JANUARY • FEBRUARY 2015

IEEE Internet Computing

SERVICES FROM THE EARTH TO THE CLOUD

Security in Internet Banking
Future Mobile Video
Email Encryption

IEEE

IEEE computer society

IEEE Internet Computing:

Call for Papers

Submit a manuscript on ScholarOne at https://mc.manuscriptcentral.com:443/ic-cs

Internet of You: Data Big and Small
(November/December 2015)

Final submissions due 1 March 2015

Please email the guest editors a brief description of the article you plan to submit by 1 February 2015.

Guest editors: Deborah Estrin and Craig Thompson (ic6-2015@computer.org)

Where our ancestors left behind few records, we are creating and preserving increasingly complete digital traces and models of almost every aspect of our lives. This special issue aims to explore technologies and issues from small user-centric models of individuals to real-time analytics on huge aggregations of user data. Some are aspiring to let you record everything about yourself and convert it into a model that's queriable, conversant, and possibly even active in gaining new experiences for itself. Others are concerned with stemming the tide of third-party data aggregation to mitigate risks that can evolve from near total information awareness.

This special issue seeks original articles that explore both small data (individual-scale data sources, processing, and modeling) and big data (community-level aggregation and analytics). Topics include

- diverse data sources and digital traces, including email, Facebook, location, images, and sound;
- methods to combine trace data into complete models, data liberation, kinds of user models, and data quality;
- methods to aggregate and process heterogeneous, datasets and stages of life ontologies;
- usage models for experience sampling;
- representation technologies;
- new applications that draw insights from data analytics;
- open architectures for personalization, the role of cloud computing, and relevant emerging standards;
- concerns regarding surveillance, and privacy and security technology safeguards; and
- social and philosophical implications for humans' conception of self.

Internet Economics
(January/February 2016)

Final submissions due: 1 May 2015

Please email the guest editors a brief description of the article you plan to submit by 2 April 2015.

Guest editors: Arpita Ghosh and Ashish Goel (ic1-2016@computer.org)

The Internet both enables online versions of traditional markets and provides a platform for a vast range of new economic activity, ranging from targeted online advertising to crowdsourcing to peer-to-peer lending and digital currencies. These economic systems pose new theoretical and data-driven research questions: How do these online markets perform, and how should they be designed? How does the potentially giant scale of these systems affect performance? How do users behave in these online platforms, and how should incentives and interfaces be designed for maximum efficacy?

This special issue will address theoretical and applied research related to the modeling, analysis, and design of Internet-specific economic activities and incentive systems. We welcome any research related to economic aspects of the Internet, including

- Internet auctions, markets, and exchanges;
- reputation and quality in online markets;
- digital media, user-generated content, and social networks;
- crowdsourcing and human computation, and online labor markets;
- P2P lending, crowdfunding, and digital currencies;
- online privacy and security, and personal data markets;
- approaches to spam/fraud control;
- e-commerce issues in cloud computing and Internet-enabled apps;
- mobile advertising and location-based e-commerce;
- decision- and game-theoretic behavior modeling;
- user-experience and interface design; and
- incentives and mechanisms for collaboration, consensus, and decision making.

All submissions must be original manuscripts of fewer than 5,000 words, focused on Internet technologies and implementations. All manuscripts are subject to peer review on both technical merit and relevance to *IC*'s international readership – primarily practicing engineers and academics who are looking for material that introduces new technology and broadens familiarity with current topics. We do not accept white papers, and we discourage strictly theoretical or mathematical papers. To submit a manuscript, please log on to ScholarOne (https://mc.manuscriptcentral.com:443/ic-cs) to create or access an account, which you can use to log on to *IC*'s Author Center and upload your submission.

www.computer.org/internet/author

IEEE Internet Computing

JANUARY/FEBRUARY 2015, VOLUME 19, NUMBER 1

DEPARTMENTS

COLUMNS

www.computer.org/internet/

This publication is indexed by ISI (Institute for Scientific Information) in SciSearch, Research Alert, the CompuMath Citation Index, and Current Contents/Engineering, Computing, and Technology. Postmaster: Send undelivered copies and address changes to IEEE Internet Computing, IEEE Service Center, 445 Hoes Ln., Piscataway, NJ 08855-1331. Periodicals postage paid at New York, NY, and at additional mailing offices. Canadian GST #125634188. Canada Post Publications Mail Agreement Number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8. Printed in the USA. Circulation: IEEE Internet Computing (ISSN 1089-7801) is published bimonthly by the IEEE Computer Society. IEEE headquarters: 3 Park Avenue, 17th Floor, New York, NY 10016-5997. IEEE Computer Society headquarters: 1828 L St. N.W., Suite 1202, Washington, D.C. 20036-5104. IEEE Computer Society Publications Office: 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, Calif. 90720; (714) 821-8380; fax (714) 821-4010. **Subscription rates:** IEEE Computer Society members get the lowest rates and choice of media option — US$48/1,300 for member/nonmember institutional print + online. For information on other prices or to order, go to www.computer.org/subscribe. Back issues: $20 for members, $173 for nonmembers.

ENGINEERING AND APPLYING THE INTERNET

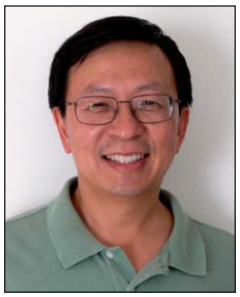
Web services have grown to influence and encompass a vast array of research and application areas, from the Internet of Things to geographic information services to cloud computing. The articles in this issue examine just some of the applications and ongoing research expanding the reach of Web services ever farther.

Cover by Giacomo Marchesi, bucket@earthlink.net

SERVICES FROM EARTH TO THE CLOUD

For more information on these or any other computing topics, please visit the IEEE Computer Society Digital Library at www.computer.org/publications/dlib.

From the Editors



The Growing Pains of Cloud Storage

Yih-Farn Robin Chen • *AT&T Labs Research*

Cloud storage is growing at a phenomenal rate, fueled by multiple forces, including mobile devices, social networks, and big data. Content is created anytime and anywhere on billions of smartphones and tablets; high-resolution photos and videos are frequently uploaded to the cloud automatically as soon as they're captured. A Gartner report predicts that consumer digital storage will grow to 4.1 zettabytes in 2016, with 36 percent of this storage in the cloud.[1] Social interactions and transactions on the Internet are frequently captured and analyzed for targeted advertising. In addition to social networks and e-commerce, big data analytics are growing in many other sectors, including government, healthcare, media, and education. An IDC forecast suggests that big data storage is growing at a compound annual growth rate of 53 percent from 2011 to 2016.[2]

The growth in cloud storage has made it an expensive cost component for many cloud services and today's cloud infrastructure. Whereas raw storage is cheap, the performance, availability, and data durability requirements of cloud storage frequently dictate sophisticated, multitier, geo-distributed solutions. Amazon Simple Storage Service (S3) offers 11 nines of data durability (99.999999999 percent), but some other services demand even more stringent requirements due to the sheer number of objects being stored in the cloud (1.3 billion Facebook users, uploading 350 million photos each day) and to the data's importance (who can afford to lose a video of their baby's first steps?). Data is frequently replicated or mirrored in multiple datacenters to avoid catastrophic loss, but copying it across datacenters is expensive. The networking cost is frequently proportional to the distance and bandwidth requirements between datacenter sites.

Traditional storage systems use dedicated hardware and networking to guarantee preservation of the quality-of-service (QoS) requirements, such as throughput, latency, and IOPS (total number of input/output operations per second). Unfortunately, these dedicated resources are frequently underutilized. Cloud computing promises efficient resource utilization by allowing multiple tenants to share the underlying networking, computing, and storage infrastructure. However, providing end-to-end storage QoS guarantees to individual tenants is difficult without mechanisms for avoiding interference. Typically, in a cloud environment such as Openstack, multiple tenants share the backend block storage (Linux's logical volume manager or a Ceph RADOS block device [RBD], for example) through a storage virtualization layer such as Cinder, which attaches virtual machines (VMs) to individual storage volumes. Providing customized storage QoS to meet different tenant needs is challenging. One exception is all-SSD storage arrays; some vendors (such as Solid Fire) let different tenants allocate storage volumes with different QoS types and dynamically change them, but all-SSD solutions (on the order of US$1,000 per terabyte) are expensive compared to HDD-based solutions. Moreover, an IOPS guarantee in the backend isn't sufficient because there might be contention for network bandwidth or CPU capacity from other tenants.

Finally, to operate any Web-scale solutions, infrastructure service providers are moving to scale-out solutions based on commodity hardware, instead of expensive storage appliances, which are frequently more expensive and difficult to adapt to changing workload or specific QoS requirements. Any cloud solution architect must understand the tradeoffs among the performance, reliability, and costs of cloud storage to provide an effective overall solution.

Emerging trends are sweeping through the storage industry to address these issues. Here,

Published by the IEEE Computer Society 1089-7801/15/$31.00 © 2015 IEEE

I discuss two software-based solutions: erasure-coded storage and software-defined storage (SDS).

Erasure-Coded Storage

Erasure coding has been widely studied for distributed storage systems. Various vendors, companies, and open source software systems have adopted it recently, including EMC, Cleversafe, and Amplidata; Facebook, Microsoft, and Google; and Ceph, Quantcast File System (QFS), and a module of the Hadoop Distributed File System (HDFS-RAID), respectively. The primary reason for this adoption is that erasure-coded storage uses less space than fully replicated storage, while providing similar or higher data durability.

To understand why erasure coding is becoming crucial in storage systems, I must explain some basics. Erasure coding is typically controlled by two key parameters: k and n. A file or file segment is typically broken into k chunks, erasure coded, and expanded into n chunks ($n > k$) that are distributed over n storage servers or hard disks. Any k chunks are sufficient to reconstruct the original file, which can tolerate up to a loss of $m = n - k$ chunks without any data loss. One way to think about erasure coding is to consider a system of over-specified linear equations. You're essentially given n linear equations to solve for k variables. Picking any k out of these n equations would be sufficient to determine the values of those k variables. We frequently refer to the first k chunks as *primary chunks*, and the m chunks as *parity chunks*. Because we can vary k and m arbitrarily, a general erasure-coded storage solution in the form of (k, n) or $k + m$ has much higher flexibility in terms of the tradeoffs between storage space and reliability compared to the popular RAID 6 system, which uses only two parity blocks and is equivalent to a $k + 2$ erasure-coded scheme.

A scalable distributed storage system, such as HDFS or Swift, stored on multiple racks or sites typically uses triple redundancy (three copies of each data block) to improve both availability and durability. As the cloud storage volume continues to grow exponentially, the triple redundancy scheme becomes expensive. As an example, the QFS system uses $6 + 3$ ($k = 6$ and $m = 3$) erasure coding and is designed to replace HDFS for MapReduce processing. HDFS uses triple replication and incurs 200 percent storage overhead, but it can only tolerate up to ANY two missing blocks of the same data. A $6 + 3$ erasure code, on the other hand, can tolerate up to ANY three missing coded blocks with only 50 percent storage overhead. Such significant cost savings, while maintaining the same or higher reliability, is why many storage systems are now incorporating erasure codes.

One concern with erasure-coded storage is the extra overhead caused

by the encoding/decoding time, which depends heavily on the erasure-coding scheme's strength. For a fixed k, higher m or n incurs more computation overhead while providing higher reliability. As computing servers gain in performance, the computation overhead of commonly used erasure codes becomes more manageable, and the bottleneck is frequently shifted to the disk or network throughput.

Another concern is the repair cost. Given that erasure coding of $6 + 3$ requires six chunks to repair one chunk, the networking cost of repairing a chunk is six times that of a simple replication scheme. Some Facebook experiments use a $10 + 4$ erasure-coding scheme, which incurs even higher repair costs (but lower storage overhead at 40 percent). Several repair schemes (such as Xorbas[3] and Hitchhiker[4]) have been proposed to reduce the repair bandwidth, with or without additional storage overhead.

As data durability becomes increasingly important for cloud storage, erasure coding can also play an important role in cloud storage geo-distribution. It allows chunks of an erasure-coded file to be placed in multiple datacenters or racks to increase data durability. For example, a $9 + 15$ or (9, 24) erasure-coded storage system could put six chunks each in New Jersey, Illinois, Texas, and California (east, north, south, and west areas of the US). Because any file can be reconstructed from nine chunks, it can be reconstructed by retrieving those chunks from any two datacenters. Thus, it will tolerate up to two datacenter failures, even during a major natural disaster such as 2013's Hurricane Sandy, which caused a loss of 68 billion dollars and affected 24 states. On the other hand, because each file retrieval requires accessing chunks from two datacenters, it might incur longer latency and significant communication costs, which is fine for archival storage, but not ideal for frequently accessed storage. Alternatively, if we know certain files' access patterns, and it turns out that most accesses come from New Jersey, we can place nine chunks in New Jersey and five chunks each in Illinois, Texas, and California. This would allow users to complete most accesses with low network latency and slightly lower reliability, given that a datacenter loss has the potential to lose nine instead of six chunks. The chunk-placement issue in erasure coding affects latency, cost, and reliability in geo-distributed storage systems and is currently an active research field.

To realize the vision of a virtualized datacenter, we need software-defined storage that virtualizes storage resources and separates storage management software from the underlying hardware.

Software-Defined Storage

Cloud computing started with the virtualization of computing resources, followed by recent advances and rapid innovations in software-defined networks (SDNs), which aim to virtualize networking resources and separate the control plane from the data plane. To truly realize and complete the vision of a virtualized datacenter, however, we need software-defined storage that virtualizes storage resources as well and separates storage management software from the underlying hardware.

Unfortunately, unlike SDN, there isn't a clear definition of what software-defined storage really is, although many storage vendors claim that they have SDS solutions. Most SDS definitions include a list of desirable attributes.[5,6] Here, I summarize those that pertain to multitenant cloud storage solutions, what I call the S.C.A.M.P. principles of SDS.

Scale-Out

SDS should enable a scale-out (horizontal scaling of low-cost, commodity hardware) instead of a scale-up (vertical scaling using more powerful hardware) storage solution as the workload grows or changes dynamically over time. A scale-out solution is best implemented in a cloud environment with large computing, networking, and storage resource pools. A cloud storage solution is never just about storage — all the necessary computing and networking resources must also scale accordingly to support common storage operations: deduplication, compression, encryption/decryption, erasure coding/replication, and so on.

Customizable

SDS should allow storage system customization to meet specific storage QoS requirements. This lets customers purchase storage solutions based on their specific performance and reliability constraints and avoid unnecessary over-engineering, which frequently happens when a cloud storage service provider tries to meet the needs of multiple customers with diverse requirements. In a multitenant cloud with a shared backend storage, guaranteeing the desired storage QoS is particularly difficult. The latest version of Openstack Cinder, which provides a block storage service, now

allows multiple backends with different QoS types (such as different IOPS or throughput numbers) to partially address this issue.

Automation

Once storage QoS requirements are clearly defined, SDS should automate the complete provisioning and deployment process without human intervention. The current practice is that a storage architect or system administrator is intimately involved in designing and installing the storage system. This process is typically error-prone and not amenable to adapting to changing workloads or requirements in real time.

Masking

SDS could mask the underlying storage system (physical or virtualized) and distributed system complexity (single or multiple-site) as long as such systems can present a common storage API (block, file system, object, and so on) and meet QoS requirements. This gives infrastructure service providers greater flexibility in restructuring their resource pools or architecting storage systems. For example, Ceph can present a block device API even though the underlying implementation is done in its RADOS object storage.

Policy Management

SDS software must monitor and manage the storage system according to the specified policy and continue to meet storage QoS requirements despite potential interference from other tenants' workloads. It must also handle failures and autoscale the system when necessary to adapt to changing workloads. As stated previously, however, guaranteeing end-to-end storage QoS in a multitenant cloud is a hard problem that requires protecting resources on the entire path from a VM to the storage volume. Microsoft's IOFlow[7] aims to provide an SDN-like controller to control storage bandwidth allocation at multiple points of such a path.

SDS Definition

By combining the S.C.A.M.P. principles, we can now define SDS: an SDS solution should automatically map customizable storage service requirements to a scalable and policy-managed cloud storage service, with abstractions that mask the underlying storage hardware and distributed system complexities.

Incidentally, erasure coding is a crucial technology that can help meet the SDS customization requirement. For a fixed k, varying n (or m, the number of parity chunks) increases the reliability and replication factor (and hence the storage cost). At the same time, it increases the overall encoding/decoding time, hence the required computation capacity, and perhaps reduced performance. This lets an automated storage architect look at the storage QoS requirements and pick particular erasure-code parameters (k and m) to meet the minimal reliability and performance requirements with the least amount of storage overhead.

The rapid growth of cloud storage has created challenges for storage architects to meet different customers' diverse performance and reliability requirements while controlling costs in a multitenant cloud environment. Erasure-coded storage and SDS could address these challenges and open up new opportunities for innovation. Moreover, erasure coding could play a crucial role in offering design tradeoffs in certain SDS solutions. These two technologies, working together, have a huge potential to address the growing pains of cloud storage and help ease the transition from traditional IT storage solutions — given that cloud storage will likely support a large portion of all IT storage needs in the future.

References

1. "Gartner Says that Consumers Will Store More than a Third of Their Digital Content in the Cloud by 2016," Gartner, press release, 25 June 2012; www.gartner.com/newsroom/id/2060215.
2. "Big Data Drives Big Demand for Storage, IDC Says," *Business Wire*, 16 Apr. 2013; www.businesswire.com/news/home/20130416005045/en/Big-Data-Drives-Big-Demand-Storage-IDC.
3. M. Sathiamoorthy et al., "XORing Elephants: Novel Erasure Codes for Big Data," *Proc. VLDB Endowment*, 2013, pp. 325–336.
4. K. Rashmi et al., "A Hitchhiker's Guide to Fast and Efficient Data Reconstruction in Erasure-Coded Data Centers," *Proc. 2014 ACM Conf. SIGCOMM*, 2014, pp. 331–342.
5. B. Earl et al., "Software-Defined Storage," *5th Usenix Workshop on Hot Topics in Storage and File Systems*, panel, 2013; www.usenix.org/conference/hotstorage13/workshop-program/presentation/earl.
6. M. Carlson et al., "Software-Defined Storage," Storage Networking Industry Assoc. working draft, Apr. 2014; http://snia.org/sites/default/files/SNIA%20Software%20Defined%20Storage%20White%20Paper-%20v1.0k-DRAFT.pdf.
7. E. Thereska et al., "IOflow: A Software-Defined Storage Architecture," *Proc. 24th ACM Symp. Operating Systems Principles*, 2013, ACM, pp. 182–196.

Yih-Farn Robin Chen is a Lead Inventive Scientist at AT&T Labs Research. His research interests include cloud computing, storage systems, mobile computing, and distributed systems. Chen received a PhD in computer science from the University of California, Berkeley. He's a vice chair of the International World Wide Web Conferences Steering Committee (IW3C2) and a member of the editorial board of *IEEE Internet Computing*. Contact him at chen@research.att.com.

Reviewer Thanks

IC Thanks Our 2014 Reviewers

The articles appearing in *IEEE Internet Computing* result from the hard work of many people. We deeply appreciate the efforts of everyone who reviewed the many articles submitted to *IC* last year. The peer review process helps maintain the magazine's revered quality. Those of us in the Internet computing community owe gratitude to people who participate in this crucial service. Readers who would like to contribute as reviewers can visit www.computer.org/internet to find out how they can get involved. —*IC's Editorial Board and Staff*

2014 Reviewers

Abdallah, Maha
Agrawal, Abhishek
Aiello, Luca
Aiello, Marco
Almeida, Jussara
Almeida, Virgilio
Alvarez-Hamelin, José
Anantharam, Pramod
Ardagna, Claudio
Arlitt, Martin
Armitage, Grenville
Baek, Joonsang
Balaguer, Federico
Baresi, Luciano
Beard, Kate
Beg, Mirza
Benatallah, Boualem
Bermudez, Maria
Bertino, Elisa
Bi, Jun
Blake, Brian
Boldi, Paolo
Bordogna, Gloria
Bousetta, Kahled
Breslin, John
Bustamante, Fabian
Cai, Hubo
Calafate, Carlos
Carminati, Barbara
Cavalcante de Oliveira, Jaudelice
Caverlee, James
Celino, Irene
Chandra, Surendar
Chang, Hyunseok
Chen, Kuan-Ta
Chen, Yih-Farn
Chesñevar, Carlos
Chou, Shao-Yan
Claypool, Mark
Coetzee, Louis
Corcho, Oscar
Cornford, Dan
Cugola, Gianpaolo
D'Elia, Alfredo
D'Mello, Sidney
da Silva, Hugo
Daniel, Florian
davies, elwyn
De Ryck, Philippe
Delling, Daniel
Dewri, Rinku
Dorn, Christoph
Douglis, Fred
Dubois, Daniel
Duquennoy, Simon
Efstratiou, Christos
Eisenhauer, Greg
Emrich, Andreas
Epasto, Alessandro
FarajiDavar, Nazli
Farrell, Stephen
Feller, Eugen
Fernández, Alejandro
Firner, Bernhard
Fowler, Scott
Fra', Cristina
Frey, Davide
Ganz, Frieder
Gao, Wei
Gauch, Susan
Georgantas, Nikolaos
Giatsidis, Christos
Giese, Holger
Gniady, Chris
Gomes Soares, Luiz Fernando
Gopalan, Kartik
Griwodz, Carsten
Groba, Christin
Gruenbacher, Paul
Guinea, Sam
Haddadi, Hamed
Han, Qi
Hancock, Jeff
Harper, Simon
Henderson, Tristan
Houben, Geert-Jan
Howell, Fred
Hsu, Cheng-Hsin
Huang, Jimmy
Huguenin, Kévin
Ignatovic, Alex
Inzinger, Christian
Iosup, Alexandru
Ishakian, Vatche
Issarny, Valerie

Published by the IEEE Computer Society 1089-7801/15/$31.00 © 2015 IEEE

IC Thanks Our 2014 Reviewers

Javadi, Bahman
Johns, Martin
Josefsson, Simon
Joshi, James
Julien, Christine
Kaltenbrunner, Andreas
Katasonov, Artem
Keller, Eric
Kermarrec, Anne-Marie
Ko, Bong-Jun
Kolozali, Sefki
Kotoulas, Spyros
Lago, Patricia
Lane, Nic
Lee, Choong-soo
Lee, Kyumin
Leiba, Barry
Leitner, Philipp
Lekies, Sebastian
Leonardi, Stefano
Leontiadis, Ilias
Levina, Olga
Li, Ang
López-Nores, Martín
Ludwig, Heiko
Magnani, Matteo
Mahanti, Anirban
Makaroff, Dwight
Margara, Alessandro
Marino, Andrea
Mascolo, Cecilia
McCann, Julie
McCreadie, Richard
Mele, Ida
Mika, Peter
Milojicic, Dejan
Minocha, Shailey
Misra, Prasant
Monsieur, Geert
Monti, Corrado
Mottola, Luca
Nardini, Franco Marie
Noulas, Anastasios
O'Reilly, Colin
Oinas-Kukkonen, Harri
Ooi, Wei Tsang
Padget, Julian
Panigati, Emanuele
Panta, Rajesh
Panzica La Manna, Valerio
Pathak, Animesh
Pathan, Mukaddim
Patrono, Luigi
Payton, Jamie
Peoples, Cathryn
Perego, Raffaele
Perkins, Colin
Picco, Gian Pietro
Pinciroli, Carlo
Porter, George
Prieur, Christophe
Quercia, Daniele
Querrec, Ronan
Race, Nicholas
Remy, Sekou
Rossi, Gustavo
Rothenberg, Christian Esteve
Rozza, Alessandro
Ruiz-Martínez, Antonio
Sahu, Sambit
saleh moustafa, iman
Santini, Massimo
Sarhan, Nabil
Sastry, Nishanth
Saukh, Olga
Schiele, Gregor
Sharma, Abhigyan
Simperl, Elena
Singh, Munindar
Sirivianos, Michael
Song, JaeSeung
Spatscheck, Oliver
Strauch, Steve
Suznjevic, Mirko
Tao, Jie
Taylor, Kerry
Thirunarayan, Krishnaprasad
Thompson, Craig
Tolk, Andreas
Tran, Huy
Triukose, Sipat
Truong, Hong-Linh
Trushkowsky, Katherine
Truta, Traian
Tuexen, Michael
Urgaonkar, Rahul
Urueña, Manuel
van Kranenburg, Rob
van Schaik, Paul
van Steen, Maarten
Vandikas, Konstantinos
Varlamis, Iraklis
Vazirgiannis, M
Vigna, Sebastiano
Vilajosana, Xavier
Vizzari, Giuseppe
Voigt, Thiemo
Wang, Yang
Wei Law, Yee
Wei, Yi
Wenning, Rigo
Westerink, Peter
Wilson, Christo
Wohlgenannt, Gerhard
Wolfe, Christopher
Xavier Parreira, Josiane
Xiang, Yang
Xiao, Zhen
Xiong, Li
Yu, Jeffrey
Zaveri, Amrapali
Zdravkovic, Jelena
Zomaya, Albert

Feature: Web Services

Extending the Devices Profile for Web Services Standard Using a REST Proxy

The Devices Profile for Web Services (DPWS) standard enables the use of Web services for certain Internet of Things (IoT) applications. DPWS is appropriate for implementing services on resource-constrained devices. However, little investigation has gone into how such services perform in IoT scenarios when it comes to features such as dynamic discovery and publish-subscribe eventing. Moreover, DPWS introduces considerable overhead due to its use of SOAP envelopes in exchange messages. To tackle these problems, the authors extend the DPWS standard using a REST proxy, creating a RESTful Web API that paves the way for developers to invest more in this technology.

Son N. Han and Soochang Park
Institut Mines-Telecom, Telecom SudParis

Gyu Myoung Lee
Liverpool John Moores University

Noël Crespi
Institut Mines-Telecom, Telecom SudParis

We're witnessing the next major Internet evolution, in which millions of devices will connect to create a new ecosystem called the Internet of Things. With advancements in technology and the arrival of numerous commercial products, the IoT has gained considerable momentum. Application-layer standards such as the Constrained Application Protocol (CoAP)[1] and the Devices Profile for Web Services (DPWS)[2] support the creation of next-generation IoT applications for the Web.

DPWS in particular enables secure Web services capabilities on resource-constrained devices, thus supporting service-oriented and event-driven applications for networked devices. DPWS has an architectural concept similar to the W3C's Web Services Architecture,[3] but it differs in several ways to better fit in resource-constrained environments (constrained nodes and low-power, lossy networks) and event-driven scenarios.

Thus far, DPWS has been widely used in automation, home entertainment, and automotive systems.[4] It's also applicable for maintaining integration with the Internet and enterprise infrastructures.[5] Strong community support makes it a promising technology for the future IoT. However, IoT systems containing huge numbers of devices, in contrast to the small numbers in industrial and home applications, make some DPWS features – such as dynamic discovery and publish-subscribe eventing – impossible in mass or even global device deployments. We must thus extend DPWS for

Published by the IEEE Computer Society
1089-7801/15/$31.00 © 2015 IEEE

Devices Profile for Web Services Implementation

Since its debut in 2004, the Devices Profile for Web Services (DPWS) has become part of Microsoft's Windows Vista and Windows Rally (a set of technologies intended to simplify the setup and maintenance of wired and wireless networked devices). It's been developed in several projects under the European Information Technology for European Advancement (ITEA) and Framework Program (FP): Service Infrastructure for Real-Time Embedded Networked Applications (ITEA 02014 SIRENA), Service-Oriented Device and Delivery Architectures (ITEA 05022 SODA), Socrates (FP6), and the ongoing IMC-AESOP (FP7) and Web of Objects (ITEA 10028 WOO) projects. Many technology giants such as ABB, SAP, Schneider Electric, Siemens, and Thales have participated in these projects. Given that they have large market shares in electronics, power, automation technologies, and enterprise solutions, their promotion of DPWS technology promises a wide range of future DPWS/IoT products. Schneider Electric and Odonata pioneered DPWS implementation, leading to the early and open source release of software stacks implementing DPWS in C and Java available at the Service-Oriented Architecture for Devices website (SOA4D.org). The Web Services for Devices initiative (WS4D.org) reinforces the implementation by providing and maintaining a repository to host several open source stacks and toolkits for DPWS. In addition, considerable recent research completes the technology. Experimental results show that DPWS can be implemented in highly resource-constrained devices such as sensor nodes with reasonable ROM footprints.[1] Others have explored technical issues such as encoding and compression,[2] integration with IPv6 and 6LoWPAN,[3,4] the scalability of service deployment,[5] and security in the latest release of WS4D DPWS stacks.

References

1. C. Lerche et al., "Implementing Powerful Web Services for Highly Resource-Constrained Devices," *Proc. IEEE Int'l Conf. Pervasive Computing and Comm. Workshops* (PERCOM Workshops), 2011, pp. 332–335.
2. G. Moritz et al., "Encoding and Compression for the Devices Profile for Web Services," *Proc. IEEE 24th Int'l Conf. Advanced Information Networking and Applications Workshops* (WAINA 10), 2010, pp. 514–519.
3. G. Moritz et al., "Beyond 6LoWPAN: Web Services in Wireless Sensor Networks," *IEEE Trans. Industrial Informatics*, vol. 9, no. 4, 2013, pp. 1795–1805.
4. I. Samaras, G. Hassapis, and J. Gialelis, "A Modified DPWS Protocol Stack for 6LoWPAN-Based Wireless Sensor Networks," *IEEE Trans. Industrial Informatics*, vol. 9, no. 1, 2013, pp. 209–217.
5. X. Yang and X. Zhi, "Dynamic Deployment of Embedded Services for DPWS-Enabled Devices," *Proc. 2012 Int'l Conf. Computing Measurement, Control, and Sensor Network* (CMCSN 12), 2012, pp. 302–306.

IoT scenarios and resolve several problems before it can successfully arrive in the IoT domain. We analyze some of these problems and propose an extension to the DPWS standard that uses a REST proxy.

DPWS and the Internet of Things

The IoT is an ecosystem in which all smart things (sensors and actuators, embedded devices, electronic appliances, and digitally enhanced everyday objects) are connected using Internet protocols to facilitate interoperability. It envisions an era of pervasive applications built on top of these networked devices. IoT scenarios require that devices both connect to the Internet and integrate seamlessly into existing Internet infrastructure, in which Web applications are predominant. The IoT could benefit from the Web services architecture using the DPWS standard. DPWS brings W3C Web services technology into the IoT by defining specifications that provide a secure and effective mechanism for describing, discovering, messaging, and eventing services for resource-constrained devices.

DPWS is based on the Web Services Description Language (WSDL; www.w3.org/TR/wsdl) and SOAP (www.w3.org/TR/soap/), which describe and communicate device services. It doesn't require any central service registry, such as UDDI (http://uddi.org/pubs/uddi_v3.htm), for service discovery. Instead, it relies on SOAP-over-UDP binding and UDP multicast to dynamically discover device services (http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/os/wsdd-soapoverudp-1.1-spec-os.html). DPWS offers a publish-subscribe eventing mechanism, WS-Eventing (www.w3.org/Submission/WS-Eventing/), that lets clients subscribe to device events — for example, a device switch is on/off or the environmental temperature reaches a predefined threshold. When an event occurs, subscribers receive notifications via separate TCP connections.

DPWS uses WSDL to describe a device, Web Services Metadata Exchange (www.w3.org/TR/ws-metadata-exchange/) to define metadata about the device, and WS-Transfer (www.w3.org/Submission/WS-Transfer/) to retrieve the service description and metadata information. Messaging occurs via SOAP, WS-Addressing (www.w3.org/Submission/ws-addressing/), and the Message Transmission Optimization Mechanism/XML-Binary Optimized Packaging

(www.w3.org/TR/soap12-mtom/) with SOAP-over-HTTP and SOAP-over-UDP bindings. It uses WS-Discovery for discovering a device (*hosting service*) and its services (*hosted services*) and the Web Services Policy (www.w3.org/Submission/WS-Policy/) to define a policy assertion and indicate the device's compliance with DPWS.

DPWS's secure Web services, dynamic discovery, and eventing features are its main advantages for event-driven IoT applications. Nevertheless, when applying DPWS on current Internet infrastructure (IPv4), developers face several problems. The main concern is with dynamic discovery — the network range of UDP multicast messages is limited to local subnets; it's impossible to carry out discovery in a large network such as the Internet. With WS-Eventing, establishing separate TCP connections when delivering the same event notification to many subscribers will generate a global, mesh-like connectivity between all devices and subscribers. This requires high memory, processing power, and network traffic and thus consumes considerable energy in devices.

Another issue is overhead from the data representation in XML and from multiple bidirectional message exchanges. This isn't a problem when DPWS devices communicate locally, but in a mass device deployment, these messages would generate heavy Internet traffic and increase latency in device and application communication. Furthermore, W3C Web services use WSDL for service description and SOAP for service communication; the former, despite being a W3C standard, requires that developers process poorly structured XML data; the latter is common mostly in stateful enterprise applications, whereas recent Web applications are moving toward the core Web concepts that REST[6] encompasses by offering stateless, unified, and simple RESTful Web APIs.

To solve these problems, we propose extending the DPWS standard using a REST proxy that would enable the following features:

- global dynamic discovery using WS-Discovery (http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01) in local networks;
- a proxy-based topology for the publish-subscribe eventing mechanism;
- dynamic REST addressing for DPWS devices;
- a RESTful Web API; and
- WSDL caching.

Our REST proxy extension of DPWS unburdens Internet traffic by processing the main load in local networks. It can also extend local dynamic discovery globally via the RESTful Web API. Developers need not parse complex WSDL documents to access service descriptions; they can use the RESTful Web API to control devices.

Use Case

The new ecosystem of networked devices makes many IoT platforms available for building a new generation of Web-based applications that aggregate services. Peter, an IoT developer, chooses DPWS technology for his Web-based home automation system. He wants to make a module controlling a newly purchased DPWS heater. The heater is equipped with a temperature sensor, a switch, memory, a processor, and networking media, and is implemented with a hosted service consisting of seven operations: check the heater status (*GetStatus*), switch the heater on or off (*SetStatus*), get the room temperature (*GetTemperature*), adjust the heater temperature (*SetTemperature*), and add, remove, and get available policy rules for defining the heater's automatic operation (*AddRule*, *RemoveRule*, or *GetRules*).

Peter connects the heater to the network and tries to control it from his IoT application. We follow Peter's development process to understand the challenges he can encounter when developing, deploying, and communicating the device from his IoT application and how the extended DPWS helps him solve these problems. This use case illustrates a common case in several consumer applications when a new device joins the network.

REST Proxy Design

We introduce the detailed design of the REST proxy to extend DPWS to achieve global dynamic discovery, publish-subscribe eventing, dynamic REST addressing, a RESTful Web API, and WSDL caching.

Global Dynamic Discovery

When an application tries to locate a device or a hosting service in a network, it uses the SOAP-over-UDP binding to send a UDP multicast message. This message carries a SOAP envelope containing a WS-Discovery *Probe* message with search criteria — for instance, the device's name. All the target devices in

the network (local subnet) that match the search criteria will respond with a unicast WS-Discovery *Probe Match* message (also using the SOAP-over-UDP binding). In our use case, the heater sends the *Probe Match* message containing the network information. The application can send a series of other messages via the same means to invoke a required operation. At this point, Peter would realize that his IoT application in the current Internet infrastructure can't dynamically discover the heater because the network range is limited to the local multicast message subnet.

If the application uses a REST proxy, it can suppress multicast discovery messages and send a unicast request to the proxy instead. Then, the proxy can representatively send *Probe* and receive *Probe Match* messages to and from the network while device behavior remains unmodified; devices still answer to *Probe* messages arriving via multicast. In networks with frequent changes in the device structure, where several *Probe* messages appear, the proxy can significantly unburden Internet traffic.

The REST proxy provides two APIs to handle discovery as follows:

PUT http://157.159.103.50:8080/discovery: update the discovery with search criteria (for example, the device name)
GET http://157.159.103.50:8080/discovery: get the list of discovered devices (157.159.103.50 is the proxy's IP address, and 8080 is the port number.)

We also propose a repository in the proxy to maintain a list of active devices. The repository updates when devices join and leave the network. In addition, the proxy periodically checks the repository's consistency — say, every 30 minutes. For a proxy with 100 devices, the repository is about 600 Kbytes, so unconstrained machines can feasibly host a proxy.

Publish-Subscribe Eventing

To receive event notifications, Peter can subscribe his application directly to the heater by sending a SOAP envelope containing a WS-Eventing *Subscribe* message (again, using the SOAP-over-HTTP binding). The heater responds by sending a WS-Eventing *SubscribeResponse* message via the HTTP response channel. When an event occurs, the heater establishes a new TCP connection and sends an event notification to the subscriber. Therefore, multisubscriber scenarios generate a high level of traffic, requiring considerable resources and causing devices to consume more energy. However, we can implement this publish-subscribe mechanism through the REST proxy to reduce the overhead of SOAP message exchanges and resource consumption, replacing global mesh-like connectivity with a proxy-based topology (see Figure 1). One API is dedicated to event subscription; instead of sending a WS-Eventing *Subscribe* message, the application sends an HTTP POST request to the subscription resource as follows:

POST http://157.159.103.50:8080/heater/event (parameter: application endpoint): subscribe to an event

The proxy receives the event notification from the device and then disseminates these messages to the applications.

Dynamic REST Addressing

DPWS uses WS-Addressing to assign a unique identification to each device (endpoint address), independent of its transport-specific address. This unique ID is used with a series of message exchanges — *Probe/ProbeMatch* and *Resolve/ResolveMatch* — to get a transport address. A client then sends another series of messages back and forth to invoke an operation. This process creates the overhead on the Internet. We define a mapping between a pair of DPWS endpoint/transport addresses and a single proxy URI, and thus replace several SOAP messages with simpler HTTP request/response ones. The mapping is carried out dynamically when the proxy discovers a device. In our DPWS heater use case, this would occur as follows:

Endpoint address: urn:uuid:800fa0d0-f5c0-11e2-80de-911c7defef4c
Transport address: http://157.159.103.50:4567/heater

mapped to

URI: http://157.159.103.50:8080/heater

The mapping is unique for each device service, and data are stored in the proxy's device repository. The repository also updates when the device status changes or periodically when

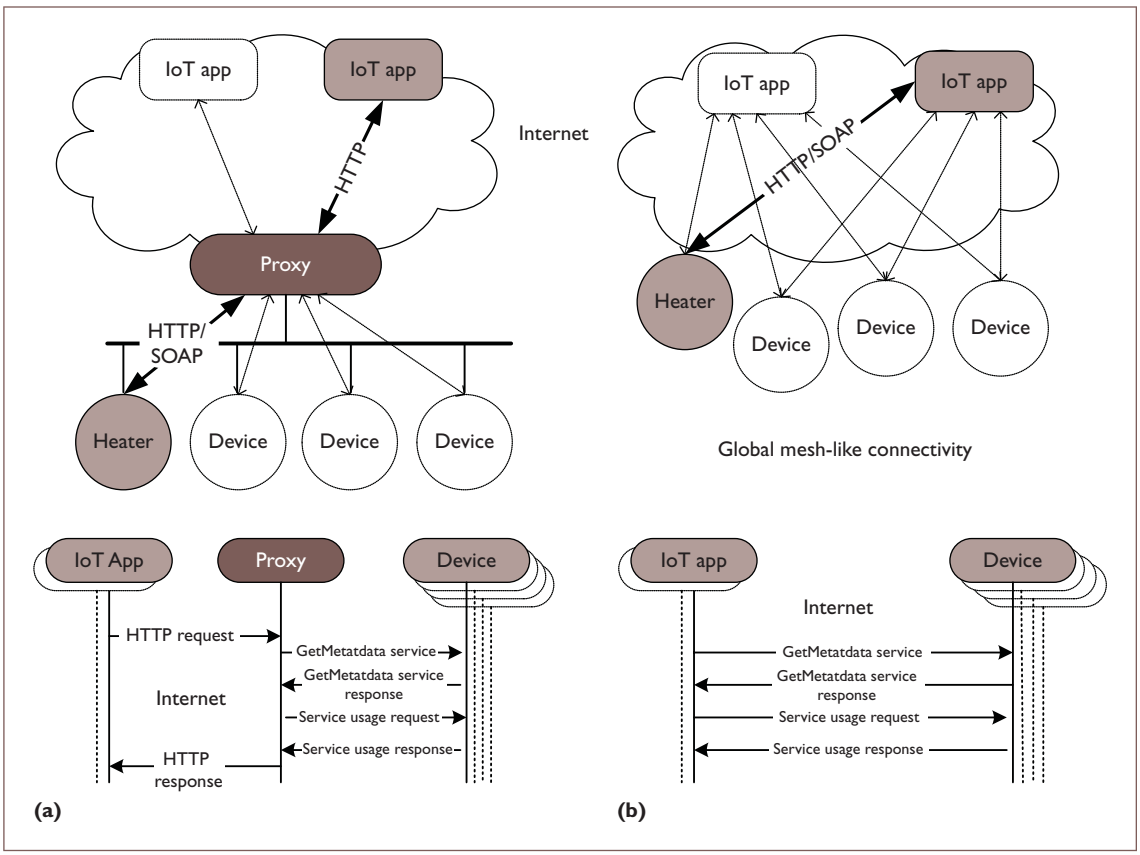


Figure 1. Experiment setup for two cases. (a) Our proposed scheme configures a proxy-based topology with local HTTP/SOAP binding. (b) The original Devices Profile for Web Services (DPWS) communication configures global mesh-like connectivity for HTTP/SOAP binding. Consequently, the original DPWS introduces higher latency and overhead.

the proxy runs a routine to check all active devices.

