COMPUTING

edge

- Software
- Digital Transformation
- Big Data
- Mobile Computing

APRIL 2020

www.computer.org

IEEE COMPUTER SOCIETY

IEEE

IEEE Internet Computing

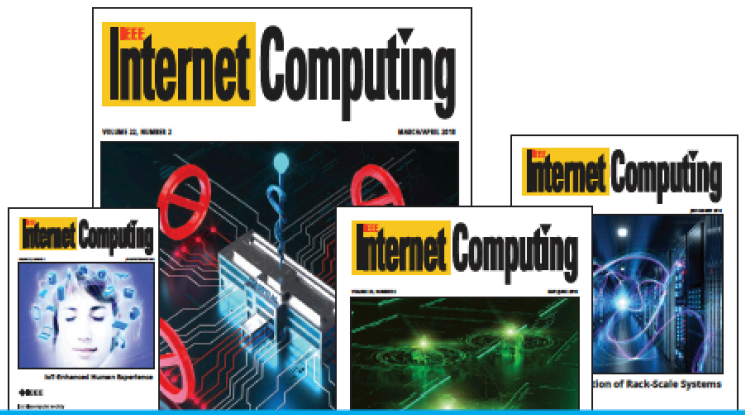
IEEE Internet Computing delivers novel content from academic and industry experts on the latest developments and key trends in Internet technologies and applications.

Written by and for both users and developers, the bimonthly magazine covers a wide range of topics, including:

- Applications
- Architectures
- Big data analytics
- Cloud and edge computing
- Information management
- Middleware
- Security and privacy
- Standards
- And much more

In addition to peer-reviewed articles, *IEEE Internet Computing* features industry reports, surveys, tutorials, columns, and news.

www.computer.org/internet



Join the IEEE Computer Society for subscription discounts today!

www.computer.org/product/magazines/internet-computing

IEEE COMPUTER SOCIETY

COMPUTING edge

IEEE COMPUTER SOCIETY computer.org

IEEE COMPUTER SOCIETY | IEEE | Printed with inks containing soy and/or vegetable oils

STAFF

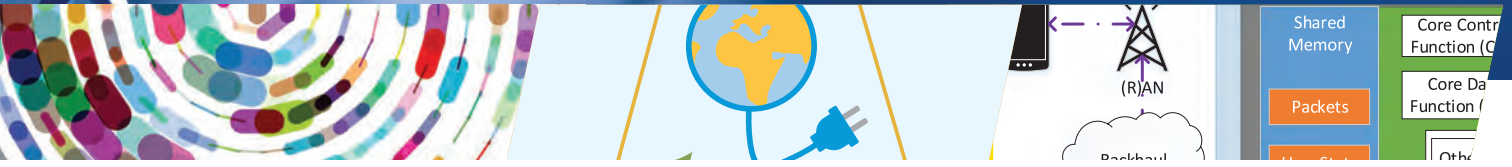
Circulation: *ComputingEdge* (ISSN 2469-7087) is published monthly by the IEEE Computer Society. IEEE Headquarters, Three Park Avenue, 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; voice +1 714 821 8380; fax +1 714 821 4010; IEEE Computer Society Headquarters, 2001 L Street NW, Suite 700, Washington, DC 20036.

Postmaster: Send address changes to *ComputingEdge*-IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Printed in USA.

IEEE Computer Society Magazine Editors in Chief

APRIL 2020 • VOLUME 6 • NUMBER 4

COMPUTING edge



Magazine Roundup

The IEEE Computer Society's lineup of 12 peer-reviewed technical magazines covers cutting-edge topics ranging from software design and computer graphics to Internet computing and security, from scientific applications and machine intelligence to visualization and microchip design. Here are highlights from recent issues.

Computer

Technology Predictions: Art, Science, and Fashion

Making accurate predictions isn't easy. However, many people have enjoyed making predictions throughout history, and even more individuals have taken part in learning about predictions. Read more in the December 2019 issue of *Computer*.

Computing in Science & Engineering

Force Sensor Model Based on FEA for the Electromagnetic Levitation System

The traditional control strategy of the electromagnetic levitation system (ELS) is a voltage or current control strategy based on the gap sensor and the current sensor. In this article from the November/December 2019 issue of *Computing in Science & Engineering*, an electromagnetic force sensor model is proposed and the mapping relation between electromagnetic force with current and levitation gap is obtained by Maxwell computation. The new force sensor model is used in ELS's control simulation, and the result shows that a new force sensor model can work well.

IEEE Annals of the History of Computing

The Spring of Artificial Intelligence in Its Global Winter

Against the dominant narrative of the history of artificial intelligence (AI), this article from the October–December 2019 issue of *IEEE Annals of the History of Computing* shows how, during the period considered to be the second global winter (1987–1993), the AI field experienced spring in South Korea. Putting the Korean language-processing problem at the center, researchers began to lay the intellectual and social foundation of AI in South Korea in the late 1980s.

IEEE Computer Graphics and Applications

Capturing and Visualizing Provenance from Data Wrangling

Data quality management and assessment play a vital role for ensuring trust in data and its fitness-of-use for subsequent analysis. The transformation history of a data-wrangling system is often insufficient for determining the usability of a dataset, lacking information on how changes affected the dataset. Capturing workflow provenance along the wrangling process and combining it with descriptive information as data provenance can enable users to comprehend how these changes affected the dataset, and if they benefited data quality. The authors of this article from the November/December 2019 issue of *IEEE Computer Graphics and Applications* present DQProv Explorer, a system that captures and visualizes provenance from data-wrangling operations.

IEEE Intelligent Systems

Non-cooperative Target Detection of Spacecraft Objects Based on Artificial Bee Colony Algorithm

Although heuristic algorithms have achieved state-of-the-art performance for object detection, they have not been demonstrated to be sufficiently accurate and robust for multi-object

detection. To address this problem, this article from the July/August 2019 issue of *IEEE Intelligent Systems* incorporates the concept of species into the artificial bee colony algorithm and proposes a multi-peak optimization algorithm named Species-Based Artificial Bee Colony (SABC). The authors apply SABC to detect the non-cooperative target (NCT) from two aspects: multi-circle detection and multi-template matching. Experiments are conducted using real cases of "ShenZhou8" and "Apollo 9" space missions, as well as the "Chang'e" camera point system developed by the Hong Kong Polytechnic University. Experimental results show that the proposed method is robust enough to detect NCT under various kinds of noise, under weak light, and in orbit, which leads to accurate detection results more quickly than other methods.

IEEE Internet Computing

Rapid Prototyping of IoT Solutions: A Developer's Perspective

Many new Internet-of-Things (IoT) devices and solutions appear in the market every day. Although commercial IoT products are the majority, Do-It-Yourself (DIY) solutions implemented by independent developers still represent a significant driving force. In this scenario, the availability of development tools for less-experienced developers and professionals to reduce the time needed to create prototypes is crucial. In this article from the July/August 2019 issue of *IEEE Internet Computing*, the authors first review the tools available to implement all the components of a typical IoT architecture in different programming languages, and then analyze how Python can be used to implement all the components of a typical IoT architecture. As a practical example, they illustrate the implementation of a smart home system built exploiting low-cost off-the-shelf hardware and programmed only through Python.

IEEE micro

Network-on-Chip Design Guidelines for Monolithic 3D Integration

Monolithic three-dimensional (M3D) integration is viewed as a promising improvement over through-silicon-via-based 3D integration due to its greater inter-tier connectivity, higher circuit density, and lower parasitic capacitance. With M3D integration, network-on-chip (NoC) communication fabric can benefit from reduced link distances and improved intra-router efficiency. However, the sequential fabrication methods utilized for M3D integration impose unique interconnect requirements for each of the possible partitioning schemes at transistor, gate, and block granularities. Furthermore, increased cell density introduces contention of available routing resources. Prior work on M3D NoCs has focused on the benefits of reduced distances, but has not considered these process-imposed circuit complications. In this article from the November/December 2019 issue of *IEEE Micro*, NoC topology decisions are analyzed in conjunction with these M3D interconnect requirements to provide an equivalent architectural comparison between M3D partitioning schemes.

IEEE MultiMedia

A Retrieval System of Medicine Molecules Based on Graph Similarity

Medicine information retrieval has grown significantly and is based on the structural similarity of medicine molecules. The chemical structural formula (CSF) is a primary search target as a unique identifier for each compound in the research field of medical information. This article from the

October–December 2019 issue of *IEEE MultiMedia* introduces a graph-based CSF retrieval system, PharmKi, which accepts the photos taken from smartphones and the sketches drawn on the tablet PCs as inputs. To establish a compact yet efficient hypergraph representation for molecules, the authors propose a graph-isomorphism-based algorithm for evaluating the spatial similarity between graphical CSFs. An indexing strategy based on the graph TF-IDF technology is also introduced to achieve a high efficiency for large-scale molecule retrieval. The results of a comparative study demonstrate that the proposed method outperforms the existing methods on accuracy, and performs well on efficiency.

IEEE Pervasive Computing

Esports Athletes and Players: A Comparative Study

The authors of this article from the July–September 2019 issue of *IEEE Pervasive Computing* present a comparative study of regular players' and professional players' (athletes') performance in Counter Strike: Global Offensive discipline. Their study is based on ubiquitous sensing, helping to identify the biometric features significantly contributing to the classification of particular skills of the players. The research provides a better understanding of why the athletes demonstrate superior performance as compared to other players.

IEEE Security & Privacy

The Security Implications of Data Subject Rights

Data protection regulations give individuals rights to obtain the information that entities have on them. However, providing such information can also reveal aspects of the entity's underlying technical infrastructure and organizational processes. This article from the November/December 2019 issue of *IEEE Security & Privacy* explores the security implications this raises and highlights the need to consider such rights in fulfillment processes.

IEEE Software

From Art to Science: The Evolution of Community Development

Community development in open-source ecosystems is increasingly complex. This article from the November/December 2019 issue of *IEEE Software* focuses on the OpenShift and CNCF ecosystems and concludes that cross-community collaboration analysis is challenging and that a more scientific approach is required.

IT Professional

Autonomous Cars: Challenges and Opportunities

We are witnessing an evolution in the automotive industry. Recent technological advances and fast proliferation of technologies are accelerating the development of smarter vehicles, including autonomous cars. The authors of this article from the November/December 2019 issue of *IT Professional* analyze the state-of-the-art results of autonomous cars, identifying the main stakeholders and their role in their success or failure. They also discuss some of the challenges and opportunities of autonomous cars and identify some applications that might justify their adoption in modern society.

Editor's Note

The Growth of Open Source

Over the past decade, the programming community has embraced open source software (OSS), which is now used in everything from mobile apps to defense systems. The sharing and collaboration that OSS allows has propelled its popularity among coders, researchers, and organizations. GitHub recently surpassed 40 million users and 100 million repositories, while 89 percent of IT leaders say that OSS is important in their organization, according to a Red Hat survey.

This *ComputingEdge* issue opens with two articles from *Computer* that focus on the role of OSS in modern software development. In "The Innovations of Open Source," the author discusses the important legal, process, tool, and business model innovations that the open source movement has produced in the software industry and beyond. "How to Select Open Source Components" provides a practical guide for choosing the right open source projects to use in your work based on requirements such as functionality, licensing, documentation, and code quality.

Open source isn't the only trend shaping the software industry; digital transformation—which refers to organizations leveraging the latest computing technology to increase their impact—will also affect the future of software. *IEEE Software*'s "Digital Transformation" posits that it "will completely reshape the landscape of software technologies and processes." *IT Professional*'s "Governing and Piloting Emerging Technologies" advises business leaders on which cutting-edge technologies to deploy in their companies.

