Service Architectures*, chapter SpoVNet: An Architecture for Easy Creation and Deployment of Service Overlays, pages 23–47. River Publishers, 2011.
- [2] C. Hübsch, C. Mayer, S. Mies, R. Bless, O. Waldhorst, and M. Zitterbart. Demo Abstract: Using Legacy Applications in Future Heterogeneous Networks with *ariba*. In *Proc. IEEE INFOCOM*, San Diego, May 2010.
- [3] C. Hübsch, C. Mayer, S. Mies, R. Bless, O. Waldhorst, and M. Zitterbart. *ariba*: Rahmenwerk für Overlay-basierte Dienste (Peer reviewed demonstrator). In *Kommunikation in Verteilten Systemen (KiVS)*, Kiel, Mar. 2011.
- [4] C. Hübsch, C. P. Mayer, S. Mies, R. Bless, O. P. Waldhorst, and M. Zitterbart. Reconnecting the Internet with *ariba*: Self-Organizing Provisioning of End-to-End Connectivity in Heterogeneous Networks. *ACM SIGCOMM Computer Communication Review*, 40(1):131–132, Jan. 2010. SESSION: Best poster & demo abstracts from ACM SIGCOMM 2009.
- [5] O. P. Waldhorst, R. Bless, and M. Zitterbart. Overlay-Netze als Innovationsmotor im Internet – Spontane virtuelle Netze: auf dem Weg zum Internet der Zukunft. *Informatik Spektrum*, 2(288):171–185, Mar. 2010. DOI: 10.1007/s00287-010-0418-6, ISSN 0170-6012 (Print) 1432-122X (Online).