RESTful Web API

Because it's based on the dynamic REST addressing mechanism we've described, our REST proxy can generate a RESTful Web API associated with each device. This means that, instead of sending several SOAP-over-HTTP binding messages involving strict and large data formats, Peter can take advantage of simple, familiar Web interfaces. The API consists of functions for discovery, subscription, and service calls in the REST architectural style. To generate this RESTful Web API from DPWS operations, we propose a design constraint on the implementation of DPWS devices based on the fact that most device services provide simple operations compared to normal Web services, which have complex I/O data structures. Our proposed constraint follows a simplified CRUD model (create, read, update, delete) to map between these services and HTTP methods: *DPWS operation prefix → CRUD action → HTTP method*. Specifically, we apply four CRUD actions to map DPWS operations to HTTP methods as follows:

Get_ → READ → GET
Set_ → UPDATE → PUT
Add_ → CREATE → POST
Remove_ → DELETE → DELETE

Table 1 shows the API that the REST proxy provides for the heater device, mapped with DPWS operations. The following illustrates request and response messages to get and return the heater's status using HTTP method GET on the URI http://157.159.103.50:8080/heater:

```
GET /heater HTTP/1.1
Host: 157.159.103.50:8080
Accept: text/html
```

Table 1. Proxy RESTful Web API for the heater.

No.	RESTful Web API	DPWS* operations	Parameters	Functionalities
1	GET http://157.159.103.50:8080/discovery PUT http://157.159.103.50:8080/discovery	Discovery	 deviceName	List devices Search for devices
2	POST http://157.159.103.50:8080/heater/event	Subscription		Subscribe to an event
3	GET http://157.159.103.50:8080/heater	GetStatus()		Get heater status
4	PUT http://157.159.103.50:8080/heater	SetStatus(String)	status	Set heater status
5	GET http://157.159.103.50:8080/heater/temp	GetTemp()		Get room temperature
6	PUT http://157.159.103.50:8080/heater/temp	SetTemp(int)	temperature	Adjust heater temperature
7	POST http://157.159.103.50:8080/heater/rules	AddRule(String)	rule	Add new rule
8	GET http://157.159.103.50:8080/heater/rules	GetRules()		List of rules
9	DELETE http://157.159.103.50:8080/heater/rules/{ruleID}	RemoveRule(int)	ruleID	Delete a rule

*Devices Profile for Web Services

```
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html
Transfer-Encoding: chunked
```

WSDL Caching

When an application knows the endpoint address of a hosted service, it can ask that service for its interface description by sending a *GetMetadata Service* message. The service might respond with a *GetMetadata Service Response* message, including a WSDL document. This document describes the supported operations and the data structures used in the device service. Some DPWS implementations (such as the Java Multi-Edition DPWS Stack, or JMEDS; http://ws4d.org/jmeds/) provide a cache repository for storing the WSDL document at runtime. After the application retrieves the WSDL file for the first time, it's cached for local use in subsequent occurrences within the DPWS framework's life cycle (start/stop). This kind of caching mechanism can significantly reduce both latency and message overhead. Our DPWS proxy can provide WSDL caching not only at runtime but also permanently in a local database. The cache is updated along with the routine of maintaining the device repository in the proxy.

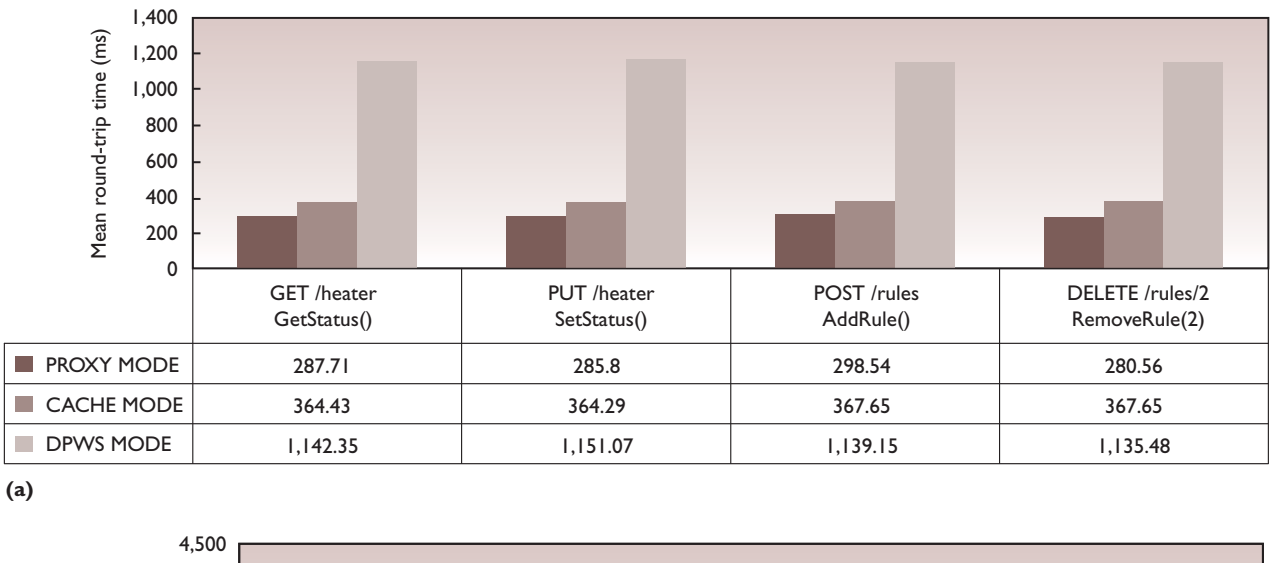
Evaluation

We set up an experiment to evaluate latency and overhead in two different scenarios: one uses our proposed REST proxy (Figure 1a), and the other uses the original DPWS (Figure 1b). In both cases, an IoT application communicates with a DPWS device (a heater) to invoke its hosted service (heater functionalities). To replicate a realistic deployment of the IoT application, we deployed it on a server running Tomcat (http://tomcat.apache.org) that used a public Internet connection and was located about 30 km away from the devices' local network.

We implemented the heater with a hosted service *SmartHeater* providing seven operations, as Table 1 shows (DPWS operations 3 to 9). These operations use simple command-line messages to indicate each operation's effect, such as "current status: on" and "new status updated: off." We implemented a REST proxy in Java using the Jersey library on Tomcat (http://jersey.java.net) to handle the heater's RESTful Web API. The IoT application either uses the API provided by the REST proxy or directly communicates with the heater (using the WS4D JMEDS library) to carry out the DPWS heater's four functionalities: checking heater status, setting heater status, adding a new rule, and deleting a rule.

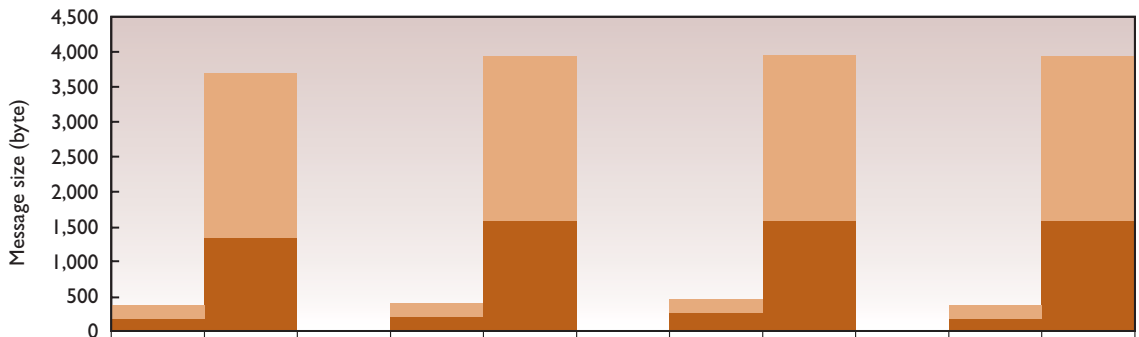
Features Comparison

For the original DPWS communication, we exclude the preprocessing phase that discovers the device information (endpoint and transport addresses). We measure round-trip time (RTT) and message size only for invoking operations. Note that the actual time of the whole process is higher and varies according to implementation strategies. You can choose to have a device discovered and its services invoked in real time, or have the information about the device



	GET /heater GetStatus()	PUT /heater SetStatus()	POST /rules AddRule()	DELETE /rules/2 RemoveRule(2)
PROXY MODE	287.71	285.8	298.54	280.56
CACHE MODE	364.43	364.29	367.65	367.65
DPWS MODE	1,142.35	1,151.07	1,139.15	1,135.48

(a)



	PROXY MODE	DPWS MODE	PROXY MODE	DPWS MODE	PROXY MODE	DPWS MODE	PROXY MODE	DPWS MODE
	Get status		Set status		Add rule		Delete rule	
RESPONSE	201	2,345	199	2,344	201	2,349	193	2,338
REQUEST	167	1,320	203	1,572	248	1,572	171	1,563

(b)

Figure 2. Latency and message overhead evaluation. (a) Mean round-trip time for 100 tests when using the REST proxy (PROXY MODE) and original Devices Profile for Web Services standard with caching (CACHE MODE) and without caching (DPWS MODE) in four cases: GET /heater, GetStatus(); PUT /heater, SetStatus(); POST /rules, AddRule(); and DELETE /rules/2, RemoveRule(2). (b) Message sizes of requests (REQUEST) and responses (RESPONSE) for the same four operations.

stored and then send requests only to invoke the device service. The real RTTs and message sizes will always be higher than those using our proposed REST proxy.

Our design extends the DPWS standard with new features, as Table 2 shows. These features, including global discovery, global messaging, and a RESTful Web API, are necessary to realize the technology for IoT applications. In the meantime, the extension preserves DPWS's publish-subscribe eventing mechanism with an even better messaging format.

Latency and Message Overhead

Figure 2a presents the mean RTTs for an application sending requests and receiving responses for the four operations of the hosted service *SmartHeater*. We used the RESTful Web API from the proxy (PROXY MODE) and original DPWS operations with and without WSDL caching (CACHE MODE and DPWS MODE, respectively). Using the proxy API significantly improves the latency by roughly 75 percent and 25 percent compared to the results in DPWS MODE and CACHE MODE. In many pervasive IoT scenarios requiring high

responsiveness, reasonable delay would improve system performance and the user experience.

Figure 2b shows the message sizes of requests (REQUEST) and responses (RESPONSE) in the four APIs (PROXY MODE) and their counterpart four DPWS operations (DPWS MODE) for fulfilling the same tasks. In DPWS MODE, the messages don't include WSDL documents because we assume that developers choose to cache these documents to preliminarily optimize the application performance (real-time processing of WSDL documents generates more messages). Message overhead improves significantly when we apply the REST proxy. For real deployments of applications and devices in original DPWS communication, nearly full-mesh connectivity (Figure 1b) is unavoidable compared to the simple and linear increments of HTTP traffic in the REST proxy scenario (Figure 1a).

DPWS was designed for use in event-driven IoT applications thanks to features such as eventing and dynamic discovery, which can't be supported natively with HTTP. They use SOAP-over-UDP multicast and SOAP-over-HTTP binding, which are, in practice, limited in network range and introduce considerable overhead by using SOAP envelopes. Instead, our REST proxy extends the DPWS standard to better integrate it into IoT applications and the Web while maintaining its advantages. Our experimental results show a significant improvement in reducing latency and overhead as well as simplifying the global topology of using a RESTful Web API. To use our REST proxy design in the future, a standard will be necessary for designing DPWS services for different devices and for the dynamic generation of a RESTful Web API. Also, we must further investigate its adoption in several scenarios with real-time constraints or in high dynamicity, such as in military and disaster monitoring applications.

Acknowledgments

This work is supported by two European ITEA projects: 10028 Web of Objects (WOO) and 11020 Social Internet of Things: Apps by and for the Crowd (SITAC).

References

1. Z. Shelby, K. Hartke, and C. Bormann, "Constrained Application Protocol (CoAP)," IETF Internet draft, June 2013.
2. *Devices Profile for Web Services Version 1.1*, Oasis standard, July 2009.
3. "Web Services Architecture," W3C working group note, Feb. 2004.
4. T. Cucinotta et al., "A Real-Time Service-Oriented Architecture for Industrial Automation," *IEEE Trans. Industrial Informatics*, vol. 5, no. 3, 2009, pp. 267–277.
5. P. Spiess et al., "SOA-Based Integration of the Internet of Things in Enterprise Services," *Proc. IEEE Int'l Conf. Web Services* (ICWS 09), 2009, pp. 968–975.
6. R.T. Fielding, "Architectural Styles and the Design of Network-Based Software Architectures," PhD dissertation, Donald Bren School of Information and Computer Sciences, Univ. of California, Irvine, 2000.

Table 2. Features comparison between DPWS* and the extended proxy.

Features	DPWS	Proxy
Global discovery	No	Yes
Publish-subscribe eventing	Yes	Yes
Global messaging	SOAP	HTTP
Global topology	Mesh-like	Proxy-based
RESTful Web API	No	Yes
Configuration module	No	Yes

*Devices Profile for Web Services

Son N. Han is a PhD student at Institut Mines-Telecom, Telecom SudParis. His research focuses on the Internet of Things. Han received an MSc in computer science from the University of Seoul. He's a student member of IEEE. Contact him at son.han@it-sudparis.eu.

Soochang Park is a research associate at Institut Mines-Telecom, Telecom SudParis. His research focuses on networking. Park received a PhD from Chungnam National University. He's a member of IEEE. Contact him at soochang.park@telecom-sudparis.eu.

Gyu Myoung Lee is a senior lecturer at the Liverpool John Moores University and an adjunct professor at the Korea Advanced Institute of Science and Technology (KAIST). His research focuses on future networks and services. Lee received a PhD from KAIST. He's a senior member of IEEE. Contact him at g.m.lee@ljmu.ac.uk.

Noël Crespi is a professor at Institut Mines-Telecom, Telecom SudParis. His research focuses on service architecture, and the Internet of Things, people, and services. Crespi received a PhD from Paris VI University. He's a senior member of IEEE. Contact him at noel.crespi@mines-telecom.fr.

Annotating Uncertainty in Geospatial and Environmental Data

The Geography Markup Language (GML) — the existing standard for encoding geospatial data — has no mechanism for annotating such data with uncertainty. To address this issue while supporting the geospatial community's existing data and service standards, the authors extend GML to enable uncertainty markup. They demonstrate this extension's use with some common geospatial data types and Web services. The result is a robust capability to share error information while maintaining compatibility with existing geospatial data clients.

Elias Ioup
US Naval Research Laboratory

Zhao Yang
University of New Orleans

Brent Barré, John Sample, and Kevin B. Shaw
US Naval Research Laboratory

Mahdi Abdelguerfi
University of New Orleans

The Open Geospatial Consortium's (OGC's) geospatial and environmental data services have seen wide adoption throughout several communities and are mandatory for governments that must disseminate these types of data. OGC standards such as the Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS), and Geography Markup Language (GML)[1] are used by diverse user groups that access the data with various geospatial clients. Despite often being mandated, OGC standards lack defined methods for representing uncertainty. Without uncertainty, the larger community can't properly use geospatial and environmental data for navigation, analysis, modeling, or visualization. The consequences of using such data without uncertainty can range from poor weather forecasting to grounding or even wrecking a ship. Our research lets geospatial services developers annotate GML — the standard exchange format for geospatial data — with uncertainty information.

Uncertainty, specifically positional and value accuracy, are among a number of complex and interrelated parameters that represent the overall quality of a geospatial dataset. Other data quality parameters include provenance, consistency, and completeness. More detailed discussions of geospatial data quality are available elsewhere.[2,3] Here, we focus primarily on data's positional and value accuracy. While beyond this article's scope, we discuss issues of data provenance (lineage) in other work.[4] The results we present here aren't a complete solution for communicating the quality of geospatial data. However, our method annotates positional and data value uncertainty in a way that's descriptive, easy to integrate into geospatial Web service architectures, and widely compatible with various client applications.

Published by the IEEE Computer Society
1089-7801/15/$31.00 © 2015 IEEE

Our work was guided by specific requirements for annotating uncertainties in geospatial data. First, any approach to describing uncertainties must support the diverse methods available for representing error distributions, including multiple probability distributions and statistics. Second, it must be simple to use for both data producers and consumers. Finally, the approach must support wide use by different Web service clients that can't be modified specifically to handle new uncertainty annotations. As such, we can't modify existing standards in a way that can break compatibility. We must also attempt to match current common practices that, while not official parts of the standard, will increase usability across a large number of users.

Our approach to adding uncertainty annotations to GML meets these requirements. GML supports both gridded and vector data products, making it a good platform for supporting a wide variety of data. Our method for annotating such products uses GML together with the Uncertainty Markup Language (UncertML), an interoperable model for describing uncertainty using XML. We further extend our method to support an uncertainty distribution that UncertML doesn't natively support, and present examples of implemented Web services that incorporate this uncertainty into their data visualizations for users.

UncertML

Our approach uses UncertML to define the data structure of error information. UncertML offers comprehensive representations for different uncertainty types, is expressive in encoding these uncertainties, and is easy to use.[5] It describes uncertainty using statistics (such as mean and standard deviation), explicit probability distributions, and samples. It adopts the philosophy that all data values are inherently uncertain (that is, they're random variables rather than values with defined quality metadata), and their uncertainty should be explicitly quantified and exchanged.[6] Various projects that require uncertainty descriptions use UncertML, including Interoperability and Automated Mapping (Intamap), Quality-Aware Visualization for the Global Earth Observation System of Systems (GeoViQua), and NetCDF Uncertainty Conventions (NetCDF-U).

Attaching Uncertainty to GML

GML provides the core data encoding capabilities we use in this work. Our primary goal is to support sharing environmental data such as bathymetry, currents, or seafloor obstructions via Web services. Environmental data is represented as both gridded and vector data. Because the latest GML standard enables encoding for grids as well as vector products, it's a perfect markup language for environmental information. However, without a method for encoding uncertainty, this vital information is removed from the underlying data when it's shared using GML and the OGC services GML supports.

Combining GML with existing UncertML schemas lets users comprehensively encode and share environmental data without removing its associated uncertainty information. Incorporating UncertML into GML requires determining locations for the new uncertainty elements that don't break the existing GML standard. GML alone doesn't provide a complete mechanism for encoding and disseminating geospatial data. Instead, it provides a set of XML schemas, which we can use to create application-specific schemas that are in turn used for data transmission. Any GML user must employ a separately created application schema; the GML standard only ensures that the geospatial elements in that schema are uniform across applications.

Thus, to achieve our objective of sharing uncertainty information, we need only ensure that the application schemas our services use support encoding it. We don't need to change the GML standard itself. Table 1 provides an outline of how to modify GML application schemas to support adding uncertainty to different geospatial data types.

Motivating Use Case: Bathymetric Data

Our initial target dataset for this work is bathymetry, with uncertainty information associated with each grid value. Our bathymetric data is stored as a rectified grid representing the continuous water-depth field over a region. Each grid location has an associated depth mean and standard deviation that specify the normal distribution used to represent the uncertainty in that grid region. We calculate the uncertainty values when multiple bathymetric datasets are merged into a single grid based on factors such as the amount of data in a grid cell.

Conventional GML encoding. The application schema we employ for our grid with uncertainty must use the native GML geospatial types as its basis. The base type for a coverage is `gml:AbstractCoverageType`, which includes a coverage's basic elements, the domain, and the range. The `gml:domainSet` and `gml:rangeSet`

Related Work in Uncertainty Support

A prime example of adding uncertainty support to the Geography Markup Language is uGML, which provides a method for attaching positional uncertainty to GML elements.[1] In uGML, an `uncertainty` sub-element is added to GML's `AbstractGeometry` element so that every geometry type can support a new metadata field, which can contain any Uncertainty Markup Language (UncertML) uncertainty element. The change is minimal, but effectively modifies the GML standard and could break clients using strict GML parsers. Our larger concern with uGML is where the uncertainty element is placed within the GML feature's geometry. Logically, this placement makes sense given that uGML is focused on the geometry's positional uncertainty. However, GML parsers can ignore the uncertainty elements and not present this information to users. Uncertainty elements encoded as properties are more likely to be parsed and presented to the user by common GML clients. uGML's authors prefer not to place the uncertainty annotation in a separate element because it moves the uncertainty away from the geometry to which it refers. Although we agree with this logic, we believe there is greater value in making uncertainty information visible to users following the more common conventions of GML application schema design. Ultimately, the inadequacies of both approaches show the necessity of amending the GML standard to formally support uncertainty.

Ashley Morris and Frederick Petry also extend GML with positional uncertainty to make a format called UGML (not to be confused with uGML).[2] Their implementation uses a fuzzy-set approach to represent a feature's positional uncertainty. Each feature in UGML contains multiple *alpha-cuts* — that is, boundaries with different associated uncertainties. The reliance on this single method to represent uncertainty is this approach's primary limitation, especially given the limited use of fuzzy-set-theory-based uncertainties with geospatial data. The extension they propose to GML is also problematic because it uses few supported GML capabilities.

Another approach adds positional uncertainty to GML using the ISO 19113 and 19115 standards for data quality and metadata.[3] This approach replaces standard GML geometry types (such as *Polygon*) with complex geometry constructs such as *MultiCurve* aggregate or *Topological* complex when positional data quality varies over a feature. This approach's limitation is that it replaces simple, commonly used GML types with complex, infrequently used types. Extensive customization is required for any application to use this method. Additionally, the ISO 19113 data quality annotations aren't as expressive as those in UncertML and aren't likely to meet the uncertainty annotation needs for many data products.[1]

The Interoperability and Automated Mapping (Intamap) project provides service for automatic data interpolation.[4] This framework originally extended GML 3.1 to include the mean and variance characteristics of requested observations using the Web Feature Service and Web Coverage Service.[4] Further extensions added UncertML support to the Web Processing Service (WPS) standard to communicate data uncertainty.[1] The WPS standard's generic nature enables this flexibility in data encoding but also requires considerable coordinated development to achieve integration between parties.

The Quality-Aware Visualization for the Global Earth Observation System of Systems (GeoViQua) project has developed tools for integrating uncertainty through the geospatial service stack (www.geoviqua.org). GreenLand, based on research originally from the UncertWeb project, is a geospatial client that can render quality-aware data. Examples include using confidence triangles, whitening, and animation to indicate a grid cell's uncertainty (https://wiki.52north.org/bin/view/Geostatistics/Greenland). On the server side, the GeoViQua project has developed an extension to the Web Map Service (WMS) called WMS-Q (for quality).[5] WMS-Q supports mechanisms to attach uncertainty information to map layers (or groups of layers) for client use. Using the UncertML vocabulary, users can annotate a map layer with its probability distribution and have sublayers that represent the parent's mean and variance.

This work has two core limitations that our research overcomes. First, many solutions modify the Open Geospatial Consortium (OGC) standards or common usage patterns for the service. These changes require modifications to existing geospatial tools already in use. The second limitation in some previous work is the use of less expressive methods for representing uncertainty. Our work maintains compatibility with existing OGC standards and common usage patterns while supporting UncertML's expressive uncertainty representation.

References

1. M. Williams, "Probabilistic Uncertainty in an Interoperable Framework," PhD dissertation, Dept. of Computer Science, Aston Univ., 2011; http://eprints.aston.ac.uk/15791/1/Williams,_Matthew_W_2011.pdf.
2. A. Morris and F.E. Petry, "UGML: An Extension of GML to Provide Support for Geographic Objects with Uncertain Boundaries," *Proc. 7th Int'l Symp. Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*, 2006, pp. 794–801.
3. A. Donaubauer, T. Kutzner, and F. Straub, "Towards a Quality Aware Web Processing Service," *Proc. 6th Geographic Information Days*, 2008, pp. 16–18.
4. M. Williams et al., "Supporting Interoperable Interpolation: The INTAMAP Approach," *Proc. Int'l Symp. Environmental Software Systems*, 2007.
5. J. Blower et al., *OGC OWS 9 Data Quality and Web Mapping Eng. Report*, Open Geospatial Consortium, June 2013.

Table 1. Annotating data uncertainty using the Geography Markup Language (GML).

Data type	Uncertainty	GML	Uncertainty annotation	Example
Geometry	Each point has individual uncertainty	Add property `positionalErrorUncertainty` Use `shared="false"`	Encodes the uncertainty of the positional error rather than the positions themselves	Sounding position
Geometry	Multiple points share uncertainty	Add property `positionalErrorUncertainty` Use `shared="true"`	Encodes the uncertainty of the positional error rather than the positions themselves	Shoreline
Property value	Value has a single uncertainty	Define property in application schema using the applicable UncertML uncertainty type	Encodes the uncertainty as part of the data value	Sounding depth
Grid	Individual cells have separate uncertainty	Extend GML coverage (rectified grid) type	Adds the uncertainty element definition to the `valueComponent` in the grid range section If cell value isn't part of the uncertainty element, adds that to the `valueComponent` Places the numerical data for the grid cells' value and uncertainty inside a tuple list	Bathymetric grid
Grid	All grid cells share uncertainty	Extend GML coverage (rectified grid) type	Adds uncertainty element as a sub-element in the new coverage feature defined in the application schema	Sea surface temperature model

properties specify the locations of the coverage points and the value at each point, respectively. The bathymetric data used in this work is a rectified grid, meaning it maps to Earth locations.[1]

Uncertainty annotation for GML grids. To add uncertainty to a grid defined in GML, we must extend the grid definition. As mentioned, this uncertainty information is normally distributed. We use the existing UncertML to extend GML. UncertML provides the `NormalDistributionType` to describe normally distributed data.

To add normally distributed uncertainty to a GML grid, we create a new application schema that defines a new type called `UncertaintyCoverage`, which extends the `DiscreteCoverageType` in GML and imports the UncertML schema elements. The uncertainty itself is included in the actual GML document's `<gml:rangeSet>` element. We specify the range set's value components using the `<un:NormalDistribution>` element, which includes sub-elements for specifying the mean and variance. The `<un:mean>` and `<un:variance>` sub-elements contain the identifier `template`, which specifies how to map the actual numerical data to the range's value components. The numerical values are subsequently stored as a list in the `<DataBlock>` of `<gml:rangeSet>`. This list contains numeric tuples in which each tuple's dimension is equal to the number of range parameters. Figure 1 shows the sample uncertainty data in GML using the uncertainty schema.

```
<gml:rangeSet>
   <DataBlock>
      <gml:rangeParameters>
         <gml:CompositeValue>
            <gml:valueComponents>
               <un:NormalDistribution>
                  <un:mean>template</un:mean>
                  <un:variance>template</un:variance>
               </un:NormalDistribution>
            </gml:valueComponents>
         </gml:CompositeValue>
      </gml:rangeParameters>
      <gml:doubleOrNilReasonTupleList>
         2698 26.98 2678 26.78 2657 26.57 2637 26.37
      </gml:doubleOrNilReasonTupleList>
   </DataBlock>
</gml:rangeSet>
```

Figure 1. Sample uncertainty data in the Geography Markup Language. The GML data range is defined using the UncertML normal distribution element. The data values in the tuple list are the mean and variance of the normal distribution for each depth location.

```
<!-- Element -->
<xs:element name="CircularErrorProbability"
   substitutionGroup="un:AbstractSummaryStatistic">
   <xs:complexType>
      <xs:complexContent>
         <xs:extension base="un2:CircularErrorProbabilityType" />
      </xs:complexContent>
   </xs:complexType>
</xs:element>

<!-- Circular error Complex Type -->
<xs:complexType name="CircularErrorProbabilityType">
   <xs:complexContent>
      <xs:extension base="un:AbstractSummaryStatisticType">
         <xs:sequence>
         <xs:element name="error" type="un2:QuantityType"/>
         </xs:sequence>
         <xs:attribute name="probability" use="required">
            <xs:simpleType>
               <xs:restriction base="xs:double">
                  <xs:minInclusive value="0.0" />
                  <xs:maxInclusive value="1.0" />
               </xs:restriction>
            </xs:simpleType>
         </xs:attribute>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>

<!-- Quantity type with unit of measure attribute (for CEP error) -->
<xsd:complexType name=”QuantityType">
      <xsd:simpleContent>
         <xsd:extension base="xsd:double">
            <xsd:attribute name="unitOfMeasure" type="xsd:anyURI"
                  use="optional"/>
          </xsd:extension>
      </xsd:simpleContent>
</xsd:complexType>

<!-- This schema sample allows importation of UnitsML elements. -->
<xsd:any minOccurs="0" maxOccurs="unbounded"
   namespace="urn:oasis:names:tc:unitsml:schema:xsd:UnitsMLSchema-1.0"
   processContents="strict"/>
```

Figure 2. XML schema defining the circular error probability (CEP) uncertainty element. The element has both an error and probability. A CEP error is a measurement, so the schema supports importing UnitsML elements, which can then be associated with the error.

Note that these changes don't fundamentally modify the GML standard, and the resulting GML documents with uncertainty remain valid for existing clients and parsers.

Extending Support to New Distributions

UncertML might not include elements necessary to describe a desired uncertainty representation. For example, the World Vector Shoreline

(WVS) dataset defines its uncertainty in terms of *circular error probability* (CEP), an intuitive measure of uncertainty defined as the radius of a circle, centered about the mean, whose boundary is expected to include a specific percentage of the population within it.

UncertML has no definition for circular error type. To represent this type of uncertainty easily, we extended UncertML to support CEP. Because the data structure of the quantile statistic in UncertML is similar to CEP, we modeled our new statistic definition on the quantile type. One modification is how we handle the error value. Because CEP's error value is a measurement with units, we added a new `QuantityType` that supports a UnitsML unit of measure reference as an attribute, following the OGC *Units of Measure Recommendation* and the *UnitsML Guide*.[7,8] The schema can include portions of UnitsML for use in these references. Figure 2 shows our new XML schema defining the CEP uncertainty element.

Next, we use the new XML schema in our GML, attaching the uncertainty to an entire collection of features rather than a single grid cell. To do this, we created a new `CEPFeatureCollection` type that extends the GML `AbstractFeatureCollectionType`. Inside this new element, we added an attribute for `CircularErrorProbability`.

Annotating Vector Data Type Uncertainty

We extended our methodology to enable annotating the uncertainty of vector data to support nongridded data (for example, depth soundings, shorelines, and hazard areas). Again, our aim is to maintain compatibility with GML. We approach the problem using the recommended method of placing the uncertainty as a property into the application schema that our data uses. For existing feature properties, we merge the uncertainty with the data value. Figure 3 shows an example of the depth property with a normal distribution. Instead of a separate value for depth, we simply encode the mean and standard deviation. The mean takes the place of a standalone depth value. We also incorporate the method of attaching units used when we defined CEP. The depth element contains a `unitOfMeasure` attribute referring to a UnitsML element listed later.

Annotating positional uncertainty is more difficult because we can't modify the geometry elements from the GML schema. Instead, we add a new property called `positionalErrorUncertainty` that provides the error uncertainty for the GML feature's geometry. Figure 4 shows an example for a point feature.

```
<gml:featureMember>
   <sounding gml:id="797439">
      .
      .
      .
      <depth unitOfMeasure="#uml_meters">
         <un:NormalDistribution>
            <un:mean>20.0</un:mean>
            <un:variance>2.5</un:variance>
         </un:NormalDistribution>
      </depth>
   </sounding>
</gml:featureMember>
<unitsml:Unit xml:id="uml_meters">
    <unitsml:UnitName xml:lang="en-US">meter</unitsml:UnitName>
   <unitsml:UnitSymbol type="ASCII">m</unitsml:UnitSymbol>
</unitsml:Unit>
```

Figure 3. Depth property with a normal distribution. The depth element is encoded as a mean and variance pair. It also contains a `unitOfMeasure` *attribute referring to a UnitsML element listed later.*

Notice that we encode the uncertainty of the error, not the uncertainty of the position itself. This approach reduces our logical discomfort of separating the data value in the GML geometry from the uncertainty annotation. The error uncertainty also indicates the geometry type that's being modified — in the Figure 4 case, a point. Should the feature include multiple geometries, `positionalErrorUncertainty` will include further geometries (matching the ordering to indicate which error goes with which geometry). The shared attribute in the error geometry specifies whether each coordinate value has an individual error (as in Figure 4) or whether they all share the same error. In the former case, each coordinate value will have its own UncertML distribution listed in the order the coordinates appear in the geometry. In the latter case, only a single uncertainty will be present and shared by all coordinates. Geometries with multiple points will have an uncertainty for each. Figure 5 shows an example for a polygon. The `exterior` and `LinearRing` elements are included in the error uncertainty to simplify the correlation between the error uncertainties and multiple boundaries in the polygon.

Uncertainty-Enabled Services

Ultimately, we include uncertainty GML application schema and instance documents to

```
<gml:featureMember>
   <sounding gml:id="797439">
      <geometry>
         <gml:Point>
            <gml:pos dimension="2">-90.9320253369377 29.2337370854158</gml:pos>
         </gml:Point>
      </geometry>
      <positionalErrorUncertainty>
         <Point shared="false">
            <un:Probability le="0.0001" ge="-0.0001">
               <un:probabilities>0.95</un:probabilities>
            </un:Probability>
            <un:Probability le="0.0002" ge="-0.0002">
               <un:probabilities>0.90</un:probabilities>
            </un:Probability>
         </Point>
      </positionalErrorUncertainty>
      .
      .
      .
</gml:featureMember>
```

Figure 4. Example for a point feature. We extend our GML feature with a new property called `positionalErrorUncertainty`*, which provides the error uncertainty for the GML feature's geometry.*

support data sharing using Web services. The bathymetry use case is meant to support data sharing via WCS. We use the WFS when we want to distribute vector data objects (points, lines, and polygons), as with shoreline data. The WMS provides a standard mechanism for sharing images.

Our architecture uses the WFS and WCS to serve raw data to clients and the WMS to serve rendered images of this raw data. The WMS itself acts as a client to the WFS and WCS. This architecture design gives users multiple options for accessing data depending on how they wish to use it.

Bathymetry

In our bathymetry application, we connect a grid visualization WMS to the underlying WCS that serves the raw bathymetry data encoded as GML. Once the uncertainty is added to the bathymetry GML, the WMS can visualize it in the images it provides clients. Here, we demonstrate a confidence triangles approach.[9] Each grid rectangle is rendered as two triangles representing the upper and lower bound to the grid value confidence interval. Figure 6 shows the bathymetry WMS's output. Figure 6a is the layer without any uncertainty visualization, whereas 6b is the same data with uncertainty identified by confidence triangles. Although we highlight only one uncertainty visualization method here, many others exist for gridded data, including chrominance modification, glyph overlays, and transparency. One benefit of this standards-based approach is that we can offer these and additional methods for visualizing the grid as different layers (or styles) of a single WMS and let users choose which method works best for their application. The detailed data value and uncertainty information are available by clicking on the map.

Coastal Mapping

Coastal mapping is another application in which proper uncertainty representation and communication are important. Here, we have multiresolution shoreline data that's rendered and displayed to users. Again, we implement the service architecture described earlier. A WFS feeds shoreline data into a WMS, which visualizes the map for the user. Properly displaying the coastline's accuracy is an important safety consideration.

```
<gml:featureMember>
   <hazardarea gml:id="1150">
      <geometry>
         <gml:Polygon>
            <gml:exterior>
               <gml:LinearRing>
                  <gml:posList dimension="2">
                     40.56793975830078 -14.211050033569336
                     40.56792449951172 -14.210745811462402 40.56840896606445
                     -14.210102081298828 40.56793975830078
                     -14.211050033569336
                  </gml:posList>
               </gml:LinearRing>
            </gml:exterior>
         </gml:Polygon>
      </geometry>
      <positionalErrorUncertainty>
         <Polygon shared="false">
            <exterior>
               <LinearRing>
                  <un:Probability le="0.0001" ge="-0.0001">
                     <un:probabilities>0.95</un:probabilities>
                  </un:Probability>
                  <un:Probability le="0.0003" ge="-0.0003">
                     <un:probabilities>0.95</un:probabilities>
                  </un:Probability>
                  <un:Probability le="0.0001" ge="-0.0001">
                     <un:probabilities>0.90</un:probabilities>
                  </un:Probability>
                  <un:Probability le="0.0002" ge="-0.0002">
                     <un:probabilities>0.90</un:probabilities>
                  </un:Probability>
                  <un:Probability le="0.0004" ge="-0.0004">
                     <un:probabilities>0.85</un:probabilities>
                  </un:Probability>
                  <un:Probability le="0.0004" ge="-0.0004">
                     <un:probabilities>0.85</un:probabilities>
                  </un:Probability>
                  <un:Probability le="0.0003" ge="-0.0003">
                     <un:probabilities>0.95</un:probabilities>
                  </un:Probability>
                  <un:Probability le="0.0002" ge="-0.0002">
                     <un:probabilities>0.90</un:probabilities>
                  </un:Probability>
               </LinearRing>
            </exterior>
         </Polygon>
      </positionalErrorUncertainty>
   </hazardarea>
</gml:featureMember>
```

Figure 5. An example polygon. The uncertainty of each coordinate in the polygon is represented in the probability elements. The probabilities match the order of the polygon position values.

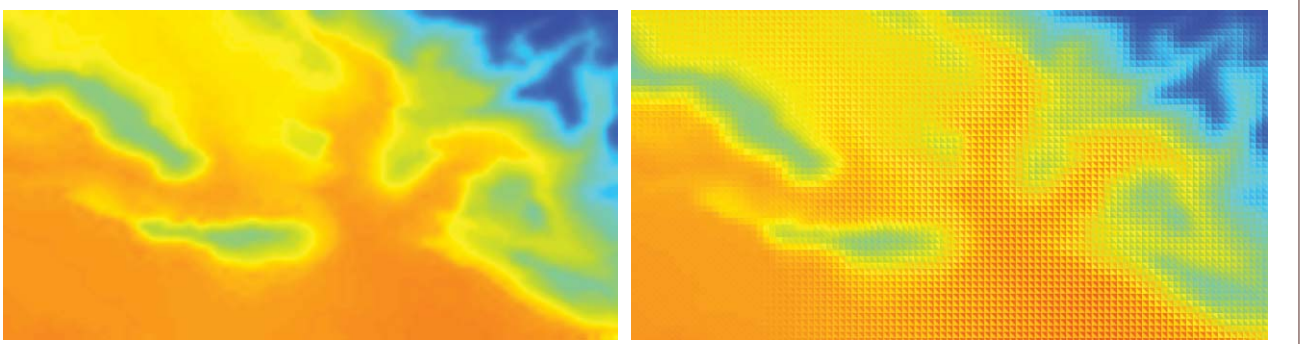


Figure 6. Map output from our uncertainty-enabled bathymetry Web Map Service (WMS). We can see (a) the rendered bathymetry with no uncertainty and (b) the uncertainty identified by confidence triangles. The WMS makes both layers available.

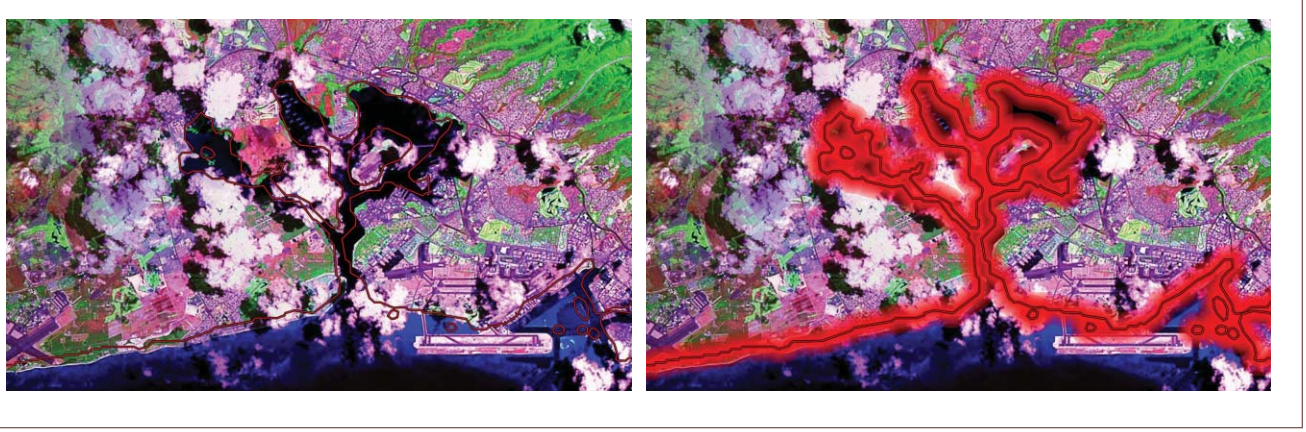


Figure 7. A visualized map based on World Vector Shoreline data. (a) The Web Map Service layer (at a scale of 1:250,000) is overlaid on Landsat imagery. (b) By visualizing the data uncertainty annotations, the error is displayed as a semitransparent buffer around the shoreline.