Another article from *IT Professional* covers digital transformation with a focus on big data. In "Big Data Solutions for Micro-, Small-, and Medium-Sized Enterprises in Developing Countries," the authors show how utilizing big data is helping businesses in developing countries improve processes and access financial services. *IEEE Internet Computing*'s "Analytics without Tears, or Is There a Way for Data to Be Anonymized and Yet Still Useful?" discusses privacy concerns related to big data analytics.

The final two articles in this *ComputingEdge* issue address different aspects of 5G cellular networks: security and performance. *IEEE Security & Privacy*'s "It Takes a Village to Secure Cellular Networks" examines security challenges and possible solutions. *IEEE Internet Computing*'s "Improving Performance and Scalability of Next-Generation Cellular Networks" investigates possible changes to system architecture and protocols that could lead to lower latency and a better user experience.

IEEE COMPUTER SOCIETY

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.

MEMBERSHIP: Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

COMPUTER SOCIETY WEBSITE: www.computer.org

OMBUDSMAN: Direct unresolved complaints to ombudsman@computer.org.

CHAPTERS: Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

AVAILABLE INFORMATION: To check membership status, report an address change, or obtain more information on any of the following, email Customer Service at help@computer.org or call +1 714 821 8380 (international) or our toll-free number, +1 800 272 6657 (US):

- Membership applications
- Publications catalog
- Draft standards and order forms
- Technical committee list
- Technical committee application
- Chapter start-up procedures
- Student scholarship information
- Volunteer leaders/staff directory
- IEEE senior member grade application (requires 10 years practice and significant performance in five of those 10)

PUBLICATIONS AND ACTIVITIES

***Computer*:** The flagship publication of the IEEE Computer Society, *Computer* publishes peer-reviewed technical content that covers all aspects of computer science, computer engineering, technology, and applications.

Periodicals: The society publishes 12 magazines and 18 journals. Refer to membership application or request information as noted above.

Conference Proceedings & Books: Conference Publishing Services publishes more than 275 titles every year.

Standards Working Groups: More than 150 groups produce IEEE standards used throughout the world.

Technical Committees: TCs provide professional interaction in more than 30 technical areas and directly influence computer engineering conferences and publications.

Conferences/Education: The society holds about 200 conferences each year and sponsors many educational activities, including computing science accreditation.

Certifications: The society offers three software developer credentials. For more information, visit www.computer.org/certification.

BOARD OF GOVERNORS MEETING

28 – 29 May: McLean, Virginia

EXECUTIVE COMMITTEE

President: Leila De Floriani
President-Elect: Forrest Shull
Past President: Cecilia Metra
First VP: Riccardo Mariani; **Second VP:** Sy-Yen Kuo
Secretary: Dimitrios Serpanos; **Treasurer:** David Lomet
VP, Membership & Geographic Activities: Yervant Zorian
VP, Professional & Educational Activities: Sy-Yen Kuo
VP, Publications: Fabrizio Lombardi
VP, Standards Activities: Riccardo Mariani
VP, Technical & Conference Activities: William D. Gropp
2019–2020 IEEE Division VIII Director: Elizabeth L. Burd
2020-2021 IEEE Division V Director: Thomas M. Conte
2020 IEEE Division VIII Director-Elect: Christina M. Schober

BOARD OF GOVERNORS

Term Expiring 2020: Andy T. Chen, John D. Johnson, Sy-Yen Kuo, David Lomet, Dimitrios Serpanos, Hayato Yamana
Term Expiring 2021: M. Brian Blake, Fred Douglis, Carlos E. Jimenez-Gomez, Ramalatha Marimuthu, Erik Jan Marinissen, Kunio Uchiyama
Term Expiring 2022: Nils Aschenbruck, Ernesto Cuadros-Vargas, David S. Ebert, William Gropp, Grace Lewis, Stefano Zanero

EXECUTIVE STAFF

Executive Director: Melissa A. Russell
Director, Governance & Associate Executive Director: Anne Marie Kelly
Director, Finance & Accounting: Sunny Hwang
Director, Information Technology & Services: Sumit Kacker
Director, Marketing & Sales: Michelle Tubb
Director, Membership Development: Eric Berkowitz

COMPUTER SOCIETY OFFICES

Washington, D.C.: 2001 L St., Ste. 700, Washington, D.C. 20036-4928; **Phone:** +1 202 371 0101; **Fax:** +1 202 728 9614; **Email:** help@computer.org
Los Alamitos: 10662 Los Vaqueros Cir., Los Alamitos, CA 90720; **Phone:** +1 714 821 8380; **Email:** help@computer.org

MEMBERSHIP & PUBLICATION ORDERS

Phone: +1 800 678 4333; Fax: +1 714 821 4641;
Email: help@computer.org

IEEE BOARD OF DIRECTORS

President: Toshio Fukuda
President-Elect: Susan K. "Kathy" Land
Past President: José M.F. Moura
Secretary: Kathleen A. Kramer
Treasurer: Joseph V. Lillie
Director & President, IEEE-USA: Jim Conrad
Director & President, Standards Association: Robert S. Fish
Director & VP, Educational Activities: Stephen Phillips
Director & VP, Membership & Geographic Activities: Kukjin Chun
Director & VP, Publication Services & Products: Tapan Sarkar
Director & VP, Technical Activities: Kazuhiro Kosuge

IEEE

revised 14 January 2020

EDITOR: Dirk Riehle, Friedrich Alexander-University of Erlangen Nürnberg, dirk.riehle@fau.de

This article originally appeared in
Computer
vol. 52, no. 4, 2019

COLUMN: OPEN SOURCE EXPANDED

The Innovations of Open Source

Dirk Riehle, *Friedrich-Alexander-Universität, Erlangen-Nürnberg*

FROM THE EDITOR

Welcome to this new column on open source! The "Open Source Expanded" column will aim to provide an insightful article every two months. These articles will be written for the software practitioner by authors from both academia and industry. Articles will be grouped by theme rather than appearing in arbitrary order. Our first theme is about open source licenses and license compliance. Even decades after open source was created, it is still a hot topic and unknown territory for many. Later themes will focus on using open source, project communities, business models, interesting cases, and what might come after open source. This issue's article, the first in its series, provides an overview of what is to come by reviewing the most important innovations that open source has provided for the software industry and beyond. If you have comments or would like to suggest future themes and articles, feel free to contact me at dirk@riehle.org. *Computer* will provide a discussion board for articles as well. — *D. Riehle*

Open source has given us many innovations. This article provides an overview of the most important innovations and illustrates the impact that open source is having on the software industry and beyond.

The main innovations of open source can be grouped into four categories: legal, process, tool, and business models. Probably the best known innovations are open source licenses, which also define the concept. Software becomes open source if users receive it under an open source license. To be an open source license, it must fulfill 10 requirements set forth by the Open Source Initiative, the protector and arbiter of what constitutes open source.[1] Most notably, the license must allow

- free-of-charge use of the software
- access to and modification of the source code
- the ability to pass on the source code and a binary copy.

Before there was open source software, there was free software. Richard Stallman defined the four freedoms of software that make it "free" as:[2]

> *the freedom to run the program as you wish, for any purpose [...], the freedom to study how the program works, and change it so it does your computing as you wish [...], the freedom to redistribute copies so you can help others [...], the freedom to distribute copies of your modified versions to others [...].*

Open source software and free software, and the people behind them, have struggled with each other at times. For all practical purposes, however, the

DOI No. 10.1109/MC.2019.2898163
Date of publication: 16 April 2019

difference is irrelevant to users. What matters is the license under which a user receives a particular software.

LEGAL INNOVATION

Licenses can be structured into permissions (the rights granted to a user), obligations (what is required to receive these rights), and prohibitions (what may not be done; for example, claiming that using the software implies an endorsement by its creator). The two legal innovations are

1. the rights grant as introduced earlier
2. a particular obligation called *copyleft*.

The rights grant helped open source spread and succeed. As research has shown, it taps into the human desire to help each other and collaborate on interesting projects.

People sometimes ask why developers do not put their work into the public domain. This misses the point: by putting something into the public domain, an author typically waives his or her rights, and most authors do not want that. Rather, they want to be specific about which rights they grant and which obligations they require.

The most famous license obligation is probably the copyleft clause. Stallman invented this clause, and it became popular through GNU General Public License v2 in 1991. It states that if you pass on copyleft-licensed code, such as part of a product that you sell, you must also pass on your own source code if it modifies the copyleft-licensed code. The specifics of this can get complicated quickly, and they will be discussed in more detail in future columns. Many companies worry that if their source code is mixed with copyleft-licensed code, they will lose their intellectual property and, hence, their competitive advantage in the marketplace.

In the past, companies have used this clause to incorrectly discredit open source software as "a virus" or "cancer" and a "communist" or "hippie undertaking." However, nobody forces anyone to use open source software. In an amazing about-face, some of the most well-known companies that fought open source only 15 years ago are now among its biggest supporters. The "Business Model Innovation" section of this article explains some of this.

PROCESS INNOVATION

The next innovation open source has brought us is engineering process innovation.[3] The open source initiative has this to say about open source software development:[1]

> *Open source is a development method for software that harnesses the power of distributed peer review and transparency of process. The promise of open source is better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in.*

This is the other definition of open source, which does not focus on licenses and intellectual property but, rather, on collaborative development. There is no single open source software engineering process because each open source community defines its own.

Through his development of the Linux kernel, Linus Torvalds was the first to explore, at scale, a truly collaborative open source process. His approach has no particular name but is often identified with his moniker, BDFL (which stands for "benevolent dictator for life"), implying a hierarchical structure. A core benefit of an open collaboration process was named after Torvalds and is called Linus' law, which states, "Given enough eyeballs, all bugs are shallow."[4] The idea is that more broadly used software matures more quickly since problems are found and solved more quickly.

The collaborative peer group, as explored by the original Apache web server team (httpd) and codified

as The Apache Way (of open source software development), is a similar but different approach that may be more popular today.[5] The software industry owes this group of developers as much as it owes Torvalds, if not more.

The Apache Way is a consensus-based, community driven governance approach to collaboration in open source projects. The Apache Software Foundation's website explains it in detail. An important aspect is the distinction between contributors, who submit work for inclusion in an open source project, and committers, who review and integrate the work. Committers are called *maintainers* in a Linux context, and they usually are developers, too, not just reviewers. Using this contributor–committer interplay, nearly all open source projects practice precommit code review to ensure the quality of the software under development.

The principles of open source software development can be summarized as three principles of open collaboration.[6]

- In open collaboration, participation is egalitarian (nobody is a priori excluded).
- Decision making is meritocratic (based on the merits of arguments rather than status in a corporate hierarchy).
- People are self-organizing (they choose projects, processes, and tasks rather than being assigned to them).

Similarly, open source projects practice open communication. This form of communication is public (everyone can see it), written (so you don't have to be there when words are spoken), complete (if it wasn't written down, it wasn't said), and archived (so that people can look up and review discussions later).

Such open collaborative processes, which are not dominated by any single entity, lead to community open source software, which is collectively owned, managed, and developed by a diverse set of stakeholders. These collaboration processes are not limited to software but spill over into adjacent areas. For example, they have brought forward many formal and de facto standards that the software industry relies on.[3] The methods for open source software development have also taken root inside companies, where they are called inner source.[7,8]

TOOL INNOVATION

Most of the tools used in open source software development are familiar to closed source programmers as well. However, the needs of open source processes have led to two major tool innovations that have since become an important part of corporate software development as well: software forges and distributed version control.

A software forge is a website that allows the creation of new projects and provides developers with all of the tools needed for software development, such as a home page, an issue tracker, and version control. What makes software forges special is that they facilitate matchmaking between those who are looking to find a useful software component and those who are offering one. They are an enterprise software product category because, even within one company, you want to have one place for all software being developed.

Distributed version control is version control in which you copy the original repository and work with your copy. Thus, you do not need commit rights or ask for permission to start work. Git and Mercurial are the two best-known examples of such software. Some may argue that distributed version control is not an open source innovation because some of its roots are in proprietary software. However, the open source community developed and refined its own solutions, which work well with how open source software is developed, and thereby popularized the concept.