The data source used in this application is WVS, and it includes coastline features for the world at different resolutions. Unfortunately, the uncertainty is rarely transmitted along with the data, leading to visualizations that are misleading or dangerously inaccurate. Figure 7a illustrates a shoreline rendered without uncertainty and overlaid on top of imagery. The difference in the coastline is readily apparent. Neither the imagery nor the WVS shoreline is intrinsically wrong; the WVS data is simply being rendered at an inappropriately large map scale. However, a user looking at the shoreline via a Web service wouldn't know this if the shoreline weren't placed on top of the imagery. Thus, uncertainty annotations are needed in the raw data for proper visualization in the resulting maps.

WVS's uncertainty is defined by a CEP, which varies depending on the shoreline's resolution. The highest-resolution shoreline — a 1:250,000 scale — has 90 percent of its features within 500 meters of the true geographic location. The WVS shoreline is represented by a polyline data type with a single uncertainty for the entire feature. Consequently, we add the `positionalErrorUncertainty` element to the feature to hold the uncertainty information. The uncertainty type is the CEP uncertainty element we created previously. A WMS then renders the resulting raw data to provide a better visualization of the shoreline's accuracy at whatever scale is rendered. Figure 7b is a version visualized with uncertainty. In the updated version, the error is displayed as a semitransparent buffer around the shoreline, which removes the discrepancy between the WVS shoreline and the imagery. The resulting visualization gives the user an intuitive notion of the data's underlying accuracy as well as the proper map scale.

The two use cases we've presented highlight our approach's usefulness and demonstrate how uncertain data can be used in a larger workflow including visualizations.

Future work in this area should focus on several different aspects of using uncertainty with geospatial and environmental data and services. As discussed in the introduction, data quality is a complex concept that includes numerous parameters. Future work should incorporate other aspects of data quality into the annotation framework, especially provenance and completeness. Further research is also necessary to investigate how downstream tools that use uncertain geospatial and environmental data incorporate these multiple aspects of data quality. Lastly, effort must go into updating the OGC standards, especially GML, to directly support the annotation of geospatial features with uncertainty information.

References

1. *OpenGIS Geography Markup Language Encoding Standard version 3.2.1*, OpenGIS implementation specification OGC 07-036, Open Geospatial Consortium, Aug. 2007.
2. X. Yang et al., "An Integrated View of Data Quality in Earth Observation," *Philosophical Trans. Royal Society A: Mathematical, Physical, and Eng. Sciences*, vol. 371, Dec. 2012.
3. P. Díaz et al., "Analysis of Quality Metadata in the GEOSS Clearinghouse," *Int'l J. Spatial Data Infrastructures Research*, vol. 7, 2012, pp. 352–377.
4. K. Shaw et al., "Passively Repurposing Hyperspectral Data by Formally Modeling Provenance," *Proc. 17th World Multi-Conference on Systemics, Cybernetics, and Informatics*, 2013, vol. 2, pp. 165–169.
5. M. Williams et al., "UncertML: An XML Schema for Exchanging Uncertainty," *Proc. 16th Ann. Conf. GIS Research*, 2008, pp. 275–279.
6. M. Williams, D. Cornford, and L. Bastin, "Describing and Communicating Uncertainty within the Semantic Web," *Uncertainty Reasoning for the Semantic Web Workshop, Proc. 7th Int'l Semantic Web Conf.*, 2008, pp. 32–43.
7. J. Bobbitt, *Units of Measure Use and Definition Recommendations*, OGC recommendation, Sept. 2002; http://portal.opengeospatial.org/files/?artifact_id=11498.
8. M. Weber, *UnitsML Guidelines Version 1.0*, Oasis specification, Sept. 2011; www.oasis-open.org/committees/download.php/42538/UnitsML-Guide-v1.0-wd01.pdf.
9. E.J. Pebesma and J.W. de Kwaadsteniet, "Mapping Groundwater Quality in the Netherlands," *J. Hydrology*, vol. 200, nos. 1–4, 1997, pp. 364–386.

Elias Ioup is a computer scientist in the Geospatial Computing section of the US Naval Research Laboratory. His research interests include high-performance geospatial data processing, geospatial and environmental Web services, and geospatial data visualization. Ioup received a PhD in engineering and applied science from the University of New Orleans. Contact him at elias.ioup@nrlssc.navy.mil.

Zhao Yang is a PhD student in the Department of Computer Science at the University of New Orleans. His research interests include geospatial systems and big data. Yang received an MS in computer science from the University of New Orleans. Contact him at zyand1@uno.edu.

Brent Barré is a computer scientist in the Geospatial Computing section at the US Naval Research Laboratory. His research interests include visualization of geospatial and environmental data, and geospatial Web services. Barré received an MS in computer science from the University of New Orleans. Contact him at brent.barre@nrlssc.navy.mil.

John Sample is the head of the Geospatial Computing section at the US Naval Research Laboratory. His research interests include techniques for high-performance geospatial data extraction, advanced caching techniques, and geospatial tiling systems. Sample received his PhD in computer science from Louisiana State University. Contact him at john.sample@nrlssc.navy.mil.

Kevin B. Shaw is the head of the Office of Geospatial Science and Technology Innovation within the Marine Geosciences Division at the US Naval Research Laboratory. His research interests include geospatial data management and the exploitation of provenance information to enable the repurposing of scientific data. Shaw has a PhD in engineering and applied science from the University of New Orleans. Contact him at kevin.shaw@nrlssc.navy.mil.

Mahdi Abdelguerfi is a professor in and chair of the Department of Computer Science at the University of New Orleans. His research interests include spatial and spatiotemporal information systems, parallel processing of geospatial data, and geosensor networks. Abdelguerfi received a PhD in computer engineering from Wayne State University. Contact him at mabdelgu@uno.edu.

Context Awareness as a Service for Cloud Resource Optimization

Cloud computing architectures involve different actors that can be divided into four categories: infrastructure- and platform-as-a-service providers, PaaS clients, and end users. These actors establish contracts with each other with the intention of optimizing their costs and offering quality service. Paying attention to end users' context and providing context awareness as a service at the PaaS provider level offers several benefits, including that it lets PaaS clients develop multimodal services and thus optimize infrastructure use.

Christophe Gravier and Julien Subercaze
Univeristé Jean Monnet

Amro Najjar
Henri Fayol Institute

Frédérique Laforest
Univeristé Jean Monnet

Xavier Serpaggi and Olivier Boissier
Henri Fayol Institute

Cloud computing is a value chain that implies synallagmatic contracts — that is, contracts with mutual obligations and rights — between the parties. Here, we consider a cloud computing architecture composed of four contracts with different (and sometimes conflicting) stakes and issues. As Figure 1 shows, the result is a four-tier model:

- The *infrastructure-as-a-service* (IaaS) provider offers cloud infrastructure to its clients through a (hopefully standards-based) API; the goal is to cover its capital expenditure (CAPEX) by maximizing the loan time of its commodity servers.
- The *platform-as-a-service* (PaaS) provider is a cloud user that instantiates virtual machines (VMs) into the IaaS provider's infrastructure through its API; the PaaS provider's goal is to satisfy its customers while minimizing its operating expense (OPEX), which is primarily the cost of renting VMs from the IaaS provider.
- The *PaaS client* is a company whose developers use the PaaS to propose online cloud-hosted services to PaaS customers using the PaaS provider's development and runtime tools; the goal is to deliver a specified service to end users (usually under best-effort conditions).
- *End users* buy services from the PaaS client, with no idea about which actor or infrastructure lies behind the PaaS client; the end user's goal is to easily access the cloud-hosted service using heterogeneous devices, ranging from smartphones to IPTV to PCs and tablets.

Existing cloud approaches generally optimize each actor's CAPEX and OPEX

 Published by the IEEE Computer Society 1089-7801/15/$31.00 © 2015 IEEE

by reacting to or predicting significant evolution in the infrastructure's quality of service to manage cloud elasticity. However, a more fine-grained adaptation is possible because end users' contexts might imply different service modalities with different resource needs. The bottleneck here is the lack of information on the context, which is required to create more efficient elasticity rules in the cloud.

We propose taking the user's context into account at the PaaS provider level, which (as Figure 1 shows) is a cornerstone of the cloud infrastructure and is directly connected to end users. The PaaS provider level — and its inherited, conflicting situation of trying to satisfy end users while optimizing its OPEX — is largely understudied. Meanwhile, PaaS is gaining momentum; Gartner forecasts a compound annual growth rate of 17.1 percent through 2018.[1]

Before describing the details of our solution here, we first offer a reference scenario and an overview of the PaaS provider's stakes and issues.

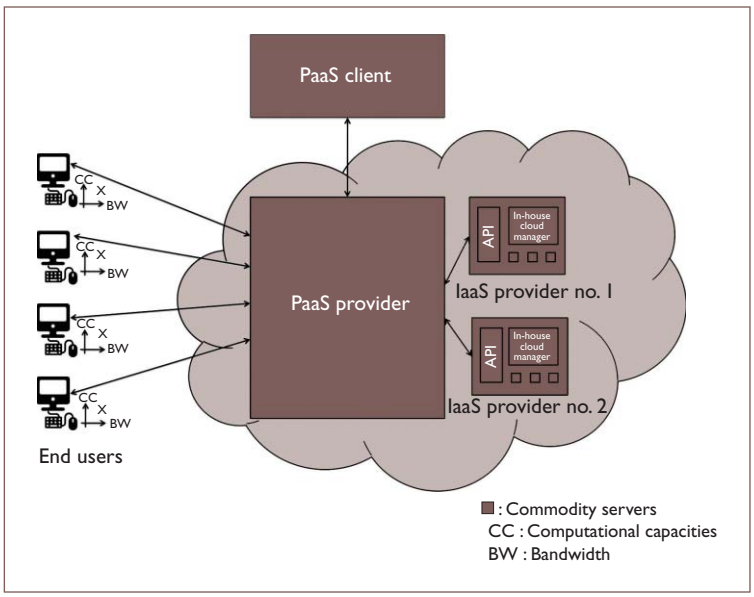


Figure 1. Global view of a classical cloud architecture using a platform as a service. The PaaS client is blind to the context of service consumption when it comes to cloud elasticity.

Scenario and PaaS Cost Model

To set the context for our discussion, we offer an example scenario and a typical PaaS provider cost model for renting resources from an IaaS provider.

Reference Scenario

Our example scenario uses the following four-tier cloud architecture:

- *IaaS provider*: Amazon Web Services (http://aws.amazon.com), a collection of cloud computing services.
- *PaaS provider*: Heroku (www.heroku.com), a PaaS that provides services in several programming languages.
- *PaaS client*: Alice, who provides a multimedia streaming service.
- *End user*: Bob, who accesses Alice's service on his smartphone.

We chose multimedia streaming as Alice's service because such services are becoming increasingly important with the advent of HTTP streaming solutions such as Dynamic Adaptive Streaming over HTTP (DASH)[2] and HTTP LiveStreaming.[3] Alice has observed that her clients sometimes connect with different devices that have different screen sizes, resolutions, computational capacities for video decoding, and so on. In her view, even if a smartphone is using Wi-Fi at home, there's no point in wasting bandwidth offering a high-quality video version because smartphone screen resolution can't take advantage of it. Sending the high-quality version also introduces latency because the video data is too large. Thus, users like Bob might have to wait for video chunks — and might ultimately give up the service as a result. Moreover, when video quality is too high for a device, it can take a toll on the batteries while decoding data that's not fully exploited and doesn't contribute to quality of experience.

This scenario can be duplicated to numerous devices, each of which typically requires its own configuration. Given this, Alice wants to provide an adaptive streaming service that can encode a video on-the-fly and perfectly fit any device that connects to her service. She expects to both deliver a better user experience and optimize her costs.

PaaS Cost Model

A key challenge for PaaS providers is to satisfy the service-level agreements (SLAs) they enter into with their customers, while decreasing the associated costs. The ongoing issue is to identify how a PaaS provider could offer the service and decrease its PaaS client's OPEX, while keeping end users' same perceived value. This perceived value depends on how end users access the service. As Figure 1 shows, however, the PaaS

provider has only a limited understanding of the service consumption context. As a consequence, to honor its SLA with end users, the PaaS client aims to provide a commodity service, which we define as a *unique service delivery setting* shared by all its customers. Regarding cloud elasticity for the PaaS provider, which in turn is a customer of the IaaS provider, the PaaS provider can create an elasticity function (specified as ε here). This function gives the number of required VMs for a given timeframe — typically based on the number of connected end users at that time. So, for a given time frame τ, σ_τ is the number of end users connected to the PaaS-hosted application and η_τ is the number of VMs to loan to the IaaS provider for this time frame τ to serve the σ_τ clients; α is the default number of VMs when no end users are connected to the PaaS at runtime. That is,

$$\eta_\tau = \varepsilon(\sigma_\tau) + \alpha \mid \eta_\tau \in \mathbb{N}.$$

In practice, the PaaS client usually writes rules for cloud elasticity based on its empirical knowledge that a single VM for a given flavor can handle a maximum of β end users. The number of different flavors is usually a limited set (such as small, medium, large, and extra-large VM configurations). In this case, $\varepsilon(x) = x/\beta$, and our equation thus becomes

$$\eta_\tau = \frac{\sigma_\tau}{\beta} + \alpha \mid n_\tau \in \mathbb{N}.$$

In pragmatic cloud computing management, the PaaS client knows how many end users a single VM can handle; this lets it set up an elasticity strategy that optimizes η_τ.

Usually, this linear relation isn't continuous but is instead discrete and encoded using business rules.[4] When triggered, these rules scale up and down the number η_τ of loaned VMs.[5] Depending on the IaaS provider, the rules are

- encoded in the PaaS management framework and applied using the IaaS provider API (a PaaS rule engine senses and executes elasticity rules at runtime); or
- filled on a Web configuration front-end by a PaaS provider operator.

IaaS provider leaders such as Amazon or RackSpace usually give their clients both means of programming cloud elasticity. However, the second configuration is chosen primarily when the IaaS provider is hosting the entire PaaS infrastructure, which tends to be the predominant model today. Prediction models that heavily use machine-learning techniques enhance these thresholds-based models.[6,7] Such models typically rely on features that are more complex than the number of connected clients, such as the server-side monitoring frameworks. In general, the optimization objective relies on the fact that the commodity service is the same for any user, so it doesn't have to handle user-specific features.

Providing Context Information on the PaaS

As we now describe, context-awareness information can positively participate in the decision-making process for cloud elasticity.

Context Awareness

It's beyond our scope here to provide a complete state of the art in context awareness, but *IEEE Internet Computing* recently published a literature review.[8] Here, we define context from the PaaS provider's viewpoint as any information that describes either the end user's situation or the rented resources needed to deliver the service. The provenance of context is therefore a fourfold source of information that encompasses

- end user devices, such as battery levels, access network, and location;
- network sensors to measure resources such as bandwidth and traffic;
- user preferences, such as preferred video resolution and pingback modality upon job completion; and
- cloud providers' APIs, such as the number of rented VMs, their flavors, and so on (the Cloud Application Management for Platforms, or *CAMP*, is a common cloud management interface[9]).

As we describe in more detail later, the three main benefits of context awareness at the PaaS level are the ability to optimize infrastructure costs; offer context awareness as a service in the PaaS, which frees developers from a certain amount of code, configurations, and runtime monitoring; and permit self-adaptation of the PaaS.

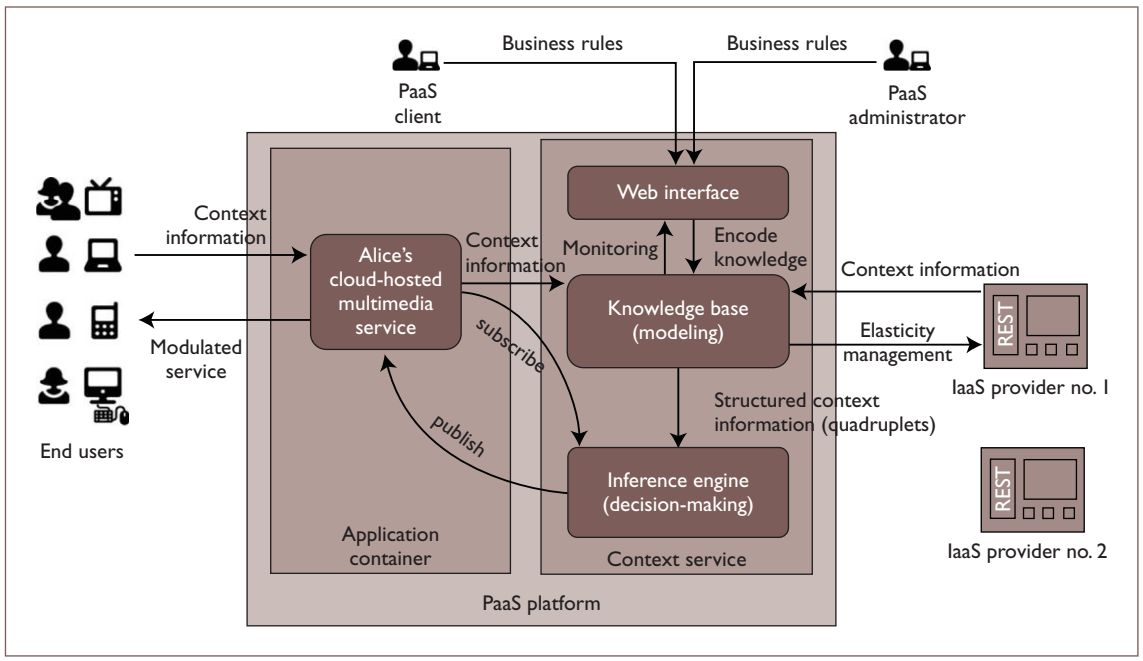


Figure 2. Management and exploitation of context awareness across the different cloud layers in a four-tiers cloud ecosystem. The PaaS platform is enhanced with context-awareness capacities so that any change of service modality delivered to the end user has an impact on the cloud elasticity management.

Managing and Extracting Context

A PaaS provider should be able to model, infer, and make decisions based on the end user's context information. If modeled correctly, the user's context can provide interesting hints on how to tune the service to better match user expectations as well as to minimize infrastructure costs. At runtime, when the PaaS deploys the service, different clients can connect.

The PaaS acts as a cornerstone in the cloud ecosystem for context awareness management. The PaaS client (when context awareness is motivated by a multimodal application) or the PaaS provider (when motivations come from improving PaaS self-awareness) encode service modality for delivery as a quadruplet. This quadruplet specifies the service modality according to the service to be modulated with respect to context information and the necessary cloud resources to be rented. We assume that this knowledge is encoded using an ad hoc Web interface on the PaaS platform.

At runtime, the PaaS platform regularly samples context information from the four different context sources. Interoperability is achieved through RESTful APIs, which the literature discusses for cloud end users[10] and providers.[9] The PaaS platform's Context Service decides whether to adapt the service delivered or rent more or less cloud resources. This Context Service implementation varies from simple yet scalable approaches, such as decision trees, to fine-grained yet hardly scalable techniques, such as one from Semantic Web standards (for ontologies and rule engines).

Once the service modality is inferred, the decision is routed to the hosted application through message routing in the application server (using the publish-subscribe paradigm) for client-side service modulation, or to the cloud provider APIs if the decision implies loaning more or less resources. Figure 2 illustrates this entire landscape.

Benefits of a Context-Awareness Service

Our assumption is that, at the PaaS level, context awareness could help enhance the development and runtime environments that a PaaS provider delivers. We now describe the benefits of context awareness from the PaaS viewpoint.

Optimize Infrastructure

Providing different application behaviors according to the end user's context lets the PaaS provider optimize the required infrastructure to run the service. Because it can offer better insight into the user's context (including computational

Table 1. Alice's proportions and relative costs for various customer contexts.[13]

Context	Proportion of clients (*P*) (%)	Relative cost (*C*)
HDTV	23	1
Laptop	38	0.93
Tablet	26	0.38
Smartphone	13	0.19

capacities, storage, and remaining batteries), the PaaS client can deploy different strategies depending on what the user's side computes and, on the PaaS-side, elasticity and so on. Few studies have investigated the exploitation of end user context data as input for decision-driven elasticity of services or infrastructures. Among these studies, Marcos D. Assuncao and his colleagues proposed a model for scheduling server-based jobs depending on local variations in the user's context.[11] They demonstrate that end user context awareness contributes to improving the overall user experience as well as rationalizing the cloud computing environment's resource use. Dan Miao and his colleagues address a similar issue for achieving maximum quality of experience for mobile users in the context of a cloud-hosted free viewpoint video-rendering application.[12] Context awareness could let developers encode rules to execute different data management and computation ability strategies at runtime based on context.

When the PaaS provider offers fine-grained architecture management at the development stage, the runtime architecture better matches the user's service-consumption context. Coarse-grained provisioning models are roughly based on using a single VM model to represent VMs' needs to shrink or enlarge the cloud infrastructure. A fine-grained representation of the end user context allows for a more optimized provisioning model.

The gain is the savings between a full-featured service blind to the end user's context, minus what each service modality can actually consume if optimized according to the PaaS client's service-adaptation strategy. In our example scenario, a lower-resolution service created when Bob uses a mobile phone would require fewer VMs for a distributed encoding algorithm; it would thus save on infrastructure costs and increase service delivered to the client (due to lower latency, better optimization, less energy consumption for decoding, and so on).

As an example, Table 1 shows the breakdown of Alice's clients in their access context and the relative associated bandwidth and VM costs. The highest costs are for the cable HDTV context. The proportion of each device is illustrative, yet not arbitrary; we derived them from a 2012 Google study.[13] This study focused on multiscreen usage on the Web and television and found that 77 percent of TV viewers don't use a television. Of that 77 percent, 49 percent use a smartphone (each with its own particular context and resolution), and 34 percent use a PC/laptop. For the sake of our example, we'll assume the remaining 17 percent are tablets. We infer costs from the fact that, in this example, Alice's service sends data to each device with different resolutions, and we decrease the bitrate (therefore bandwidth cost) as the resolution changes. These costs are therefore proportional to the difference in resolution between the device considered and the reference HDTV device:

- HDTV: 1920 × 1080
- Laptop: 1620 × 1200
- Tablet: 1024 × 768
- Smartphone: 800 × 480

Obviously, this is just an illustration; the costs might not be this linear, and they depend mainly on the application.

Table 1 lets us determine Alice's savings after she has encoded elasticity rules involving her end users' context elements. We can precisely compute the optimal gain *G* as follows:

$$G = 1 - \sum_{i=1}^{i=|Context|} P_i \times C_i = 0.293$$

If context awareness were provided as a service to Alice's developers, she could allocate and therefore pay for about 30 percent of her resources, while maintaining acceptable service quality. This ultimately results in better OPEX optimization of Alice's cloud service. As a corollary, Alice is more likely to choose a PaaS provider that offers context awareness as a service.

Multimodal Cloud-Developed and -Hosted Applications

Context awareness not only helps optimize OPEX for PaaS clients, it also makes it possible to enhance the service delivered to the end user. This

is key, given that cloud market competition has already driven prices to near zero levels,[14] therefore minimizing OPEX costs; consequently, PaaS clients need another differentiator (mainly, quality of service).[15] By using the end user context as a data source — that is, a *context source* — in the PaaS toolkit, the PaaS provider can develop and host applications that better match users' service consumption contexts. For instance, by giving the developer information about a user's social network and geolocation, the PaaS provider could build applications that adapt behavior according to the social proximity of the user's friends.

Other systems could decrease their bandwidth consumption and thus preserve battery life in end user devices. Because personal and interactive applications are increasingly moving to the cloud, providing a user-tailored service is highly important.[15]

For Alice, a multimodal cloud-developed and -hosted adaptive video-streaming application could be more easily created and deployed using a context framework provided by her PaaS provider. Alice's application would likely better fit the end user's service-consumption context because it would let Alice maximize the tradeoff between perceived service and device capabilities by exploiting user context. Thus, for Alice, providing an adaptive cloud service means adding value to her service.

Improving the PAAS Self-Awareness

Adaptation decisions based on an end user's context can go beyond business logic modulation to enable the PaaS provider itself to perform self-adaptation.

By accounting for the end user's context, the PaaS could go as far as adapting itself to this context. This could serve various purposes. For example, assuming the application was multimodal, a PaaS could be personalized at runtime, and size its memory and CPU footprints accordingly. Thus, the PaaS provider could activate or deactivate a service on the PaaS side depending on the user's needs. This "a la carte" PaaS service activation would enhance the servers' memory footprint. The cumulative effect of this memory savings per VM might mean that more PaaS VM instances could fit on the same host machine because they would presumably be more lightweight than the full-featured activated ones.

Other scenarios of PaaS adaptation are also possible. Let's say Alice delivers videos of soccer games, and the European Soccer Cup features two different semifinal games simultaneously at two different venues. During half time, spectators at both games would likely use their handheld devices to watch the highlights of the other semifinal game. From the PaaS provider's viewpoint, all spectators at each game share the same geographical context; Alice's cloud service runtime environment on the PaaS could automatically detect this and therefore instantiate more VMs in a datacenter close to each stadium. Previous studies show that the cloud's geographic proximity to end users can significantly impact system performance. In one study,[16] for example, latency was cut in half when the VM was moved closer to end users. The PaaS middleware's self-adaptation is a win–win for both end users and the PaaS provider.

Adding context awareness to the PaaS development toolkit and runtime environment serves multiple objectives, including that it's added value for the PaaS provider because it lets the provider's clients provide multimodal cloud services as well as scale their infrastructure needs based on user context.

Of the four tiers in the cloud architecture, one tier — the PaaS provider — is at the heart of our work. The PaaS provider must constantly manage its cost model (primarily, the resources it rents from its IaaS) and its revenue model (its service delivery to end users anytime and anywhere). Right now, the service is delivered with some elasticity, because the cloud is naturally dynamic. However, it's also possible for the service to be modulated from full-featured to a lightweight service with user-specified features. Given their position in the cloud infrastructure, PaaS providers could offer a way to tune up or tune down their loaned cloud resources or service modalities, depending on the policies written by the application developers or PaaS administrators.

Acknowledgments

Our work was supported by the OpenCloudware project (http://opencloudware.org), which is funded by the French Fonds National pour la Société Numérique (FSN) and Pôles Minalogic, Systematic, and Solutions Communicantes Sécurisées (SCS). We also thank the three reviewers for their careful reading of this article and their helpful comments.

References

1. F. Biscotti et al., *Market Trends: Platform as a Service, Worldwide, 2013-2018, 2Q14 Update,* report, Gartner Group, 19 June 2014; www.gartner.com/doc/2773418?srcId=1-2819006590&tpcp=itg.
2. T. Stockhammer, "Dynamic Adaptive Streaming over HTTP – Standards and Design Principles," *Proc. ACM Conf. Multimedia Systems*, 2011, pp. 133–144.
3. R. Pantos and W. May, eds., "HTTP Live Streaming, version 10," IETF Internet draft, work in progress, Oct. 2012.
4. L. Rodero-Merino et al., "From Infrastructure Delivery to Service Management in Clouds," *Future Generation Computer Systems*, vol. 26, no. 8, 2010, pp. 1226–1240.
5. M. Armbrust, "A View of Cloud Computing," *Comm. ACM*, vol. 53, no. 4, 2010, pp. 50–58.
6. A. Quiroz, "Towards Autonomic Workload Provisioning for Enterprise Grids and Clouds," *Proc. 10th Int'l Conf. Grid Computing*, 2009, pp. 50–57.
7. A. Ganapathi, "Statistics-Driven Workload Modeling for the Cloud," *Proc. IEEE Int'l Conf. Data Eng. Workshops*, 2010, pp. 87–92.
8. P. Mehra, "Context-Aware Computing: Beyond Search and Location-Based Services," *IEEE Internet Computing*, vol. 16, no. 2, 2012, pp. 12–16.
9. *CAMP, Cloud Application Management for Platforms, version 1.0*, Oasis, Aug. 2012.
10. J.H. Christensen, "Using RESTful Web-Services and Cloud Computing to Create Next Generation Mobile Applications," *Proc. 24th ACM SIGPLAN Conf. Companion Object-Oriented Programming Systems Languages and Applications,* 2009, pp. 627–634.
11. M.D. Assuncao et al., "Context-Aware Job Scheduling for Cloud Computing Environments," *Proc. Int'l Conf. Utility and Cloud Computing*, 2012, pp. 255–262.
12. D. Miao et al., "Resource Allocation for Cloud-Based Free Viewpoint Video Rendering for Mobile Phones," *Proc. ACM Int'l Conf. Multimedia*, 2011, pp. 1237–1240.
13. D. Pham, "Navigating the New Multi-Screen World: Insights Show How Consumers Use Different Devices Together," Google Mobile Ads Blog, 29 Aug. 2012; http://googlemobileads.blogspot.com/2012/08/navigating-new-multi-screen-world.html.
14. D. Durkee, "Why Cloud Computing Will Never Be Free," *Comm. ACM*, vol. 53, no. 5, 2010, pp. 62–69.
15. T. Hobfeld, "Challenges of QoE Management for Cloud Applications," *IEEE Comm.*, vol. 50, no. 4, 2012, pp. 28–36.
16. A. Shakimov et al., *Vis-á-Vis, Online Social Networking via Virtual Individual Servers*, tech. report TR-2008-05, Duke Univ., 2008.

Christophe Gravier is an associate professor of computer science at Université Jean Monnet. His research interests are in context-aware systems, cloud computing, distributed systems, and Web information extraction. Gravier received a PhD in computer science from Université Jean Monnet. Contact him at christophe.gravier@univ-st-etienne.fr.

Julien Subercaze is a senior researcher at Université Jean Monnet. His research interests include distributed systems and data mining, focusing on Web information extraction algorithms and latent semantic-preserving hashing schemes for Web documents. Subercaze received a PhD in multiagent systems from the Institut National des Sciences Appliquées de Lyon. Contact him at julien.subercaze@univ-st-etienne.fr.

Amro Najjar is a doctoral student at the Institut Fayol, Ecole Nationale Suprieure des Mines of Saint-Etienne. His research interests include quality of experience-driven elasticity policies and multiagent systems for cloud computing architectures. Najjar received a master's degree in computer science from ENS Mines Saint-Etienne. Contact him at najjar@@emse.fr.

Frédérique Laforest is a professor at Télécom Saint Etienne, the engineering school of Université Jean Monnet, where she leads the Satin Team, working on social network recommendation and semantic information management on the Web and in cloud architectures. Laforest received a PhD in computer science from the Institut National des Sciences Appliquées de Lyon. Contact her at frederique.laforest@univ-st-etienne.fr.

Xavier Serpaggi is an associate professor at Ecole Nationale Suprieure des Mines of Saint-Etienne, France. His research interests include quality of experience in the cloud, trust management systems, and the Web of Things. Serpaggi received a PhD in image processing from ENS Mines Saint-Etienne. Contact him at serpaggi@emse.fr.

Olivier Boissier is a professor of computer science at Ecole Nationale Suprieure des Mines of Saint-Etienne, France, where he coordinates the Informatique pour les Systèmes Coopératifs, Ouverts et Décentralisés research group in computer science. His research interests are in multiagent systems. Boissier received a PhD in computer science from INP Grenoble and an HdR from ENS Mines de Saint-Etienne and Université Jean Monnet. Contact him at boissier@emse.fr.

JTangCSB

A Cloud Service Bus for Cloud and Enterprise Application Integration

Cloud and enterprise application integration (CEAI) is a new challenge as more enterprise applications migrate successfully to cloud environments. However, achieving CEAI properties such as multitenancy, cloud-level scalability, and environmental heterogeneity is nontrivial for enterprise applications originally built under the assumption of centralized management. A concrete implementation of the cloud service bus approach, called JTangCSB, addresses the challenge of handling these properties and demonstrates its feasibility in practice.

Jianwei Yin and Xingjian Lu
Zhejiang University, China

Calton Pu
Georgia Institute of Technology

Zhaohui Wu and Hanwei Chen
Zhejiang University, China

To offer maximum value to their users, enterprise applications in cloud environments must not only cooperate with other cloud-based services but also keep data synchronized with on-premises software. According to a recent survey by Oracle, 81 percent of 1,355 surveyed business mangers cite full integration as an important requirement for achieving the benefits of cloud computing (https://emeapressoffice.oracle.com/imagelibrary/downloadmedia.ashx?MediaDetailsID=2905&SizeId=-1).

However, cloud and enterprise application integration (CEAI) is a significant challenge due to the fundamental management assumptions in traditional enterprise-computing environments (centralized) versus cloud environments (open and distributed). Traditional EAI and its associated software tools such as the enterprise service bus (ESB)[1] work well for single companies and stable workloads, but as enterprise applications migrate to the cloud, important assumptions no longer hold. In this article, we examine three properties that are relatively uncommon in traditional EAI but considered critical in cloud environments in general and CEAI in particular:

- *Multitenancy*. We focus on the software-as-a-service (SaaS) model, with cloud services supporting multiple tenants in the same cloud. CEAI challenges related to sharing and isolation management include multitenancy identification, access control, data storage, and service executions, all of which affect performance, reliability, and security.
- *Cloud-level scalability*. With the number and scale of cloud services constantly

 1089-7801/15/$31.00 © 2015 IEEE Published by the IEEE Computer Society

Related Work in Cloud and Enterprise Application Integration

Significant work addresses the challenges of individual cloud and enterprise application integration (CEAI) properties in the context of services computing. We describe some representative examples here for illustration purposes, but new papers are published every year in conferences such as the International Conference on Web Services (ICWS), IEEE Conference on Services Computing (SCC), IEEE International Conference on Cloud Computing (IEEE CLOUD), and ACM Symposium on Cloud Computing (SOCC), as well as in journals such as *IEEE Transactions on Services Computing*. For multitenancy, Steve Strauch and his colleagues investigate the requirements for multitenant enterprise service bus (ESB) solutions, propose an implementation-agnostic ESB architecture that addresses these requirements, and discuss the prototype realization with a performance evaluation.[1,2] For cloud-level scalability, Luis Vaquero and his colleagues discuss the initiatives, relevant efforts, pending challenges, and trends in whole-application scalability in cloud environments.[3] For environmental heterogeneity, Nikolaos Georgantas and his colleagues introduce interoperability solutions based on abstracting and merging the common high-level semantics of interaction paradigms in the future Internet, where complex applications will be composed from extremely heterogeneous systems.[4,5]

Additional research combines two of the three CEAI properties — for example, Françoise Baude and his colleagues illustrate scalability and heterogeneity through a prototype implementation of an Internet-wide service cloud, providing a fully transparent means to access, compose, and deploy services through a federation of distributed ESBs.[6] CloudScale combines scalability and multitenancy in a prediction-driven, elastic, resource-scaling system for multitenant cloud computing;[7] its lightweight and application-agnostic properties make it suitable for large-scale cloud systems. Jiacai Ni and colleagues propose an adaptive database schema design method for multitenant applications that achieves good scalability and high performance with low space.[8] Sanjeev Kumar Pippal and Dharmender Singh Kushwaha combine multitenancy and heterogeneity in a simple, robust, query-efficient, scalable, and space-saving multitenant database architecture for heterogeneous environments.[9]

To the best of our knowledge, JTangCSB is the first research product that explicitly addresses all three CEAI properties through a careful combination of multitenancy-aware data storage and service execution,[1,2,8,9] distributed integration engine and service bus topology,[6,10,11] efficient autoscaling algorithms and targeted performance optimization mechanisms (for cloud-level scalability),[3,7,12,13] and integrated data semantics, as well as various kinds of adaptors (for environmental heterogeneity).[4,5,9]

In industry, some companies and open source organizations also focus on CEAI — for example, MuleSoft provides an integration-platform-as-a-service (iPaaS) product called CloudHub (www.mulesoft.org), and Jitterbit has various integration solutions for small to medium-sized enterprises (www.jitterbit.com). Aobing Sun and colleagues developed a public software-as-a-service (SaaS) platform by using the cloud service bus to support service-oriented architecture-based software and easily combine the primitive mechanisms in SaaS.[11] However, these products primarily focus on the connection problem, with only partial support for the three CEAI properties.

References

1. S. Strauch et al., "ESB MT: Enabling Multi-Tenancy in Enterprise Service Buses," *Proc. IEEE CloudCom*, 2012, pp. 456–463.
2. S. Strauch et al., "Implementation and Evaluation of a Multi-Tenant Open-Source ESB," *Proc. European Conf. Service-Oriented and Cloud Computing* (ESOCC), 2013, pp. 79–93.
3. L.M. Vaquero et al., "Dynamically Scaling Applications in the Cloud," *ACM SIGCOMM Computer Communication Rev.*, vol. 41, no. 1, 2011, pp. 45–52.
4. N. Georgantas et al., "A Coordination Middleware for Orchestrating Heterogeneous Distributed Systems," *Advances in Grid and Pervasive Computing*, Springer, 2011, pp. 221–232.
5. N. Georgantas et al., "Service-Oriented Distributed Applications in the Future Internet: The Case for Interaction Paradigm Interoperability," *Proc. European Conf. Service-Oriented and Cloud Computing* (ESOCC), 2013, pp. 134–148.
6. F. Baude et al., "ESB Federation for Large-Scale SOA," *Proc. ACM Symp. Applied Computing*, 2010, pp. 2459–2466.
7. Z. Shen et al., "CloudScale: Elastic Resource Scaling for Multi-Tenant Cloud Systems," *Proc. 2nd ACM Symp. Cloud Computing*, 2011, article no. 5.
8. J. Ni et al., "Adapt: Adaptive Database Schema Design for Multi-Tenant Applications," *Proc. 21st ACM Int'l Conf. Information and Knowledge Management*, 2012, pp. 2199–2203.
9. S.K. Pippal and D.S. Kushwaha, "A Simple, Adaptable and Efficient Heterogeneous Multi-Tenant Database Architecture for Ad Hoc Cloud," *J. Cloud Computing*, vol. 2, no. 1, 2013, pp. 1–14.
10. J. Yin et al., "A Dependable ESB Framework for Service Integration," *IEEE Internet Computing*, vol. 13, no. 2, 2009, pp. 26–34.
11. A. Sun et al., "CSB: Cloud Service Bus Based Public SaaS Platform for Small and Median Enterprises," *Proc. Int'l Conf. Cloud and Service Computing*, 2011, pp. 309–314.
12. H.C. Zhao et al., "A Unified Modeling Framework for Distributed Resource Allocation of General Fork and Join Processing Networks," *Proc. ACM SIGMETRICS*, 2010, pp. 299–310.
13. H. Chen, J. Yin, and C. Pu, "Performance Analysis of Parallel Processing Systems with Horizontal Decomposition," *IEEE Cluster*, 2012, pp. 220–229.

expanding, CEAI must be scalable in multiple dimensions. Efficient and proactive autoscaling mechanisms must be developed to ensure scalability, not only to increase the number of cloud services offered but also to maintain each service as integration needs grow.

- *Environmental heterogeneity.* Different platforms, languages, APIs, and communication

styles increase the pressure on seamless CEAI. Achieving it for heterogeneous systems will require various adaptors, unified integration semantics, and environment-aware management mechanisms.

The cloud service bus (CSB) approach builds on ESB and service-oriented integration (SOI) concepts to bridge the gap between EAI and CEAI by supporting the provisioning, consumption, and execution of available integration services.[2] CSB is designed to support CEAI properties such as multitenancy, cloud-level scalability, and environmental heterogeneity, and it also includes a marketplace to make global integration services available to CEAI components.

We developed a practical CSB implementation called JTangCSB as a platform for efficient and cost-effective integration service delivery. Its multitenancy-aware mediation components and flows support integration with SaaS in clouds. Distributed integration engine and service bus nodes make the architecture more scalable to the increasing number of cloud services and their expanding scale; autoscaling algorithms and targeted performance optimization mechanisms support large-scale integration and ensure quality-of-service (QoS) requirements. JTangCSB's various adaptors and integrated data semantics also support environmental heterogeneity. We tested JTangCSB with a realistic case study: a company trying to integrate a customer relationship management (CRM) service in an SaaS platform (Salesforce) and an enterprise communications portal (ECP) system in a platform-as-a-service (PaaS) infrastructure (offered by China Telecom) with some on-premises software.

JTangCSB: A Concrete CSB Implementation

As Figure 1 shows, JTangCSB consists of four layers: infrastructure, core assets, business logic, and presentation.

The first layer — infrastructure — contains all the basic resources (computing, network, disk, and so forth) to support JTangCSB's daily operation. It consists of several *JTC-Rendezvous*, each of which represents a distributed resource center over the Internet. The basic unit of JTC-Rendezvous is JTC-VM, which is used to host CSB containers.

The second layer — core assets — contains all the software resources (components, mediation flows, and services). The components are deployed on *CSB containers* — concrete processing nodes that provide a runtime environment for the components. Component instances, mediation flows, and services are all stored in a global semantic space, with mediation flows connecting the component instances (to support cloud-level scalability, components and mediation flows have a cluster implementation). Services represent the cloud applications or on-premises software that are registered with the mediation flows. The component instances and mediation flows are multitenancy-aware, meaning they know which tenants are invoking them and can take appropriate tenant-specific actions. Multitenancy support consists of registries for services, components, and mediation flows, along with a registry center that contains tenant information. A set of tenant and configuration registries stores configuration information for tenants and mediation flows, as well as the mappings to tenants and permissions.

The third layer — business logic — is responsible for processing management functions. By using distributed and scalable bus nodes, the *distributed CSB engine* provides a runtime environment for the mediation flows, including data transformation, message routing, flow orchestration, and pubish-subscribe middleware. It also provides configuration management and runtime monitoring to collect statistical information and logs. To fulfill the defined integration logics, the *integration controller* is in charge of managing integration-related resources. The *mediation flow manager* can define and manage integration scenarios by using other resource managers (service, component, tenant, and configuration managers) to generate multitenancy-aware components and organize them to be multitenancy-aware mediation flows. The *container manager* manages CSB containers, and a component framework OSGi handles the lifecycle management for the mediation components in all the containers. The load balancer evenly distributes workloads in these containers, and the auto-scaling component satisfies variable user demands by adjusting the number of CSB containers. Finally, the *access controller* handles the identification and authentication of all the tenants and users, managing all authorized accesses transparently. It supports the Lightweight Directory Access Protocol

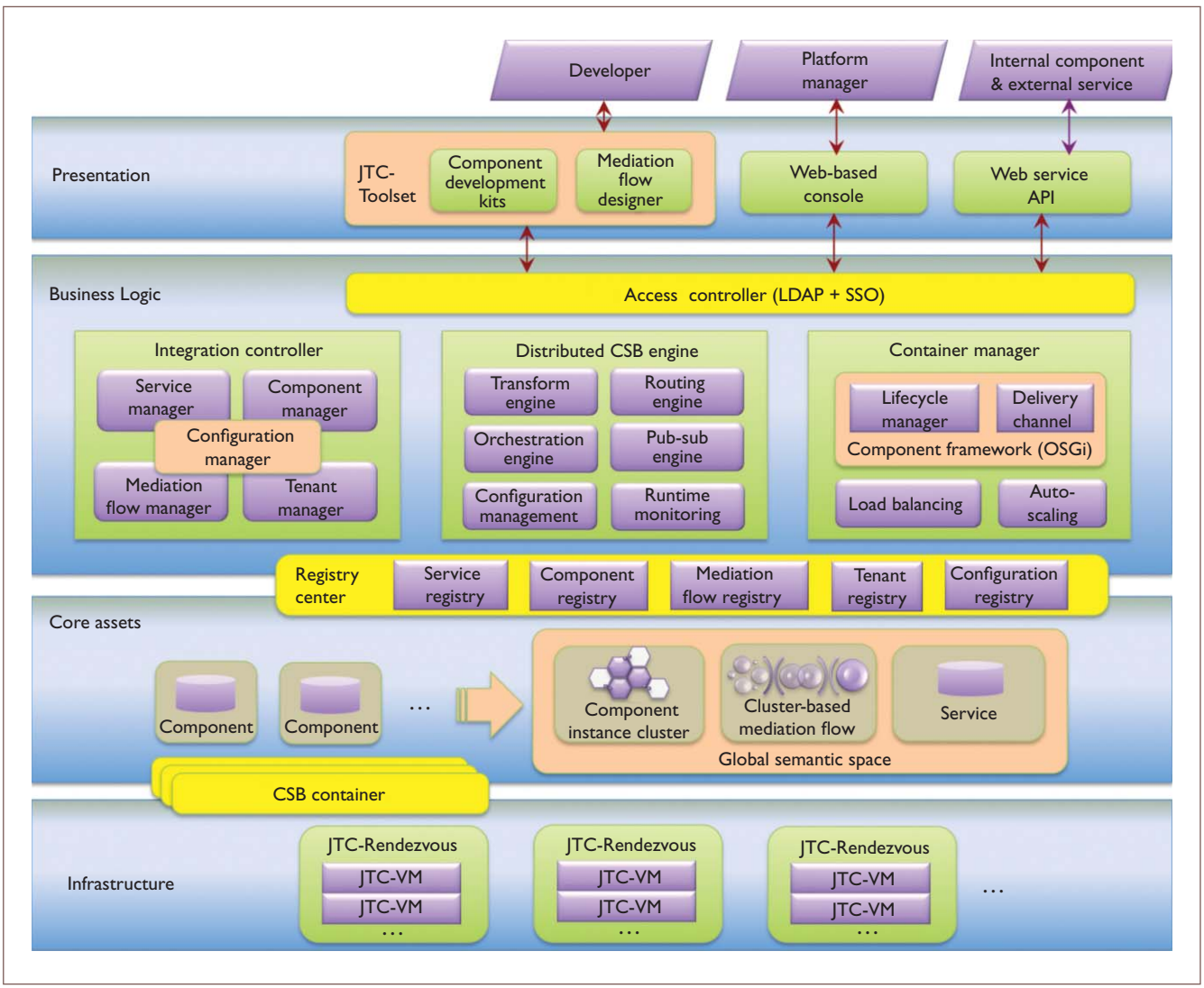


Figure 1. The JTangCSB architecture has four layers: infrastructure, core assets, business logic, and presentation. The infrastructure layer contains all the basic resources (computing, network, disk, and so forth); the core assets layer contains all the software resources (components, mediation flows, and services); the business logic layer is responsible for processing management functions; and the presentation layer contains three components that allow utilization, administration, and interaction with JTangCSB.

(LDAP) and single sign-on services, which can be used to provide role-based access by login once in a session.

The last layer — presentation — contains three components that allow utilization, administration, and interaction with JTangCSB. The *JTC-Toolset* facilitates development tasks for CEAI by providing a mediation component development kit and a mediation flow designer. The two tools are Eclipse plugins for component development and mediation flow design, respectively. The *Web-based console* provides administration, management, and monitoring functionalities for platform managers. Finally, the *Web service API* enables the integration and communication of internal components and external services (cloud applications or on-premises software).

Implementation and Functionality

JTangCSB supports the three CEAI properties through a variety of design and implementation techniques.

Multitenancy Support

To create a multitenancy environment for CEAI, we developed a module in the tenant manager

component based on the architecture that Steve Strauch and his colleagues proposed.[3] Specifically, we deployed a shared LDAP service (based on OpenLDAP) to consistently manage tenant and user information and to provide role-based access in the access controller component. We also implemented single sign-on to let users login once to gain access to all their authorized assets.

To achieve data isolation, we used a shared database for storing multitenancy data. By adding a tenant column for each resource table, the shared registry is multitenancy-aware and provides isolation across tenants.

To support multitenancy service executions, tenant-based deployment and configuration for mediation flows are part of the mediation flow manager's implementation. A tenant can create a new mediation flow instance or register itself with an existing shared instance. JTangCSB can also support each tenant's behavior by adding tenant-specific configurations to the registry center when required. Besides tenant-based deployment and configuration, we designed all adaptors and message processers to be multitenancy aware by providing an identity-distinguishing service in front of the core functionality. With this service, they can identify different tenants and users by their ID and then perform operations according to tenant-specific configurations.

An important consideration in multitenancy support is performance. To support performance scalability as the number of tenants and requests increases, JTangCSB provides a cluster implementation for each component instance (starting from one node). Because component instances of the same mediation flows can have different workload intensities, the finer-grained cluster solution might be more efficient and cost-effective. For each component instance cluster, the autoscaling module will dynamically adjust the number of nodes to adapt to changing workloads. Within a cluster, the load balancer selects the node with the shortest queuing length to serve a request. Due to the variety of workload patterns in a production environment, we also provide interfaces for each component instance cluster to adopt its own customized autoscaling and load-balancing algorithms.

Cloud-Level Scalability Support

As Figure 2 shows, JTangCSB uses a distributed service bus architecture to extend traditional ESB functionality with a bus node's scalable hierarchical organization. Similar to the ESB runtime environment, each PeerBusNode is equipped with a normalized message router (NMR) to handle local communication. The NMR provides the mediated message exchange infrastructure to allow components (adaptors and processers) to interoperate in a standard way via binding components (BCs) and service engines (SEs). BCs provide connectivity between external services and the CSB environment (Web services and so on), and SEs provide business logic and transformation services to other components. A DistributedNMR implements messaging between components hosted in different PeerBusNodes. The DistributedNMR is a virtual cross-node NMR, connected with JMS (Active MQ). Thus, a large number of messages can be handled by PeerBusNodes in a distributed manner, making JTangCSB's messaging capability highly scalable.

Because component instances of the same mediation flows can have different workload intensities, the finer-grained cluster solution might be more efficient and cost-effective.

JTangCSB containers can be deployed in different PeerBusNodes. Because relative physical distances between containers often result in different data transfer rates, the placement of these containers can help achieve high performance, especially for large-scale CEAI projects. JTangCSB does this by reducing the amount of data being transferred via an automated container placement algorithm:

1. The algorithm determines the average message size between any two adjacent nodes based on system logs.
2. The algorithm models each possible container topology as a queuing network (QN). The container is represented by a service center, whereas the Internet between containers is modeled as a load-dependent service center whose service time is based on message size and network bandwidth.

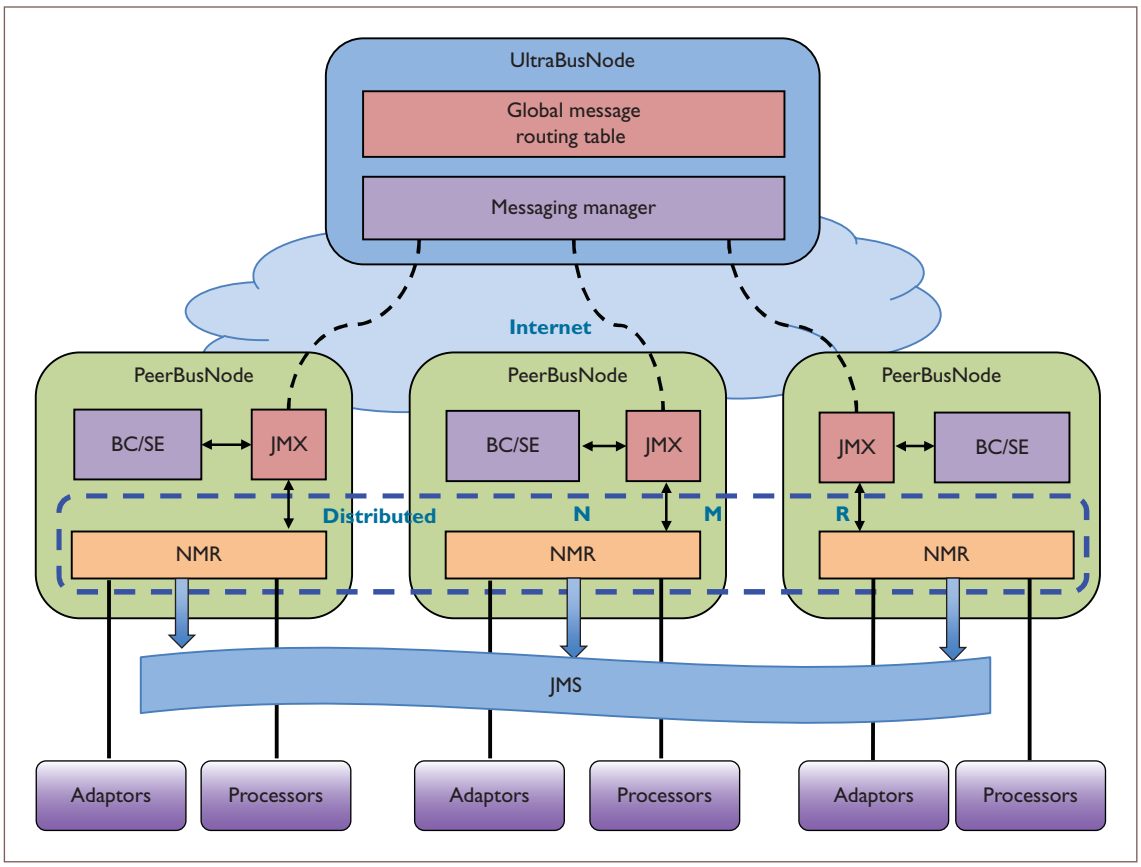


Figure 2. JTangCSB's distributed service bus architecture. Each PeerBusNode is a stand-alone runtime environment for handling communications among local mediation components. The UltraBusNode manages and coordinates communications among PeerBusNodes based on Java Management Extensions (JMX) and the global message routing table.

3. The algorithm calculates each QN's response time.
4. The algorithm selects the topology with the shortest response time and redeploys the containers by using VM migration techniques. The actual topologies can include fork and join operators.[4]

Multiclass fork-join queuing network (FJQN) models are needed for the second and third steps. We proposed an approximated calculation method called horizontal decomposition to solve this kind of problem.[5]

Environmental Heterogeneity Support

JTangCSB provides a variety of adaptors that can be implemented by different languages through OSGi to enable communication with heterogeneous services, thus connecting cloud applications and on-premises software. The SOAP WS, RESTful WS, and database adaptors also translate data formats between Web services within CSB and distributed (heterogeneous) services. Assuming semantic compatibility, the various services connected to different adaptors become connected and integrated, despite different hardware and software platforms, languages, APIs, and tenants.

JTangCSB also supports various communication styles, including message-based, pub-sub, and event-driven — specifically, the CSB engine handles common communication components such as Apache Active MQ, Opensplice DDS, and Esper. To make this heterogeneous data machine understandable, we integrated semantics into the global storage space to support automatic sharing and exchange of heterogeneous data via the open source OWL API and Jena toolset. It's this global semantic space that makes automatic sharing and exchange more efficient for heterogeneous data, despite differences in languages, interfaces, and platforms.

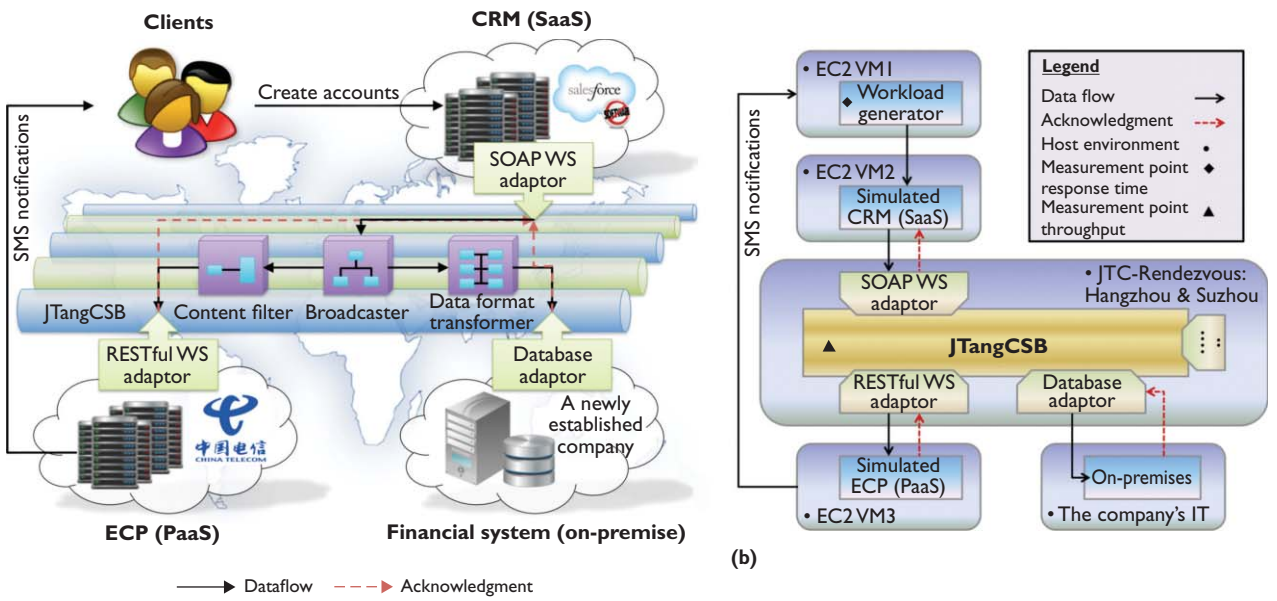


Figure 3. Case study. (a) The integration scenario for client registration: step 1, client creates an account in CRM; step 2, CRM synchronizes client information with an on-premises financial system; and step 3, the case study company's system sends the registration result to the client using ECP's short message service. (b) The evaluation setup. This practical environment helped us evaluate JTangCSB for an integration scenario.

Experimental Evaluation

We tested JTangCSB with a realistic case study, and the experimental results show the feasibility and effectiveness of the CSB approach.

Application Integration Scenario and Evaluation Setup

Using JTangCSB, our case study company defined seven mediation flows with 17 mediation components (six adaptors and 11 message processers). As a representative workload, we used the client registration process in Figure 3a to evaluate JTangCSB's performance. Figure 3b shows the implementation, which includes a CRM adaptor, an ECP adaptor, and a database adaptor to connect with CRM, ECP, and an on-premises financial system, respectively. The adaptors use a broadcaster, a content filter, and a data format transformer to translate and exchange data among them.

To evaluate the configuration in Figure 3b, we deployed JTangCSB as a private integration platform in our two JTC-Rendezvous located in Hangzhou and Suzhou (two cities in China, about 160 km apart). Each JTC-Rendezvous consists of 48 Dell PowerEdge R710 servers; the physical servers within each JTC-Rendezvous were connected by 1-Gbit Ethernet, and the two JTC-Rendezvous were connected by 10-Mbit Internet. Due to publication restrictions on the production environment, we replaced the Salesforce CRM and China Telecom ECP services with equivalent request/response services that simulate CRM and ECP. We deployed the simulated services on Amazon EC2 (VMs on Compute Optimized C1 Medium instances) to maintain their distributed nature – ditto the workload generator (emulated clients). We measured performance metrics such as response time via the workload generator (from request to response) and the throughput via the last component of each running mediation flow.

Multitenant Overhead Evaluation

To evaluate JTangCSB's overhead, we chose the multitenancy support and measured system performance with 2, 10, and 50 tenants. The workload ranged from 800 to 2,000 clients, and the benchmark consisted of requests that invoke one of the seven mediation flows. The invoking ratio among the seven flows is 3:7:10:12:13:20:35, derived from the logs of the company's production system, which was deployed as a commercial application

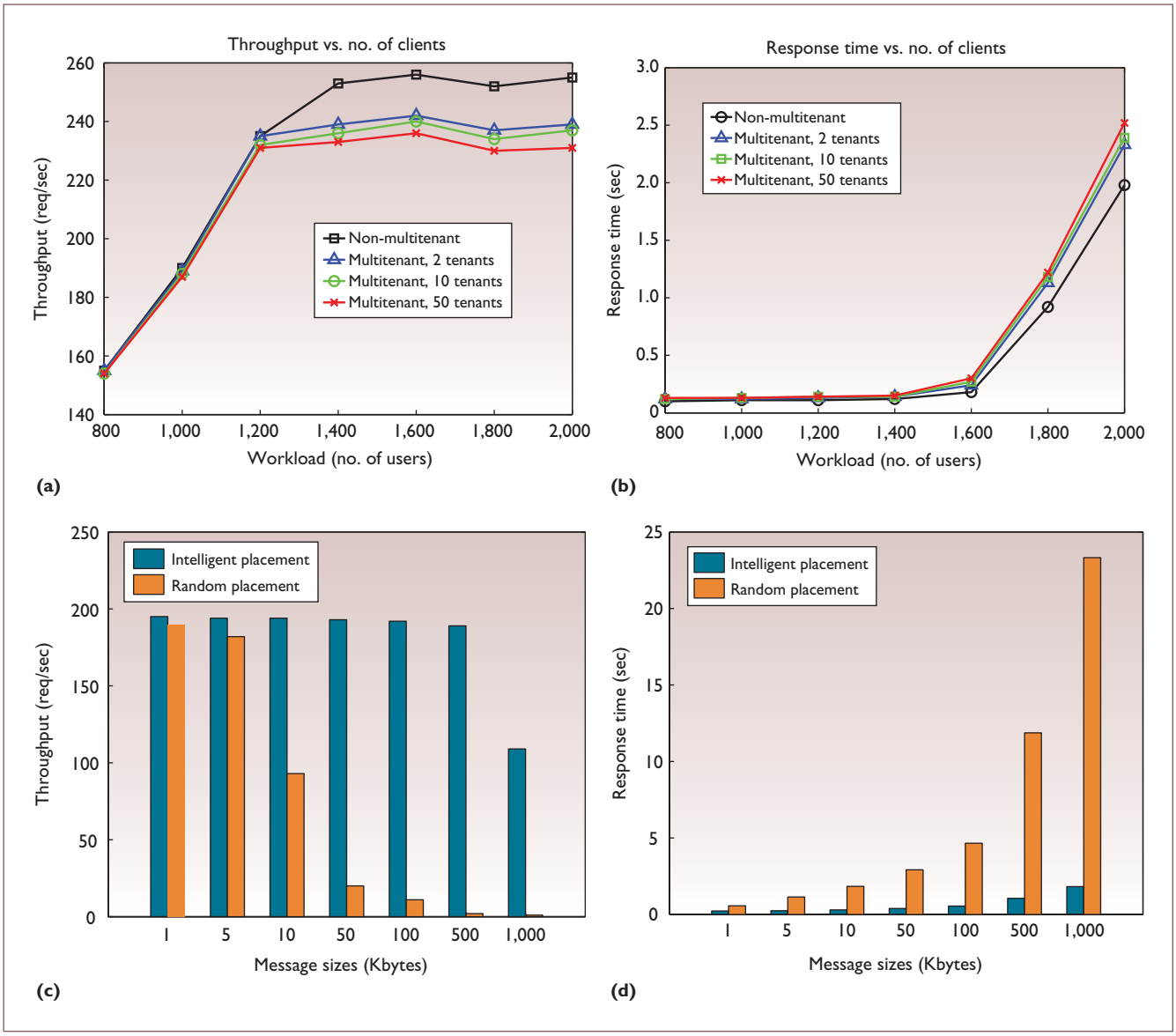


Figure 4. Experimental results of multitenancy overhead and placement optimization: (a) throughput (workload), (b) response time (workload), (c) throughput (message size), and (d) response time (message size).

supported by JTangCSB. Figures 4a and 4b present the measured throughput and response time from the experiments separately.

The multitenant support overhead compared the cases of 2, 10, and 50 tenants with the non-multitenant-aware implementation. The multitenant support overhead is unnoticeable until workload 1,200; for workloads at 1,800 and higher, the average response time becomes too long (higher than 1 second) for realistic service-level agreements. The cases between 1,200 and 1,600 show that multitenant support achieves the same response time, but a small throughput penalty (between 10 and 20 percent). Performance differences for the increasing number of tenants (2, 10, and 50) are quite small (less than 10 percent). Our experimental results show that the overhead introduced by multitenancy in JTangCSB is acceptable, demonstrating effective support of multitenancy.

Container Placement Optimization Evaluation

Evaluating a sophisticated software platform such as JTangCSB is an ongoing process and a subject of active research. To determine whether

JTangCSB could achieve cloud-level scalability, we compared its intelligent container placement method with a simple random placement method. In this experiment, the workload was constant at 1,000 clients, and the message size in the client registration flow varied from 1 Kbyte to 1,000 Kbytes. Figures 4c and 4d show the throughput and response time for the different message sizes. Using the intelligent placement method, the throughput remains high until the message size reaches 1,000 Kbytes. In comparison, the random method causes the throughput to decline as message size grows larger than 10 Kbytes. The response time measurements show the same trend.

The main reason for this response time (and throughput) difference is message delivery overhead via the Internet. In this flow, original client-generated messages will be sent on two paths: one for the ECP and the other for the on-premises financial system. The message size for ECP is always small (less than 1 Kbyte) because of the data transformation filter. However, the message for the financial system is almost the same size as the original message (only the format's been changed). Thus, our intelligent container placement method avoids large amounts of data being transferred over the Internet, giving better performance than the random one.

The growing success of migrating enterprise applications to cloud environments is increasing the pressure to integrate enterprise applications with other distributed services on clouds. Our future work will evaluate JTangCSB in more detail by conducting extensive experiments and adapting it to more application domains, not just the enterprise-computing environment.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant no. 61272129, the National High-Tech Research Program of China (no. 2013AA01A213), the New-Century Excellent Talents Program by the Ministry of Education of China (no NCET-12-0491), the Zhejiang Provincial Natural Science Foundation of China (LR13F020002), and the Science and Technology Program of Zhejiang Province (no. 2012C01037-1).

References

1. J. Yin et al., "A Dependable ESB Framework for Service Integration," *IEEE Internet Computing*, vol. 13, no. 2, 2009, pp. 26–34.
2. F. Baude et al., "ESB Federation for Large-Scale SOA," *Proc. ACM Symp. Applied Computing*, 2010, pp. 2459–2466.
3. S. Strauch et al., "ESB MT: Enabling Multi-Tenancy in Enterprise Service Buses," *Proc. IEEE 4th Int'l Conf. Cloud Computing Technology and Science* (CloudCom), 2012, pp. 456–463.
4. H.C. Zhao et al., "A Unified Modeling Framework for Distributed Resource Allocation of General Fork and Join Processing Networks," *Proc. ACM SIGMETRICS*, 2010, pp. 299–310.
5. H. Chen, J. Yin, and C. Pu, "Performance Analysis of Parallel Processing Systems with Horizontal Decomposition," *IEEE Cluster*, 2012, pp. 220–229.

Jianwei Yin is a professor in the e-Service Research Center at Zhejiang University, China. His research interests include service computing, cloud computing, and healthcare IT. Yin received a PhD in computer science from Zhejiang University. Contact him at zjuyjw@cs.zju.edu.cn.

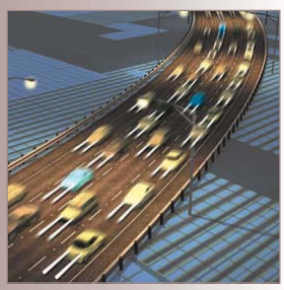
Xingjian Lu is a PhD student in the e-Service Research Center at Zhejiang University; he is the corresponding author for this article. His research interests include service computing and performance engineering. Lu received a BS in computer science from Xidian University. Contact him at zjulxj@zju.edu.cn.

Calton Pu is the John P. Imlay Jr. Chair in Software at the Georgia Institute of Technology. His research interests include cloud computing and Internet data management. Pu received a PhD in computer science from the University of Washington.Contact him at calton@cc.gatech.edu.

Zhaohui Wu is a professor in the e-Service Research Center at Zhejiang University. His research interests include service computing, embedded systems, and pervasive computing. Wu received a PhD in computer science from Zhejiang University. Contact him at wzh@cs.zju.edu.cn.

Hanwei Chen is a research and development engineer at the Industrial & Commercial Bank of China. His technical interests include event-based system and Web services. Chen received a PhD in computer science from Zhejiang University. Contact him at chw@cs.zju.edu.cn.

Track: Best Conference Papers



I Know Where You've Been

Geo-Inference Attacks via the Browser Cache

To provide more relevant content and better responsiveness, many websites customize their services according to users' geolocations. However, if geo-oriented websites leave location-sensitive content in the browser cache, other sites can sniff that content via side channels. The authors' case studies demonstrate the reliability and power of geo-inference attacks, which can measure the timing of browser cache queries and track a victim's country, city, and neighborhood. Existing defenses cannot effectively prevent such attacks, and additional support is required for a better defense deployment.

Yaoqi Jia
National University of Singapore

Xinshu Dong
Advanced Digital Sciences Center

Zhenkai Liang and Prateek Saxena
National University of Singapore

Geolocation is valuable, privacy-sensitive information about users: websites have a strong interest in obtaining it to provide personalized services and advertisements, and hackers[1] want to exploit it for spear phishing and social engineering attacks. Geolocation leakage can cause tremendous damage to a user's privacy.

A traditional way for websites to identify users' locations is through IP addresses,[2] but this method is often unreliable. First, IP address-based geolocation tracking isn't accurate for mobile networks.[3] For example, one recent study[3] found that more than 90 percent of the mobile devices in Seattle could be associated with IP addresses located more than 600 miles away from the city. Second, users can intentionally use anonymization services, such as VPN and Tor (www.torproject.org), to hide their real IP addresses. Recent advancements in mobile devices let websites obtain geolocation information from GPS sensors, but modern browsers disable the access to this data by default to protect user privacy. Nevertheless, even though mobile browsers require users' explicit permission to access GPS data, we show here that attackers can exploit side channels to infer their geolocations without that explicit permission.

Prior research has unveiled numerous privacy-leaking side channels via the browser cache.[4–8] Specifically, timing attacks on the browser cache were introduced to sniff browsing history more than a decade ago;[4] Andrew Bortz and his colleagues later deployed similar timing attacks on additional Web applications and scenarios.[5] The geolocation inference, or *geo-inference*, attacks that we describe in this article are based on the simple assumption that users usually visit location-oriented websites for places where they currently live or plan to visit — for example, when visiting Google's main page, users are

Published by the IEEE Computer Society
1089-7801/15/$31.00 © 2015 IEEE

IEEE Internet Computing | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page | Qmags

automatically redirected to their specific country page for Google, such as www.google.com.sg in Singapore. Starting from this assumption, we conducted experiments on three popular websites, Google, Craigslist, and Google Maps, and found that by conducting geo-inference attacks via the browser cache, attackers can reliably infer a user's country, city, neighborhood, and even home address. After running experiments on five mainstream browsers, we also found that Chrome, Firefox, Safari, Opera, and Internet Explorer are vulnerable to such attacks. Aided by a multicountry proxy service (Hotspot Shield), we then browsed the Alexa Top 100 websites in five different countries (US, UK, Australia, Japan, and Singapore) and found that 62 percent of them contain location-sensitive resources and are susceptible to geo-inference attacks.

Security researchers have proposed various defense solutions against browser-cache sniffing, including cache segregation, which restricts browsers to caching only same-origin resources.[9] Here, we discuss other potential defenses for geo-inference attacks and propose a server-side solution to mitigate such attacks.

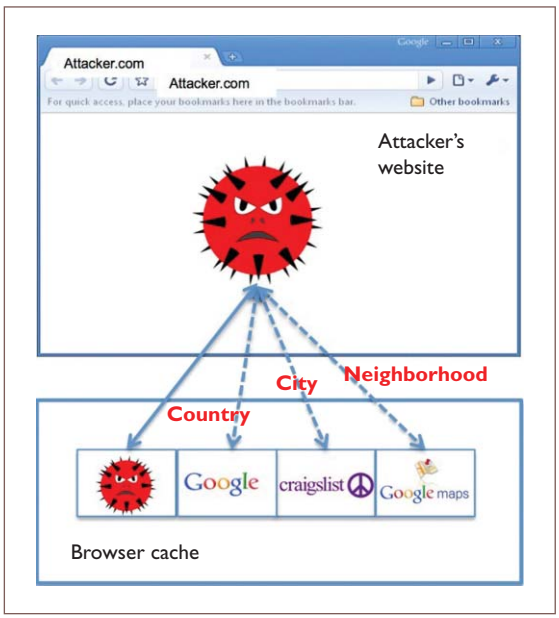


Figure 1. Geo-inference attacks. The malicious website sniffs location-sensitive resources left by location-oriented sites such as Google, Craigslist, and Google Maps via timing side channels in the browser cache.

Problem Definition

The adversary in a geo-inference attack[1] is someone who controls at least one Web server and hosts a malicious site (such as attacker.com) that can run JavaScript on the victim's browser but can't compromise browser code or bypass the same-origin policy.[10] To advertise the malicious site, the attacker can promote it via search engine optimization (SEO) and disseminate its shortened URL through emails, social networks, and advertisements.

Let's assume that Alice is an ordinary Web user who doesn't clear her browser cache frequently; she usually visits sites specific to her geolocation to take advantage of customized local services. When Alice visits the attacker's malicious site in the browser on her desktop, laptop, or mobile device, she denies the unfamiliar site's request to access her device's location. However, the malicious payload in attacker.com can sniff any location-sensitive resources left by sites such as Google, Craigslist, and Google Maps via side channels, as Figure 1 shows.

Browser Cache Timing Channels

To reduce websites' page-load time, today's mainstream browsers utilize memory or disk cache to save infrequently changed resources such as image, JavaScript, and CSS files.

Although browser cache improves page-loading efficiency for users, it also introduces side channels to attackers. By measuring the load time for a specific site twice in a victim's browser, the attacker can determine whether the user has visited the site previously — if the difference is smaller than a certain threshold, for example, the user has been to the site at least once.[4]

Geo-Inference Attacks via the Browser Cache

As part of a case study on browsing history, Gilbert Wondracek and his colleagues de-anonymized social network users by analyzing their visited URLs.[6] In this article, we show the implication of timing attacks in de-anonymizing the user's geolocation. In geo-inference attacks, the attacker makes guesses about the victim's geolocation and then queries the cached resources corresponding to that geolocation. By utilizing the resources left by location-oriented sites in the browser, geo-inference attacks provide an oracle for attackers to verify the user's country, city, or neighborhood, as Figure 1 shows.

For example, let's say that for an anonymous submission to the IEEE Symposium on Security and Privacy, a paper author wants to check which of the 20 program committee members visits her supplementary website (as cited in the paper). Let's suppose that the PC members are from 20 different but known cities: PC members' information is usually public on the conference's website. The author can mount various geo-inference attacks on her supplementary website (which is only available to reviewers) to identify the PC members by querying whether the site's visitors come from any of the 20 cities. The author has a fair chance of successfully inferring a paper reviewer, even if the visit is made through a Web proxy.

Case Studies

To study geo-inference attacks at various granularities on mainstream browsers, we set up a website that contains scripts for exploring the side channels the browser cache has created. We used two well-known timing channels: page-load time and resource-load time. We also used two other vectors for querying that haven't been widely explored: cross-origin resource sharing (CORS) and <img>'s complete property (www.comp.nus.edu.sg/~prateeks/papers/Geo-inference.pdf). Using them together, we can explore these channels to first find the victim's country via Google, infer via Craigslist the city in which the victim is located in that country, and then finally determine his or her neighborhood via Google Maps.

Locate Your Country: Google

To infer a user's country, we must first determine which page from Google's geography-specific domains is in the browser cache. To achieve this, we start with Google's logo image, the URL for which consists of Google's local domain — for example, www.google.com.sg/images/srpr/logo11w.png.

Image-load time is the interval between requesting the image file and when the browser finishes rendering the image. We measured the image-load time of Google logos from Google's 191 regional domains to determine the victim's geolocation, setting the start time as an attribute of the image tag and the end time in the `onload` event handler. We made three measurements for image-load time: the first without cache and the other two with. If the first image-load time was significantly larger, we inferred that the image wasn't cached in the user's browser, implying the user hasn't visited the country-specific site recently. If the time difference was sufficiently small — say, 10 ms — we inferred that the image hits cache, implying the user recently visited that site. Because each logo represents one local domain, and one domain is hosted in each country, where each country's logo is cached out of the 191 options indicates the country in which the victim lives or recently visited.

Below is the code segment we used to measure image-load time:

```
var image = document.createElement(`img');
image.setAttribute(`startTime', (new
Date().getTime()));
image.onload = function()
{
    var endTime = new Date().getTime();
    var loadTime = endTime - parseInt
    (this.getAttribute(`startTime'));
    ......
}
```

We found that the image-load time in each Google domain without cache was much larger than that with cache. Figure 2a shows this difference. Because there's only one request for Google's logo image, the browser directly reads it from the cache when reloading the cached image file, and the average image-load time is usually minimum, say, 1 or 0 ms.

Locate Your City: Craigslist

Several websites offer city-specific content — Craigslist, for example, offers classified advertisements for 712 cities worldwide. All of Craigslist's subsites are city-oriented (such as newyork.craigslist.org) and can be loaded in frames; to locate the user's city, we measured page-load time, setting the start time as an attribute of the iframe and the end time in the `onload` event handler. Specifically, we measured the page-load time of Craigslist's city-oriented websites in the user's country that we inferred from the previous step three times in iframes. If the difference between the page-load time in the first attempt and the last two was significantly large — say, 500 ms — we inferred that the page wasn't cached in the user's browser, implying the user recently hasn't visited the city-specific page; otherwise, the page had been visited. Suppose that the user always

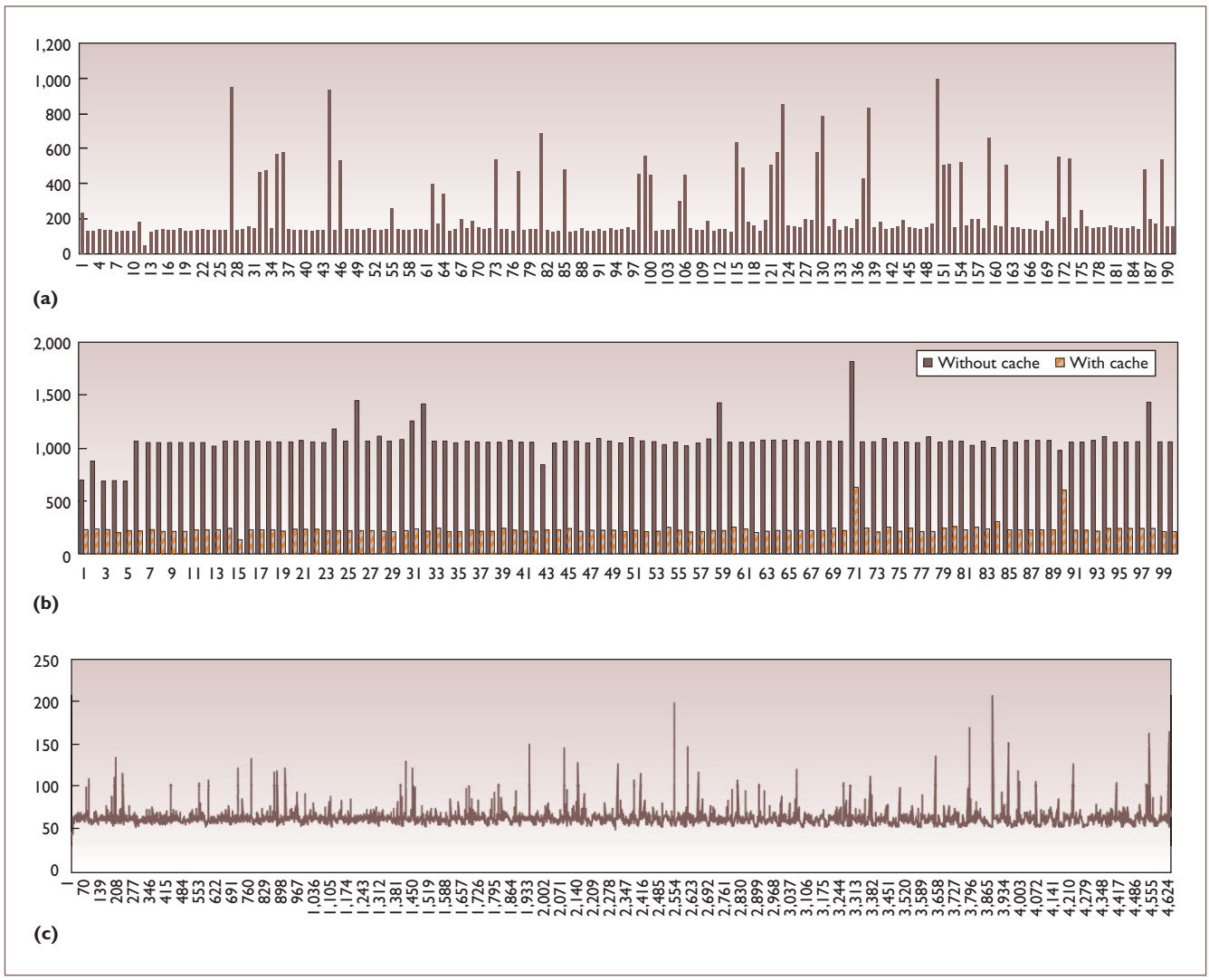


Figure 2. Reliability of geo-inference attacks. (a) Difference in image-load time (in ms) without cache (> 129 ms) versus with cache (0 ∼ 1 ms) for Google's 191 regional domains on a Mac OS X running Chrome in Singapore. (b) Difference in page-load time (in ms) without cache (> 1,000 ms) versus with cache (approximately 220 ms) for 100 Craigslist sites on a Mac OS X running Chrome in Singapore. (c) Difference in image-load time (in ms) without cache (> 50 ms) versus with cache (0 ∼ 1 ms) for 4,646 map tiles of New York City from Google Maps on a Mac OS X running Chrome in Singapore.

buys or sells goods on Craigslist locally in his or her city; from the recently visited page (cached page), the attacker could determine the user's city.