Comparing distributed version control with branching misses the point. Having your own repository allows developers to work using their own style, free of any centralized decisions on how to use branches.

Distributed version control was popularized by being the main version control software underlying a new generation of software forges, most notably Github and Gitlab. As such, companies are adopting both forges and distributed version control at a rapid pace.

BUSINESS MODEL INNOVATION

Open source is changing the software industry by how it makes new business models and breaks old ones. For instance, it lays the legal foundation for open collaboration between individuals and companies, defines more effective collaboration processes with higher productivity than closed-source approaches,

and invents the tools to support it. Open source itself may not be a business model, but it is a potent strategy and a tool to use in competitive environments.

For-profit models

There are different approaches for classifying business models enabled by open source, but I like to put them into five categories. Three are for-profit business models, and two are nonprofit models. The for-profit business models are as follows.

1. *Consulting and support business models*: In this conventional model, a company earns money by providing consulting and support services for existing open source software. They do not sell a license, but they service the software anyway. The original open source service company was Cygnus Solutions, which serviced the GNU set of tools. More recent examples are Cloudera and Hortonworks, which service Hadoop.
2. *Distributor business model*: In this business model unique to open source, a company sells subscriptions to software and associated services that are partly or completely based on open source software. This model only works for complex software that consists of tens or hundreds and sometimes thousands of possibly incompatible components that a customer wants to use.

   The most well-known examples are Linux distributors like Red Hat and Suse, but many other smaller companies provide distributions of other kinds. The competitively differentiating intellectual properties of a distributor are its test suites, configuration databases, and compatibility matrices, which they typically do not open source.
3. *Single-vendor open source business model*: In this model, a company goes to market by providing a sometimes reduced, sometimes complete, version of its product as open source. The company never lets go of full ownership of the software and sets up various incentives for users to move from the free open source version to a paid-for, commercially licensed version. The most common incentives are support and update services, but it often also includes a copyleft license that users would like to replace with a proprietary one.

   If done correctly, both the company and its products benefit from the help of the community of nonpaying users. The company typically does not get code contributions, but it does get lively discussion forums, bug reports, feature ideas, and word-of-mouth marketing. The most well-known example of this model was MySQL, the database company, but there are many more recent ones, such as SugarCRM, MongoDB, and Redis Labs.

The distributor and single-vendor models are especially important because they enable returns on investment that are attractive to venture capitalists. Thus, they are the main conduit through which billions of dollars have been invested into open source software.

Open source foundations

There are two more models that determine how the development of open source software is being funded. They are actually two variants of the same idea: the open source foundation.

An open source foundation is a nonprofit organization tasked with governing one or more open source projects, representing them legally, and ensuring their future. In the past, open source foundations were set up to ensure the survival of unsupported community open source projects, but companies are increasingly coming together to set up a foundation with the goal of developing new open source software.

The two variants of open source foundations are as follows.

1. *Developer foundations*: This type of nonprofit foundation is run by software vendors (developers) who decide to join forces to ensure the survival and health of the open source software they depend on. By ensuring broadly shared ownership of the software, the vendors make certain that no one can monopolize this particular type of component and reap all of the profits from software products that rely on it. This is why Linux was supported against

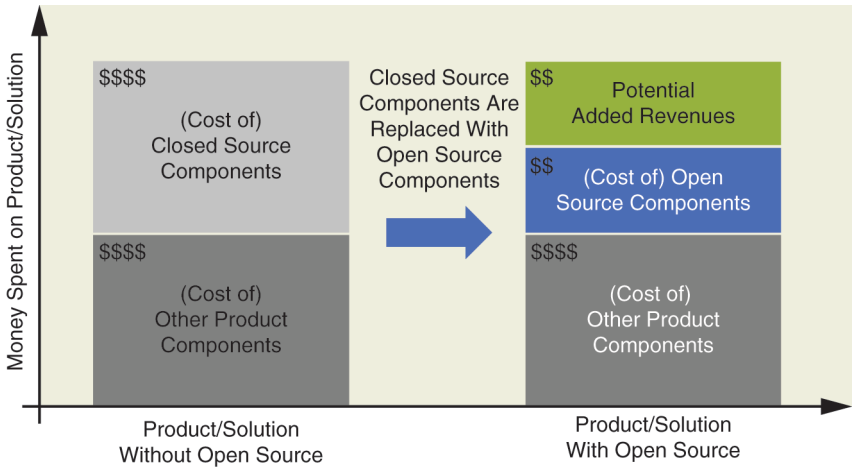


FIGURE 1. The economic logic of community open source software.

Microsoft Windows, Eclipse against Microsoft Visual Studio, and, more recently, OpenStack against Amazon Web Services.

2. *User foundations*: This type of nonprofit is predominantly run by companies that are not software vendors but rely on the software managed by the foundation, either as part of their operations or directly as part of a product that is only partly software. Examples are the Kuali Foundation for software to run universities, the GENIVI foundation for automotive infotainment software, and the openKONSEQUENZ foundation for software for the (German) smart energy grid (the last of which I helped create).

Figure 1 shows how replacing a closed source component in a product with an open source component shifts profits between the different component suppliers and generally leaves more profit for the vendor, which integrates the components and sells the final product. Because of this economic logic, I expect to see more product vendors and service suppliers from outside the software industry get in on the game. They will fund the development of open source components they need, taking money out of the market for such components and moving it to places where they can more easily appropriate it. Therefore, in the future, we can expect funding for open source software development to increase by a couple of orders of magnitude.

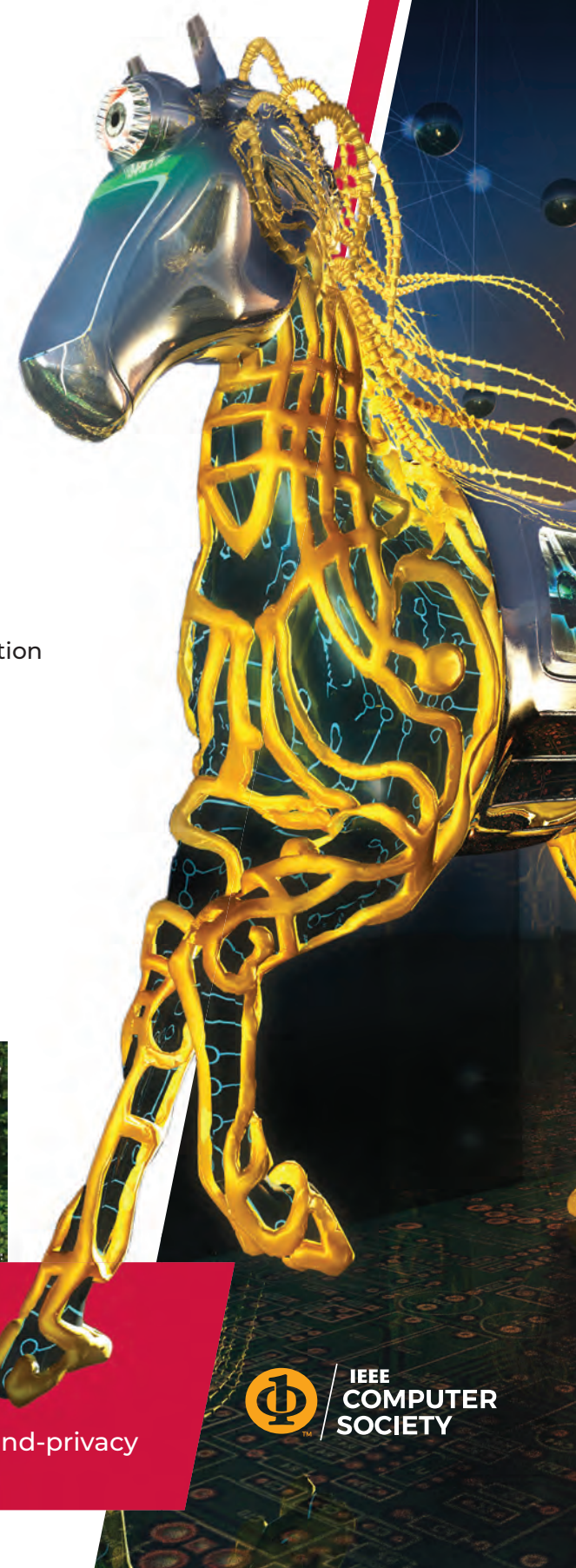
REFERENCES

1. Open Source Initiative. 2018. [Online]. Available: https://opensource.org
2. Free Software Foundation, "GNU operating system." 2018. [Online]. Available: https://www.gnu.org/philosophy/free-sw.en.html
3. C. Ebert, "Open source drives innovation," *IEEE Softw.*, vol. 24, no. 3, pp. 105–109, 2007.
4. E. Raymond, "The cathedral and the bazaar," *Knowledge, Technol. Policy*, vol. 12, no. 3, pp. 23–49, 1999.
5. The Apache Software Foundation. 2018. [Online]. Available: https://www.apache.org/foundation/how-it-works.html
6. D. Riehle, "The five stages of open source volunteering," in *Crowdsourcing*, W. Li, M. N. Huhns, W.-T. Tsai, and W. Wu, Eds. New York: Springer-Verlag, 2015, pp. 25–38.
7. J. Dinkelacker, P. K. Garg, R. Miller, and D. Nelson, "Progressive open source," in *Proc. 24th Int. Conf. Software Engineering*, 2002, pp. 177–184.
8. D. Riehle, M. Capraro, D. Kips, and L. Horn, "Inner source in platform-based product engineering," *IEEE Trans. Softw. Eng.*, vol. 42, no. 12, pp. 1162–1177, Dec. 2016.

DIRK RIEHLE is the professor for open source software at the Friedrich Alexander-University of Erlangen Nürnberg. Contact him at dirk@riehle.org.

IEEE

SECURITY & PRIVACY



***IEEE Security & Privacy* is a bimonthly magazine communicating advances in security, privacy, and dependability** in a way that is useful to a broad section of the professional community.

The magazine provides articles with both a practical and research bent by the top thinkers in the field of security and privacy, along with case studies, surveys, tutorials, columns, and in-depth interviews. Topics include:

- Internet, software, hardware, and systems security
- Legal and ethical issues and privacy concerns
- Privacy-enhancing technologies
- Data analytics for security and privacy
- Usable security
- Integrated security design methods
- Security of critical infrastructures
- Pedagogical and curricular issues in security education
- Security issues in wireless and mobile networks
- Real-world cryptography
- Emerging technologies, operational resilience, and edge computing
- Cybercrime and forensics, and much more

www.computer.org/security

Join the IEEE Computer Society for subscription discounts today!

www.computer.org/product/magazines/security-and-privacy

IEEE COMPUTER SOCIETY

EDITOR: Dirk Riehle, Friedrich Alexander-University of Erlangen Nürnberg, dirk.riehle@fau.de

This article originally appeared in
Computer
vol. 52, no. 12, 2019

COLUMN: OPEN SOURCE EXPANDED

How to Select Open Source Components

Diomidis Spinellis, *Athens University of Economics and Business*

FROM THE EDITOR

Welcome back! Open source gives you high-quality software for free. What's not to like about this? But wait a second: You need to choose the right open source component. Making a poor choice for using an open source component in your products or projects can create serious problems. In this article, well-known open source expert Diomidis Spinellis takes us through the process of selecting the right open source component for your needs. Significant thought should be spent on such a decision, because using an open source component creates a dependency that needs to be managed, and some dependencies are easier to manage than others. But more on this topic in one of the next columns. As always, happy hacking! — *Dirk Riehle*

With millions of open source projects available on forges such as GitHub, it may be difficult to select those that best match your requirements. Examining each project's product and development process can help you confidently select the open source projects required for your work.