Our attacks reliably identified whether a city-specific page was cached in the user's browser. We considered the average page-load time of the latter two measurements as the time with cache. As Figure 2b demonstrates, the difference is quite large, implying that it's easy for the attacker to distinguish whether the page is cached in the user's browser or not.

Locate Your Neighborhood: Google Maps

To extend our study to a finer granularity, we exploited the cached resources from Google's online map service, Google Maps. These maps are actually tessellations of map tiles, with each tile localized at a predicable URL. After manually analyzing the URLs of the requests for map tiles, we determined that the URLs have patterns that are a function of real geolocation coordinates. For example, we can derive the coordinates (12627, 23720) of "Grand Loop Rd, YellowStone National Park, WY 82190, USA"

Table 1. Prevalence of geo-inference attacks on mainstream browsers.

	Locate country with image-load time	Locate country with cross-origin resource sharing	Locate country with <img> property	Locate city with Craigslist	Locate neighborhood with Google Maps
Chrome (Linux, Windows, OS X + Android, iOS)	√	√	–	√	√
Firefox (Linux, Windows, OS X + Android)	√	–	√	√	√
Safari (Windows, OS X + iOS)	√	–	√	√	√
Opera (Linux, Windows, OS X + Android, iOS)	√	√	–	√	√
Internet Explorer (Windows)	√	–	√	√	√

from the requesting URL (https://www.google.com.sg/maps/vt/pb=!1m5!1m4!1i15!2i12627!3i23720!4i128!2m1!1e0!3m3!5e1105!12m1!1e47!4e0). Although the coordinates aren't the real geographic coordinates, we can predict the URLs of one specific area's map tiles corresponding to the coordinates, as we explain next.

We recorded the URLs of tiles on the map's edge of the user's city, which we verified via Craigslist; next, we derived the coordinates from the URLs and calculated the coordinates of the map tiles within the city's area. We further predicted the URLs of all the map tiles in the user's city. After that, we measured the image-load time of the map tiles in the city. When the user starts to browse the Google Maps website on his or her desktop, laptop, or mobile device, Google Maps usually automatically zooms in to the user's current location based on the GPS sensor or IP address. Thus we can locate the victim's recently visited streets and neighborhood by sniffing the victim's visited map tiles.

We set the start time as an attribute of the image and the end time in the `onload` event handler. In our experiments, we measured the image-load time for all 4,646 map tiles in New York City on Google Maps three times. If the image-load time at the first attempt is much larger than the latter two — say, 40 ms — it indicates a cache miss for the image, implying the user hasn't recently visited a page that contains this map tile. If the difference between each time is quite small — say, 10 ms — the image is in the cache.

We found that the image-load time of map tiles from Google Maps without cache was much larger than that with cache. Therefore, we can reliably determine the user's location by measuring the image-load time of map tiles in the user's city. As Figure 2c shows, the huge difference makes it quite straightforward to distinguish whether the specific map tile is cached in the browser or not. Our technique detects a set of tiles corresponding to city areas recently browsed by the user, which always includes the user's actual location because Google Maps automatically centers the map there when first accessed.

In this case, Google Maps has 4,646 map tiles for the entire city — so, worst case, the attacker needs to make 4,646 queries to locate the victim. According to our experiments, one Web attacker can verify approximately 5 to 10 geolocations every second (as Figure 2c shows), which amounts to eight minutes' time in a single-threaded execution, assuming the attacker doesn't speed up the process with parallel queries. Because multiple map service sites exist, such as Bing Maps and Yahoo Maps, the attacker can also probe the map tiles for these sites with the same technique.

Prevalence of Geo-Inference Attacks

In addition to our case studies with Google, Craigslist, and Google Maps, we found that mainstream browsers on different platforms as well as other popular websites are susceptible to geo-inference attacks.

Susceptible Browsers and Platforms

We conducted additional experiments on five mainstream browsers — Chrome, Firefox, Safari, Opera, and Internet Explorer — on different platforms. As Table 1 shows, the mainstream browsers on both desktop and mobile platforms

are susceptible to geo-inference attacks. To show that geo-inference attacks aren't affected by the user's location, we switched the region to US, UK, Australia, Singapore, and Japan with Hotspot Shield, but we didn't find any large differences in the results for these browsers. Due to limited space, we only show the Singapore-based results in Chrome in this article.

Location-Sensitive Resources in the Alexa Top 100 Websites

To demonstrate the susceptibility of websites to geo-inference attacks, we analyzed the Alexa Top 100 websites and identified location-sensitive sites and resources. We exclude 45 domains that fall into one of the following categories:

- sites highly related to specific countries (google.de or yahoo.co.jp, for example);
- sites known to contain pornographic content (such as xvideos.com and xhamster.com); and
- unreachable sites (such as akamaihd.net and googleusercontent.com).

Using Hotspot Shield, we visited the 55 remaining websites in five different countries (US, UK, Australia, Japan, and Singapore) and recorded the URLs of cached resources for each website. We also automatically compared the collected URLs from different countries with our analysis tool, sorted out the URLs that vary in different countries, and then manually verified the resources with the same functionality but a URL that changes from country to country.

As Table 2 shows, 62 percent of the websites we studied have location-sensitive resources, such as domain and logo, with 34 of them having country- or region-specific URLs, and 20 containing location-sensitive logos. Thus, any user who has recently visited these websites is vulnerable to exposing geolocation information to attackers.

Predictable URLs of Map Tiles in Popular Map Services

We also investigated 11 websites that provide map services: Google Maps, Bing Maps, Yahoo Maps, arcGIS, HERE, OpenStreetMap, ViaMichelin, MapQuest, WikiMapia, Navionics, and Mappy in May, 2014. As Table 3 shows, the URLs of map tiles from these sites are all statically predictable — in the example URL column, the red parts can be predicted based on exact coordinates in the map. Thus, for one area, an attacker can predict all the map tiles in this area and further conduct geo-inference attacks to infer the user's visited streets and likely neighborhood.

Defenses

Technically speaking, geo-inference attacks can be prevented by existing defenses against privacy leakage via the browser cache. However, there are trade-offs to consider when deploying such defenses.

Pros and Cons of Existing Defenses

Before describing our server-side solution, let's review existing defenses.

Private browsing mode is not the answer. Private browsing mode (Private Browsing in Safari and Firefox, Incognito Mode in Chrome, Private Window in Opera, and inPrivate Browsing in Internet Explorer) prevents browsers from permanently storing any history, cookies, or other client-side states for websites. But contrary to what the name suggests, it doesn't prevent the caching of Web resources during users' private browsing sessions. Instead, all cache incurred during private browsing mode is automatically cleared after the user closes the window: users are still susceptible to geo-inference attacks via the browser cache.

Can VPN or Tor prevent attacks? Geo-inference attacks are based on timing attacks against the browser cache, so they aren't affected by VPN services that replace the user's original IP addresses. Although the current version of Tor Browser Bundle (version 3.5.2.1) disables disk cache in browsers by default, memory cache is still active, making it similar to private browsing mode. Browser cache is available until the user closes the browser, at which point the cache stored in memory is invalidated. For browser caches, we also find that TorBrowser adds an additional "domain=string" property to label every cache entry with the top-level window's domain (www.torproject.org/projects/torbrowser/design/#identifier-linkability). Therefore, TorBrowser can protect users from geo-inference attacks when the malicious page's top-level URL is different from the targeted

Table 2. The Alexa Top 100 websites that contain location-sensitive resources.

Domain	Location-sensitive domain name	Location-sensitive logo or image	Domain	Location-sensitive domain name	Location-sensitive logo or image
google.com	√	√	facebook.com	–	–
youtube.com	–	–	wikipedia.org	√	√
yahoo.com	√	√	baidu.com	√	√
qq.com	√	√	amazon.com	√	√
live.com	–	–	twitter.com	–	–
taobao.com	√	√	linkedin.com	√	–
blogspot.com	√	–	wordpress.com	√	–
ebay.com	√	√	bing.com	√	–
vk.com	–	–	tumblr.com	–	–
weibo.com	–	–	pinterest.com	–	–
msn.com	√	√	ask.com	√	√
tmall.com	√	–	microsoft.com	√	√
paypal.com	√	√	apple.com	√	–
instagram.com	–	–	imdb.com	–	–
craigslist.org	√	–	neobux.com	–	–
stackoverflow.com	–	–	adobe.com	√	–
alibaba.com	√	–	fc2.com	–	–
ifeng.com	√	–	imgur.com	–	–
cnn.com	√	√	huffingtonpost.com	√	√
vube.com	–	–	conduit.com	–	–
wordpress.org	√	–	espn.go.com	√	–
flickr.com	–	–	adcash.com	–	–
reddit.com	√	√	aliexpress.com	√	√
xinhuanet.com	√	√	about.com	√	√
godaddy.com	√	√	youku.com	–	–
netflix.com	√	–	dailymotion.com	√	–
sogou.com	–	–	cnet.com	√	√
vimeo.com	–	–			

site. However, for mashup websites, all the embedded sites in frames share the same top-level window's domain – that is, the mashup's domain. The malicious embedded site can query the cache status of other embedded sites' cached location-sensitive resources to infer the user's geolocation. Thus only adding "domain=string" property on the cache entry is insufficient. To prevent timing attacks, other researchers[9] consider two observers: the site embedding the content and the host of the content to label the cache entry. In contrast, TorBrowser only uses the top-level window's domain as the "domain" field to enforce the caching mechanism. Therefore, the researchers' solution defeats geo-inference attacks in the mashup scenario, but TorBrowser does not. We've run our experiments on TorBrowser v3.5.2.1 and found that TorBrowser can still leak users' geo-location information for mashup websites.

Segregating browser cache works but is expensive. Current browsers don't have explicit boundaries or restrictions among different

Table 3. Map service websites with predictable URLs.

	Predictable URLs	Example URL
Bing Maps	√	http://ak.dynamic.t1.tiles.virtualearth.net/comp/ch/132232211021301?mkt=enus&it=G,VE,BX,L,LA&shading=hill&og=37&n=z
Google Maps	√	https://www.google.com/maps/vt/pb=!1m4!1m3!1i16!2i51661!3i32532!2m3!1e0!2sm!3i253000000!2m24!1e2!2spsm!4m2!1sgid!2sWAxEEKreqXIZcPPYKArZA!4m2!1ssp!2s1!8m15!13m14!1b1!2sgp!3b1!8i1!9b0!12m3!2b0!3b1!4b0!13b1!15b1!16b0!18m1!1b0!3m9!2sen!3s!5e1105!12m1!1e47!12m1!1e1007!12m1!1e38!4e0!20m1!1b1
Yahoo Maps	√	https://2.base.maps.api.here.com/maptile/2.1/maptile/365f4c78eb/normal.day/20/826503/520390/256/png8?lg=ENG&token=TrLJuXVK62IQk0vuXFzaig%3D%3D&requestid=yahoo.prod&app_id=eAdkWGYRoc4RfxVo0Z4B
HERE	√	http://4.base.maps.api.here.com/maptile/2.1/maptile/365f4c78eb/normal.day/20/826729/520562/256/png8?lg=ENG&app_id=SqE1xcSngCd3m4a1zEGb&token=r0sR1DzqDkS6sDnh902FWQ&xnlp=CL_JSMv2.5.3.2
arcGIS	√	http://server.arcgisonline.com/ArcGIS/rest/services/World_Topo_Map/MapServer/tile/19/260259/413276
OpenStreetMap	√	http://a.tile.openstreetmap.org/19/413273/260260.png
ViaMichelin	√	http://md0.viamichelin.com/mapsgene/dm/mapdirect;YXNpX2RfMDAwNms=;MDAwMDAwMzE0MTAwMDAwMDYzOTE=?
MapQuest	√	http://mtile01.mqcdn.com/tiles/1.0.0/vy/map/18/206695/130117.png
WikiMapia	√	http://i5.wikimapia.org/?x=206640&y=130129&zoom=18&r=1214936&type=map&lng=0
Navionics	√	http://d2hcl9zx8watk4.cloudfront.net/tile/16/51661/32536?LAYERS=config_1_1_0&TRANSPARENT=FALSE
Mappy	√	http://slab3.axe.mappy.com/1v1/slab/get.aspx?viewmode=map& sx=419024&sy=206558&zoom=12&auth=aRjYy8tqadXy4N6l8MngEoBT/CrRrS48 cjm87Bf2Y+VjRPBN1P0OqFtCA9PEFo1mhCGkrCTQAzqjPM7jaFvkuw==

domains. Previous research[9] deployed a same-origin policy on the browser cache to prevent CORS from loading resources stored by the original sites. In principal, such a solution would defeat geo-inference attacks, but to our knowledge, this solution hasn't been adopted by modern Web browsers as a default setting. Performance could be one of the reasons why, so we implemented a prototype of the cache segregation policy in Chromium (version 34) and measured the estimated performance overhead incurred from it.

Current browsers can store all cacheable resources, but with a cache segregation policy, Chromium could only cache same-origin resources, not CORS. After all cacheable resources were stored in the vanilla Chromium and same-origin resources cached in Chromium with the cache segregation policy, we measured the page-load time for the Alexa Top 100 websites (except three unreachable sites: googleusercontent.com, akamaihd.net, and thepiratebay.sx). As Figure 3a shows, the same-origin caching policy triggered more than a 50 percent performance overhead, indicating that the same-origin caching policy can affect page-load time significantly.

Our Approach

Geo-inference attacks are ultimately caused by Web applications' increasing complexity and the amount of information in them. Accordingly, Web applications (or their developers) should pay attention to privacy in addition to functionality. Perhaps location-sensitive resources, such as country-specific logos, shouldn't be identified in Web sessions or cached in browsers. On the server side, Web developers can set the "Cache-Control: no-cache" directive in the HTTP response header for location-sensitive resources to instruct browsers not to cache those resources (although identifying them is a difficult task for today's complex Web applications).

To address this issue, we designed a prototype tool to help Web developers identify location-sensitive resources. Essentially, it collects the URLs of the cached resources that we found

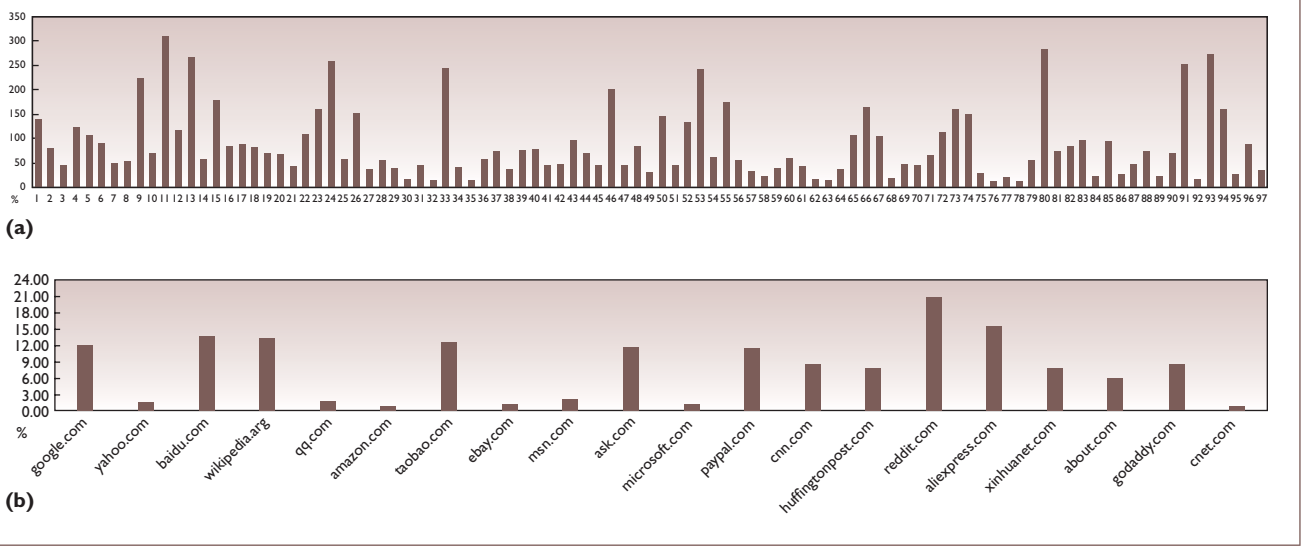


Figure 3. Our solution compared to segregating browser cache. (a) Enforcing a same-origin caching policy adds 20 to 300 percent performance overhead to load each site compared with the vanilla version without that policy. (b) Enforcing our solution of not caching location-sensitive logos only adds 0.7 to 20.7 percent performance overhead compared with the vanilla version.

during our manual testing of Web applications from different geolocations (behind VPN services) and then compares and automatically labels those URLs that are potentially location-sensitive. Web developers can verify these resources corresponding to this scaled-down set.

We evaluated our solution on 20 applications that contain country-specific logos (see Table 2). We first measured the page-load time for 20 applications in Chromium with all cacheable resources cached, and then measured it again without those logos cached. As Figure 3b shows, no caching for location-sensitive resources only adds a 0.7 to 20.7 percent performance overhead compared with the loading time of each site with cached logos, which is much better than simply segregating the browser cache as in Figure 3a. Compared to segregating the browser cache (which prevents sites from loading all cross-origin cached resources), our approach sets restriction only on location-sensitive resources, which are usually just a small portion of non-mapping sites. However, if a website contains numerous location-sensitive resources, such as Google Maps, this solution might trigger a huge performance overhead as the trade-off for privacy.

As we've demonstrated here, browser cache is a known and still open channel to leak privacy information to unintended parties. Geo-inference attacks are one example of leaking geolocations to malicious websites. Beyond browser cache, many more resources are shared between different Web origins in the browser, including the network stack, storage, and cookies. Any "leftover" state of one Web origin can leak information to another Web origin. Future research can look into how to exhaustively identify all potential leakage channels in different shared states in browsers and how to defend against these channels.

Acknowledgments

We thank the reviewers and editors for their insightful feedback and Chunwang Zhang, Hong Hu, and Shweta Shinde for their comments on an early presentation of this work. This work is supported by the Ministry of Education, Singapore, under grant no. R-252-000-495-133. Any opinions, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the Ministry of Education, Singapore.

References

1. D. Akhawe et al., "Towards a Formal Foundation of Web Security," *Proc. Computer Security Foundation Symp.*, 2010, pp. 290–304.
2. I. Poese et al., "IP Geolocation Databases: Unreliable?" *ACM SIGCOMM Computer Communication Rev.*, vol. 41, no. 2, 2011, pp. 53–56.

3. M. Balakrishnan, I. Mohomed, and V. Ramasubramanian, "Where's That Phone? Geolocating IP Addresses on 3G Networks," *Proc. 9th ACM SIGCOMM Conf. Internet Measurement*, 2009, pp. 294–300.
4. E.W. Felten and M.A. Schneider, "Timing Attacks on Web Privacy," *Proc. 7th ACM Conf. Computer and Communications Security*, 2000, pp. 25–32.
5. A. Bortz and D. Boneh, "Exposing Private Information by Timing Web Applications," *Proc. 16th Int'l Conf. World Wide Web*, 2007, pp. 621–628.
6. G. Wondracek et al., "A Practical Attack to De-Anonymize Social Network Users," *Proc. IEEE Symp. Security and Privacy*, 2010, pp. 223–238.
7. Z. Weinberg et al., "I Still Know What You Visited Last Summer: Leaking Browsing History via User Interaction and Side Channel Attacks," *Proc. Symp. Security and Privacy*, 2011, pp. 147–161.
8. R. Kotcher et al., "Cross-Origin Pixel Stealing: Timing Attacks Using CSS filters," *Proc. ACM SIGSAC Conf. Computer and Communications Security*, 2013, pp. 1055–1062.
9. C. Jackson et al., "Protecting Browser State from Web Privacy Attacks," *Proc. 15th Int'l Conf. World Wide Web*, 2006, pp. 737–744.
10. J. Ruderman, "Same Origin Policy for JavaScript," Mozilla, 2009; https://developer.mozilla.org/En/Same_origin_policy_for_JavaScript.

Yaoqi Jia is a PhD student in the Department of Computer Science at the National University of Singapore. He works on discovering new security vulnerabilities and attack vectors in Web and mobile applications, as well as on exploring new solutions to eliminate them from today's applications. Jia obtained his BS in computer science from the Huazhong University of Science and Technology. Contact him at jiayaoqi@comp.nus.edu.sg.

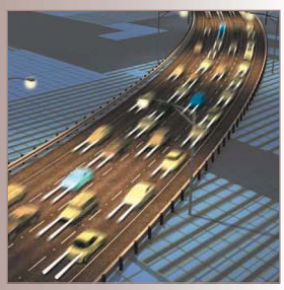
Xinshu Dong is a postdoctoral fellow at the Advanced Digital Sciences Center and a research affiliate in the University of Illinois's Coordinated Sciences Laboratory. His research interest spans the broad areas of security in software systems and cyber-physical systems. Dong obtained his PhD in computer science from the National University of Singapore. Contact him at xinshu.dong@adsc.com.sg.

Zhenkai Liang is an associate professor at the National University of Singapore. His research interests include system and software security, Web security, mobile security, and software testing. Liang obtained his PhD in computer science from Stony Brook University. Contact him at liangzk@comp.nus.edu.sg.

Prateek Saxena is an assistant professor at the National University of Singapore. His research interests include computer security and its intersection with formal methods and programming languages. Saxena obtained his PhD in computer science from the University of California, Berkeley. Contact him at prateeks@comp.nus.edu.sg.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.

Track: Best Conference Papers



The Effectiveness of Security Images in Internet Banking

Many Internet banking websites use security images as part of the login process, under the theory that they can help foil phishing attacks. Previous studies, however, have yielded inconsistent results about whether users notice that a security image is missing. This article describes an online study of 482 users that attempts to clarify the extent to which users notice and react to the absence of security images. The authors found that most participants entered their password even when the security image and caption weren't present.

Joel Lee and Lujo Bauer
Carnegie Mellon University

Michelle L. Mazurek
University of Maryland

As a security measure, many banking websites display a security image and caption each time a user logs into his or her account.[1] When users first register for an account, they're prompted to pick a security image from a list of available images, and to create a caption to accompany the image. The user is presented with the security image and caption on all subsequent logins, and instructed not to log in if the image or caption is missing or incorrect. This strategy is expected to help protect users from phishing attacks: If a phishing website mimics a real one in all ways except that it doesn't show the user's chosen security image, a vigilant user might notice the absence of the image and refuse to log in. Bank of America, for example, uses an image, an image title, and three challenge questions, together known as the SiteKey.[2] PNC Bank displays a user-selected personal security image and a caption created by the user[3]; and Santander Bank's approach is similar.[4]

Despite the almost ubiquitous use of security images on banking sites, their effectiveness at preventing phishing attacks is uncertain. Even setting aside strategies that a sophisticated attacker might use to show the correct security image on a phishing site, we don't have a good understanding of users' ability to notice that an expected image is missing and then refuse to log in.

Previous studies of the effectiveness of security images have reached divergent conclusions (see the sidebar). In one, 92 percent of participants proceeded to log into their real bank accounts even when the security image was absent.[5] In another, 60 percent of users of an online assignment-submission system noticed missing security images and refused to log in.[6] These studies used different methodologies, making it difficult to reconcile their results or isolate specific reasons for the divergence. Additionally, both studies occurred in settings sufficiently different from real-world online banking

Published by the IEEE Computer Society
1089-7801/15/$31.00 © 2015 IEEE

Related Work in Security Indicators

Several studies have shown that visual security indicators, including special toolbars and SSL warnings, are often ineffective.[1,2] Min Wu and his colleagues additionally found that many users either don't know about phishing attacks (which security images are intended to prevent), or don't realize how sophisticated such attacks can be.[1]

To our knowledge, only two prior studies, with divergent results, specifically examine the effectiveness of security images. Stuart Schechter and his colleagues evaluated several security measures commonly used in online banking, including security images.[3] They divided participants into three conditions: role-playing, security-primed, and using their own real bank accounts. Only two of 60 total participants (both in the real account condition) refused to enter their passwords when the security image was removed.

In this study, participants were primarily university students, and many of those recruited opted out, possibly biasing the sample toward less-security-conscious users. In addition, Schechter and his colleagues structured the study as a series of tasks, with participants unable to proceed until they'd completed the current task. Finally, the experimental lab setting might have influenced users to feel safer, obey authority figures, and complete the tasks despite any misgivings.

Amir Herzberg and Ronen Margulies examined phishing detection using an assignment-submission system in a university computer science department.[4] Over three semesters, they simulated several phishing attacks, offering students incentives for correctly detecting attacks. When they used interactive security images — that is, participants were required to click on the image during login — almost 60 percent of participants successfully detected phishing when the image was absent. In this study, all participants were computer science students (with presumably high levels of technological literacy), were primed to anticipate attacks, and were given incentives to detect phishing. Moreover, the authors specifically removed participants who didn't attempt to detect attacks. These methodological differences could account for the divergent results between the two studies.

Our study attempts to address some shortcomings of the two prior studies, while examining not just whether security images are effective but also which factors (including size, interactivity, and habituation) impact their effectiveness. We also varied participants' security priming and their compensation to test how these changes affect the results. We use the Amazon Mechanical Turk crowdsourcing service (MTurk) to recruit participants who are significantly more diverse than typical student samples.[5]

References

1. M. Wu, R.C. Miller, and S.L. Garfinkel, "Do Security Toolbars Actually Prevent Phishing Attacks?" *Proc. SIGCHI Conf. Human Factors in Computing Systems*, 2006, pp. 601–610.
2. J. Sunshine et al., "Crying Wolf: An Empirical Study of SSL Warning Effectiveness," *Proc. 18th Usenix Security Symp.*, 2009, pp. 399–432.
3. S. Schechter et al., "The Emperor's New Security Indicators: An Evaluation of Website Authentication and the Effect of Role Playing on Usability Studies," *Proc. 28th IEEE Symp. Security and Privacy*, 2007, pp. 51–65.
4. A. Herzberg and R. Margulies, "Forcing Johnny to Login Safely," *Proc. 16th European Symp. Research in Computer Security*, 2011, pp. 452–471.
5. M. Buhrmester, T. Kwang, and S.D. Gosling, "Amazon's Mechanical Turk — A New Source of Inexpensive, Yet High-Quality, Data?" *Perspective on Psychological Science*, vol. 6, no. 1, 2011, pp. 3–5.

scenarios that it's difficult to generalize from their results.

We conducted an online study of security images' effectiveness that addresses some limitations of previous studies by accounting for habituation, addressing participant motivations, and varying security images' visual characteristics. We found that security images are only marginally effective – overall, 73 percent of our participants logged in even when the security image was absent. Moreover, our results suggest that varying the visual characteristics of security images has little to no impact on their effectiveness. The effectiveness of security images is similarly unchanged with varying levels of user habituation and motivation.

Study Methodology

We built a banking website with a similar look and feel to an actual banking website, shown in Figure 1. We recruited participants using the Amazon Mechanical Turk (MTurk) crowdsourcing service. To avoid priming, we didn't inform them about the study's true purpose; instead, we told participants they were to test the website's direct deposit functionality. To complete the study, participants were instructed to "report deposits" by logging in to their accounts, clicking a button, and entering the amount that was deposited into their bank account. During login, participants entered their user ID, then entered their password on the screen with their security image and caption. After logging in, participants were shown the amount deposited into their account. We required most participants to report five deposits over at least five days; in one condition, we shortened the study to two deposits over at least two days. Participants could report each deposit no sooner than

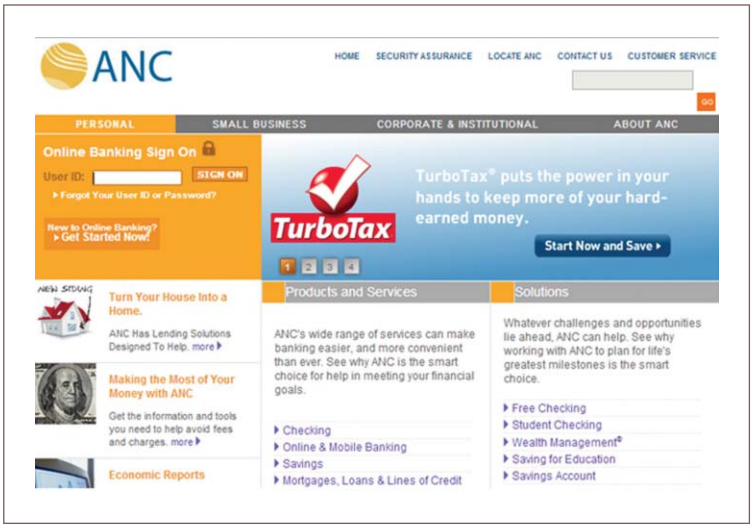


Figure 1. Fictitious bank website used in the study, designed to closely mimic a real banking website. After entering their user ID, users were taken to a second page where their chosen security image and caption were displayed.

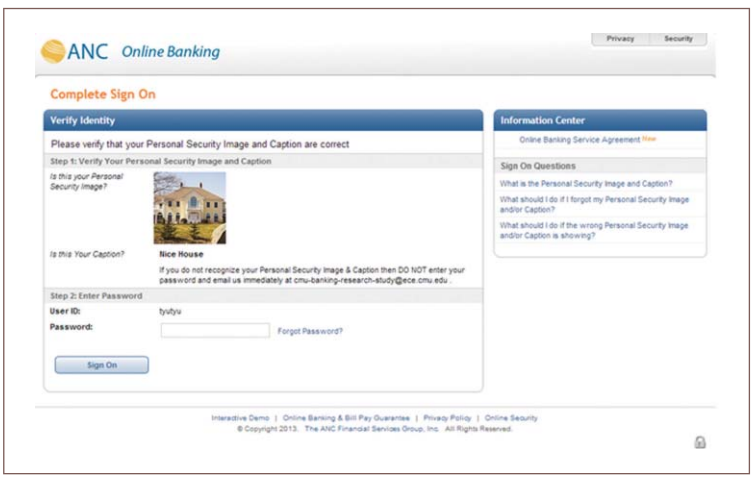


Figure 2. Login screen with security image and caption as shown in the control condition. We instructed users not to enter their password unless their chosen security image and caption were present.

24 hours after the previous deposit (or after the start of the study). To remind participants to report the next deposit, we sent a reminder email with a link to the website to each participant 24 hours after each deposit report (or after starting the study).

We required participants to report five deposits to simulate habituation to security images and captions. To promote realism, participants who completed the study received as compensation the total amount that was "deposited" into their accounts. Each time they logged in, participants were shown the following message: "If you do not recognize your Personal Security Image & Caption then DO NOT enter your password and email us immediately at [email address]." This message was similar to that displayed at an actual banking website. Figure 2 shows the login screen with the security image, caption, and the message.

When the participant accessed the site to report the final deposit, the security image and caption were replaced with an "under maintenance" image. We restored the security image and caption for any login attempt 5 or more minutes after that time. This simulates a real-life scenario when the user doesn't see the security image upon accessing a phished website.

We recorded whether participants entered their passwords in the 5-minute period during which the security image and caption weren't displayed. After participants reported the final deposit, the site directed them to an exit survey, which asked several questions about security images and participant demographics.

Conditions

We defined 12 conditions, which fall into seven categories. The first category consists of our control condition, whereas the others explore specific factors that could influence security images' effectiveness: appearance, interactivity, the ability to customize images, the lack of a caption, and methodological variations. In addition, one category explores the effects of varying multiple factors at the same time. The study had a between-subjects design, meaning we assigned each participant to exactly one condition. We assigned participants randomly to one of the first nine conditions. Conditions 10–12 tested variations in study methodology; we solicited participants for these separately (in parallel with soliciting participants for the other conditions) because the methodological variations being tested required small changes to the study advertisement.

Condition 1, our control condition, closely mimics PNC Bank's security image implementation. The study site displays the security image chosen by the participant at 100 × 100 pixels.

In conditions 2 and 3, the image differs in appearance. Using these conditions, we explored whether security images with different visual

features make it more likely that a participant will notice that a security image is missing:

- In condition 2, *large image*, the chosen security image is shown at 300 × 300 pixels, or 9 times larger than in the control.
- In condition 3, *blinking image*, the security image is programmed using JavaScript to blink repeatedly to draw the user's attention.

Conditions 4–6 require different interaction from the user. These conditions test whether requiring participants to interact with the security image makes it more likely that they will refuse to log in when the security image is missing:

- In condition 4, *interactive image*, participants must click on the security image before they can enter their passwords.
- In condition 5, *copy random word*, participants must copy a random word placed within the security image before they can enter their passwords.
- In condition 6, *copy caption*, participants must copy the caption displayed with the security image before they can enter their passwords.

Condition 7, *custom image*, reflects differences in customization. Rather than choosing from a list of available images, participants upload an image of their choice. This condition tests whether letting users customize their security image increases its effectiveness.

Condition 8, the *multifeature* condition, reflects differences in customization, appearance, and interactivity. Participants upload an image of their choice. The image blinks continuously (using JavaScript), and participants must click on it before they can enter their passwords. This condition tests whether the simultaneous presence of features present individually in other conditions improves security images' effectiveness.

In condition 9, *no caption*, we didn't ask participants to create a caption during account registration, and no caption is shown during login. Security images are commonly accompanied by a caption. However, we wanted to decouple the security image's effect from the caption's effect.

Conditions 10–12 follow different study methodologies. We designed these conditions to test the effects of study duration (and habituation) and monetary and other incentives:

- In condition 10, *two logins*, we asked participants to log in to the account twice, instead of five times as in other conditions. The second time they logged in, we removed the security image.
- In condition 11, *more pay*, we paid participants twice the amount of money as in the base condition.
- For condition 12, *more security conscious*, we put in the following message in the consent form and instructions page, in an attempt to make participants more security conscious: "Recently, internet banking websites have been under attack. If your account is compromised, you will not receive payment for the study. It is important for you to take the necessary security measures, such as to choose a hard-to-guess password." During a debriefing at the end of the study, we explained to participants that the message was fictitious and its purpose.

We conducted the study in April and May 2013. Of the 569 individuals who signed up for an account on our website, 482 (85 percent) completed the entire study by reporting five deposit amounts over five days (or two deposit amounts over two days for users in the *two logins* condition). Participants who reported at least one deposit amount tended to complete the entire study — only 15 didn't finish the study after reporting one or more deposits. The remainder of this article focuses on the 482 participants who completed the entire study.

Study Results

We received numerous emails from participants about the "under maintenance" security image, asking us whether to log in given the absence of their security image and caption. We replied by telling participants not to log in, to try again after a few minutes, and to only log in when the correct security image and caption appears. We discuss participant email more later.

Security Image Effectiveness

Across all conditions, 352 of 482 participants (73 percent) entered their passwords when their security image and caption weren't displayed. The remaining 130 participants (27 percent) didn't. Table 1 shows results by condition.

Table 1. Percentage and number of participants who entered their passwords without the security image.

Condition number	Condition name	Entered password (percent)	No. of participants who entered password/total participants	*p*-value
1	Control	75.00	30/40	
2	Large image	86.84	33/38	0.185
3	Blinking image	57.14	24/42	0.088
4	Interactive image	74.36	29/39	0.948
5	Copy random word	63.64	21/33	0.292
6	Copy caption	69.77	30/43	0.595
7	Custom image	82.50	33/40	0.412
8	Multifeature	74.36	29/39	0.948
9	No caption	78.05	32/41	0.746
10	Two logins	68.42	26/38	0.519
11	More pay	77.78	35/45	0.763
12	More security conscious	68.18	30/44	0.490
Total		73.03	352/482	

Notes: Shading groups conditions that fall under the same category: conditions 2 and 3 (differing in appearance), 4–6 (differing in interaction), and 10–12 (differing in study methodology). The *p*-value reflects a X^2 comparison between each condition and the control.

We used an X^2 test ($\alpha = 0.05$) to compare the 11 experimental conditions to the control condition. No condition showed a significant difference from the control, but the blinking image condition showed the most improvement in participants declining to log in when the security image wasn't shown.

We also found no statistically significant difference in security images' effectiveness based on participants' gender, country, major/degree/job, level of education, or security experience.

Sentiment toward Security Images

At the end of the study, we asked participants to rate (on a 5-point Likert scale) their agreement or disagreement with five statements about their security image. We compared the results between conditions (X^2, $\alpha = 0.05$), binning "strongly disagree," "disagree," and "neutral" responses as one response group and "agree" and "strongly agree" as another.

The following are some of the statements we asked participants to respond to.

Of the participants in the control condition, 5 percent agreed or strongly agreed with the statement, "Using a security image as part of the login process was annoying." Across the experimental conditions, agreement varied from 2.2 to 18.2 percent of participants. Figure 3 shows participants' responses. Although no condition showed a statistically significant difference from the control, the *copy random word* and *copy caption* conditions (18.2 and 16.3 percent, respectively) appear to be most annoying, and could perhaps be significant at a larger sample size. Condition *copy random word* required users to type in a random word placed in the image, whereas the *copy caption* condition required users to type in the security caption shown beneath the image each time they logged in to the account. This requires additional effort and slows the login process, so it was consistent with our expectations that sentiment could be negatively affected.

Participants were relatively evenly split on the statement, "I wish that my bank's website used a similar security image": 42.5 percent of participants in the control condition agreed or strongly agreed, and agreement or strong agreement in experimental conditions ranged from 42.4 percent (*copy random word*) to 71.8 percent (*multifeature*). The only statistically significant result was in the *multifeature* condition ($p = 0.009$), in which 71.8 percent of participants agreed or strongly agreed. This might have been due to a combination of noticeability and convenience: the image was supplied by the participant, blinked, and had to be clicked on before a participant could log in, but the amount of effort required during the login process was minimal (an extra click).

The vast majority of participants reported paying attention to the security image: 92.5 percent

in the control condition, and between 84.2 (*large image* condition) and 100 percent (*multifeature* condition) in experimental conditions did not agree with the statement, "I did not look at the security image before I entered my password." Notably, this is a much higher fraction of participants than the fraction that declined to log in when the security image was absent. As with the previous statement, the condition that varied most from the control was the *multifeature* condition. Although this difference wasn't statistically significant at this sample size, it does suggest that the multiple additional features might have made more participants look at the security image before they entered their password.

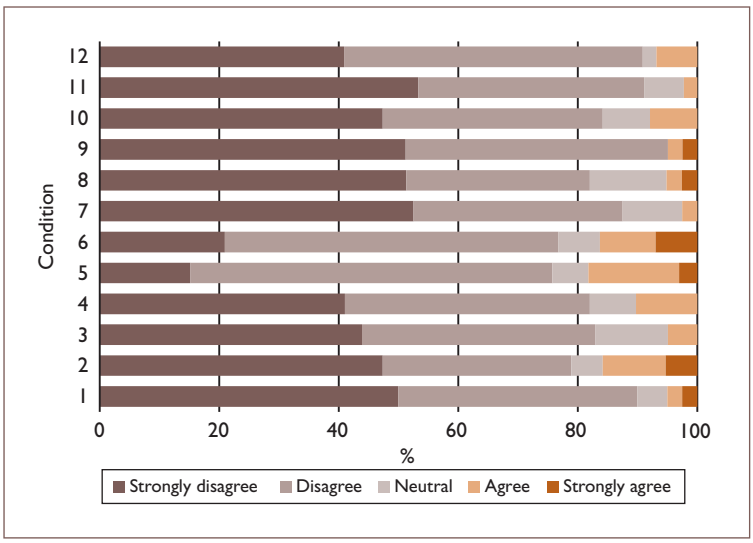


Figure 3. Participants' responses to the statement that "using a security image as part of the login process was annoying."