If most of the code comprising your product or service isn't open source software, it's highly likely that you're wasting effort and cash reinventing the wheel. Yet with millions of open source projects available on forges such as GitHub, it may be difficult to select those that best match your requirements. Examining two facets of each candidate project, the product and its development process (see Table 1), can help you select with confidence the open source projects required for your work.

PRODUCT

Functionality

Begin by assessing the functionality of the project under consideration and determine whether it covers both current needs and future strategic directions. For instance, if you are selecting a message queue, consider whether the underlying messaging protocol matches the one prevalent in your industry and whether the system can scale in the future to cover your projected needs. It is equally important to evaluate whether the project's functionality is egregiously excessive compared to your needs. For example, if you simply want to compress data that you store in a file, you may not want to use a multiformat data archiving library. Selecting a small, focused project over a larger one has many advantages. In typical cases, such a choice will offer a reduced storage footprint for your system, fewer transitive third-party dependencies, a lower installation complexity, and a smaller surface vulnerable to malicious attacks.

If an open source project's functionality nearly fits your organization's needs and no other project can

DOI No. 10.1109/MC.2019.2940809
Date of current version: 22 November 2019

	Attribute	Deal breaker	Areas that may require investment	Areas important for in-house development
Product	Functionality	Partial		
	Licensing	Full		
	Nonfunctional properties	Partial		
	Popularity			
	Documentation			
	Code quality			
	Build system			
Process	Development process			
	Code commits			
	Project releases			
	Support			
	Issue management			
	Acceptance of contributions			

TABLE 1. Judging the open source project selection criteria.

satisfy them completely, you can still use it and make the required changes on your own. However, under this scenario, you must more stringently evaluate the elements I outline later on regarding source code changes and contributions. See the last column in Table 1.

Licensing

Narrow down your search by examining whether the project's licensing[1] is compatible with your business model, mission, or other software you are using. Within your project's source code, if you directly incorporate elements licensed under the GNU General Public License, then you also must distribute your code under the same license. This may be undesirable if your business model depends on keeping your product's source code under wraps; in this case, you should be looking for projects that use more permissive licenses, such as the Berkeley Software Distribution and Apache ones. Similar concerns apply if you are offering software as a service, and you plan to use software licensed under the Affero General Public License. As another example, software released under version 1.1 of the Mozilla Public License cannot be linked together with code licensed under the GNU General Public License.

Nonfunctional properties

Evaluate the project's fit with your requirements by also looking at its nonfunctional properties. Is it compatible with your product's processor architecture, operating system, and middleware? Will it accommodate your future expansion plans and directions? For example, if your product works on macOS but you're also eyeing the Windows market, then you should be using open source libraries supported on both systems. Is the product's performance compatible with your requirements? This is especially important when selecting a database or a big data analytics infrastructure. If performance is critical, do not assume particular performance outcomes; rather, benchmark with realistic workloads.

Popularity

Then consider the project's popularity. Popularity is important because it can determine how likely it will be

for your questions to receive answers on public forums, for volunteers to contribute fixes and enhancements, and for the project to continue to evolve if its original developers veer off course (namely, losing interest or steering the project toward an undesirable direction). Simple metrics, such as GitHub stars, the number of StackOverflow questions with the corresponding tag, the download count, and the number of Google query results are all usually sufficient to discern the cases that really matter.

Documentation

The project's documentation is another aspect that should be examined. Although most answers regarding a software's operation ultimately lie in the source code, resorting to such digging for everyday operations is undesirable. Therefore, judge how well the software is documented, both at the technical (installation and maintenance procedures) and user levels (tutorials and reference manuals). Although nearly all mature open source software projects are well documented, some smaller ones suffer in this dimension. There are Unix command-line utilities, for example, that lack the traditional manual page. I try to avoid such projects, both to keep my sanity (life is too short to waste on hunting down command-line options) and because such a level of indifference toward the end user is often a sign of deeper problems.

SOURCE CODE

This brings me to another product characteristic you should check, namely the project's source code and the code's quality. If you anticipate adjusting the project to your needs, then select projects written in programming languages with which you are familiar. Even if you don't plan to touch the project's source code, low code quality can affect you through bugs, security vulnerabilities, poor performance, and maintenance problems. Again, there's no need to dig deeply to form a useful opinion. In most cases, your objective is to avoid problematic projects, not to perform thorough due diligence of the code. Look at the project's source code files. Are they named and organized into directories following the conventions of the project's programming language? Is there evidence of unit testing? Does the repository also contain elements that it shouldn't, such as object and executable files? Open and browse a few files. Are methods or functions short and readable? Are identifiers well chosen? Is the code reasonably commented? Is the formatting consistent with the language's coding conventions? Again, serious deviations are often indicators of more important hidden flaws.

Build process

The quality of a project's build process is important for two reasons. Some organizations reuse open source code projects through binary distributions, as libraries, that they link with their other code or as components that run on their infrastructure. If your organization works like this, at some point you may need to build the binary from source code to fix a bug or add a feature required by your organization. Other organizations (mostly larger ones) have strict rules against using random binaries off the Internet and have processes for building everything internally from source (at least once).

SOFTWARE RELEASED UNDER VERSION 1.1 OF THE MOZILLA PUBLIC LICENSE CANNOT BE LINKED TOGETHER WITH CODE LICENSED UNDER THE GNU GENERAL PUBLIC LICENSE.

Whatever the case, it's sensible to check how easy it is to perform a project build. Is the procedure documented? Does it work in your environment? Will you need some rarely used build tools, an unsupported integrated development environment, or a compiler for an exotic programming language? For critical dependencies, evaluate these requirements in the same way that you're evaluating the primary open source project under consideration.

PROCESS

No matter how shiny the open source project appears to your eyes, you also should invest some time to examine how it is produced and managed. This will affect your experience with it in the long term and also may uncover potential pitfalls that weren't discernible from the product's examination.

Development process

Start by evaluating the quality of the project's development process. Does the project practice continuous integration? You can easily determine this by looking for corresponding configuration files (for example, .travis.yml or Jenkinsfile) in the project's root directory. Examine what the continuous integration pipeline exercises. Does it, for example, include static analysis of the code as well as unit testing? Does it build and spell-check the documentation? Does it calculate testing code coverage? Does it enforce coding standards? Does it check for up-to-date dependencies? A shortcut for answering these questions are badges appearing in the project's GitHub page, though their significance is not always a given.[2]

A LACK OF FRESH COMMITS MAY IMPLY THAT NOBODY WILL STEP IN TO ADDRESS NEW REQUIREMENTS OR BUGS.

Code commits

Then look at code commits to the project's revision management repository. Are commits regularly made by a diverse group of committers? Unless the project is very stable and likely to remain so (consider a numerical library), a lack of fresh commits may imply that nobody will step in to address new requirements or bugs. Similarly, commits by a single author or very few signal that the project suffers from a key person risk. Also known as a *bus factor*, this identifies the danger the project faces if, for example, a lead developer is hit by a bus.[3] Also, look at the details of a few commits. Are they clearly labeled and appropriately described? Do they reference any documented issues that they have addressed using a standard convention? Is there evidence that code changes and additions have been reviewed and discussed?

Project releases

Down the road, see how these commits translate into complete project releases. Are these sufficiently recent and frequent? For cutting-edge projects (say, a deep-learning library), you want to see regular updates; for more stable ones, you're looking for evidence of maintenance releases. In some cases, frequently integrating new releases of an open source component into your code base can be disruptive, due to the risks and additional work of this process. To avoid these problems, check for a separate release channel for obtaining only security and other critical fixes. Additionally, to minimize the disturbance associated with bringing in major updates, see if there are so-called long-term support releases and determine whether their time horizon matches your project's pace.

Support channels

Source code availability is an excellent insurance policy for obtaining support because it allows you to resolve issues and fix bugs within your organization; "Use the source, Luke," to paraphrase a line from *Star Wars*. Such measures, however, are typically extreme. When using open source software, a helpful support forum is usually the most practical way to resolve such problems. Consequently, look for the project's available support channels. Is there an online forum, a mailing list, or a chat group where you can ask questions? Do useful answers arrive quickly? Are respondents supportive and friendly? In my experience, the quality of a project's technology and its support are orthogonal. Some projects with mediocre quality code offer excellent support and vice versa. For enterprise scenarios where it's not prudent to rely on volunteer help for resolving critical issues, you may also wish to examine the quality of paid support options offered through specialized companies, consultants, or products.

Handling issues

Inevitably, at some point, you're likely to encounter a bug in the open source project you're using. Therefore, it's worth examining how the project's volunteers handle issues.[4] Many open source projects offer access to their issue management platform, such as GitHub Issues, Bugzilla, or Jira, which allows you to look under the hood of issue handling. Are issues resolved quickly? How many issues have been left rotting open for ages? Does the ratio between open and closed issues appear to be under control, that is, in line with the number of project contributors?

Contributing fixes and enhancements

Another scenario down the road concerns the case where you make some changes to the project's source code, either to fix a bug or add a new feature that your organization requires. Although you can keep your changes to yourself, integrating them into the upstream project safeguards their continued availability and maintenance alongside new releases (in addition to it being the proper thing to do).

Evaluate how you'll fare in this case by examining how easy it is to contribute fixes and enhancements. Is there a contributor's guide? If you're using the project as a binary package, is it easy to build and test the project from its source code? Through what hoops do you have to jump to get your contribution accepted? Is there an efficient method by which to submit your changes, for example, through a GitHub pull request? Does the project regularly accept third-party contributions? Note that some organizations release projects with an open source code license but allow little or no code to be contributed back to their code base.

All 13 evaluation criteria I've outlined in Table 1 are important, and taking them into account can spare you unpleasant surprises and the cost of switching from one project to another. Furthermore, you can use Table 1 as guidance on how crucial some criteria are in specific contexts. Specifically, those identified in the first colored column can be deal breakers. In addition, if you identify problems with the yellow-marked criteria in the second column, this means that you'll need to build in-house capacity to support the corresponding open source project. Finally, if you decide to support the project with in-house resources, then the green-marked components in the third column become more important. Ultimately, all of these checks will help to ensure a long, happy, and prosperous relationship with the open source components you're selecting for your work.

ACKNOWLEDGMENTS

I thank Zoe Kotti and Alexios Zavras, who made many helpful suggestions in an earlier version of this article.

REFERENCES

1. A. Morin, J. Urban, and P. Sliz, "A quick guide to software licensing for the scientist-programmer," *PLOS Comput. Biol.*, vol. 8, no. 7, pp. 1–7, 2012.
2. A. Trockman, S. Zhou, C. Kästner, and B. Vasilescu, "Adding sparkle to social coding: An empirical study of repository badges in the *npm* ecosystem," in *Proc. 40th Int. Conf. Software Engineering*, 2018, pp. 511–522.
3. K. Yamashita, S. McIntosh, Y. Kamei, A. E. Hassan, and N. Ubayashi, "Revisiting the applicability of the Pareto principle to core development teams in open source software projects," in *Proc. 14th Int. Workshop Principles of Software Evolution*, 2015, pp. 46–55.
4. T. F. Bissyandé, D. Lo, L. Jiang, L. Réveillère, J. Klein, and Y. L. Traon, "Got issues? Who cares about it? A large scale investigation of issue trackers from GitHub," in *Proc. IEEE 24th Int. Symp. Software Reliability Engineering (ISSRE)*, 2013, pp. 188–197.

DIOMIDIS SPINELLIS is a professor of software engineering and head of the Department of Management Science and Technology, Athens University of Economics and Business. His most recent book is *Effective Debugging: 66 Specific Ways to Debug Software and Systems*. He is a Senior Member of the IEEE and ACM. Contact him at dds@aueb.gr.

EDITOR: Christof Ebert, Vector Consulting Services, christof.ebert@vector.com

This article originally appeared in
IEEE Software
vol. 35, no. 4, 2018

DEPARTMENT: SOFTWARE TECHNOLOGY

Digital Transformation

Christof Ebert and Carlos Henrique C. Duarte

FROM THE EDITOR

Digital transformation is a technology-driven continuous change process of companies and our entire society. Its cornerstone is ubiquitous embedded computing, connectivity, and flexible value streams. Obviously, there's no digital transformation without software. But what are the underlying platforms? What lessons can we take from actual projects? In this instalment of Software Technology, Carlos Duarte and I highlight lessons learned from transformation projects and the underlying technologies and platforms. I look forward to feedback from both readers and prospective article authors. —*Christof Ebert*

Digital transformation (DX) is about adopting disruptive technologies to increase productivity, value creation, and the social welfare. Many national governments, multilateral organizations, and industry associations have produced strategic-foresight studies to ground their long-term policies. By proposing the implementation of public policies regarding DX, such groups expect to achieve the goals listed in Table 1.

DX is forecasted to have high annual growth and fast penetration.[1–3] But there are barriers slowing its dissemination, such inadequate or overly heterogeneous company structures or cultures, the lack of DX strategies and ROI (return on investment) visibility, and even the perception of cannibalization of existing businesses (the "innovator's dilemma"[4]). External barriers also exist, such as the lack of recognition of how DX will benefit all of society, a shortage of skills and a qualified labor force, lacking or insufficient infrastructure, missing or inadequate regulation and consumer protection, and poor access to funding, particularly for small and medium businesses.

THE INDUSTRY PERSPECTIVE ON DX

Industry is moving to adopt holistic business models, completely redesign products and services, and establish closer interactions with suppliers and long-term partnerships with customers.[5,6] The widespread implementation of DX will profoundly affect the industry business environment—for example, by providing better value-chain integration and new-market exploitation, with competitive-advantage gains.

DX is driven by a flood of software technologies. Embedded electronics such as microdevices with sensors and actuators connected through the IoT facilitate ubiquity. Data analytics, cloud storage and services, convergent interactivity and cognition, augmented reality with visualization and simulation, pattern recognition, machine learning, and AI are facilitating a convergence of IT and embedded systems.[2,5] Underlying these, we've identified enabling methods, techniques, and tools, such as agile development for flexible systems, blockchains and Hyperledger to ensure security and trust in distributed transactions, and microservices and open APIs supporting software architectures.

Let's look at automotive technologies, where digitization is ramping up fast. A modern car incorporates 50 to 120 embedded microcontrollers and is connected over various external interfaces to a variety of cloud and infotainment technologies. Onboard software is in the range of hundreds of millions of lines of code (MLOC) and is still growing exponentially. Automotive software product lines and variants are some of the largest and most complex in industry. It's said that the automobile is rapidly becoming a "computer on wheels."

 Published by the IEEE Computer Society

Automotive original-equipment manufacturers (OEMs) are equipping next-generation production processes and vehicles with connected embedded sensors and actuators to obtain better intelligence and control. They adapt information and communication technology workflows from their IT systems to each car. Vertical integration is attained by ensuring that product-lifecycle-management systems, enterprise-resource-planning systems, production-planning-and-control systems, and manufacturing-execution systems work in coordination with capital goods on plant floors. Concerning horizontal integration, vehicle parts are delivered with RFID tags to guarantee production traceability.

TABLE 1. Digital transformation (DX) goals.

Perspective	Objective
Social	Foster the development of a more innovative and collaborative culture in industry and society.
	Change the education system to provide new skills and future orientation to persons so that they can achieve excellence in digital work and society.
	Create and maintain digital-communication infrastructures, and ensure their governance, accessibility, quality of service, and affordability.
	Strengthen digital-data protection, transparency, autonomy, and trust.
	Improve the accessibility and quality of digital services offered to the population.
Economic	Implement innovative business models.
	Increase income generation, productivity, and added value.
	Improve the regulatory framework and technical standards.

OEMs work with suppliers that have the same focus, to ensure that the acquired parts come with self- or distance-monitoring facilities. Examples include highly interconnected electronic control units from companies such as Bosch, Continental, Denso and ZF; mechatronic systems from Aptiv, Magna, Mahle, and Schaeffler; and head units and infotainment systems from companies such as Harman, Valeo, Panasonic, and Visteon. In factories, robots from ABB, Denso, Kuka, and Yaskawa assemble complete vehicles from parts with exact monitoring and logging of, for example, screw load torque to ensure compliance with production and safety standards. All software is individually configured for each car by modern IT systems, both in production and after sales with over-the-air upgrades. These movements toward a digital automobile world have already rationalized costs and investments. For example, according to David Powels, former CEO of Volkswagen's Latin American operations, in the three-year period ending in 2016, the group obtained 30 percent productivity gains in some factories, with a focus on digital process and competencies.[7]

Other industries are following fast toward DX. Vivo, a company in the Spanish Telefonica group, is adopting the agile-squad model and open innovation as the bases of its DX implementation. The company developed a social software robot called Vivi, which helps customers formulate requests. Ten million sessions have already been opened, and 94 percent of them have been solved in an automated way.

Hospital Samaritano and Hospital Sírio-Libanês, two leading São Paulo institutions, have consistently invested in DX to improve the patient experience and operational performance. Both keep integrated secure electronic health records of patients, which are used in procedures, treatments, prevention, and healthcare planning and decisions.

DX IMPACTS

DX has been a source of continuous entrepreneurship and business dynamism, particularly in technology-intensive industries. These companies have reorganized themselves to operate simultaneously in two distinct modes. The standard mode keeps traditional businesses and operations running, while a disruptive mode seeks additional opportunities to exploit new markets and innovate in technologies, processes, products, or services. Figure 1 illustrates that value is now created not only in traditional ways (the yellow arrows) but also through digitization (the green arrows).

Software technology today is both the driver and effect of disruption. The market leaders are ahead of their competitors because they develop and commercialize new technologies to address customers' future performance needs. However, these companies don't want to cannibalize their current cash cows. So, they're rarely in the forefront of commercializing

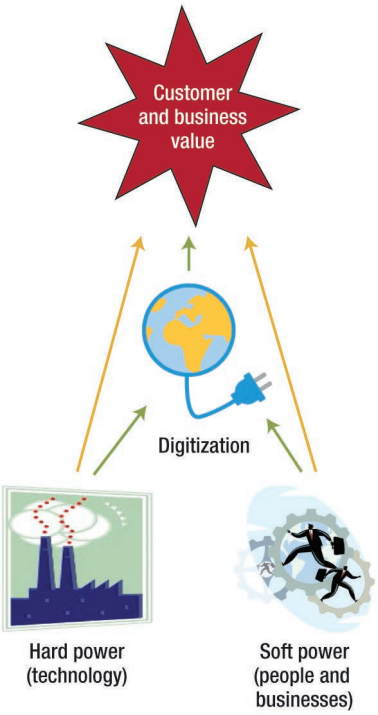


FIGURE 1. Digital transformation is a convergence of hard and soft forces and movements from which additional value emerges. Value is now created not only in traditional ways (the yellow arrows) but also through digitization (the green arrows).

new technologies that don't initially meet the needs of mainstream customers and that appeal to only small or emerging markets.

So, disruptive companies explore the occupation gaps left by the market leaders. This is a source of innovation and market change, which Clayton Christensen illustrated using price and performance data from the hard-disk-drive industry.[4]

We performed a systematic classification of the DX technology offerings; Table 2 presents some details. Although we haven't compiled quantitative evidence of the disruptions caused by the analyzed hardware technologies, Organisation for Economic Co-operation and Development studies have recognized that robots, 3D printing, and connected devices have disrupted productivity in their respective markets.[1,2,5]

Likewise, some of the software technologies we studied have been disruptive, but this is due only to their strategic significance, resulting from their penetration, adoption, and perceived value in distinct market segments vis-à-vis the initial markets. Finally, the remaining software technologies we studied can't be considered disruptive because they haven't obtained value recognition outside their initial markets (and therefore aren't included in Table 2). We structured our findings in the form of a knowledge map,[6,8] but here we presented just some branches in textual, pictorial, and tabular form. We hope this compilation will help software engineering (SE) practitioners and researchers develop and implement DX.

THE MUTUAL INFLUENCES OF DX AND SE

With software being key to any DX, mutual influences between DX and SE must exist. But DX disruptions due to SE innovations might emerge at any time and are almost impossible to predict. So, we can only speculate about the implications for SE.

To perform initial verification and validation of our ideas, we organized a debate panel during the 2016 IEEE Requirements Engineering Conference.[5] During the panel, researchers and practitioners discussed the impact and relationship of requirements engineering and DX in industry and research institutions. The panelists came from companies such as Intel (US), Nokia (Finland), Denso (Japan), Civic (China), and CI&T (Brazil).

The participants agreed that software technologies at the core of DX disruptions have been around for some time. These technologies have caused disruptions because of

- early or timely value delivery (agile methods),
- usage at larger scales (APIs, microservices, and IPv6),
- applications in new domains (3D modeling and printing, control software, and blockchains), and
- unpredicted technology combinations (cognitive computing, which combines computer vision, voice recognition, natural-language processing, and machine learning).

So, DX has not led to the development of radically new software technologies. Instead, it has given rise to new software technology applications, owing to the additional requirements that must be satisfied.

Technological solutions' complexity and scale have increased substantially, leading to software systems with many MLOC, usually binding together old and new, in-house and third-party developments. However,

TABLE 2. DX technologies.

Technology type	Inherent nature and attributes	Distribution and significance	Early-adopter experience	Adopted technology	Ease of adoption	No. of alternatives
					URLs	
Collaborative equipment (drones and robots)	Hardware capable of limited interactivity with moving parts and remote or embedded controls having typical sensor or actuator functions in heavy industry, space, or military applications	Adoption of cognitive computing expanded this technology's applicability from routine tasks to those requiring adaptability or autonomy, enabling its commercial use in precision agriculture, logistics, consumable-product industries, and services.	Alibaba manages retail warehouses in China using teams of unmanned shelf-carrying robots, which load and unload at multifunctional workstations.	Quicktron self-charging robots with QR code readers, laser or LIDAR anti-collision sensors, adaptive routing, and Wi-Fi connectivity with back-end software	Hard	Few
					http://translate.google.com/translate?js=n&sl=auto&tl=en&u=http://www.flashhold.com/page/16.htm	
Additive manufacturing and 3D printing	3D object creation from digital models, using printer heads driven by software-controlled stepper motors, for polymerization, jetting, extrusion, fusion, lamination, or deposition	Advances in image processing, precision mechanics, and new materials decreased the price of printers and printed objects, making them accessible to businesses and consumers for rapid prototyping and small-scale or customizable production.	BioArchitects supplies FDA-certified 3D-printed prostheses to customers in Brazil and the US, for training doctors and planning surgery procedures.	GE Arcam machines, which support additive high-power Electron Bean Melting production of titanium prostheses from CAD models generated using diagnostic-imaging exams	Hard	Very few
					http://www.arcam.com/products/arcam-q10	
IoT connected devices	Hardware with embedded digital electronics, software, and network connectivity enabling its unique identification, data collection, and data exchange	Implementation of IPv6 and reduced device costs enabled the massive dissemination of connected devices in machine-to-machine transactions and the IoT.	Volkswagen uses an IoT solution based on RFID tags to manage supply chain traceability in factories worldwide.	A Kathrein IoT distributed antenna system with customized software and standardized UHF RFID tags and transponders to ensure end-to-end order traceability	Hard	Some
					https://www.kathrein-solutions.com/solutions/logistics	
Agile development	A software development approach based on adaptive planning, evolutionary development, early delivery, and continuous improvement through collaboration of self-organizing cross-functional teams	The rapid-prototyping development approach evolved to widespread agile development owing to user involvement and rapid compliance with requirements, time-to-market reduction, and early value delivery.	Lloyds Bank adopted design thinking, agile methods, and a cloud-based microservice architecture to transform 10 customer journeys, which paid back in three years.	IBM Bluemix, a hybrid cloud platform-as-a-service architecture, used to support Scrum and a minimum-viable-product development methodology	Medium	Very many
					https://www.ibm.com/cloud-computing/bluemix	
Blockchain or Hyperledger	Continuously growing lists of decentralized information blocks, linked and secured through cryptography, used in recording financial transactions between parties efficiently, verifiably, and permanently	This technology has been disseminated to many other application domains that require secure fault-tolerant event record management, such as the arts, law, accounting, commerce, and healthcare.	A blockchain open source platform has been used to manage things ranging from World Food Program vouchers for Syrian refugees to a collaborative decentralized news network.	The Ethereum blockchain app platform, a decentralized framework with programmable virtual-machine and peer-to-peer protocols for defining and running distributed secure transactions	Medium	Some
					https://www.ethereum.org	
Open APIs and microservices	APIs and distributed services allowing system architectures to be structured in modular and open configurations	This technology's use in developing enterprise application ecosystems out of business functionalities, with decoupled deployment and operation, maximizes value for money.	Equinix Cloud Exchange provides cross-cloud application integration and scalable services by using an open API platform.	Google Apigee, a Java-based service platform to develop, deliver, manage, and analyze APIs via their proxies	Easy	Many
					http://www.apigee.com	
AI	A set of algorithmic tools for data analysis, representation, inference, deduction, and heuristics-based behavior	The coupling of AI to big data, cloud computing, natural-language processing, computer vision, and voice recognition enabled the scalable resolution of real problems in many application domains.	Telefonica launched its AURA AI service to help customers with any bureaucratic, communication, and interactive-content demand.	The Microsoft Bot Framework and LUIS, the respective IDE for creating and deploying software robots and natural-language-understanding integrated services	Easy	Many
					https://dev.botframework.com https://www.luis.ai/home	