Qualitative Feedback

We received nearly 300 emails from participants on a range of topics. Most were procedural or administrative — users requiring a password reset, asking for clarifications on how the amount would be deposited into the account, asking when the email with information on the next deposit would be sent, or simply confirming that they reported the deposit amount. In addition, we received numerous emails from participants who informed us when their security image was replaced with an "under maintenance" image. Lastly, several users provided feedback on the study.

We received email from 32 participants informing us that their security image and caption wasn't shown when it was changed to an "under maintenance" image. Of these participants, a minority reported logging in despite noticing that the security image was absent. For example, a participant commented that "when I realized that my security image was missing, I came very close to sending you an email . . . but then I thought that maybe the study wanted to see if I would proceed with login." Another participant commented that "I did notice the under construction and wasn't sure what to do. Had this been my actual bank account, I would have exited and tried again later. But since I knew I was supposed to check in every day I didn't want to chance it so I continued through, clicked on it, and entered my password." Yet another participant wrote, "The wrong security image came up today. I got an under maintenance sign. But I signed in with my password anyway because I was afraid I would not receive my $3.50 if I did not sign in." This feedback suggests that although we tried to make the study as realistic as possible, participants' awareness that they were taking part in a study and their desire to complete the study might have led them to log in despite the "under maintenance" image at a greater rate than they would have in practice, based on these email reports. On the other hand, users in real-world scenarios might have their own compelling reasons for logging in even when they notice something amiss.

Evaluating Password Strength

We examined the strength of the passwords created by the 482 participants who completed the entire study by subjecting the passwords to a password-cracking algorithm.

Our method for cracking passwords simulated an attacker with moderate knowledge of current password-cracking tools. The specific tool we used was oclHashcat-1.20, a popular tool known for its support of many hashing functions and its exploitation of parallelization through GPUs. We used oclHashcat's most popular attack mode, a rule-based attack, in which an attacker uses mangling rules to transform dictionary entries into additional guesses. The dictionary we chose consisted of three public leaks (RockYou, Yahoo, and MySpace), the commercial Openwall dictionary, an inflection dictionary (different tenses of natural-language words), and a Google Web corpus. We used roughly 300,000 randomly generated mangling rules supplied with oclHashcat to define the

transformations. These rules are a subset of a large number of mangling rules generated by oclHashcat developers, selected and ordered based on their effectiveness when used against the RockYou dataset. It's reasonable to assume that an attacker with moderate familiarity with oclHashcat could perform a similar, or similarly effective, attack, though a more experienced attacker could use a more sophisticated and effective approach. We ran the cracking algorithm to 5.6×10^{12} guesses, which corresponds to anywhere from an hour to days of work for an attacker, depending on available hardware and hash functions used to protect the passwords.

Overall, we cracked 54.1 percent (261) of participants' passwords. Of these participants, 80.1 percent (209) entered their passwords when the security image was missing, compared to 64.7 percent (143) of participants whose passwords weren't cracked. This difference is statistically significant (X^2, $p = 0.0002$). This shows that participants who created stronger passwords (not cracked within 5.6×10^{12} guesses) paid more attention to security images. One possible explanation for this is that participants who create strong passwords are likely to be more security conscious or knowledgeable than those who create weaker passwords, so it would be reasonable for those same participants to also pay more attention to security indicators such as the security image and caption.

Implications

As with other studies, the one described in this article has several limitations. For example, our study used fake Internet banking accounts, rather than real ones. Although we tried to make the experience as realistic as possible in terms of look and feel, participants' motivation to log in within the study would have differed from their motivation in a real setting. As with other biases potentially caused by the study framing, this one would likely affect all conditions equally, minimizing its impact on comparisons between conditions; however, this bias might have influenced the overall finding that security images are ineffective.

Our methodology varied both the level of habituation and the strength of motivation. In practice, however, both effects would likely be stronger, because users interact with their online banking accounts over a period of years rather than days, and their real accounts are of significantly higher value than we could simulate in the study.

In practice, phishing attacks come in many forms. Our methodology involved mimicking one such attack; others could be more (or less) successful, potentially leading to different login rates in the presence of an attack than we observed.

In our control condition, 75 percent of participants entered their passwords even when their security image and caption weren't shown. This result differs from the study by Stuart Schechter and his colleagues, which found (in a lab setting) that 92 percent of participants using their own online banking accounts did so.[5]

Our *interactive image* condition, in which participants were required to click on their security image before logging in, is comparable to the method Amir Herzberg and Ronen Margulies used in their study.[6] We found that 74 percent of participants in this condition entered their passwords despite the missing security image; in contrast, only 40 percent of participants in the Herzberg study did so.

In each case, we attribute the difference in observed user behavior to differences in methodology. The Schechter study took place in a setting in which participants might have been particularly likely to ignore security concerns, whereas in the Herzberg study, participants were incentivized to detect phishing attacks. We believe our methodology strikes a balance that is more consistent with real-world use of security images. Consistent with this, the fraction of our participants that detected phishing attacks falls between the ranges reported in previous work.

Overall, our results suggest that security images are generally not very effective, especially when compared to other more secure, albeit expensive, methods, such as using a security token for two-factor authentication.

In general, and perhaps surprisingly, our results suggest that requiring additional login tasks doesn't lead to significantly greater effectiveness, but can create greater annoyance. For example, participants who had to type in a word that appeared in the image or type in the security caption before they could enter their password weren't more successful at evading simulated phishing attacks. However, these participants experienced greater levels of annoyance with

the login process, suggesting that adding nontrivial complications or tasks to the login process doesn't improve the effectiveness of security images and similar security measures.

Interestingly, the *multifeature* condition, in which participants defined the security image, the image blinked, and participants had to click on the image, proved no more effective than conditions with much subtler image effects and fewer attention-grabbing features. This is despite the fact that more participants in the *multifeature* condition stated that they looked at the security images before entering the password than in the control condition. This highlights a gap between reported noticeability and the likelihood of logging in when the security image was absent, which we also observed across conditions.

None of the interactive conditions, in which users had to click or type something related to the security image before they could enter a password, significantly affected the security image's effectiveness. This result diverges from that of Herzberg and Margulies.[6]

In the *interactive image* condition, participants might have clicked on the image without noticing whether it was correct. In the two copy conditions, users reported more annoyance than in the control condition, suggesting an increased awareness of the security image. However, the added inconvenience might have made participants glad to find the site "under maintenance" because it let them proceed more quickly.

Also surprisingly, participants who uploaded their own images to use as their security image — instead of choosing from a list of website-provided images — weren't significantly more effective at noticing the image's absence.

Finally, to the extent that our study exposed these factors, habituation, the financial compensation to participants, and the amount of security priming participants received didn't significantly affect participants' ability to notice and effectively react to missing security images.

Given our findings that security images aren't especially effective, banks and other companies that currently use them should consider alternative approaches, such as two-factor authentication (using a one-time password sent via text message, generated through software installed by the customer, or provided by a hardware token).

We also found that users who created weaker passwords paid significantly less attention to security images. Although we can't definitively conclude that this group is less security conscious in general, anecdotal evidence suggests they might be. This suggests that a specific subset of users who generally adopt poor security practices constitutes a weak link in securing an organization or company. Perhaps these users can be identified and encouraged to improve their security behavior.

Acknowledgments

This work was supported in part by US National Science Foundation award CNS-1018211. We thank Cristian Bravo-Lillo, Limin Jia, Sean Segreti, and Blase Ur for their input and help in various phases of the research.

References

1. J. Kirk, "Study: Users Ignore Bank Security Features," *Computerworld*, Feb. 2007; www.computerworld.com/s/article/9010283.
2. "SiteKey FAQs," Bank of America, 2013; www.bankofamerica.com/privacy/faq/sitekey-faq.go.
3. "An Added Layer of Security," PNC Bank, 2014; www.pnc.com/en/security-assurance/preventing-fraud/added-layer-security.html.
4. "SSA Makes Online Banking Even More Secure," Santander Bank, 2014; www.santanderbank.com/us/personal/banking/online-and-mobile-banking/security-center/ssa-learn-more.
5. S. Schechter et al., "The Emperor's New Security Indicators: An Evaluation of Website Authentication and the Effect of Role Playing on Usability Studies," *Proc. 28th IEEE Symp. Security and Privacy*, 2007, pp. 51–65.
6. A. Herzberg and R. Margulies, "Forcing Johnny to Login Safely," *Proc. 16th European Symp. Research in Computer Security*, 2011, pp. 452–471.

Joel Lee is an assistant manager in the Singapore Public Service. His research interests focus on usable security, examining how users react to security features, and how these security features can be improved. Lee has an MS in information security policy and management from Carnegie Mellon University. He participated in this research as part of his MS degree. Contact him at joellee@gmail.com.

Lujo Bauer is an associate research professor at Carnegie Mellon University. His research interests include

building usable access-control systems with sound theoretical underpinnings, developing languages and systems for runtime enforcement of security policies on programs, and generally narrowing the gap between a formal model and a practical, usable system. Bauer received a PhD in computer science from Princeton University. Contact him at lbauer@cmu.edu.

Michelle L. Mazurek is an assistant professor in the Computer Science Department at the University of Maryland. Her research interests include human factors in computer security, with an emphasis on designing systems from the ground up to support usable security. Mazurek has a PhD in electrical and computer engineering from Carnegie Mellon University. Contact her at mmazurek@cs.umd.edu.

Focus on Your Job Search

IEEE Computer Society Jobs helps you easily find a new job in IT, software development, computer engineering, research, programming, architecture, cloud computing, consulting, databases, and many other computer-related areas.

New feature: Find jobs recommending or requiring the IEEE CS CSDA or CSDP certifications!

Visit www.computer.org/jobs to search technical job openings, plus internships, from employers worldwide.

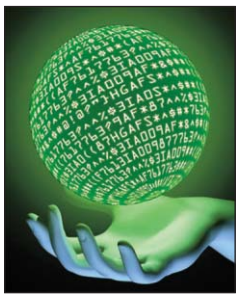
http://www.computer.org/jobs

IEEE computer society | JOBS

The IEEE Computer Society is a partner in the AIP Career Network, a collection of online job sites for scientists, engineers, and computing professionals. Other partners include Physics Today, the American Association of Physicists in Medicine (AAPM), American Association of Physics Teachers (AAPT), American Physical Society (APS), AVS Science and Technology, and the Society of Physics Students (SPS) and Sigma Pi Sigma.

Spotlight

Editor: Gustavo Rossi • gustavo@lifia.info.unlp.edu.ar



An Architecture and Guiding Framework for the Social Enterprise

Vanilson Burégio • *Federal University of Pernambuco, Brazil*

Zakaria Maamar • *Zayed University, United Arab Emirates*

Silvio Meira • *Federal University of Pernambuco, Brazil*

Interest is growing in how enterprises should capitalize on social technologies such as social networks, blogs, and wikis. A social enterprise is one that strives to open up new communication channels with stakeholders using such technologies. The authors suggest an architecture and a guiding framework to help integrate social technologies into enterprise operations.

According to Gartner, "many large companies are embracing internal social networks, but for the most part, they're not getting much from them."[1] Inadequate sponsorship and overemphasis on technology are the most cited reasons for this limited return on investment (ROI). However, we identify another reason undermining social network adoption: the lack of tangible benefits that demonstrate how these networks add value to enterprise operations. Enterprises still aren't sure about the ROI of Web 2.0 technologies,[2] although expenditures on these technologies are skyrocketing, and were expected to reach US$4.6 billion globally by 2013 (http://readwrite.com/2008/04/20/enterprise_20_to_become_a_46_billion_industry/). With such substantial investment, enterprises need coaching on how their business applications can capitalize on Web 2.0 technologies to draw from relevant information such as markets trends, consumer comments, and supplier strategies. Usually, getting this information requires processing large volumes of structured and unstructured online data (big data).

A *social enterprise* is one that, in addition to having an online presence, strives to open up new communication channels with various stakeholders using Web 2.0 technologies such as social networks, wikis, and blogs. Andrew McAfee was the first to introduce "Enterprise 2.0" to describe the use of emergent social software platforms within or between companies and their partners or customers.[3] According to Gartner's Forecast Analysis, enterprise social software has seen 24.4 percent growth since 2009.[4] This makes it the fastest-growing software market, but it must work hand in hand with regular business processes to ensure Enterprise 2.0's success (see www.bluekiwi-software.com/en/bk-blog/report-the-state-of-social-business-2013). As Rob Cross and his colleagues state, "Enterprise 2.0 only works if it is part of a business process. It's great to work in new ways, but it's not enough. To make it real, it has to be very practical."[5] Today's enterprises shouldn't miss out on riding the Web 2.0 wave — many opportunities are still untapped. However, they should ride this wave in a gradual and controlled way. According to one study, "A recent survey of 1,160 business and IT professionals shows that while 46 percent of the organizations increased their investments in social technologies in 2012, only 22 percent believed that managers are prepared to incorporate social tools and approaches into their processes."[6]

Our proposed architecture lays down the social enterprise's foundations, while our framework

offers guidelines on how to identify, evaluate, select, and roll out Web 2.0 initiatives that work to transform the enterprise into a social one.

The Social Enterprise

A social enterprise shouldn't restrict itself to just social relations (such as friendship and collegiality) between people only. Using appropriate social relations, an enterprise could connect other components, such as tasks,[7] machines,[7,8] and databases.[9] With support from peers, any enterprise component should be able to form or join the networks upon which end users can perform social queries.[10] Supporting our recommendations that today's objects have the capacity to socialize, Wei Tan and his colleagues state that

> Most social networks connect people or groups who have similar interests or features. In the near future, we expect that such networks will connect other entities, such as software components, Web-based services, data resources, and workflows. More importantly, the interactions among people and nonhuman artifacts have significantly enhanced data scientists' productivity."[11]

Given this, let's look at our architecture for the social enterprise.

Enterprise Architecture

An enterprise architecture describes the necessary components on which an enterprise operates to offer services to or make products for customers. The selected architecture type affects the enterprise's business model. Indeed, an enterprise with an online presence through a website expects that customers will submit orders online. In contrast, another enterprise might require that customers come in person to place orders. Different variations of online and offline business models are possible depending on the enterprise itself, the nature of services and products, what

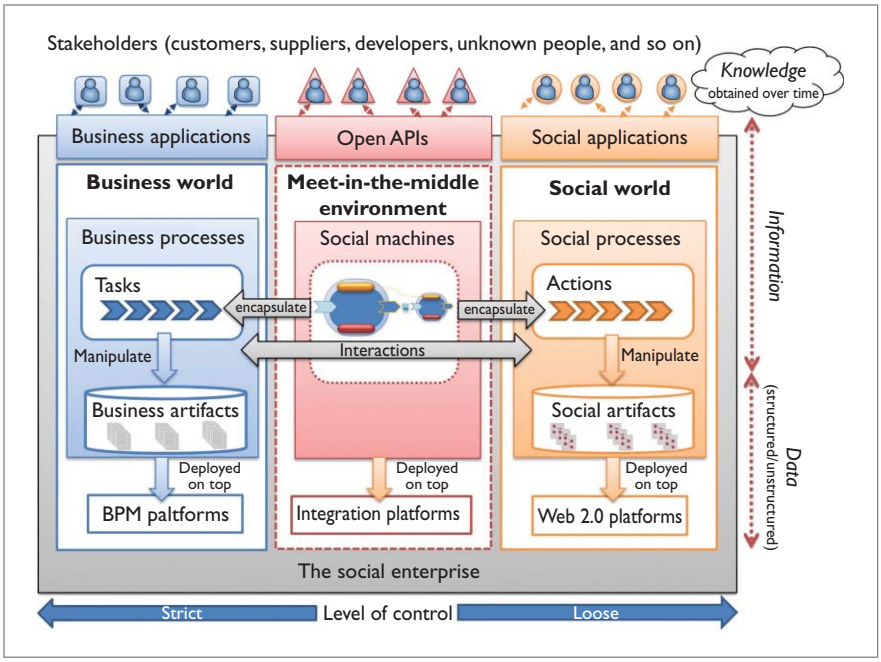


Figure 1. Architecture for the social enterprise. Structured and unstructured data are processed into information that becomes knowledge over time.

legislation is in place, IT infrastructure status, and so on.

A social enterprise architecture must be strongly coupled with Web 2.0 technologies to support the enterprise in reaching out to stakeholders, including customers, suppliers, competitors, and regulatory authorities. Unknown people (such as discussion group members) can also be stakeholders and can thus interact with the social enterprise. This isn't the case with traditional (that is, nonsocial) enterprises, in which stakeholders must be known in advance. The *business world* refers to structured business processes that the enterprise rolls out in response to internal and external events, such as receiving customers' orders. The *social world* backs the business world by offering new forms of communication with the enterprise's stakeholders. We can connect these worlds through a meet-in-the-middle environment.

Figure 1 illustrates our social enterprise architecture. The level of control swings from strict in the business world to loose in the social world, showing how much control an enterprise can exercise over the operations it initiates. In the business world, processes consist of tasks that manage business artifacts such as bill, customer, and order. A business artifact is "a concrete, identifiable, self-describing chunk of information that can be used by a business person to actually run a business."[12] Dedicated business process management platforms assist engineers to design, develop, deploy, and track processes. In the social world, these processes are built on the fly (that is, unstructured) in response to actions (tasks) that Web 2.0 applications allow users to execute (posting comments, inviting friends, chatting online, and so on) in order to manage social artifacts. These social artifacts abstract specific objects or events associated with Web 2.0 applications such as a Facebook invite, a Twitter updating account, or photo tagging in Instagram. To bridge the gap between the business and social worlds, the meet-in-the-middle environment acts as an integrator through connectable building

Table 1. Terminology for a social enterprise architecture.

Business world		Social world	
Term	**Definition**	**Term**	**Definition**
Process	Constitutes the enterprise's know how	Process	Is built on the fly in response to actions that Web 2.0 applications allow users to execute
Artifact	Consists of a chunk of information that business people use	Artifact	Abstracts objects/events associated with Web 2.0 applications
Knowledge	Establishes what the enterprise knows about the environment through official channels (legislation and sales across the country); knowledge helps define the enterprise's goals (to increase sales volume by 20%)	Knowledge	Establishes what the enterprise gets to know about the environment through Web 2.0 applications (such as tweets on new products)
Task	Defines what the enterprise executes to reach its goals; tasks are put together to form business processes	Action	Defines what the enterprise can execute over Web 2.0 applications: creating an account, inviting people, and so on
Stakeholder	Defines those who interact with the enterprise through official channels and affect its business goals; business stakeholders are known by default	Stakeholder	Defines those who use Web 2.0 applications to interact with the enterprise; anonymous stakeholders are accepted

blocks called *social machines* (SMs).[8] These SMs provide specialized APIs that encapsulate both the business world's tasks and the social world's actions; the business world can now act over the social artifacts (as when a marketing business process launches a new campaign on Facebook) and vice versa (online comments on the campaign are used to adjust the marketing business process). We can see from Figure 1 how structured and unstructured data are processed into information that becomes knowledge over time.

Table 1 shows a taxonomy of terms associated with our social enterprise architecture that helps further bring the business and social worlds together.

Guiding Framework

Our framework is a set of guidelines that help the future social enterprise to embrace Web 2.0 technologies. These guidelines frame this journey from three perspectives. The *technology* perspective identifies the future Web 2.0 applications (built on Web 2.0 technologies) that will sustain an enterprise's growth and support the enterprise in achieving its goals. The *organization* perspective puts in place the necessary procedures to ensure an efficient use of future Web 2.0 applications (to avoid misuses). Finally, the *management* perspective identifies the relevant metrics (or key performance indicators) that will help an enterprise evaluate this efficient use to determine tangible benefits. The organization and management perspectives are our response to the overemphasis of technology we discussed in the introduction.

Technology

Most available Web 2.0 technologies were launched first in the open Web and have since been customized in response to enterprises' needs and requirements. However, this customization occurs without a careful analysis of these technologies' benefits and limitations. Blindly copying the competition is the main reason for enterprises to adopt Web 2.0 technologies. From a technology perspective, an enterprise that wishes to embrace such technologies should thus analyze what's available in the market that could be in line with its mission (see Table 2). Among other things, this analysis requires

- understanding the different types of Web 2.0 technologies in terms of pros and cons;
- setting the necessary functional and nonfunctional criteria that will assist the enterprise in selecting the best technologies depending on its goals and the applications it wants to develop;
- defining the technical specifications of the computing resources on which the applications will operate;
- defining the applications' operation mechanisms, such as maintenance and upgrade frequency; and
- developing a risk analysis ("what if") of the applications' impact on the enterprise operations.

Table 2 lists some social technologies and their corresponding potential benefits and possible risks, based on the extensive practical experience one of us (Vanilson Burégio) has gained when deploying Web 2.0 technologies in different Brazilian companies, including the Recife Center for Advanced Studies

Table 2. Web 2.0 technologies: benefits and risks.

Types	Potential benefits	Possible risks
Wikis	Intensify information sharing and foster collaboration among employees to co-create knowledge through shared content.	Require strong commitment to keep content updated. Most people will not take the time to do this.
Online social networks (OSNs)	Enable quicker access to expertise and resources once users have personal profiles that expose their interests (experience, education, activities, and so on). OSNs can also support the process of identifying promising leaders in the enterprise.	Become helpful only once a "good" number of people are connected. Enterprises might face difficulties during first-time adoption (cold start). Having online presence on external OSNs exposes enterprises to a high level of open criticism on their products and services.
Microblogging	Encourages interactive discussion by creating a quick problem-solving space. Allows employees to stay on top of their day-to-day information and communication needs in an open, yet informal way.	Given the huge variety of topics possible, risks creating a space with too much unstructured content, causing information overload.
Social bookmarking	Promotes explicit assessment of the value or usefulness of several external and internal information resources.	Open access to everyone's bookmarks raises confidentiality concerns over specific projects. Within the enterprise, teams might want to create shared bookmarks that are visible only to the group.
Social customer relationship management (CRM)	Allows brand monitoring and derives meaning from social data through analytics. Uses the power of social interactions to get closer to customers to improve revenue and efficiency.	Risks consumers' limited-engagement in social media if no tangible value is added to their experience.

and Systems (www.cesar.org.br), the Brazilian Federal Data Processing Service (www.serpro.gov.br), and the Brazilian State Agency for Information Technology (www.ati.pe.gov.br).

Management

Without tangible benefits, accurate performance indicators, and proper unstructured content, any enterprise will have difficulty backing its Web 2.0 investment. Hence, the management perspective should establish the value Web 2.0 applications add to the enterprise. This requires

- evaluating how the social enterprise can leverage Web 2.0 applications through tangible benefits (or key performance indicators);
- monitoring the application's activity level over time (for example, the number of active members and posted messages);
- assessing applications' ROI (such as the number of new customers and increases in sales volume);
- harnessing the available content on Web 2.0 applications into content suitable for decision making; and
- adopting a proper online reputation-management plan to monitor and promptly react to what people say in social media about the enterprise and its brand, products, and services.

These items could be converted into policies defining why an enterprise should go social.

Organization

The practical experiences we've had in adopting Web 2.0 technologies in enterprises show that becoming a real social enterprise is more a cultural than a technological issue. Often, using Web 2.0 technologies and applications efficiently requires significant changes in how people work, communicate, and collaborate. The organization perspective should thus establish the necessary procedures that frame the use of Web 2.0 applications in accordance with the enterprise's policies and regulations. This requires

- indicating how, when, and where employees can engage in Web 2.0 operations;
- defining the nature of content that can be discussed via applications;
- setting policies for reaching out to application respondents; and
- educating employees through training and guidance to support cultural change.

As with the management perspective, these items could be converted into policies defining what

an enterprise's employees can and can't do. Without proper awareness and education, "I didn't know" could become the default response to actions taken over Web 2.0 applications.

There is no doubt that we must revisit existing enterprise practices to prepare for the transition from the traditional to the social enterprise. As Mike Vizard says, "Most new technology doesn't live up to its potential because the people and processes that need to adjust to its introduction aren't ready for the magnitude of change involved."[6]

Our guiding framework will help establish the CORE characteristics of social enterprises. *Connection* means converting ad hoc relations into long-lasting ones and promoting different forms and levels of interaction among the enterprise's stakeholders and services. The social enterprise should be a truly connected business, relating its employees, customers, partners, and services with each other and with the market. *Open* means creating new conversation channels with the business world. Open online APIs, for example, help achieve this goal by exposing an enterprise's internal capabilities to the external world. *Reachable* means facilitating ubiquitous accessibility to the social enterprise. It involves, among other things, being more responsive to different forms of social interaction, such as posting notes, chatting, and updating content. As the number of stakeholder devices (such as laptops, desktops, smartphones, tablets, and consoles) increases significantly, we must think about ways to efficiently create adaptable and user-friendly online social applications. Finally, *engagement* means creating a culture of community that relies on collaboration, sharing, and participation. Social applications can sustain this engagement with online communities, crowdsourcing, and social customer relationship management.

Acknowledgments

This work was partially supported by the National Institutes of Science and Technology for Software Engineering (INES) in Brazil, funded by CNPq and FACEPE grants 573964/2008-4 and APQ-1037-1.03/08. Zakaria Maamar's work was partially supported by the ZU Research Incentive Funds Program (grant R12045). The information and views set out in this article are those of the authors and do not necessarily reflect the official opinion of the Brazilian companies cited herein.

References

1. C. Kanaracus, "Gartner: Social Business Software Efforts Largely Unsuccessful for Now," *Infoworld*, 29 Jan. 2013; www.infoworld.com/d/applications/gartner-social-business-software-efforts-largely-unsuccessful-now-211813.
2. S.C. Seshadri, "Exploring Business and Collaboration with Web 2.0," *SETLabs Briefings J.*, vol. 7, no. 3, 2009, pp. 3–10.
3. A.P. McAfee, "Enterprise 2.0: The Dawn of Emergent Collaboration," *MIT Sloan Management Rev.*, vol. 47, no. 3, 2006, pp. 21–28.
4. *Forecast Analysis: Software as a Service, Worldwide, 2009–2014*, Gartner report, 2009.
5. R. Cross et al., "The Collaborative Organization: How to Make Employee Networks Really Work," *IEEE Eng. Management Rev.*, vol. 39, no. 1, 2011, pp. 59–68.
6. M. Vizard, "IBM: Business Processes Need to Get Social in 2013," *ITBusinessEdge*, Dec. 2012; www.itbusinessedge.com/blogs/it-unmasked/ibm-business-processes-need-to-get-social-in-2013.html.
7. E. Kajan et al., "The Network-Based Business Process," *IEEE Internet Computing*, vol. 18, no. 2, 2014, pp. 63–69.
8. V.A. Burégio et al., "Moving Towards 'Relationship-Aware' Applications and Services: A Social Machine-Oriented Approach," *Proc. 17th IEEE Int'l EDOC Conf. Workshops* (EDOCW 13), 2013, pp. 43–52.
9. A. Badia, "Databases as Social Entities," *IEEE Intelligent Systems*, vol. 27, no. 5, 2012, pp. 70–73.
10. J. Montemayor and C.P. Diehl, "Social Query: Looking for Social Signals from Online Artifacts," *Johns Hopkins APL Technical Digest*, vol. 30, no. 1, 2011, pp. 41–46.
11. W. Tan et al., "Social-Network-Sourced Big Data Analytics," *IEEE Internet Computing*, vol. 17, no. 5, 2013, pp. 62–69.
12. A. Nigam and N.S. Caswell, "Business Artifacts: An Approach to Operational Specification," *IBM Systems J.*, vol. 42, no. 3, 2003, pp. 428–445.

Vanilson Burégio is a software architect for the Brazilian Federal Data Processing Service (Serpro) and a lecturer in the College of Information Systems and Technology at Unibratec. His research interests include software engineering, social machines, Web-oriented architectures, enterprise 2.0, and software reuse. Burégio received a PhD in computer science from the Federal University of Pernambuco, Brazil. Contact him at vaab@cin.ufpe.br.

Zakaria Maamar is a full professor in the College of Information Technology at Zayed University, United Arab Emirates. His research interests include Web services, social networks, and enterprise computing. Maamar received a PhD in computer science from Laval University, Canada. Contact him at zakaria.maamar@zu.ac.ae.

Silvio Meira is a full professor in the Center for Informatics at the Federal University of Pernambuco, Brazil, chief scientist of the Recife Center for Advanced Studies and Systems (C.E.S.A.R.), and founder and current chairman of the board of Porto Digital. His research interests include social machines, social networks, and open source software. Meira received a PhD in computing from the University of Kent, UK. He is the chair of the National Institute of Science and Technology for Software Engineering in Brazil. Contact him at silvio@meira.com.

Editor: George Pallis • gpallis@cs.ucy.ac.cy

CometCloud

Enabling Software-Defined Federations for End-to-End Application Workflows

Javier Diaz-Montes, Moustafa AbdelBaky, Mengsong Zou, and Manish Parashar • *Rutgers University*

Emerging applications, from big science to the Internet of Things, increasingly involve dynamic and data-driven end-to-end workflows with large and often heterogeneous requirements. CometCloud aims to provide infrastructure and programming support for enabling such workflows via flexible, software-defined synthesis of custom cyberinfrastructure through the autonomic, on-demand federation of geographically distributed compute and data resources.

Emerging applications increasingly involve dynamic and data-driven end-to-end workflows. For example, end-to-end science and engineering applications integrate data sources (such as monitoring, observations, and experiments) with computational modeling, analytics, and visualization. These workflows typically have large and often heterogeneous requirements, and necessitate platforms that dynamically combine resources across systems and data centers — for example, to aggregate capacity and capabilities. Furthermore, emerging pervasive computational ecosystems, powered by the Internet of Things (IoT), and the resulting proliferation of data sources could enable a new class of application workflows that can fundamentally transform our ability to manage and optimize our lives and environment. However, these applications will once again require that we seamlessly and opportunistically combine pervasive digital data sources and computational power.

The CometCloud project at the Rutgers Discovery Informatics Institute (RDI2) aims to provide infrastructure and programming support for such end-to-end application workflows. CometCloud enables flexible, software-defined synthesis of custom cyberinfrastructure through the autonomic, on-demand federation of geographically distributed compute and data resources. The CometCloud team has been working closely with scientists and engineers from different domains to understand application workflows and their requirements, and to explore appropriate usage modes and abstractions. This has led CometCloud to expose a software-defined federated cyberinfrastructure using cloud abstractions to support various programming paradigms and application requirements.

Here, we present CometCloud's architecture and design, and employ sample use cases to illustrate how CometCloud supports dynamic application workflows.

CometCloud Overview

CometCloud is an autonomic framework designed to enable highly heterogeneous, dynamically federated computing and data platforms that can support end-to-end application workflows with diverse and changing requirements.[1] This occurs through autonomic, on-demand federation of geographically distributed compute and data resources as applications need them, and by exposing the federation using elastic cloud abstractions and science-as-a-service platforms. Consequently, CometCloud can create a nimble and programmable environment that autonomously evolves over time, adapting to changes in infrastructure and application requirements. Figure 1a shows an overview of the CometCloud architecture, including its three key layers: infrastructure/federation, autonomic management, and programming/interface.

The infrastructure/federation layer manages dynamic resource federation and provides

 1089-7801/15/$31.00 © 2015 IEEE Published by the IEEE Computer Society

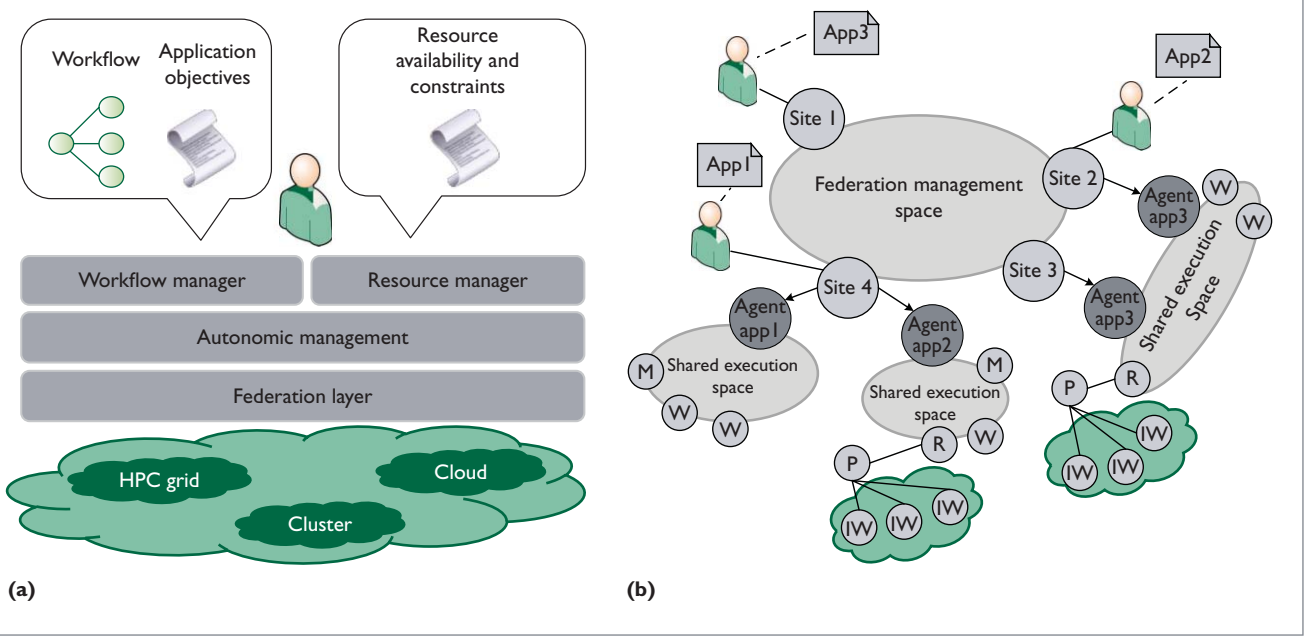


Figure 1. CometCloud. (a) The CometCloud architecture comprises three key layers: infrastructure/federation, autonomic management, and programming/interface. (b) The CometCloud coordination model is used to orchestrate different aspects of the federation.

essential services. Several key components make up this layer. The first is an information lookup system built on a content-based distributed hash-table (DHT) based on a structured peer-to-peer overlay. This system is used for information discovery (for instance, to locate resources using their attributes). It maintains content locality and guarantees that content-based information queries, specified using keywords and wildcards, are satisfied with bound costs (in terms of the number of hops required to find the data).

The second is a scalable, decentralized, shared coordination space, built on top of the DHT,[2] called CometSpace, which all resources in the federation can access associatively. CometSpace provides tuple-space-like abstraction for coordination and messaging, and enables coordination in the federation model (Figure 1b). Specifically, we define two types of coordination spaces. First, a single-management space spans across all resource sites, creating and orchestrating the federation. Second, multiple shared execution spaces are created on-demand during application workflow executions to satisfy computational or data needs. Execution spaces can be created within a single resource site, or can burst to others, such as public clouds or external high-performance computing (HPC) systems.

CometCloud federation is created dynamically and collaboratively; resources or sites can join or leave at any point, identify themselves (using security mechanisms such as public/private keys), negotiate the federation terms, discover available resources, and advertise their own resources and capabilities.[3]

Next, the autonomic management layer lets users and applications define objectives and policies that drive resource provisioning and application workflow execution while satisfying user constraints (such as budgets and deadlines) and application requirements (types of resources). The autonomic mechanisms in place not only provision the right resources when needed, but also monitor the execution's progress and adapt it to prevent violations of established agreements.[4]

Finally, the programming/interface layer provides interfaces to independently describe application workflows and resources. Currently, we can describe application workflows, using XML documents, as a set of stages defining input and output data, dependencies to other stages, and scheduling policies, and possibly annotated with specific objectives and policies.

The CometCloud resource interface lets providers and users declaratively specify resource availability as well as policies and constraints to regulate their use. For example, a provider can offer resources only at certain times of the day; users might require certain resource capabilities or capacities, or specific connectivity; users might prefer certain resource types over others (such as HPC versus clouds), or want to use cloud resources only if they're within a desired price range; or users

might want to only use resources within a specific geographic region due to data movement regulations. These constraints collectively define the set of resources that are federated at any instant in time and that the application can use.

CometCloud Use Case Scenarios

We next present use cases to illustrate how CometCloud and its federation model can be used to enable real-world applications.

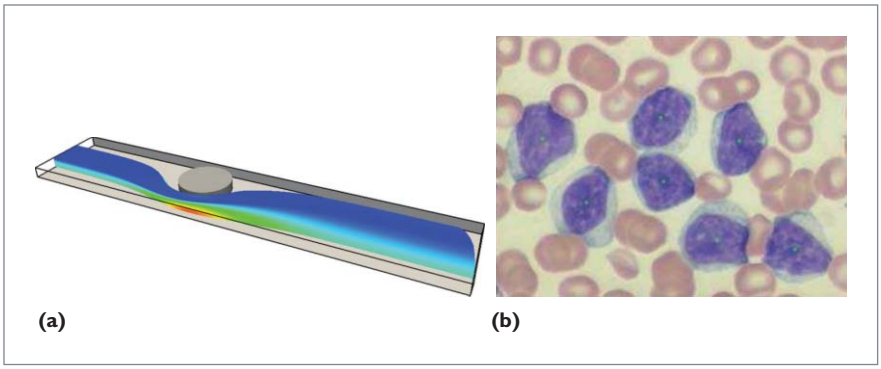


Figure 2. CometCloud use cases for large-scale science and engineering. We can see sample results for (a) the fluid flow in microchannel[3] and (b) cancer detection use cases.

Large-Scale Science and Engineering

The complexity of many science and engineering problems requires computational capacity exceeding what an average user can expect from a single computational center. The analysis of high-dimensional parameter spaces or uncertainty quantification by stochastic sampling are just two examples of a broad class of problems that are becoming increasingly important in a wide range of application domains. Although we can view many of these problems as a set of independent tasks, their collective complexity easily requires millions of core hours on any state-of-the-art supercomputer, and throughput that a single multi-user queuing system can't sustain. We've used CometCloud to enable a range of applications in this area,[5] and highlight two specific use cases here.

In the first use case, we explored using aggregated HPC resources to solve a large-scale engineering problem. We showed how CometCloud can help build a computational federation that's elastic, resilient, scalable, and easy to use by end users. We focused on understanding fluid flow in microscale devices. During the experiment, we opportunistically federated resources as they became available and released any that experienced problems or ran out of allocations. We federated 10 resources across six institutions from three countries. This experiment lasted 16 days and executed 12,845 tasks. It consumed 2,897,390 core hours and generated 398 Gbytes of data. Our federation model's extreme flexibility enabled a sustained performance (see Figure 2a).[3]

In the second use case, we explored how to enable data-driven applications on top of national cyberinfrastructure to efficiently support state-of-the-art analytics across distributed image databases. We focused on content-based histopathology image retrieval (CBIR), in which datasets can be geographically distributed. In this experiment, we created a hybrid federation of HPC and cloud resources to perform parallel searching of content-wise similar images in several datasets. We showed not only the feasibility but also a significant reduction in the overall computational time, which ensures our proposed solution's practical utility for near real-time medical diagnosis (see Figure 2b).[6]

Smart Infrastructure

The rise of Internet-connected instrumentation and the extraordinary growth of digital data sources have led to an unprecedented amount of data that usually can't be predetermined. Due to data and resource's heterogeneous and distributed nature, we must rethink how we store, process, and analyze them to provide insight in a timely manner. Examples include the wide variety of sensor-network-based applications, in which sensors interface with real-world artifacts and must respond to unpredictable physical phenomena. In this context, we discuss two specific CometCloud examples.