the industry objectives for DX—an improved customer experience and operational excellence—have placed time to market, quality, and affordability at the forefront. So, practical DX problems have become tractable with software only with effective development management, reusability, and requirements-engineering methods, techniques, and tools.

The SE branches we've described have many interfaces, which deal with unproven metrics, hard complexity bottlenecks, and imprecise artifacts. The human factor is central to addressing these issues, but the required key competences for problem solving, managing complexity, and dealing with high abstraction levels are often lacking or insufficient.

DX requires software engineers to organize their work efficiently, act on their own initiative, have excellent communication skills, and successfully perform tasks involving emotion, intuition, creativity, judgment, trust, empathy, and ethics.[2]

At higher organizational levels, SE managers are expected to change their mind-sets and abandon command and control, moving to more leadership-oriented, risk-taking, and mistake-tolerant approaches. Corporate leaders need to motivate, direct, support, and inspire their autonomous teams, while learning along with them. They must be prepared to face business environments in which hyperawareness, informed decision making, and fast execution rule.

How can people obtain such skills? DX challenges traditional SE education systems to change their methods and content to a digitally transformed reality. Apart from the classroom and learning-by-doing approaches, continuous, just-in-time, and innovative learning methods—such as massive open online courses, gamification, and simulation—will be increasingly demanded.

To meet DX demands, SE will transform completely, leading to changes in how SE education treats human factors. In this new scenario, human resources will be extremely valuable, possibly becoming more important than the underlying technologies.

DX today is the megatrend across industries. However, DX is challenging because it demands a new set of competences, combining embedded-systems development with IT and cybersecurity. Software thus is the cornerstone of DX. In its convergence of classic IT with embedded-systems engineering, DX will completely reshape the landscape of software technologies and processes. (For more on DX and systems engineering, see the sidebar.)

With industry, home, healthcare, and automotive applications being major drivers, IT will converge with embedded systems such as the IoT and Industry 4.0. At the same time, embedded industries will evolve toward IT with cloud solutions and dynamic over-the-air upgrades. Critical industries such as the automotive industry involve practically all the quality requirements, such as safety, cybersecurity, usability, performance, and adaptability. The underlying software components cover anything from embedded real-time firmware to complex secured cloud solutions. Failure to meet any of those quality requirements results in expensive callback actions and legal lawsuits. These challenges will soon reach across industries.

DX is opening the doors for technology innovation, new business models, and cross-industry collaboration. The future is arriving while some are just running in their hamster wheels. Thus, we should be cautious, along the lines of what technology strategist Herman Kahn already observed several decades ago: "Everybody can learn from the past. Today it's important to learn from the future."[9]

ACKNOWLEDGMENTS

The assumptions, views, and opinions in this article are solely the authors' and don't necessarily reflect the official policy, strategy, or position of any Brazilian government entity.

REFERENCES

1. H. Demirkan, J.C. Spohrer, and J.J. Welser, "Digital Innovation and Strategic Transformation," *IEEE IT Professional*, vol. 18, no. 6, Nov. 2016, pp. 14–18.
2. C. Ebert, "Looking into the Future," *IEEE Software*, vol. 32, no. 6, 2015, pp. 92–97.
3. M. Gebhart, P. Giessler and S. Abeck, "Challenges of the Digital Transformation in Software Engineering," *Proc. 11th Int'l Conf. Software Eng. Advances (ICSEA 16)*, 2016, pp. 136–141.
4. C.M. Christensen, *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*, Harvard Business Rev. Press, 2013.
5. C. Ebert and C.H.C. Duarte, "Requirements Engineering for the Digital Transformation: An Industry Panel,"

DIGITAL TRANSFORMATION AND SYSTEMS ENGINEERING

Christof Ebert

At Vector Consulting Services, we've supported many companies in their journey toward digital transformation (DX), with both software technologies and the necessary agile business and development techniques. One key observation from projects is the fast-growing relevance of requirements engineering and systems engineering.

Requirements engineering, in its link with systems engineering, is the decisive success factor for the efficient development of hardware and software systems. Agile systems engineering supports the continuous development of requirements up to validation. It creates understanding of critical dependencies and provides methodological support to address growing complexity. Because systems engineering provides an increased number of abstraction levels and consistent handling of dependencies across a system, specifications become clearer, simpler, and less redundant. This not only increases development speed but also ensures clearly understood domain concepts in a project.

Systems engineering for DX includes

- cloud services for environment awareness, location-based services, online apps, remote diagnosis, continuous software updates, and emergency functions;
- cybersecurity, usability, and performance for modern software systems;
- service-oriented advanced OSs with secure communication platforms;
- machine learning and AI—for instance, in multisensor fusion, picture recognition, and data analytics for automated processes such as autonomous driving, medical-surgery support, and industry-scale predictive maintenance; and
- system-level modeling, testing, and simulation with models in the loop, and ensuring quality requirements such as safety.

As complexity and scale increase, quality must be ensured end-to-end at the system level. This demands professional methods and tools to ensure robustness, dependability, functional safety, cybersecurity, and usability. Security and robustness tremendously affect business models and potential liability. The more we share and network, the more we're exposed to attacks of all kinds. Usability is an interesting example of how quality factors are increasingly crucial. Across industries and applications, insufficient usability is a major source of hazards, operational failures, and critical failures.

Because quality, deadlines, and cost are pivotal across industries, the push for even better processes and project management is continuing at a fast pace. The demand for more agility and flexibility is rising. System users expect the same adaptive behaviors and continuous-delivery models that they have with their mobile devices. Rapid advances toward autonomous driving and open vehicle communication demand short-cycle recertification after over-the-air software updates. We've observed that only novel development paradigms such as service-oriented architecture can cope with the growing need for flexibility.

Proc. 24th IEEE Int'l Requirements Eng. Conf. (RE 16), 2016, pp. 4–5.

6. C.H.C. Duarte, "Patterns of Cooperative Technology Development and Transfer for Software-Engineering-in-the-Large," *Proc. 2nd Workshop Software Eng. Research and Industrial Practice (SER&IP 15)*, 2015, pp. 32–38.
7. C. Silva, "'Na crise, nossa produtividade cresceu 30%'" ("'In the crisis, our productivity grew 30%'"), *Estado de São Paulo*, 9 Dec. 2017 (in Portuguese); http://economia.estadao.com.br/noticias/negocios,na-crise-nossa-produtividade-cresceu-30,70001976321.
8. M.E. Fayad et al., *Software Patterns, Knowledge Maps, and Domain Analysis*, CRC Press, 2014.
9. H. Kahn and A.J. Weiner, *The Year 2000: A Framework for Speculation on the Next Thirty Years*, Macmillan, 1967.

CHRISTOF EBERT is the managing director of Vector Consulting Services. He is on the *IEEE Software* editorial board and teaches at the University of Stuttgart and the Sorbonne in Paris. Contact him at christof.ebert@vector.com.

CARLOS HENRIQUE C. DUARTE is a senior member of the technical staff at the National Bank of Social and Economic Development in Rio de Janeiro, and an IEEE Computer Society Distinguished Visitor. Contact him at cduarte@bndes.gov.br and carlos.duarte@computer.org.

EDITOR: Stephen J. Andriole, Villanova University School of Business, steve@andriole.com

This article originally appeared in
IT Professional
vol. 20, no. 3, 2018

COLUMN: LIFE IN THE C-SUITE

Governing and Piloting Emerging Technologies

Stephen J. Andriole, *Villanova University School of Business*

The governance of emerging technologies is different than the governance of operational technologies. No one governs plumbers the way they govern architects, which is not to derisively equate databases, enterprise resource systems, or communications networks with plumbing, but rather to equate architects with creativity.

C-suiters must first identify the emerging technologies that might disrupt their businesses. Then they must decide which ones to pilot to determine just how disruptive each might be to their business processes and whole business models. The governance of emerging technologies thus consists of "identify-and-pilot," rather than the traditional governance of "identify-deploy-manage."

EMERGING TECHNOLOGIES AND WHY THEY MATTER

There are at least 10 emerging technology clusters C-suiters should track (and possibly pilot).

Business and robotic process automation (B/RPA).

These are essential to identifying, describing, and improving the processes that run your business. If you don't know what your processes look like and how they behave, you cannot improve them by leveraging existing or emerging technologies. B/RPA is a blueprint for change—change is sometimes good and sometimes bad, but you must know the details of your processes and how they might be modified to make you more efficient and more competitive. BPA maps the key processes and RPA mimics the processes in simulated software functions. They are both enabled by automated tools that facilitate the description of existing processes and simulations of how modified processes can lead to more efficiency, lower costs, and digital transformation. You cannot improve what you cannot model or optimize. B/RPA is the first step to technology optimization.

Artificial intelligence (AI) and machine learning.

These technologies can shorten, improve, and replace routine and important processes and tasks, and can save you a lot of money by reducing headcount and improving throughput. The technology matters because it represents alternative methods, tools, and techniques for collecting, organizing, and analyzing data and information, and generating deductive and inductive knowledge. Put another way, AI can mimic—and improve—your best salespersons, underwriters, and brokers, among other professionals in your organization. AI also enables conversational speech, which means you will soon be able to augment intelligent bots, which you're probably already using. AI can also diagnose problems and suggest solutions in real time across vertical industries. The results of your BPA and RPA projects will suggest exactly where "intelligence" might be leveraged.

Blockchain.

This technology enables trusted, verifiable transactions without the cost or complexity of transactional intermediaries. According to the Forbes Technology Council, "blockchain is a public register in which transactions between two users belonging to the same network are stored in a secure, verifiable and permanent way. The data relating to the exchanges are saved inside cryptographic blocks, connected in

 Published by the IEEE Computer Society

a hierarchical manner to each other. This creates an endless chain of data blocks—hence the name blockchain—that allows you to trace and verify all the transactions you have ever made … in the case of Bitcoin, the blockchain serves to verify the exchange of cryptocurrency between two users, but it is only one of the many possible uses of this technological structure. In other sectors, the blockchain can certify the exchange of shares and stocks, operate as if it were a notary and 'validate' a contract or make the votes cast in online voting secure and impossible to alter."[1]

Cryptocurrency.

This is a virtual (digital) currency that relies on cryptography to make the currency safe and secure. Cryptocurrency is not issued by a central bank or a government that guarantees its authenticity. You likely already know about Bitcoin and probably Ethereum. You might also know that cryptocurrency is gaining transactional traction: you can buy houses with Bitcoin.[2] Cryptocurrency can also be used to execute transactions you might want to hide. Depending on the business you're in, cryptocurrency might already be an optional currency. Watch adoption rates closely. If they rise sharply, you might have to respond accordingly. That said, it's important to track methods, tools, and techniques that enable transactional autonomy.

Internet of Things (IoT).

There's value in connecting everything in your company (and supply chain) with sensors that can "talk" to everything else. The data, information, and knowledge collected and exchanged among active sensors in real time (if necessary) can be as diagnostic as you need. The IoT enables all sorts of applications like smart buildings, smart cities, smart power grids, engine monitoring and maintenance, and refrigerators that automatically fill your Instacarts. The key is two-way connectivity and the data exchanges enabled by this connectivity. If your company has opportunities to connect people, processes, devices, and components in real time with diagnostic information, then the IoT matters.

AUGMENTED ANALYTICS IS POTENTIALLY OF HUGE IMPORTANCE TO ALL ANALYTICS EFFORTS ACROSS ALL VERTICAL INDUSTRIES.

Augmented analytics.

This is analytics 2.0, where "regular" analytics (advanced statistical analysis focused on describing, explaining, predicting, and prescribing events and conditions) is augmented by AI tools, such as machine learning, natural language processing, and neural network modeling. This augmentation promises to accelerate descriptive, explanatory, predictive, and prescriptive analytics. It also enables the processing of massive databases and the production of heretofore undiscovered deduction and induction. Augmented analytics is potentially of huge importance to all analytics efforts across all vertical industries. If you're investing in analytics, you will find value in augmenting your investment with AI. If you're investing in both analytics and AI, then the offspring of the marriage should be obvious.

Virtual and augmented reality.

These technologies enable you to change the perspective and the experience of your customers and

all those who comprise your value chain. In simple terms, augmented reality (AR) facilitates the overlaying of aspects of the digital world onto the real world. Virtual reality (VR), on the other hand, simulates a completely virtual world. The opportunities for sales and marketing are extensive, especially if you're selling physical objects whose features can be enhanced with augmented reality or represented digitally in virtual reality. Remember that consumers of AR and VR are demographically available and that the form factor will eventually include conventional glasses, making the technology much more accessible to consumers.

Mobile everything.

If you understand value-chain demographics, you already know that mobile computing is ubiquitous. It will continue to grow as more and more data is transmitted across multiple devices, though most of the growth will be with smartphones (and future incarnations of today's smartphones). The exploding adoption of mobile applications is driving all this growth. Many business processes already exist on your customers' and suppliers' phones. It's safe to say that more and more of your business processes will be delivered and optimized on mobile devices.

Wearables.

Wearables come in various forms, including clothes, jewelry, shoes, wristwatches, and hats, not to mention all the fitness trackers out there. Some wearables are embedded under the skin of humans and animals. Many transmit data continuously. The possibilities here are endless. Any person, pet, or object can be "activated" with a wearable.

Cybersecurity and privacy.

First and foremost, you have no choice but to continuously and heavily invest in cybersecurity. Cybersecurity spans your networks, databases, and applications. Breaches are expensive and can severely damage your brand. Privacy is an expectation, though here demographics are on your side. Millennials are less concerned with privacy than their parents and grandparents. That said, C-suiters need to watch what happens with the rollout of the General Data Protection Regulation (GDPR) across Europe, which went into effect in May 2018.

PILOTING

C-suiters should select the technologies most likely to impact their businesses and industries and determine which ones to pilot. The first step is to conduct some industry intelligence to see what others in your industry are doing. Vendors are a rich source of information, as are professional technology forecasters like Gartner, Forrester Research, and the Cutter Consortium.

Executives should also structure pilots narrowly to determine the impact deployments might have on the targeted business processes and models. Vendors should be asked to sponsor pilots (with appropriate expectations about future work should the pilots succeed).

The first pilot should be a B/RPA project designed to map corporate processes and whole business models, which will identify the best opportunities for leveraging emerging technologies for corporate gain. After that, C-suiters can pilot the most promising technologies informed by rigorous process mapping.

REFERENCES

1. J. Giordani, "Blockchain—What Is It And What Is It For?," *Forbes*, blog, 28 March 2018 ; www.forbes.com/sites/forbestechcouncil/2018/03/28/blockchain-what-is-it-and-what-is-it-for/#57f3e49f1a16.
2. S. Mishkin, "You Can Buy This Stunning Southern California Lake House For 32 Bitcoins," *Money*, blog, 19 December 2017; http://time.com/money/5055888/real-estate-bitcoin.

STEPHEN J. ANDRIOLE is the Thomas G. Labrecque Professor of Business Technology in the Villanova School of Business at Villanova University, where he teaches courses in strategic technology and innovation and entrepreneurialism. Contact him at steve@andriole.com.

This article originally appeared in
IT Professional
vol. 21, no. 5, 2019

Big Data Solutions for Micro-, Small-, and Medium-Sized Enterprises in Developing Countries

Diana Rojas-Torres, *Universidad de La Sabana, Colombia*
Nir Kshetri, *University of North Carolina at Greensboro*

The use of big data solutions to make better and faster business decisions is no longer limited to large organizations. In recent years, micro-, small-, and medium-sized enterprises (MSMEs) in developing countries are increasingly benefiting from big data solutions (see Table 1). Such solutions have helped developing world-based MSMEs to improve business processes and market intelligence. Big data solutions have also helped them to increase access to financial services, such as loans, credit, and insurance.[1]

MSMEs make a significant contribution to national economic development in developing countries. The developing world is estimated to have 365–445 million formal and informal MSMEs (https://tinyurl.com/y5vffr3p). Formally registered MSMEs are estimated to contribute up to 45% of total job creation and 33% of gross domestic product in developing economies. These proportions would dramatically increase when informal MSMEs are included. MSMEs are likely to play especially important roles in reducing rural poverty among women and other disadvantaged groups.[2] Big data diffusion among MSMEs is, thus, likely to bring tremendous economic and social benefits to developing countries.

BIG DATA SOLUTIONS TO IMPROVE BUSINESS PROCESSES AND MARKET INTELLIGENCE

An important use of big data solutions has been in improving MSMEs' business processes and market intelligence. To take an example, Collective Intelligence Agriculture (CI-Agriculture), a subsidiary of Indonesia's big data analytic firm Mediatrac, has developed precision farming techniques for the Indonesian context. Small holder farmers can use the technique. Its Crop Accurate system uses data from diverse sources such as satellite, drone, and sensors for smart farming. The system analyzes soil condition, weather, and growth progress to give farmers advice regarding the best time to plant, fertilize, and use pest control. Farmers can make more efficient use of fertilizer and pesticides.[3] CI-Agriculture also learned about local farming practices and supply chains to develop the system.[4] The technology is scalable, which means that it is possible to use sensors for a large area. Agricultural data are collected and analyzed on a regular basis to predict crop yields. At the end of each season, smart farming system analyzes the data and provides recommendation to improve farming in the next season.

CI-Agriculture's another solution, Agritrack system, links farmers with supply chain partners such as distributors, market, and end customers.[5] Each party of the supply chain provides data via an app. Real-time information on key indicators, such as commodity prices, is provided, which can help predict prices and demand of farmers' produces.

To take another example, China's Alibaba has attracted vendors to its e-commerce websites Taobao Marketplace and Tmall.com by promoting big data-based advertising and other services. These solutions provide deep insights into shoppers' preferences.[6] Taobao has 666 million monthly active users (https://fortune.com/longform/ping-an-big-data) and over nine million vendors (https://www.azoyagroup.com/blog/index/view/chinas-new-e-commerce-law-bad-for-daigou-good-for-cross-border-e-commerce/).

DOI No. 10.1109/MITP.2019.2932236
Date of current version 11 September 2019.

TABLE 1. Some examples of big data solutions from MSMEs in developing countries.

Launched by	Big data solution	Key functions
Indonesia's Collective Intelligence Agriculture	Mobile App CI Agriculture	Give advice regarding the best time to plant, fertilize and use pest control based on the analysis of soil condition, weather, and other factors Predict prices and demand of crops Increase farmers' access to low cost loans and insurance
China's Alibaba	Ling Shou Tong for small physical stores	Help store owners make decisions related to product procurement and sales
	Big data-based advertising and other services to online vendors	Provide deep insights into shoppers' preferences
Kenya's FarmDrive	DigiFarm	Help smallholder farmers get low-cost loans combining their records revenues and expenses with other categories of information to generate credit scores

Alibaba has also developed a big data-based retail-management platform known as Ling Shou Tong for small physical stores in China. The solution aims to help store owners in making decisions related to product procurement and sales. In 2017, Alibaba started providing the platform to Chinese retail shops. The shops get the platform for "free" but they are required to use their storefronts as Alibaba's fulfillment-and-delivery centers. They also need to provide data on their customers' shopping habits and patterns (https://tinyurl.com/y5mpzgle).

BIG DATA SOLUTIONS TO IMPROVE ACCESS TO KEY RESOURCES

Big data solutions can also help to improve MSMEs' access to key resources. Especially access to credit is extremely difficult for smallholder farmers in the developing world. For instance, less than 1% of farmers in Kenya are reported to have access to formal credit.[7] To address this, the Kenya-based social enterprise FarmDrive's big data solution DigiFarm helps unbanked and underbanked smallholder farmers to receive credit. The process is simple. Smallholder farmers keep a record of their revenues and expenses. An app installed in the phone tracks these records. This information is combined with data generated from other sources such as satellite, agronomic data such as crop yields, pests and diseases, and local economic data.[8] In addition to agriculturally relevant data, DigiFarm also uses know your customer data to identify and verify the identity of the farmer as well as advanced behavioral analytics (https://tinyurl.com/y66awka5). The information is used to generate credit scores and assess their creditworthiness for loans (https://tinyurl.com/yyun3mes). Banks can use this information to provide loans to farmers and customize a farmer's payback timeline in order to match with harvests. As of 2018, over 200,000 farmers were using DigiFarm on a daily basis and 7,000 had successfully received loans to buy seeds, fertilizers, and pesticides (https://tinyurl.com/yyx4u8sz).

Likewise, big data solutions of Indonesia's CI Agriculture discussed above are expected to reduce loan costs for small-holder farmers. Data from satellite, drone, and sensors are used to calculate a field's production potential with a higher level of accuracy. These data can also be utilized to make more efficient use of fertilizer and pesticides.[3] CI-Agriculture provides insurance to farmers, which is based on calculations and schemes on smart farming technology, sensor systems, and analysis of other categories of data (https://tinyurl.com/yxrtgn2o). Insurance models are based on an analysis of weather data for up to 10 years.[4]

KEY CHALLENGES AND OPPORTUNITIES

MSMEs in developing countries have a number of options available to utilize big data solutions. MSMEs can benefit from open-source software such as Hadoop and Spark. For instance, Hadoop-based applications help MSMEs take advantage of real-time analytics from diverse sources and types of data. These include

data from external sources such as social media, machine generated data, as well as data from video, audio, email, and sensors. Many global technology companies such as Microsoft, IBM, EMC, Google, and Amazon Web Services provide Hadoop-as-a-Service (HaaS) to MSMEs in many developing countries. The HaaS providers help MSMEs in the management, analytics, and storage of data (https://tinyurl.com/y5vagmu7). MSMEs are increasingly adopting HaaS (https://tinyurl.com/y23hhj4r). These and other big data solutions targeted at MSMEs are getting more user-friendly. This is a rich community of users. Tutorials, tools, and other services are more easily accessible to MSMEs (https://tinyurl.com/yyun3mes).