The first use case targets data analysis from electricity meters to support smart (power) grids, such as electric vehicle charging stations. This scenario generates multiple datastreams in which raw data is continuously transmitted at variable and unpredictable rates. We combined CometCloud with a reference net (a particular type of Petri net) based interpreter to support simultaneous datastream processing and enable on-demand scaling up or down of heterogeneous computational resources to ensure promised quality of service (throughput) for each datastream (see Figure 3a). CometCloud lets us easily extend this model to, for example, exploit geolocation and provisioning resources at the closest datacenter.[7]

The second use case is in the context of smart buildings, where sensors and actuators control the building's energy consumption. Energy optimization requires the real-time use of sensor data, with several parameters needing optimization based on

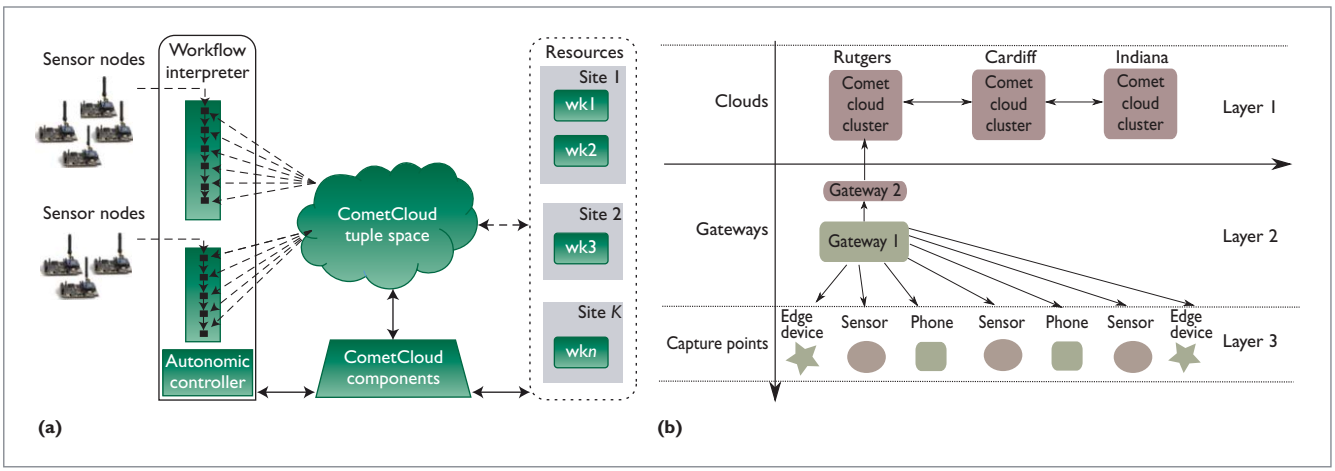


Figure 3. CometCloud use cases in the smart infrastructure domain. We can see the architectures for the (a) streaming data[7] and (b) building analytics[8] use cases.

a particular building representation. Because sensors can provide readings in 15- to 30-minute intervals, any simulation or optimization must be carried out with a similar interval. We've developed a multilayer cloud infrastructure that exploits the available computation at the cloud's edge by distributing processing over sensing nodes, multiple intermediate or gateways nodes, and cloud resources. Using this infrastructure, we explored questions such as where processing should be carried out, what processing should be undertaken centrally versus at an edge node, and how processing can be distributed across multiple datacenter locations to achieve QoS and cost targets.[8]

Analogous to how clouds have revolutionized the way we acquire and use IT resources, we believe that software-defined environments can fundamentally affect how resources (compute, data, and networking) are federated and customized. These environments will be especially important in an era where digital data sources proliferate (including geographically distributed sensors, mobile devices, and instrumented infrastructure) and nontrivial computational power is ubiquitous. Dynamic, data-driven applications can potentially transform our ability to understand and manage our lives and our environment — we can envision data-driven and information-rich pervasive computational ecosystems that seamlessly and opportunistically federate these data and computing power to model, manage, control, adapt, and optimize virtually any realizable subsystem of interest.

We're currently exploring usage modes where data is processed and insights gleaned close to the source rather than necessarily transporting it to remote data processing resources. Specifically, we're looking at application formulations where data is processed in situ (at the edge devices) and in transit (along the data path), transforming real-time data into knowledge that can drive critical decisions.

Acknowledgments

This work is supported in part by the US National Science Foundation under OCI-1339036, OCI-1310283, OCI-1441376, and IIP-0758566, and by IBM via Open Collaboration Research (OCR) and Faculty awards.

References

1. H. Kim and M. Parashar, "CometCloud: An Autonomic Cloud Engine," *Cloud Computing: Principles and Paradigms*, Wiley, 2011, pp. 275–297.
2. Z. Li and M. Parashar, "Comet: A Scalable Coordination Space for Decentralized Distributed Environments," *Proc. Int'l Workshop on Hot Topics in Peer-to-Peer Systems*, 2005, pp. 104–111.
3. J. Diaz-Montes et al., "Federated Computing for the Masses — Aggregating Resources to Tackle Large-Scale Engineering Problems," *Computing in Science & Eng.*, vol. 16, no. 4, 2014, pp. 62–72.
4. J. Diaz-Montes et al., "Data-Driven Workflows in Multi-Cloud Marketplaces," *Proc. IEEE Int'l Conf. Cloud Computing*, 2014, pp. 168–175.
5. M. Parashar et al., "Cloud Paradigms and Practices for Computational and Data-Enabled Science and Engineering," *Computing in Science & Eng.*, vol. 15, no. 4, 2013, pp. 10–18.
6. X. Qi et al., "Content-Based Histopathology Image Retrieval using CometCloud," *BMC Bioinformatics*, vol. 15, no. 287, 2014, pp. 1–17.
7. R. Tolosana-Calasanz et al., "Extending CometCloud to Process Dynamic Data Streams on Heterogeneous Infrastructures," *Proc. 2014 Int'l Conf. Cloud and Autonomic Computing* (CAC 2014), 2014, pp. 196–205.
8. I. Petri et al., "In-Transit Data Analysis and Distribution in a Multi-Cloud Environment using CometCloud," *Proc. Int'l Workshop Energy Management for Sustainable Internet of Things and Cloud Computing*, 2014, pp. 1–6.

Javier Diaz-Montes is an assistant research professor at Rutgers University and a member of the Rutgers Discovery Informatics Institute (RDI2). His research interests are in parallel and distributed computing and include autonomic computing, cloud computing, virtualization, and scheduling. Diaz-Montes received a PhD in computer science from the Universidad de Castilla-La Mancha, Spain. Contact him at javidiaz@rdi2.rutgers.edu.

Moustafa AbdelBaky is a PhD candidate in the Electrical and Computer Engineering Department at Rutgers University, a member of RDI2 and the NSF Cloud and Autonomic Computing Center at Rutgers, and a research intern at IBM T.J. Watson Research Center. AbdelBaky received the IBM PhD fellowship three years in a row, and is a member of IEEE and SIAM. Contact him at moustafa.a@rutgers.edu.

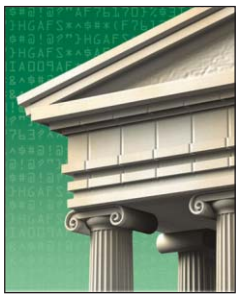
Mengsong Zou is a PhD student in the Computer Science Department at Rutgers University, and a member of RDI2. His research interests lie in parallel and distributed computing, cloud computing, and scientific workflow management. Zou received an MS in computer science from Huazhong University of Science and Technology, China. Contact him at mengsongzou@gmail.com.

Manish Parashar is a professor in the Department of Computer Science at Rutgers University; the founding director of RDI2), the NSF Cloud and Autonomic Computing Center at Rutgers, and the Applied Software Systems Laboratory, and is associate director of the Rutgers Center for Information Assurance. Parashar is an IBM Faculty Award, Tewkesbury Fellowship, and Enrico Fermi Scholarship recipient. He's a fellow of AAAS and IEEE/IEEE Computer Society, and a senior member of ACM. Contact him at parashar@rutgers.edu.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.

Internet Governance

Editor: Virgilio Almeida • virgilio@dcc.ufmg.br



The Origin and Evolution of Multistakeholder Models

Virgilio Almeida • *Federal University of Minas Gerais*

Demi Getschko • *Pontificia Universidade Católica de São Paulo*

Carlos Afonso • *Instituto Nupef, Rio de Janeiro*

Various domains have adopted multistakeholder models (MSMs) to address and deal with global challenges, such as sustainability, environment, climate, and Internet governance. Here, the authors examine the use of MSMs and their historical evolution, fundamentals, and characteristics. They also present examples of how such models are used in the global Internet governance ecosystem. Finally, the article presents a series of research questions that can be tackled to improve the efficiency of multistakeholder processes.

Multi-stakeholder processes aim to bring together all major stakeholders in a new form of communication, decision-finding (and possibly decision-making) on a particular issue; are based on recognition of the importance of achieving equity and accountability in communication between stakeholders; involve equitable representation of three or more stakeholder groups and their views; are based on democratic principles of transparency and participation; and aim to develop partnerships and strengthened networks between and among stakeholders.[1]

The Internet has enabled politics, the economy, work, entertainment, and personal relationships to develop increasingly in cyberspace. No longer just a technology, the Internet now has a strong and broad social and economic impact on all countries. Cyberspace has become strategic for development in most nations. As a result, countries have been building local policies and frameworks for cybersecurity and Internet governance.

The next couple of years will be crucial in redrawing the map of global Internet governance. Edward Snowden's revelations of massive surveillance and the announcement from the US National Telecommunications & Information Administration (NTIA) that it will seek to withdraw its role as the administrator of the IANA contract could change the global Internet governance ecosystem.

The Working Group on Internet Governance (WGIG), set up by the UN secretary-general in 2003, introduced the first working definition for the term: "the development and application by governments, the private sector, and civil society, in their respective roles, of shared principles, norms, rules, decision-making procedures, and programs that shape the evolution and use of the Internet."[2] Since then, different countries have adopted different models. These vary from utopian self-governing models of individual liberty beyond the reach of government control, to models in which Internet-related activities are subject to regulation through governments and regulatory agencies. Many variations of Internet governance models rely on the concepts and ideas introduced by multistakeholder processes.[3,4]

The recent Global Multistakeholder Meeting on the Future of Internet Governance (NETmundial) in Brazil produced an outcome document recommending that *multistakeholder models* (MSMs) be the central axis for the evolution of Internet governance.[5] This document presents the Internet governance framework as a distributed and coordinated ecosystem involving various organizations and fora. Governance bodies must be inclusive, transparent, and accountable, and their

 Published by the IEEE Computer Society 1089-7801/15/$31.00 © 2015 IEEE

structures and operations must follow an approach that enables all stakeholders to participate to address the interests of all who use the Internet as well as those who aren't yet online. So, the reader might be wondering – what is the origin of the multistakeholder concept, and how has it been applied to practical matters?

Here, we present an overview of MSMs, their use, and their historical evolution, and examine their adoption in various domains, particularly the global Internet governance ecosystem.

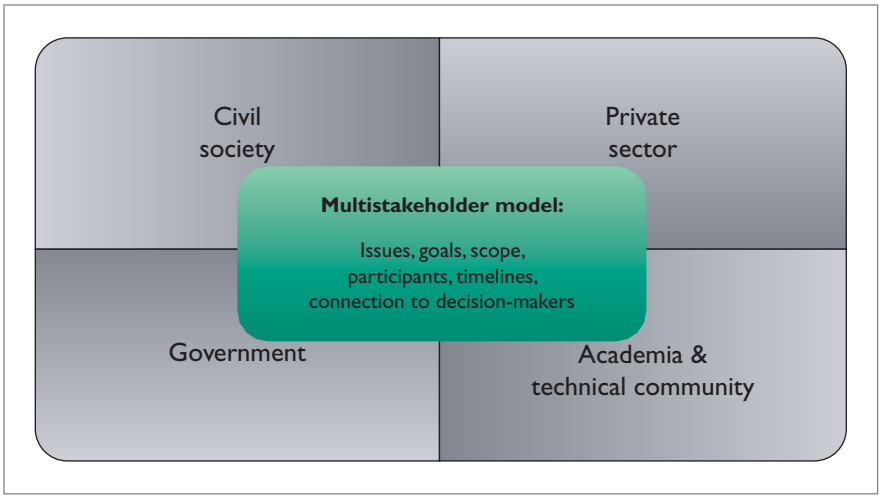


Figure 1. Main components of a multistakeholder model. We can see the different stakeholders that are typically involved in the operation of an MSM.

The Origin and Fundamentals of MSMs

In 1992, the World Conference on Environment and Development (UNCED), held in Rio de Janeiro,[6] alerted the world to several global environmental and developmental problems and placed sustainability on the agenda of the international community, national governments, and representatives from various sectors.[1] To achieve broad support for sustainable principles, various segments of society clearly had to learn how to listen to each other and integrate different views and interests to achieve practical solutions that would lead to a more sustainable world. These environmental discussions emphasized the roles of *stakeholders*: individuals or groups that have an interest in a particular decision because they can either influence or be affected by it.

The very first organization to recognize the relevant role of multiple stakeholders in the discussion of global issues was the International Labour Organization (ILO), which in 1919 created a model with representatives from governments, employers, and unions.[1]

More recently, multistakeholder discussions took place at the UN Commission on Sustainable Development (CSD), which introduced the concept as an engagement model within the UN for sustainable development issues. "Agenda 21" for 1992's UNCED is the first UN document to include different stakeholders' roles in a global agreement.[6]

The adoption of multistakeholder processes has been slow because many governments and intergovernmental bodies don't feel comfortable with the growing influence of certain stakeholders, viewing them as unelected representatives who lack legitimacy.[1] But the advantages of MSMs surpass their difficulties and create mutual benefits for the whole of society. MSMs have the potential to promote better decisions through broader inputs.

Several features are common to existing MSMs. In general, MSMs vary with regard to the issues being addressed, which range from healthcare, poverty, and gender equity to Internet governance. Figure 1 shows a typical MSM composition, including stakeholders and the model's main components: goals, participants, scope, timelines, and connection to official decision makers.

Goals

MSMs can be designed to reach goals that would be unachievable if each stakeholder worked alone. For instance, the goal of preserving a unified Internet that's unfragmented, interconnected, interoperable, open, inclusive, secure, stable, resilient, and trustworthy wouldn't be possible if only governments were involved in the agreement.

Participants

Commonly, MSMs involve representatives from different groups interested in or affected by the issue under examination. Their composition should thus be highly diverse. For example, in the case of Internet governance, the main stakeholders are civil society, government, the private sector, and technical and academic communities.

Scope

MSMs can help address issues at national, regional, or international levels. For instance, ICANN is a multistakeholder body that operates at the international level. The five Regional Internet Registries (RIRs) manage the distribution of number identifiers allocated by IANA. They are multistakeholder bodies that operate regionally.

Timelines

MSMs can be constructed for single events or open-ended processes, depending on the issue under examination. For example, NETmundial was designed to be a one-time event

organized by a multistakeholder committee.[5] The Internet Governance Forum (IGF) has a mandate from the World Summit on the Information Society (WSIS) with regard to convening an annual forum for multistakeholder policy dialogue. ICANN is a permanent organization with a multistakeholder structure for coordinating the Internet's naming system.

Connection to Decision-Makers

Multistakeholder bodies can interact in different ways with official decision-making processes at the international, regional, or national levels. Some MSM bodies are purely informative. Others can develop best practices concerning a particular issue and present them to governments.

Some MSM bodies are purely informative. Others can develop best practices concerning a particular issue and present them to governments.

Multistakeholder bodies can also conduct participatory monitoring of issues that affect society, such as a deforestation index or the quality of Internet access provided by telecommunications operators.

Multistakeholder Bodies for Internet Governance

From a historical viewpoint, the opportunities for various stakeholders to participate in governance processes increased with the end of the Cold War in the early 1990s. UNCED was one of the first multistakeholder conferences. Coincidentally, it was the first UN event to use the Internet, enabling the first opportunities for online participation. This was especially useful for civil society organizations that couldn't afford to be in Rio de Janeiro, and even for some governments from Africa that had no other adequate means of remote communication with their local bases. This was the beginning of a sequence of global conferences that now use multistakeholder presence in their discussion threads. Of special note was the second World Conference on Human Rights (WCHR) in Vienna (June 1993), which resulted in the Vienna Declaration on Human Rights approved by 171 countries and then adopted by the UN General Assembly (UNGA). The conference resulted in the creation of the High Commissioner for Human Rights of the UN in December of the same year.

Multistakeholder participation has since been a feature of UN conferences and their agencies. One interesting question is, what are the limits of this participation? Pluralistic processes involve civil society, the private and public sectors, academic and technical communities, and other interest groups that join, conscious of their distinct roles and responsibilities, for a stated common goal. In the past decade, we can point to important participation from organized civil society with relevant interests in events such as campaigns for the right to communication in the information society, and the strong presence in the WSIS process, where for the first time the "Internet governance" concept appeared and was elaborated with significant depth.

In fact, the Internet governance domain offers several examples of multistakeholder processes. For instance, the IGF is a multistakeholder forum for policy dialogue on Internet governance issues.[2] It's open and inclusive, bringing all stakeholders together to exchange information and share best practices on Internet-related public policy issues. Another example is the NETmundial meeting, which was prepared as a multistakeholder conference to discuss Internet governance's future development.[7] The conference's importance stems from how it was organized and executed. The meeting's multistakeholder nature involved civil society segments, governments, private companies, and academic and technical communities worldwide. Representatives from more than 100 countries approved by rough consensus a document of principles and a roadmap for the evolution of the Internet governance ecosystem.[5]

Pluralistic Processes in Internet Resource Management

Pluralistic decision-making processes, with their specific limitations, are present in the structures of Internet resource management. Evolving Internet technologies are coordinated by organizations such as the IETF, which proposes standards and parameters through recommendations adopted by consensus after they're discussed in open forums. The global coordination of IP number distribution is executed in practice by all five RIRs. This group constitutes a coordinating forum that seeks worldwide consensus for its policies: the Number Resources Organization (NRO). These policies are developed in pluralistic dialogues at regular meetings, open to participation from all sectors. Even though the central stock of network numbering resources falls formally under ICANN through the IANA function and is under contract between ICANN and the NTIA, in practice, the number distribution mechanisms are governed by the RIRs.

ICANN's focus is coordinating mnemonics network addressing (domain names). Its multistakeholder participation structure is well organized through support organizations and committees in which governments (Government Advisory Committee), industry, registries, registrars (Generic Names Supporting Organization and Country Code Names Supporting Organization), and civil society (Non-Commercial Stakeholders Group and At-Large Advisory Committee) strongly participate and elect representatives to ICANN's board. A certain number of board members are nominated by committee.

Some cases reveal pluralist approaches' limitations — in particular, the decision-making processes aren't entirely derived from such participation. All stakeholders recognize that they play a more consultative and advisory role in the organization's decision-making processes. The board always makes the final decisions. This doesn't mean that different stakeholders accept the status quo. The different sectors represented in ICANN seek ways to expand their influence on decisions in a context where imbalances are obvious — for example, due to economic power or political leverage that could favor some stakeholders.

Network Users as Stakeholders

Computer networks have existed with a wide variety of features and forms since the early 1970s. In contrast to this variety, the user community was, until the mid-80s, almost homogeneously composed of academics looking to remotely use computing resources. The popularization of PCs led ordinary people to use them in their own organizations and thus stimulated a growing individual involvement among communities with specific interests for exchanging data and ideas. The old bulletin board systems, which initially provided a way to exchange information in an isolated environment, demonstrated the yearning for direct communication between users on many issues. In the early 1980s, another platform generated yet more synergy: the deployment of USENET. Thousands of machines based on the UNIX standard protocol (UUCP) brought integration to user groups. The spread of many forums on various topics (USENET News) enabled open discussions and the creation of interest groups. The spread of electronic mail and mailing lists definitely brought the second wave of network users: individuals and nongovernmental organizations (NGOs) quickly joined academics in taking advantage of the new media for a faster, cheaper, and more efficient way to communicate.

In Brazil, for example, academic networks arose in the late 1980s. In 1991, civil society became the second wave, and the wide diversity of protocols (Bitnet, UUCP, DECnet, X.28, X.25, X.400, and so on) quickly coalesced toward a single solution: TCP/IP — the Internet. The creation of the World Wide Web 25 years ago shaped a new scenario in which users not only had access to information but also found efficient ways to be active in cyberspace, expressing their views and fully participating in the network. Interestingly, before the Web, little talk focused on security and privacy threats, or spying on data traffic. Even spam attacks and malicious code were rare at the time, in part because of the analog nature of communication in narrow bandwidth links, and in part because participants were academics and nonprofits. The Internet's expansion outside the limits of these initial adopters, together with its disruptive characteristics and ability to extend beyond national boundaries, changed the networking scenario. Many countries perceived the Internet as something different from the traditional and highly regulated telecommunications world and began to work on ways to govern it.

An Early MSM

The need for permanent discussions about governance models for the Internet stems from its impressive growth, both in number of users and strategic importance. The creation of a multistakeholder body for Internet governance began initially within a country and then became a feasible and globally applicable model. By examining the Brazilian case, we can make this quite clear. The Brazilian Internet Steering Committee (CGI.br) was created in 1995. It anticipated some of the features ICANN (established in 1998) exhibits, and also became a reference for the discussions introduced at WSIS 2005.

Many countries perceived the Internet as something different from the traditional and highly regulated telecommunications world and began to work on ways to govern it.

In 1995, the Brazilian government created CGI.br as a multistakeholder, nonregulatory governance body. Two years later, Brazilian telecommunications legislation defined the Internet as a "value-added service" that made it different from the telecommunications infrastructure that supports it. This innovative approach let the Internet grow quickly in Brazil. The CGI board has 21 members: nine from government organizations, four from civil society, four from the private sector,

and four from the academic and technical communities. The government members are appointed, and all other members are elected by their respective communities. Note that no single sector, even government, has a majority of votes on the board. Everything has to be negotiated among the participants. The CGI board's composition clearly reflects the Internet's multistakeholder nature. It works without public funds; the community supports CGI when registering under the .br domain (that is, the ccTLD). Any budget surplus is used in the harmonious development of the Internet in Brazil. The innovative MSM and its nongovernmental nature isn't always well understood by the public. The same observation applies to its "nonregulatory" behavior, which is always contrasted with the traditionally regulated environment in the telecommunications industry.

Internet Governance and Sustainable Development

Internet governance and sustainable development are processes that share some similarities. The concept of sustainable development refers to development that meets present needs without compromising the needs of future generations. The concept that underpins Internet governance also refers to the principles, norms, rules, and procedures that will shape tomorrow's Internet. So, the two processes work with values that are essential for future generations.

Both Internet governance and sustainable development require a process of dialogue and consensus building from all stakeholders to construct viable solutions, work to implement them, and monitor and assess the outcomes. MSMs are central to both processes, which face global challenges with strong social and economic impacts.

Consensus Building

Consensus building is a key activity for multistakeholder governance bodies. Stakeholder representatives present their views and positions on a particular issue. Then, they engage in a dialogue to achieve mutual understanding of problems. Based on this improved understanding, the body's chair or mediator seeks a consensus. The quest for consensus in MSMs is almost never an organized or orderly process. Because all stakeholders participate on equal footing, discussions are usually messy, with unpredictable developments. This equal footing basis is an essential MSM characteristic that aims at reducing specific groups' traditional influence and power, such as economic and political influence. Every stakeholder has the right to be heard based only on their perspective on the problem. In a consensus-building process, the different stakeholders work to design solutions that minimize their differences. Although participants might not be in accord with all aspects of the agreement, consensus is reached if all stakeholders are willing to accept the decision and participate in its implementation.

MSMs are always evolving. They are a new species in the biodiversity of structures for governing complex issues. Climate change, Internet governance, and water management are just some of the pressing global issues that have been experimenting with different forms of MSMs to find agreements that could lead to sustainable solutions.

However, MSMs aren't simple to implement. There are inherent difficulties in running a multistakeholder body. The implementation should tailor the process to multiple stakeholders' specificities, such as decision timing, representativeness, and language. These values are essential to achieve credibility and legitimacy within the different communities — a mandatory characteristic for making the multistakeholder decision processes viable and doable. Additionally, the decision process is sometimes too cumbersome and depends to great extent on the governing body's leadership. Clearly, a need exists for common learning on multistakeholder processes.

Many questions regarding the structure and dynamic of MSMs remain unanswered:

- How do we identify the most adequate set of stakeholders to work on a particular issue?
- How do we define the mechanisms and criteria for selecting representatives from different groups?
- How do we avoid letting influential NGOs and corporate power capture the multistakeholder process?
- How can crowdsourcing techniques be used to provide input into the dialogues of difficult issues?
- What technologies could help stakeholder representatives "feel the pulse" of their constituencies?
- What technologies could allow multistakeholder governance bodies to monitor the results of their agreements?
- What kind of technological framework will facilitate dialogue in a multistakeholder body so that a minimum consensus can be achieved?
- What type of technology could be developed to accelerate the decision process in multistakeholder organizations?
- What kind of theoretical model will support consensus building and decision making in multistakeholder environments?

These questions represent an opportunity for the research and development of new technologies that bring more efficiency to Internet governance processes.

References

1. M. Hemmati, *Multistakeholder Processes for Governance and Sustainability: Beyond Deadlock and Conflict*, Earthscan, 2002.

2. *Report of the Working Group on Internet Governance*, June 2005, www.wgig.org/docs/WGIGREPORT.pdf.
3. L.W. Fransen and A. Kolk, "Global Rule-Setting for Business: A Critical Analysis of Multi-Stakeholder Standards," *Organization*, vol. 14, no. 5, 2007, pp. 667–684.
4. L. DeNardis and M. Raymond, "Thinking Clearly about Multi-Stakeholder Internet Governance," *Proc. 8th Ann. Conf. Global Internet Governance Academic Network* (GigaNet), 2013; http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2354377.
5. "NETmundial Multistakeholder Statement of Sao Paulo," Apr. 2014; http://netmundial.br/wp-content/uploads/2014/04/NETmundial-Multistakeholder-Document.pdf.
6. "Agenda 21," *United Nations Conf. Environment & Development*, 1992; http://sustainabledevelopment.un.org/content/documents/Agenda21.pdf.
7. V. Almeida, "The Evolution of Internet Governance: Lessons Learned from NETmundial," *IEEE Internet Computing*, vol. 18, no. 5, 2014, pp. 65–69.

Virgilio A.F. Almeida is a professor in the Computer Science Department at the Federal University of Minas Gerais, Brazil. His research interests include large-scale distributed systems, Internet, social computing, and performance modeling and analysis. Almeida received a PhD in computer science from Vanderbilt University. He's chairman of the Brazilian Internet Steering Committee (CGI.br). Contact him at virgilio@dcc.ufmg.br.

Demi Getschko is an associate professor at Pontifícia Universidade Católica de São Paulo. His research interests include computer architecture and computer networks. Getschko received a PhD in electronic engineering from the University of Sao Paulo, Brazil. He's a board member of CGI.br, CEO of the Brazilian Network Information Center (NIC.br), and an inductee to the Hall of Fame of the Internet in the "Global Connectors" category. Contact him at trieste@gmail.com.

Carlos Afonso is an Internet pioneer in Brazil and an active representative of global civil society in Internet governance processes. His research interests include economics and sustainable development, theory of the state, and impacts of the Internet in politics and society. Afonso holds a masters degree in economics and has concluded doctoral studies in social and political thought from York University in Toronto. He's a founding member and former chair of the Association for Progressive Communications, and a board member for CGI.br. Contact him at ca@cafonso.ca.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.

Standards

Editor: Barry Leiba • barryleiba@computer.org

Cipher-Suite Negotiation for DNSSEC: Hop-by-Hop or End-to-End?

Amir Herzberg • *Bar Ilan University*

Haya Shulman • *Technische Universität Darmstadt*

Cipher-suite negotiation lets parties negotiate the use of the best cryptographic algorithms. A lack of cipher-suite negotiation in DNS Security Extensions (DNSSEC) introduces several problems. The authors discuss two proposed designs for cipher-suite negotiation in DNSSEC: hop-by-hop and end-to-end.

A *cipher suite* is an ordered set of one or more cryptographic algorithms, each implementing a scheme (function) used by a cryptographic protocol. For example, the `RSA_WITH_RC4_128_MD5` cipher suite is used by protocols such as Transport Layer Security (TLS) and IPsec, and employs RSA for key exchange, RC4 with a 128-bit key for bulk encryption, and MD5 for hashing.

Cipher-suite negotiation refers to the process of selecting the cipher suite that a protocol will use when running between parties, and is useful when participants support multiple cipher suites. Many standard cryptographic protocols, such as Internet Key Exchange (RFC 2409), Secure Shell (RFC 4253), Secure Sockets Layer (RFC 6101), and TLS (RFC 5246), use cipher-suite negotiation to ensure that the parties select the "best" cipher suite they jointly support and thus use better (more secure and efficient) algorithms.

DNS Security Extensions (DNSSEC; RFCs 4033–4035) is an exception among the IETF cryptographic standards. Although it supports multiple signature algorithms and hash functions,[1] such as RSA and elliptic curves, DNSSEC doesn't support cipher-suite negotiation — that is, no mechanism enables the endpoints to identify the best set of algorithms, keys, and signatures to use in securing DNS records. Instead, DNSSEC requires responses to contain all the keys and the signatures the target zone supports, even if some of those algorithms are unsupported or unneeded by the resolver. This can result in significant bloat; DNS responses containing DNSSEC extensions often exceed the 512-byte maximal packet size of non-extended responses. Almost a third are larger than the maximum transmission unit (MTU), causing fragmentation.[2] In particular, DNSSEC responses for ANY-type queries can reach 5,000 bytes or more, whereas plain ANY-type responses are less than 1,000 bytes.

To address these issues, we review two approaches for cipher-suite negotiation in DNSSEC, hop-by-hop and end-to-end, and discuss their advantages and shortcomings. The main difference between them pertains to traversing legacy DNS proxies, and we discuss how each approach handles this.

DNS Security Extensions

When no protection is employed, DNS requests and responses can be altered by attackers. DNSSEC (RFCs 4033–4035) was designed to prevent cache poisoning by providing data integrity and origin authenticity via cryptographic digital signatures over DNS resource records. These digital signatures enable the receiving resolver, which supports DNSSEC validation, to verify that the data in a DNS response is the same as the data published in the zone file of the target domain.

Published by the IEEE Computer Society
1089-7801/15/$31.00 © 2015 IEEE

Why Not Send All Signatures?

The DNSSEC standard specifies that name servers should send all relevant signatures and keys to the requesting resolvers, and the resolvers should verify all the ciphers that they support. This seems to provide maximal security. Indeed, as long as one algorithm is secure, an attacker is prevented from forging records (this is a simple case of a robust combiner[3]).

However, the lack of cipher-suite negotiation implies that when a zone supports multiple signature algorithms, significant computation and communication overhead will result. In addition, large responses inflict interoperability problems with the Internet infrastructure and expose it to attacks — for example, it can be abused to flood victim networks with large packets. These considerations can impede DNSSEC adoption as well as the support of more secure or more efficient algorithms.

More Signatures Can Reduce Security

DNSSEC responses are often larger than the MTU of the networks in the path from the name server to the resolver. Thus, they become fragmented by the name server or a router along the path. Ironically, for IP defragmentation cache-poisoning, an attacker can exploit fragmented DNS responses to inject spoofed DNS records or disrupt resolver-to-name-server communication. Specifically, by sending spoofed fragments, the attacker can trick the defragmentation mechanism into reassembling them together with the fragments from the legitimate source (name server).[4,5]

In addition, attackers often abuse DNSSEC signed responses to launch amplification denial-of-service (DoS) attacks to clog victim networks and hosts. In such an attack, the attacker sends many requests to one or more DNS servers, using a spoofed (fake) source IP address for the victim. Name servers respond to these requests by sending (much larger) responses to the IP address that originated the DNS request.

The amplification factor is the ratio between the number of response bytes sent by the amplifying (benign) DNS server to the number of bytes sent by the attacker's hosts in the corresponding requests. With DNSSEC signed responses, the ratio can be as high as a hundred.[6] Indeed, although DNSSEC deployment is still limited, it's already been abused in the largest DoS attacks in recent years, with reported bandwidths of 100 Gbps in 2010, 60 Gbps in 2011 and 2012, and 300 Gbps in 2013 launched against Spamhaus and Cloudflare (see www.arbornetworks.com for details).

The Impact of No Negotiation

To avoid interoperability problems and overhead, zone administrators usually use a limited number of (weak) algorithms and typically short keys, such as 1024-bit RSA (a measurement study of the algorithms the signed zones support is available elsewhere[7]). Indeed, because the US National Institute of Standards and Technology (NIST) and IETF standards (RFCs 4033–4035) mandate 1024-bit RSA — it's the default choice in DNSSEC implementations' key-generation procedures — most zones support just that option. Unfortunately, 1024-bit RSA is already vulnerable, especially in light of powerful, nation-state adversaries (such as the Great Firewall of China or the recent NSA intrusions). Hence, for better security, the zones should also adopt more secure algorithms (or larger keys).

Other algorithms — in particular, elliptic curve (EC)-based — are believed to be secure even when used with significantly shorter keys and signatures. In particular, 160-bit EC signatures are considered more secure than 1024-bit RSA. EC signatures were in fact standardized for DNSSEC (RFC 6605).

Due to a lack of cipher-suite negotiation, zones supporting EC signatures must also send the (larger) RSA signatures, and hence end up sending more bits to all the resolvers while providing added security to only those few that actually support EC-signature validation. This situation causes a vicious cycle: only a few resolvers validate EC signatures, but zones can't sign using only EC signatures, because this will likely be incompatible with most resolvers, essentially preventing the zones from adopting new algorithms and hence causing significant security exposure. Ultimately, performance degrades due to the use of less-efficient algorithms and the need, once security sufficiently deteriorates, to add support for new algorithms.

Using Cipher-Suite Negotiation in DNSSEC

We recently proposed two approaches for a cipher-suite negotiation for DNSSEC that would let name servers send responses containing only those keys and signatures the requesting resolver requires.[7,8] Both proposals address three main goals: security, efficiency, and interoperability.

The main security challenge is to prevent downgrade attacks in which the attacker tries to trick a client and server into using a weak cryptographic mechanism, even though they both support and prefer a stronger one. Such an attack has occurred, for example, against the cipher-suite negotiation mechanism in SSL version 2.[9]

The efficiency challenges include avoiding additional round trips in the DNS resolution process, which occurs over UDP and involves only a single request and response. In addition, we want to minimize computation overhead (having to validate additional signatures, for example) as well as storage and communication overhead (due to requirements for additional requests).

Finally, the main interoperability challenge is to support legacy intermediate proxy resolvers that haven't deployed the new mechanism. Proxies are common among recursive resolvers[10,11] as well as name servers.[12] When a proxy is used between a resolver and

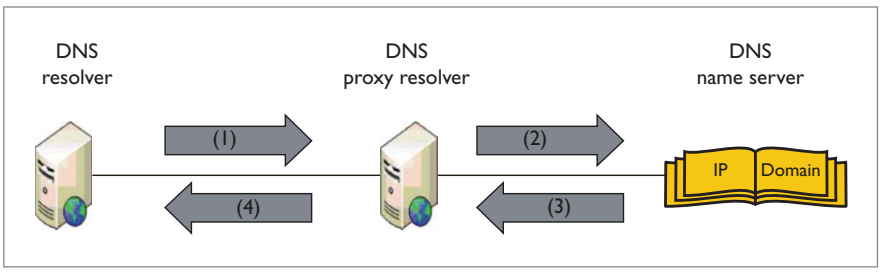


Figure 1. Cipher-suite negotiation in DNSSEC. A recursive resolver is connected to a name server over a caching proxy. All the queries traverse the proxy, and records that are not in the cache are relayed by the proxy to the name server.

a name server (see Figure 1), all the DNS requests are sent via the proxy, which either serves the requested records from its cache or forwards the requests to the name server. Caching reduces traffic volume to the name servers and improves the latency of client requests.

We can reasonably assume that, initially, most proxies won't support cipher-suite negotiation. This requirement that the intermediate proxies support cipher suite negotiation is the main difference between the cipher-suite negotiation design in our two proposals. In one, cipher-suite negotiation is hop-by-hop[7] — that is, if a legacy intermediary is present, there is simply no cipher-suite negotiation (all the signatures and keys are sent, as in current specifications). In contrast, end-to-end cipher-suite negotiation[8] occurs between the resolver and the name server, when the resolver requests the appropriate resource records, in a way that the intermediate proxies transparently support. However, this might result in additional overhead, mainly to the proxy, hence motivating the adoption of cipher-suite negotiation.

Hop-by-Hop Negotiation

Hop-by-hop cipher-suite negotiation[7] enables directly communicating DNS endpoints (see Figure 1) — such as the proxy resolver and name server, or resolver and name server (if they aren't connected via a proxy) — to agree on the optimal cryptographic material, which is used to protect the DNS transaction.

Our design uses the DNS packets themselves to signal the supported cryptographic options and thus doesn't add an additional round trip of communication to the DNS transaction. To signal the list of supported ciphers to the next hop, the resolvers can use a hop-by-hop transport-layer `EDNS0` mechanism (RFC 6891). The `EDNS0` record lets DNS resolvers and name servers support new mechanisms such as DNSSEC by allowing the exchange of arbitrary flags and options. These options aren't cached by the resolvers, and are used mainly for coordinating transport-layer parameters. Using `EDNS0` doesn't require changes to the DNS software or protocol because almost all resolvers and name servers already support it.

In hop-by-hop cipher-suite negotiation, the resolver adds a list of ciphers to the `EDNS0` record in the DNS request (step 1 in Figure 1). To signal support of cipher-suite negotiation, the entities add a flag, `CO (cipher-suite ok)`, in the `EDNS0` record. When a DNS request arrives, the proxy resolver checks whether the record is in the cache, and whether the (cached) signatures and keys correspond to the priorities the requesting resolver has signaled. If so, the proxy resolver responds from the cache without relaying the request to the name server. Otherwise, it issues a DNS request to the name server and copies the priorities and ciphers the resolver has signaled to the `EDNS0` record in the request (step 2). A name server that supports cipher-suite negotiation would extract the list of ciphers and identify the best one to use considering the list the resolver has signaled and its own priorities. In the response, the server returns only the cryptographic material that corresponds to that best cipher (step 3). The proxy caches the records in the response, along with the keys and signatures, and forwards it to the resolver (step 4).

To enable the resolvers to detect downgrade attacks in which the attacker removes strong ciphers from the list, the name server echoes back in the response a list of its own supported ciphers. This list is signed with the zone's key-signing key (KSK), which is in turn signed by the parent domain's signing key. This lets the resolver recompute the best cipher and confirm that the negotiation was successful.

The algorithms that the resolver and name server signal in the `EDNS0` field during the cipher-suite negotiation phase are encoded as numbers. Each algorithm in DNSSEC contains a code number, and these codes signal to validating resolvers which cryptographic algorithms were used to generate the digital signature and the hash digest. Next to each cipher number, the resolver also adds a priority number in ascending order (highest priority is 1).

To ensure a deterministic outcome from the cipher negotiation process, the cipher-suite client must assign different priorities to each cipher that it supports (and can't assign the same priority to two distinct ciphers, even if it considers them equally secure).

If neither the name server nor the DNS proxy support cipher-suite negotiation, then they ignore the relevant `EDNS0` options and return all the keys and signatures. Unfortunately, in this case, the host-by-host cipher-suite negotiation mechanism isn't effective, and the efficiency and security drawbacks we discussed earlier apply. These include vulnerability to DoS and poisoning attacks, as well as susceptibility to benign failures, as when firewalls block large or fragmented DNS

responses. Hence, hop-by-hop negotiation is most applicable when the resolver and the name server communicate directly, without intermediate proxies, or when the intermediate proxy also supports negotiation. For early adoption of negotiation by a resolver and name server connected via proxy, as in Figure 1, it might be better to deploy the end-to-end negotiation mechanism.

End-to-End Negotiation

To overcome the challenge of interoperability with legacy DNS proxies, the signaling should be transparent to the proxies. The crux of the solution is for the resolver to encode the preferred ciphers as a query subdomain. However, a careful design is needed to avoid a few potential pitfalls.

It might be easiest to understand the issues by first considering the following "naive" design, and the problems it would generate. In this design, the resolver prepends the ciphers it supports in order of preference. For instance, if the request is for `foo.bar`, and the resolver supports `cipher1`, `cipher2`, `cipher3`, then it will send a request for the query `cipher1.cipher2.cipher3._cs_.foo.bar`. The ciphers are ordered by the priorities, and `_cs_` is the delimiter, signaling to DNS entities that the request's originator supports cipher-suite negotiation.

Unfortunately, this naive design has two significant drawbacks. First, it isn't interoperable with name servers that don't support cipher-suite negotiation; the name servers would respond to such queries with nonexisting domain responses (requiring the resolver to resend the query without signaling support of cipher-suite negotiation). Second, this design would not allow the utilization of the caching offered by the intermediate proxies; in particular, resolvers supporting different priorities would cause the proxies to trigger distinct queries for each unique cipher combination. In addition, this design doesn't defend against degradation attacks.

Our design tackles these challenges.[13] To mitigate the increase in latency for resolvers and the volume of queries to the name servers when the DNS transaction involves legacy servers, we switch the signaling direction. Specifically, the name servers, rather than the resolvers, signal support of cipher-suite negotiation; this occurs in a backward-compatible manner using the DNSKEY records, which are already exchanged whenever DNSSEC is used.

The DNSKEY records are designed to contain the cryptographic public-verification keys that the zones support (encoded by the algorithm's number as assigned by IANA). The resolvers use these keys to verify DNSSEC signatures. The end-to-end cipher-suite negotiation mechanisms will use a new, dedicated algorithm number; instead of public keys, this number will contain the ordered list of algorithms the zone supports. Note that this doesn't cause a problem for legacy resolvers that don't recognize this new algorithm number because resolvers simply ignore any unknown number.

Suppose a resolver must send a query to a name server authoritative for `foo.bar`. Once the resolver, enhanced with a cipher-suite negotiation, receives the ciphers list in this special DNSKEY record, it learns that the name server is enhanced with the cipher-suite negotiation mechanism, and it also obtains the ordered list of ciphers that the zone supports. The resolver responds by sending to the name server (or proxy) a query for domain `foo.bar`, prepending to it as a subdomain the preferred cipher — for example, `cipher._cs_.foo.bar` (step 1 in Figure 1). The DNSKEY record is cached, and subsequent requests to that domain will incorporate the preferred cipher in the query. To name servers that don't signal cipher-suite negotiation support — that is, that don't send the DNSKEY record with the special algorithm number — the resolver won't encode the cipher in the subdomain. An intermediate proxy checks whether the record is cached, and if not, forwards the request to the name server (step 2). If the name server identifies the delimiter in the query, it responds with the cryptographic signatures that correspond to the option the resolver has signaled (step 3). The proxy caches the records and forwards the response to the resolver (step 4). Subsequent requests for the same record, with the same cipher preference, will be served from the cache.

The indirect signaling by the name servers to the resolvers implicitly supports interoperability with legacy resolvers and legacy name servers that don't support the cipher-suite negotiation mechanism, while not significantly increasing the amount of DNS requests to the legacy DNS servers.

The validity of the "list of algorithms" record is established by the resolver as it validates the signature over the entire resource record set — namely, using the zone's (already validated) KSK stored in a DNSKEY record, or the parent zone's DNSKEY record. This ensures that the resolver establishes a chain of trust all the way from the root zone to the target domain, and prevents downgrade attacks.

Cipher-suite negotiation in DNSSEC can solve critical problems pertaining to interoperability with the Internet infrastructure and can prevent attacks as well as a large fraction of the benign failures.

Acknowledgments

This research was supported by grant 1354/11 from the Israeli Science Foundation (ISF), by the Ministry of Science and Technology, Israel, and the German Federal Ministry of Education and Research (BMBF) within EC SPRIDE, and by the Hessian LOEWE excellence initiative within CASED.

References

1. "Domain Name System Security (DNSSEC) Algorithm Numbers," Internet Assigned Numbers Authority, Mar. 2014; www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml.

Standards

2. A. Herzberg and H. Shulman, "Retrofitting Security into Network Protocols: The Case of DNSSEC," *IEEE Internet Computing*, vol. 18, no. 1, 2014, pp. 66–71.
3. A. Herzberg, "Folklore, Practice, and Theory of Robust Combiners," *J. Computer Security*, vol. 17, no. 2, 2009, pp. 159–189.
4. A. Herzberg and H. Shulman, "Fragmentation Considered Poisonous: Or One-Domain-to-Rule-Them-All.org," *Proc. 2013 IEEE Conf. Comm. and Network Security*, 2013, pp. 224–232.
5. H. Shulman and M. Waidner, "Fragmentation Considered Leaking: Port Inference for DNS Poisoning," *Applied Cryptography and Network Security*, LNCS 8479, 2014, pp. 531–548.
6. C. Rossow, "Amplification Hell: Revisiting Network Protocols for DDoS Abuse," *Proc. Network and Distributed System Security Symp.*, 2014.
7. A. Herzberg, H. Shulman, and B. Crispo, "Less is More: Cipher-Suite Negotiation for DNSSEC," *Proc. Ann. Computer Security Applications Conf.* (ACSAC), 2014.
8. A. Herzberg and H. Shulman, "Negotiating DNSSEC Algorithms over Legacy Proxies," *Proc. 13th Int'l Conf. Cryptology and Network Security* (CANS), LNCS 8813, 2014, pp. 111–126.
9. T. Jager, K.G. Paterson, and J. Somorovsky, "One Bad Apple: Backwards Compatibility Attacks on State-of-the-Art Cryptography," *Proc. Network and Distributed System Security Symp.*, 2013.
10. A. Herzberg and H. Shulman, "Vulnerable Delegation of DNS Resolution," *Proc. 18th European Symp. Research in Computer Security*, 2013, pp. 219–236; http://dx.doi.org/10.1007/978-3-642-40203-6_13.
11. K. Schomp et al., "On Measuring the Client-Side DNS Infrastructure," *Proc. 2013 ACM Conf. Internet Measurement*, 2013, pp. 77–90.
12. H. Shulman, "Pretty Bad Privacy: Pitfalls of DNS Encryption," *Proc. 13th Ann. ACM Workshop on Privacy in the Electronic Society*, 2014.
13. A. Herzberg and H. Shulman, "DNS Authentication as a Service: Preventing Amplification Attacks," *Proc. Ann. Computer Security Applications Conf.*, 2014.

Amir Herzberg is a tenured associate professor in the Department of Computer Science at Bar Ilan University, Israel, where he heads the Secure Communication and Computing ("Cyber") group. His research interests include network security, especially Internet protocols, applied cryptography, privacy, anonymity, covert communication, usable security, and social-engineering attacks. Herzberg received a DSc in computer science from the Technion, Israel. He's a member of IEEE. Contact him at amir.herzberg@gmail.com.

Haya Shulman is a Claude Shannon researcher at Technische Universität Darmstadt. Her research interests are in network and cybersecurity, in particular, routing and DNS security, secure channel protocols, and cloud security. Shulman has a Ph.D in computer science from Bar Ilan University, Israel. Contact her at haya.shulman@gmail.com.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.

Beyond Wires

Editor: Yih-Farn Robin Chen • chen@research.att.com

Mobile Videos

Where Are We Headed?

Moo-Ryong Ra • *AT&T Labs Research*

Video is one of the richest forms of data that mobile devices produce. However, such devices, along with existing network infrastructure, can support only rudimentary functions: uploading, downloading, and a little bit of processing. Here, the author looks at the challenges of enabling more advanced processing capabilities and highlights future opportunities for video captured by mobile devices.

The advent of smart mobile devices with built-in sensors has produced an influx of rich contextual data and enabled many useful context-aware applications, including location-based services, motion-sensor-based games, and health applications. Among all the data mobile devices have produced, video is probably the richest. Combined with state-of-the-art computer vision and machine learning technology, this spatio-temporal visual information in a highly compressed form has tremendous potential. Notably, mobile video traffic is rapidly growing. Cisco reports that it will consume nearly 11 exabytes per month among the 15.9 exabytes per month passing through mobile networks, and predicts that it will generate 69 percent of total mobile traffic between 2013 and 2018.[1]

Along with market growth, mobile devices are also getting powerful. They now have multicore CPUs and GPUs, at least tens of gigabytes of memory, and various sensors, including high-resolution cameras. Despite these powerful capabilities, the way we use mobile phones to consume video is still rudimentary. Basically, we can take video with resolutions up to 4K, upload it to an Internet-connected server for backup or share it with family and friends via services such as YouTube, and download or stream it from cloud-based services such as Hulu and Netflix. In current practice, if you want to perform any sort of nontrivial processing on video, you must rely on more powerful computing devices, such as desktops or servers in the office or the cloud.

Clearly, as technology evolves, we will be able to do better. For instance, in the near term, we will likely be able to use an image filter on video as we've done successfully with Instagram for photos (see http://instagram.com). Also, researchers are envisioning advanced functions, including real-time composition of multiple video streams from different sources or application of complex computer-vision-based recognition algorithms on real-time videos.

Here, I focus on the following two questions: First, 5 to 10 years from now, what kinds of capabilities will we have for videos generated by mobile devices? And to support those demands, what challenges and opportunities will infrastructure providers face?

Current Status and Future Demands

As of 2014, smart mobile devices, such as smartphones and tablets, commonly have cameras with resolutions of 8–13 megapixels and can record 4K video at 30 frames per second (fps) (see http://en.wikipedia.org/wiki/List_of_4K_video_recording_devices). Videos are typically stored in a compressed form, and today's mobile devices use only a handful of video coding standards, such as H.264. These standards have been evolving mainly to improve coding efficiency and thus reduce video data's footprint.[2] Hence, mobile devices can produce high-quality visual data while consuming small amounts of storage space.

With these technical advances, people can produce high-quality videos of their everyday lives with a couple of clicks. The only remaining

Published by the IEEE Computer Society
1089-7801/15/$31.00 © 2015 IEEE

challenge is probably reducing energy consumption, given that video recording tends to put mobile devices in the highest power state. After producing and storing videos locally, people can upload and share them through services such as YouTube. (Sharing generates data security and privacy issues, which I discuss more later.) Others can consume these shared videos, and some commercial video content, on their own mobile devices. Cloud-based video streaming services are representative examples. Companies such as Netflix, Hulu, and U-verse provide on-demand video streaming services for mobile devices.

In the past decade, we've made tremendous progress in terms of mobile video quality; however, what applications can do using the videos is highly dependent on built-in codecs available on the corresponding hardware. In terms of processing and manipulating video data, going beyond what the device already provides is difficult for several reasons. First, the state-of-the-art video codecs must be highly optimized for a given hardware to fulfill requirements — for example, 30 fps with 1280×1080 resolution. Thus, open source software for processing videos (see http://sourceforge.net/projects/ffmpeg4android/) might still be too slow to use in practice. Second, when it comes to processing videos, mobile platforms, including iOS and Android, have limited tools and APIs exposed to the application layer. Third, mobile devices are still resource-constrained, especially when we consider battery life.

To overcome these obstacles and enable novel functionalities, support from infrastructure is essential. For instance, offloading computation to the cloud infrastructure could enable advanced processing capabilities, among other benefits. Also, decomposing the aforementioned complex video compression techniques and jointly optimizing them for higher-level tasks might be useful in some cases, which I discuss more later.

Advanced Processing

What kinds of processing capabilities will we need for mobile video in 5 to 10 years? To facilitate this discussion, let's look at some example capabilities. One thing worth mentioning is that, at least in the near term, these capabilities will require close interaction with cloud infrastructure before they can be realized locally on mobile devices.

The first capability that we can easily predict is near real-time processing of demanding computation on video data, such as transcoding, stabilization, applying filters to impose visual effects, and so on. As mentioned, because video-compression technology evolved primarily to enhance the compression ratio — making a pixel-level manipulation isn't in the equation, for example — it's inherently hard for us to process video in real time if we use the out-of-box codec. With proper support from infrastructure, we might be able to get the results of such processing in near real time.

We've made tremendous progress in terms of mobile video quality; however, what applications can do using the videos is dependent on built-in codecs.

The second capability is real-time sharing and composition of multiple video streams from the corpus of mobile devices. This is particularly useful for sharing real-life events in a timely manner. Suppose that a big sports event is going on, such as the Super Bowl or World Series. Many people among the stadium crowds will have mobile devices, and the number of those devices is likely to be huge. Each person will have a unique viewpoint of the event, and it would be interesting if, after they share those views, others could easily composite them in a personalized way. For instance, in a baseball stadium, Bob's seat is behind first base. He might want to see views from the outfield and third base using his smartphone. One crucial feature to have will be time synchronization among multiple video streams.

The third example is applying computer-vision-based machine learning algorithms to real-time video streams. These algorithms extract users' contexual information automatically — for instance, recognizing who (person) or what (object) is in the scene — or receive user input without using external hardware, as with hand-gesture recognition. Unfortunately, applying these algorithms to real-time videos is a computationally demanding task. One way to enable such applications is to split the application into two pieces and offload the computationally demanding part to the cloud.[3]

Although these three examples are interesting, many other capabilities could open the tremendous potential of mobile videos.

Sharing Videos and Data Privacy

It has become common to share videos on social networking sites. YouTube provides a platform for sharing video content and is used in different ways: content providers and some individuals use it to conveniently reach a large audience, whereas others use it as a rendezvous point for sharing private videos. Unfortunately,

a big concern is that, once we share the content, we immediately lose control over it. My previous work, P3, addresses this concern for photos by dividing the image into two parts (private and public) and encrypting the private part (which has the most visual information) so that cloud service providers can't extract meaningful information but still maintain their processing capability.[4] For videos, we must cope with the complexity of video coding standards. Compared to image compression standards, such as JPEG, tweaking frame-level variables, as P3 does for photos, can severely affect interframe compression performance. Although it isn't clear whether YouTube automatically mines videos to recognize faces or other subjects, it certainly possesses the data to do so. Thus, it's important to develop a technology similar to P3 for video, to ensure that video service providers (VSPs) can't mine private videos, and thereby protect users' privacy.

Requirements and Opportunities for the Next Generation

To enable advanced processing capabilities and address the privacy concern, we must certainly develop new technologies and infrastructure for mobile video. To achieve this goal, innovations are necessary to both software systems and video coding techniques. Here, I present some beneficial examples.

Requirements for Offloading Infrastructure

Despite tremendous progress in the past decade, mobile devices are still resource-limited. One useful technique to overcome this limitation is offloading. At least in some cases, this technique can significantly improve the performance — that is, the latency (ms) and throughput (fps) — of demanding applications when it comes to video data.[3] Other infrastructure-side technical concerns include timely and quality-of-service provisioning of compute and network resources, careful network failure handling, and the scalability of such services. At the same time, because we're dealing with visual information, any offloading infrastructure must handle data security and privacy issues. For instance, if the whole or partial visual information of an input video or its associated features exposes personal information from the cloud, users will have serious privacy concerns. Because these new workloads (with associated capabilities) have a high data rate and require a crisp response time, existing techniques using differential privacy and selective encryption might not be sufficient. Moreover, if we consider the compression pipeline, the issue becomes more complicated. For instance, both H.264 and Google's VP8 standard, which are increasingly dominant, use advanced adaptive arithmetic coding techniques; this makes bitstream manipulation[4] much more challenging. For example, simply removing bits from the bitstream would lead to an undecodable public video, and thus wouldn't be processable in the cloud.

If performance and data privacy concerns are resolved, users will only benefit from offloading infrastructure.

Finally, providing a relevant incentive mechanism for infrastructure providers is also important. From a user perspective, if performance and data privacy concerns are resolved, users will only benefit from offloading infrastructure. Fortunately, for most cloud service providers, this fact might implicitly motivate them to adapt the technology to increase their user base.

Cross-Layer Optimization for Advanced Processing

As mentioned, current video coding standards are composed of a series of computational steps to optimize coding efficiency. You can find detailed information elsewhere.[5,6] However, to better support future demands, it might be better to decompose such standards and take finer-grained computational building blocks into consideration for system-level optimization processes. Let's look at one example that jointly optimizes coding efficiency using other available video. Mobile videos are often stored in remote storage spaces outside the mobile device, such as a backup drive at home or distributed cloud storage such as Dropbox, Box, or AT&T Locker. When it comes to considering storage space, videos (and photos also) are especially tricky to store efficiently because the common techniques to optimize storage space, such as deduplication and compression, won't work well. This is mainly because the redundancy imposed on raw video data has already been eliminated by its carefully designed encoding process. However, if we can find additional redundancy across multiple videos at an earlier phase in the encoding process, it might be a different story. For instance, applying the deduplication technique across the corpus of video files before the entropy coding stage — for example, adaptive arithmetic coding — might let us improve coding efficiency further, thereby reducing storage capacity requirements. In particular, benefits would be maximized when videos are

captured under similar circumstances—for instance, from the same building or the same set of people. Arguably, these cases are more common for the videos mobile devices generate.

Today's mobile devices already have advanced capabilities, and future devices could have even more that are focused on processing and sharing. Before we can realize these capabilities, however, we must further evaluate the idea of offloading infrastructure to provide performance guarantees and preserve privacy. The potential for cross-layer optimization must also be verified with concrete experimental evidence. Despite these challenges, I hope this article can stimulate the community and result in active research efforts in this field.

References

1. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2013–2018*, Cisco, Feb. 2014; www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html.
2. J. Ohm et al., "Comparison of the Coding Efficiency of Video Coding Standards Including High Efficiency Video Coding (hevc)," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 22, no. 12, 2012, pp. 1669–1684.
3. M.-R. Ra et al., "Odessa: Enabling Interactive Perception Applications on Mobile Devices," *Proc. 9th Int'l Conf. Mobile Systems, Applications, and Services*, 2011, pp. 43–56.
4. M.-R. Ra, R. Govindan, and A. Ortega, "P3: Toward Privacy-Preserving Photo Sharing," *Proc. 10th Usenix Conf. Networked Systems Design and Implementation*, 2013, pp. 515–528.
5. I.E. Richardson, *The H.264 Advanced Video Compression Standard*, John Wiley & Sons, 2011.
6. T. Wiegand et al., "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, 2003, pp. 560–576.

Moo-Ryong Ra is a Senior Inventive Scientist in the Mobile and Pervasive Systems Research group at AT&T Labs Research. He is broadly interested in solving challenging problems related to mobile and cloud systems. Ra received a PhD in computer science from the University of Southern California. Contact him at mra@research.att.com.

Practical Security

Why Won't Johnny Encrypt?

Hilarie Orman • *Purple Streak*

Some 30 years have passed since I first used encrypted email, and I was reflecting on the near-absence of its adoption by ordinary individuals, even those who are security conscious. Given the knowledge that we now have about extensive government surveillance over Internet traffic, we should conclude that end-to-end encryption is the only technology that can assure email privacy. This has led me to ponder some questions, such as, "Are there readily available tools for using cryptography in email? Should we be encouraging our correspondents to turn on the 'encrypt' flag?"

Fifteen years ago, Doug Tygar and Alma Whitten wrote an interesting paper, "Why Johnny Can't Encrypt."[1] In it, they discussed the usability of encryption tools. They found that the predominant tool at the time, Pretty Good Privacy (PGP), failed to be "usable," in part because users didn't understand how cryptographic keys worked, where to get them, and how to use them. Are tools today more usable, and are there new usability issues? How do smartphones, tablets, and "bring your own device" policies change the picture?

The Internet is a much more complicated place than it was at the dawn of encrypted email, and despite its ubiquitous nature, perhaps email is passè. Should we worry more about new messaging apps, such as Facebook, Twitter, Skype, and Snapchat? What about Bitcoin?

I can't answer all these questions in this small space, but this article intends to show that end-to-end encryption is somewhat understandable, accessible with some effort, possibly extensible to social media, and in the end, perhaps saved by the global reach of social networks.

Why Do We Need End-to-End Encryption?

Although email encryption standards were developed 30 years ago, other than for a few experimental purposes, only two people have ever asked to communicate with me via encrypted email. On the other hand, I have observed that some people routinely use digital signatures to ensure their email's integrity and authenticity. I was curious about how much email is protected, and based on a simple analysis of non-spam messages that I've received in the past several months, I estimate that fewer than one person in a thousand signs messages. Because digital signatures are part of encrypted email, I assume that this represents the fraction of people who have configured their email software to use cryptography. This means that if I were to start encrypting email, I would have an uphill battle to convince my correspondents to start doing the same, because it seems that few of them have encryption enabled or available.

In talking about email privacy, I'm focusing on measures that defend email against any kind of snooping. The strongest guarantee of privacy for email is *end-to-end* encryption. This means that no one but the sender and the receiver can decrypt the communication. The email provider can't decrypt the messages because only the two communicating parties have the necessary keys. No eavesdropper can understand the messages, and no one with access to files on the email server can read them.

Without end-to-end encryption, eavesdroppers have any number of ways to read email. They can read data from communication lines as the email goes from a user to an email server or from one server to another. They can copy data from email servers or the repositories that hold email in "folders." Eavesdroppers can be unethical employees at datacenters, criminals who have gained illegal access to servers, or government agents with specific or blanket search warrants. Even if service providers try to protect communications using point-to-point encryption (for example, STARTTLS for the Simple

Published by the IEEE Computer Society 1089-7801/15/$31.00 © 2015 IEEE

Mail Transfer Protocol [SMTP]), errors in configuration might occur, or the facility might be monitored in response to a search warrant.

Thus, end-to-end encryption is the solution to surveillance avoidance, but with great privacy comes great responsibility. The cryptographic operations must be done on the endpoint device, not on a server. This means that email software must have cryptographic support of some kind — an auxiliary app, a crypto library in the mail client, or Java running in a browser. Furthermore, losing a cryptographic key can mean losing access to the email forever. Users with several devices, particularly mobile ones, must keep their keys on each device, but this increases the likelihood of keys being accidentally disclosed.

This makes for a security conundrum, but it doesn't fully explain why almost no one uses email encryption today. With the exception of the US military and some enterprises that enforce security through their organizational gateways and email clients, there appears to be little interest in email privacy. All of which leads me to ask, are the tools there, and are they usable?

Email Privacy: What You Need to Know

First, let's step back and review what's needed for end-to-end email privacy. The sender and the receiver want to know that the message is actually from the sender, it is intended for the receiver, no one but the sender could have sent the message, and no one but the sender and the receiver can decrypt it. These goals are usually accomplished through a combination of public-key algorithms and symmetric encryption.

Six pieces of information are prerequisites for secure email communication: the sender's identity, the intended recipient's identity, a cryptographic binding between the message and the sender, a cryptographic binding between the encrypted message and the intended recipient, assurance that the message hasn't been altered during transmission, and a transformation of the message from "plaintext" to "ciphertext" using a function that's extremely difficult to reverse without a message key that only the sender and receiver know.

In most cryptographic systems today, the identities are embodied in public/private key pairs. Only the identity's owner knows his or her private key, but the public key is, well, public. The sender chooses a unique message key, encrypts the message, applies a hash function to get a short representation of the message, encrypts the message key using the intended recipient's public key, and uses his or her own private key to create a digital signature over all the data. This data (signature, encrypted message key, encrypted data, and hash value) is encoded in a data structure that's compatible with SMTP, and the whole thing is then delivered to the recipient as an email message. The recipient can then check that the hash value matches the received data, verify the signature against the signed data, decrypt the message key, and decrypt the data.

If the sender only wants the recipient to check the message's authenticity (that is, that it was really sent by the entity appearing in the "From:" line) without keeping the message contents secret, then the sender can sign the message's hash without establishing a message key or encrypting. The "signature only" messages are simpler than encrypted messages because the sender doesn't need to know the recipient's public key. This is a useful way to let people know that you have a public key and that email associated with your public key is authentically "you."

Why You Need a Certificate

Email systems generally represent public keys as part of a complicated data structure called a *certificate*. A certificate is a way of saying something like "Entity Alice asserts that the string of bits XYZ is the public key of the entity known as Bob with email address bob@example.com, and Alice has attached her digital signature S to this statement." If an email user thinks that Alice is trustworthy, then the certificate binding bob@example.com to XYZ can be saved in the key-management store on the recipient's machine. If Carol has Bob's certificate, she can use his public key to send secure email to him.

End-to-end encryption is the solution to surveillance avoidance, but with great privacy comes great responsibility.

If Bob wants to use his own reputation to establish a key, then he can create a self-signed certificate that says "Entity Bob, known as bob@example.com, asserts that the string of bits XYZ is the public key of the entity known as Bob, and Bob has used XYZ to create his signature S on this statement." This demonstrates that Bob has the private key that matches XYZ. For ordinary users, this level of trust is usually sufficient. After all, if you've been in the habit of sending email to "Bob," then you'll probably be happy enough to trust a certificate that he sends you.

Many companies have email systems that take care of key issuance and management, and all the complexity might be hidden from users through a gateway machine.

The company might have a database of keys, and the user email clients might look these keys up and use them without having to bother the user. These solutions are especially important for organizations such as the US military; for example, I regularly get press releases from a local National Guard unit, and their messages always have a digital signature.

Getting Started With Secure Email

Ordinary users, especially those in a "bring your own device" world, need to have keys that work with multiple devices and a variety of email clients, and they need to carry their correspondents' public keys in a "Contacts" list that's easily portable. Let's look at the starting point of a grassroots "let's encrypt our email" effort.

"Why Johnny Can't Encrypt" lays out the minimal requirements for a usable encryption system.[1] Such a system should help users understand and accomplish the following:

- encrypt email, and decrypt email received from other people;
- sign email, and verify signatures on email from other people;
- generate a key pair;
- publish the public key;
- acquire other people's public keys;
- avoid such dangerous errors as accidentally failing to encrypt, trusting the wrong public keys, failing to back up their private keys, and forgetting their pass phrases; and
- succeed at all of the above within a few hours of reasonably motivated effort.

Do modern email clients have a "usable" interface to end-to-end encryption? In my observation, most of them support the first three items fairly well, though many users will have some trouble with the "understanding" requirement, because public-key technology isn't an intuitive concept; it takes some experience to become comfortable with the components. Neophytes always have trouble with the next two items, because the notions of public and private keys, certificates, and email attachments can become mind-numbing. The caveats implicit in the sixth item will always haunt us; there are just too many ways to lose information. Despite all this, "a few hours" will probably suffice to get almost anyone started on secure email.

Many users will have some trouble with the 'understanding' requirement, because public-key technology isn't an intuitive concept.

Configuration, Import, Export

Linux, Apple OSX and iOS, and Microsoft all support signed and encrypted email through their widely used email clients. Google and Yahoo have announced that support for their email clients is on the way. So, in theory, nothing stands between a user and end-to-end privacy. But users probably won't find a "Secure email" tab that does all the configuration automatically. It's best to start by acquiring the cryptographic keys that establish an email *identity*. An identity is associated with an email account, and you will need separate cryptographic keys for each of your email accounts.

The first obstacle that encrypted email neophytes face is getting their email name connected to a pair of related cryptographic keys (the public and private keys discussed previously). These are sometimes called the certificate, but this is a terminology conflation. The key pairs consist of two large numbers that are generated by cryptographic software libraries and that must be saved in two separate files. The public key is encoded into a certificate, whereas the private key is usually protected using a passphrase that the user makes up. The passphrase is then turned into an encryption key, which encrypts the private key, and the result is stored in a file. It's essential to make note of the passphrase!

Apple's OSX operating system has a key manager with a reasonable GUI, and it can generate keys for email. If you're already familiar with the key manager, you might naturally start there, but if you start with the email client, you'll spend some time fumbling around without getting anywhere — so start with the key manager (hint: there's an icon with a keychain graphic). Once you've generated the keys, you'll need to "export" them into a format that an email client can read. The usual exchange format is called PKCS #12 (see www.emc.com/emc-plus/rsa-labs/standards-initiatives/public-key-cryptography). The associated file extension is *p12*. Because the file contains the all-important secret key, it's encrypted with a key derived from the username and that user-supplied passphrase that you should not forget.

Once I had all that out of the way, I was able to import the key file into an email client on an Apple mobile device. Getting the client to actually sign messages wasn't too hard, but I couldn't have done it without a little help from the Web. The option is buried deep in "Settings" for the email app, under "Advanced" (of course), and at the bottom of a menu list that's covered by the virtual keyboard. With this done, I could

immediately send digitally signed messages. Moreover, other email clients on Apple devices detected the signature and marked the message as verified. Yay!

Using Microsoft Outlook started out as a more intuitive process, but it went awry. Having found email signing under "Tools," then "Accounts," and then the "Advanced" (of course) settings for one account, the application noticed that I didn't have a public key, and it offered links to several certificate providers, including one that required no payment. All went well with the Web registration, and shortly after that I received an email with a link for my certificate and private-key package. Clicking on the link didn't download a p12 file as I had expected, but it did cause the browser to consume the data into its key storage. When I tried to export the private key from there, I ran into access permission problems that seemed unsolvable. I gave up on that certificate and instead attached the p12 file from an Apple device to an email message to myself, and read it with Outlook, which happily consumed that data into its key storage (note that it needed to know the passphrase for the private key).

Linux supports two independent software bases that are very helpful for generating keys: the GNU Privacy Guard (GPG; www.gnupg.org) and OpenSSL (www.openssl.org), each with extensive cryptographic capabilities. These use a command-line interface, and the incantations are baroque (are classical musicians attracted to Linux?). Fortunately, on Ubuntu Linux, the Seahorse key manager does most of the work using GPG through a simple GUI.

Although Seahorse can generate keys for email, if you want to use them on other devices, you need to know that GPG prefers to use its own for keys. The *armored ascii* format is the default, using the file extension *asc*. Armored ascii could be unpalatable to systems that default to the public-key cryptography standards. To export the key to that format, a user might need to invoke `gnugpg` through the command line and specify the option `-export-format pkcs12`.

Users who want detailed control over their key generation can use either GPG or OpenSSL from the command line. The first step in OpenSSL is to create a *certificate authority* with a public and private key. This step is followed by creating a new key pair for email and signing it using the certificate authority. Finally, you can export the newly created email key pair to the p12 format. Microsoft Outlook seemed to consume that file happily.

Many Accounts, Many Keys

When I first dabbled with email encryption, I had only one computer and one email account to configure. Today I have five computers that I use regularly, each one with a different email app running on a different operating system. I've also got a handful of different email accounts. It's hard enough to keep my contact lists in sync, and the idea of keeping my keys current is daunting. However, to have the flexibility of using whatever device is most convenient while still assuring end-to-end privacy, the keys associated with each account must reside in each computer. That's why the information about importing and exporting the cryptographic identities is so important.

As mentioned earlier, end-to-end security requires secret keys on each device, but having more copies of secret keys increases the risk of their exposure. Recently, I was talking to experts at a computer forensics company, and they told me that in 99 percent of their investigations, passphrases or passwords are found lodged somewhere in the device memory. Should you trust your mobile phone or tablet with the keys for your email identity? The risk might be acceptable for your "social" email account, but what about the one you use for professional work? Or, you might trust a device that you use for work but not games. There's no simple answer; maybe someday we will have verified, trustworthy operating systems on trustworthy hardware, but until then, caveat keyholder.

Maybe someday we will have verified, trustworthy operating systems on trustworthy hardware, but until then, caveat keyholder.

However, once you've decided on these measures and have loaded your p12 or asc files with your digital identities into your endpoint email clients, then you're ready to start sending signed email. If you can convince your correspondents to do the same, then your email to one another will be signed, and as a side effect, the certificates will be email attachments. The PKCS #7 format for signatures has the extension *p7s* (usually smime.p7s), and this contains the certificate chain for your identity, your public key, and your digital signature of the whole message's hash. Most email clients recognize these attachments and automatically incorporate them into your key manager. When you compose a new email message to a user whose public-key certificate is in your key storage, the email client will recognize it and can encrypt the email to that recipient.

Of course, there are some gotchas. You'll need to read the email

on each of your devices to get your correspondents' keys into the key storage, or else you will need additional software to add them to synchronized contact lists. After a year or so, the certificates will expire, and you'll need to generate new key pairs and send them to all your correspondents, and they'll need to do the same. Anytime you get a new email account, you'll need to get new keys and let all your correspondents see your new certificate, and they'll be doing the same thing. This is a real usability killer, one of the major reasons that people give up on strong authentication.

Long ago, there was an expectation that global directories would simplify the problem of finding keys. Some people envisioned a hierarchical structure based on organizational units of government and industry, and other people thought that a collection of easy-to-update servers and a "web of trust" would predominate. But neither one has caught on for the majority of Internet users today. Consequently, key management is relegated to enterprise solutions or to a few dogged, security-obsessed individuals.

Social Media to the Rescue?

Generally, social media doesn't have end-to-end encryption, though any text-based messaging system could, in theory, be "crypto-ized." Some third-party applications offer help with this by providing, for example, their own database for storing and retrieving encrypted Twitter messages and handling the keys and crypto operations automatically. We might need to wait a few decades for this to become an integral part of messaging apps. But there could be a place for social media in the salvation of key management. Instead of global servers for publishing public keys, why not a social network?

The not-yet-in-alpha Keybase system (https://keybase.io/) proposes maintaining a website for bootstrapping key advertisements from registered users, but their software will do much more than hold a directory. They will also provide open source software that can run independently of the website, using it only for bootstrapping and occasional checks for new key types.

The exciting thing about Keybase is that it recognizes that public keys are used for diverse applications on the Internet. You can announce a public key on Twitter, for example. If you were to register as a Keybase user, and then registered your Twitter handle with them, Keybase would look for your announcement of your public key in your Twitter feed and would verify the key by checking the signature on it.

Your friends, should they also be Keybase users, would contact the Keybase server and look up your Keybase entries. Seeing that you had a Twitter handle, their Keybase clients would learn that you had posted a public key. The clients don't act as zombies; they independently fetch the posting with the key and validate it. Each client can sign these keys as "trusted" and save them on the Keybase server. Using a new device? You can pick up your trusted keys from the Keybase server and start watching for updates. It sounds as though you could manage trusted keys yourself, using other tools, but you'd want to keep announcing your new keys on the Keybase server and watching for your friends to start using new apps.

Because Keybase is built on GPG, it has all the crypto utility functions for key import and export and cipher suites available to its client software. Many operating systems, particularly Linux-based systems, have GPG installed by default; your mileage may vary with others.

Will this work for Bitcoin accounts? Keybase says it will, and they include this interesting statement on their webpage: "We're now embedding signed announcements in the Bitcoin blockchain." If everyone starts doing this, I think we'll need to increase the world's energy production dramatically. We want nuclear fusion now!

End-to-end encryption for email is or will soon be available for almost all commonly used computing devices. Users who are interested in it can start with signed email and move gradually to encrypted email if they can convince their friends and colleagues to join them in the quest for privacy. This is an era of social media, in which privacy takes a backseat to openness, but perhaps social media can help spread the word about the importance and accessibility of email communication privacy. Privacy — can it be the next viral movement?

Reference

1. A. Whitten and J.D. Tygar, "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0," *Proc. 8th Conf. Usenix Security Symp.*, 1999, pp. 14–14; http://dl.acm.org/citation.cfm?id=1251421.1251435.

Hilarie Orman is a security consultant and president of Purple Streak. Her research interests include applied cryptography, secure operating systems, malware identification, security through semantic computing, and personal data mining. Orman has a BS in mathematics from the Massachusetts Institute of Technology. She's a former chair of the IEEE Computer Society's Technical Committee on Security and Privacy. Contact her at hilarie@purplestreak.com.

cont. from p. 96

shown themselves to be unexpectedly preserved for similarly varying time periods.

What could prove possible as we grow into this digital era is that we might be able to regularize our efforts to preserve digital content in the form of digital objects held in various archives or repositories, such as those contemplated in the Digital Object Architecture developed at the Corporation for National Research Initiatives (CNRI; www.cnri.reston.va.us/papers/OverviewDigitalObjectArchitecture.pdf).

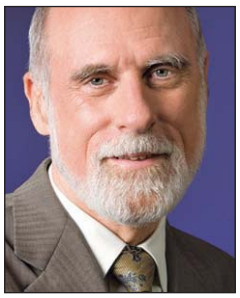
Padcasting and anycasting have other implications. For one thing, the traditional content consumers have become producers as well, raising the question of whether we can distinguish between these two roles anymore. Despite the cloud's attractions, we can imagine — in a world of symmetric, high-speed Internet access — that users will become both the consumers and the suppliers of content in varying amounts. Business and regulatory models that treat consumers and suppliers as distinct might find it more difficult to maintain the distinction, which could affect how policy develops for managing and protecting intellectual property.

The 21st Century is getting more interesting by the millisecond!

Vinton G. Cerf is vice president and chief Internet evangelist at Google, and past president of ACM. He's widely known as one of the "fathers of the Internet." He's a fellow of IEEE and ACM. Contact him at vint@google.com.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.

Backspace



Padcasting

Vinton G. Cerf • *Google*

Not long ago, I was chatting with a colleague about the process of media convergence on the Internet. We were talking about "podcasting," and I misheard him as saying "padcasting." When he corrected my misunderstanding, I continued to think about this neologism. As data rates on the Internet increase in both wired and wireless operation modes, and as we move toward an ill-defined but surely real "Internet of Things," it seems apparent that distinctions among various kinds of display devices will continue to erode. Whether we're speaking of a mobile, pad, laptop, television screen, video projector, Google Glass, Oculus Rift, or perhaps even a wristwatch, we will likely see all media becoming available on or through all devices. Where there are mismatches (such as no speaker), we might see automatic provisioning of captions, when available.

In some sense, we could be approaching a sort of "anycasting" moment (not to be confused with IP anycasting), when anything can serve as a receptor for any and all media. Some services already let users move from one device to another while seamlessly receiving the same streaming video or audio. Many of these devices let you read, compose, and reply to email or other social messaging media. Videoconferencing is no longer confined to specially equipped rooms. You can do it from mobiles (when there is adequate capacity), laptops, pads, desktops, and so on. We are also starting to see some devices with cameras interpreting gestures. Holding up a hand, palm out, can be interpreted as wanting to interrupt or ask a question, and the device can flash a symbolic hand against, say, a red background to attract attention.

The presence of cameras and microphones in addition to displays, mice, touch pads, touch displays, styli (the plural of stylus for all you pedants out there), and keyboards on so many devices makes it possible to anycast from anywhere to anyone. The casual use of increasingly rich media for everyday communication seems like a sea change in how we think about our interactions.

There are side effects of this proliferation of opportunities to express ourselves and stay connected. We see the same or similar expressions showing up as tweets, to say nothing of "re-tweeting." Waves of comments, images, and videos show up on Facebook and Google+. It is as if our lives are becoming the subject of commentary and debate, not unlike the formal and informal glosses on the parchment texts of old. The potential permanence and public nature of much of the record has opened up new areas for research and analysis, as we find in studies of the "twitterstream." At the same time, there is no guarantee that anything in the Web is at all permanent, leading to my usual "bit rot" rant about the impermanence of digital information — degradation of the medium, loss of reading devices, loss of correct information about format, loss of metadata needed to correctly interpret the data, loss of websites and inability to resolve URLs, and so on. The Internet Archive (see www.internetarchive.org) represents one among several efforts to preserve digital information for the future. Another is found at Carnegie Mellon University in the form of Project Olive (olivearchive.org), which aims to preserve various kinds of executable code. The irony of these efforts and effects is that the permanent could become evanescent and the evanescent, permanent (or at least preserved) in a sporadic sort of way. In some sense, this isn't new — all older media have shown themselves to be evanescent over varying time periods and have also

cont. on p. 95

Published by the IEEE Computer Society 1089-7801/15/$31.00 © 2015 IEEE

2015–2016 Editorial Calendar

Building Internet of Things Software (Mar/Apr 2015)

As we equip people, places, and commodities with Internet-connected embedded devices that can sense information about the environment and subsequently take action, we will create the Internet of Things. The IoT will improve society and quality of life, but making this vision a reality requires interdisciplinary efforts in a range of scientific domains. Specifically, enabling the design, implementation, validation, and real-world use of IoT software requires that we embrace diverse contributions in coherent and practical development frameworks, possibly based on current and future standards.

Physical-Cyber-Social Computing (May/June 2015)

Physical-cyber-social (PCS) computing involves a holistic treatment of data, information, and knowledge from the physical, cyber, and social worlds to integrate, understand, correlate, and provide contextually relevant abstractions to humans and the applications that serve them. PCS computing extends current progress in cyber-physical, socio-technical, and cyber- social systems. This emerging topic seeks to provide powerful ways to exploit data that are available through various IoT, citizen and social sensing, Web, and open data sources that are seeing explosive growth.

Continuous Digital Health (July/Aug 2015)

The way we deal with our health is undergoing a major transformation, not only because mobile Internet technology has given us continuous access to personal health information, but also because breaking the trend of ever-growing healthcare costs is increasingly necessary. Estimates indicate that more than 70 percent of the world population will have a smart phone by 2017. Connectivity, interoperability, sensing, and instant feedback through smartphones all provide new opportunities for gaining insights into our health behavior.

Small Wearable Internet (Sept/Oct 2015)

Conservative estimates show that the wearable electronics market will represent more than US$2 billion in revenue worldwide by 2018. This technology's high potential is such that both big, established players and small startups are actively involved in developing new devices, applications, and protocols for the wearable electronics market. Likewise, several other stakeholders, including users, mobile application and hardware developers, network operators, content providers, and regulatory authorities, are interested in better understanding and leveraging the capabilities of wearable devices.

Internet of You (Nov/Dec 2015)

Where our ancestors left behind few records, we are creating and preserving increasingly complete digital traces and models of almost every aspect of our lives. This special issue aims to explore technologies and issues from small user-centric models of individuals to real-time analytics on huge aggregations of user data. Some are aspiring to let you record everything about yourself and convert it into a model that's queriable, conversant, and possibly even active in gaining new experiences for itself. Others are concerned with stemming the tide of third-party data aggregation to mitigate risks that can evolve from near total information awareness.

Internet Economics (Jan/Feb 2016)

The Internet both enables online versions of traditional markets and provides a platform for a vast range of new economic activity, ranging from targeted online advertising to crowdsourcing to peer-to-peer lending and digital currencies. These economic systems pose new theoretical and data-driven research questions: How do these online markets perform, and how should they be designed? How does the potentially giant scale of these systems affect performance? How do users behave in these online platforms, and how should incentives and interfaces be designed for maximum efficacy?

IEEE Internet Computing

www.computer.org/internet/