MSMEs obviously experience a number of barriers in the adoption of big data solutions. Big data solutions provided by big companies are unaffordable and out-of-reach for many MSMEs in developing countries. For instance, it was reported that due primarily to Alibaba's high advertising rates, most vendors on Taobao were making losses.[9] According to an article published in the Chinese language newspaper Enterprise Observer in August 2013, over 80% of sellers on Taobao did not make a profit. It was also reported that thousands of shops on Taobao close down every day.[10]

In many cases the benefits to MSMEs of so-called free big data solutions provided by technology giants are not clear. The providers of such solutions tend to use them as a useful instrument to promote their own interests rather than those of MSMEs'. For instance, while the use of Ling Shou Tong may make it easier to run stores, many small stores worry about unfair competition from Alibaba's online marketplace, which has a huge selection of products to choose from. This means that these small stores' customers may decide to take advantage of the convenience of online shopping on Alibaba's online marketplace and pick up the products from these stores (https://tinyurl.com/y5mpzgle). Furthermore, Alibaba is in a position to make a better utilization of data on these stores' customers that these stores are required to provide.

Effective utilization of big data requires organizational capability to handle cooperation across different units and departments. Organizations in many developing countries may lack capabilities to organize and manage such multidisciplinary teams. A further challenge is MSMEs' lack of human resources to utilize big data effectively. Indeed, even large enterprises face such challenges in developing countries. Especially there has been a severe lack of big data manpower with high-level strategic thinking capabilities in developing countries. For instance, compared to many other developing countries, China has a rich endowment of big data human resources, thanks to an abundant supply of engineers. The country, however, lacks experts at the executive level (https://tinyurl.com/y5ayxkwl). Likewise, the lack of strategic leadership and the lack of idea of where to start the implementation of solutions are noted as a main reason why Colombian companies have not taken advantage of big data.[11]

More broadly, the big data labor market in developing economies faces challenges on two fronts.[5] First, there is a severe lack of engineers and scientists in order to perform analytics. Second, many analytics consultants lack skills and capabilities to understand, interpret, and put the data to work. Some estimates suggest that India would experience a shortage of 1 million data consultants (http://www.techrepublic.com/article/indias-high-demand-for-big-data-workers-contrasts-with-scarcity-of-skilled-talent/).

SUMMARY

The reliance on big data to make better and faster decisions is, thus, no longer limited to large companies. While only a tiny fraction of MSMEs in the developing world are currently taking advantage of big data solution, such solutions are getting popular among these enterprises. They increasingly depend upon big data. Data-driven decisions are gradually becoming the norm among these enterprises.

There are growing and encouraging signs of big data's positive impacts on MSME in the developing world. Big data-based innovations such as low-cost crop insurance and low-cost loans have benefitted micro enterprises and small holder farmers in the developing world. In addition to access to these strategically valuable resources such as finance and insurance, big data-based solutions have also increased the quality of their entrepreneurial activities with improved business processes and market intelligence.

Since MSMEs are critical for job creation and economic growth, it is important to deploy policy measures to facilitate the adoption of big data by

these enterprises. For instance, broad national policy measures directed toward increasing competition in the big data industry may lead to the availability of affordable solutions to MSMEs. This can be done by attracting foreign big data companies and facilitating the growth of local companies in this sector.

REFERENCES

1. N. Kshetri, et al., *Big Data and Cloud Computing for Development: Lessons from Key Industries and Economies in the Global South*. New York, NY, USA: Routledge, 2017.
2. "SMEs and SDGs: Challenges and opportunities, "Apr. 23, 2019. [Online]. Available: https://oecd-development-matters.org/2019/04/23/smes-and-sdgs-challenges-and-opportunities/
3. F. Kuo, "Can technology save ASEAN's food supplies from climate change?" *The Interpreter*, Blog. 2019. [Online]. Available: http://www.lowyinterpreter.org/the-interpreter/can-technology-save-asean-s-food-supplies-climate-change
4. N. Freischlad, "Drones over the rice paddy: Ci-Agriculture brings smart tech to the field," *Techinasia*, Blog. 2015. [Online]. Available: https://www.techinasia.com/ci-agriculture-precision-farming-indonesia
5. N. Kshetri, *Big Data's Big Potential in Developing Economies: Impact on Agriculture, Health and Environmental Security*. Wallingford, U.K., Centre for Agriculture and Biosciences International (CABI) Publishing, 2016.
6. L. Lorenzetti, "Alibaba's first public earnings reveal major revenue growth," *Fortune*, Blog. 2014. [Online]. Available: https://fortune.com/2014/11/04/alibabas-first-public-earnings-reveal-major-revenue-growth/
7. S. Maina, "Safaricom's DigiFarm aims to put more coins in farmers' pockets through technology,"*Techweez*, Blog. 2018. [Online]. Available: https://techweez.com/2018/07/23/safaricom-digifarm-more-coins-farmers/
8. L. Burwood, "FarmDrive raises funding to help Africa's smallholder farmers get finance with credit scoring algorithm,"*Agfundernews*, Blog. 2017. [Online]. Available: https://agfundernews.com/farmdrive-raises-funding-to-help-africas-smallholder-farmers-get-finance-with-credit-scoring-algorithm.html
9. C. Clover, "Alibaba has almost single-handedly brought ecommerce to China,"*Financial Times*, Blog. 2014. [Online]. Available: https://www.ft.com/content/11022ce8-a61a-11e3-8a2a-00144feab7de#axzz3V2tjJstS
10. R. Lu, "Tea leaf nation: For Alibaba's small business army, a narrowing path,"*Foreign Policy*, Blog. 2014. [Online]. Available: https://foreignpolicy.com/2014/09/10/for-alibabas-small-business-army-a-narrowing-path/
11. B. Mario Augusto, "En Colombia apuestan por una mejor utilización del big data,"Feb. 1, 2016. [Online]. Available: http://www.cioal.com/2016/02/01/en-colombia-apuestan-por-una-mejor-utilizacion-del-big-data/

DIANA ROJAS-TORRES is an Assistant Professor of entrepreneurship and innovation with the International School of Economic and Administrative Sciences, Universidad De La Sabana, Chia, Colombia. Contact her at diana.rojas7@unisabana.edu.co.

NIR KSHETRI is a Professor of management with the Bryan School of Business and Economics, University of North Carolina at Greensboro, Greensboro, NC, USA. Contact him at nbkshetr@uncg.edu.

EDITOR: Yong Cui, cuiyong@tsinghua.edu.cn

COLUMN: STANDARDS

This article originally appeared in
IEEE Internet Computing
vol. 22, no. 3, 2018

Analytics without Tears or Is There a Way for Data to Be Anonymized and Yet Still Useful?

Jon Crowcroft, *University of Cambridge and The Alan Turing Institute*
Adrià Gascón, *Warwick University and The Alan Turing Institute*

In this article, we discuss the new requirements for standards for policy and mechanism to retain privacy when analyzing users' data. More and more information is gathered about all of us, and used for a variety of reasonable commercial goals—recommendations, targetted advertising, optimising product reliability or service delivery: the list goes on and on. However, the risks of leakage or misuse also grow. Recent years have seen the development of a number of tools and techniques limit these risks, ranging from improved security for processing systems, through to control over what is disclosed in the results. Most of these tools and techniques will require agreements on when and how they are used and how they inter-operate.

HISTORICAL AND TECHNICAL CONTEXT

We have always kept data about ourselves—maybe household accounts to divvy up the food bill among students in a shared flat each month, or maybe our baby's weight and height. It's often easier to let someone else look after that data, a bank or our doctor for example, since they can use it for our benefit and keep it safe. They then store many peoples' records, and need a way to find our particular one, via some primary key, an identifier that uniquely fishes out our information—perhaps a mix of our name, birthday and postcode.

The two-sided market of cloud analytics emerged almost accidentally, initially from click-through associated with users' response to search results, and then adopted by many other services, whether webmail or social media. The perception of the user is a free service (storage and tools for photos, video, social media, etc.) with a high-level of personalization. The value to the provider is untrammeled access to the users' data over space and time, allowing upfront income from the ability to run recommenders and targeted adverts, to background market research about who is interested in what information, goods, and services, when and where. User data might be valuable in many contexts, especially when aggregated from several sources. This created a market for data, making suitable for the Internet a point made in the 70s regarding television[1]: "If you are not paying for the product, you are the product."

In this context, we've experienced a shift in how decision-making processes are approached by enterprises and governments: crucial decisions in areas as diverse as policy making, medicine, law enforcement, banking, and the workplace are informed to a great extent by data analysis. For example, decisions related to which advertisements and promotions we see online, which of our incoming emails are discarded, what type of TV series are produced, what conditions are attached to our insurance, or what new drugs are developed, are all informed to a great extent by the analysis of electronic data.

This trend is far from stabilizing. As digital surveillance grows apace, the advent of personal data gleaned not just from social media and online services but from sensors in smart homes, cars, cities, and health devices, becomes more and more intrusive. Something has to give.

There are both technical and socio-economic reasons why this has to change, and this has been recognized amongst regulators and in industry. Privacy failures risk personal and corporate wealth and safety. Theft of credit card data, identity and trade secrets is a real and present danger, increasing with the extended "attack surface" presented in the surveillance society. The volume of detail available admits of inference about people and institutes in ways not recognized or necessarily intended. In some cases, this even can lead to threats to personal safety. As a result, new laws and new technology have been proposed and enacted, which might go some way toward alleviating this. However, the roadmap is quite complex and involves choices, as well as agreements between parties. Some of the technical choices will have implications for standards, which may in turn reflect back on how regulation and legislation will evolve. One thing appears clear, that organizations are keen to retain the value of all that big data now being gathered and analyzed, whether for entertainment, health, security, or profit. Hence there is a need for careful harmonization among the new regulations and the new technology that can support the old value chains.

The General Data Protection Regulation (GDPR) and the ongoing e-privacy regulation effort are significant steps in regulating the protection of sensitive information by placing obligations on data controllers and data processors, as well as specifying user's rights. However, no specific algorithms are mentioned, and hence we are far from effective standardization guidelines. This is justified, as the technology is not quite there, and more research is needed both from a theoretical and applied perspective. However, its potential has been recognized both in industry and government. The recent report by the US Commission on Evidence-based Policymaking[2] describes differential privacy and multi-party computation as emerging technologies and states, "New privacy-protective techniques [...] may allow individuals in the Federal evidence-building community to combine data and conduct analysis without directly accessing or storing information."

In general, similarly to general security, privacy has proven to be a slippery concept, that requires a robust, mathematically rigorous approach. Nevertheless, very promising advances have been made in the last decade, both from a practical and a theoretical perspective. In this scenario, privacy-preserving analytics has emerged as a very active research topic simultaneously in several fields such as machine learning, databases, cryptography, hardware systems, and statistics. The main challenges include several applications currently being simultaneously pursued within research communities in these fields, such as finding secure ways of providing public access to private datasets, securely decentralizing services that rely on private data from individuals, enabling joint analyses on private data held by several organizations, and securely outsourcing computations on private data.

There are several alternative directions that this will evolve in the future.

ASPECTS OF PRIVACY-PRESERVING ANALYTICS: COMPUTATION AND DISCLOSURE

The general goal of research into privacy-preserving data analysis is to develop techniques that get the best utility out of a dataset without violating the privacy of the individuals represented in it. However, there are several interpretations of what we may mean by privacy in this context.

First of all one has to realize that if you belong to a certain population, and an analysis on that population is disclosed, then your privacy has been breached and there is nothing you can do—or could have done—about it. For example, assume that a predictive model about mobility in London is made public. Let's say that the model is able to accurately predict the location of London tube users, given some of their characteristics.

Regardless of whether that model was trained with their data or not, the privacy of all London tube users is breached to some extent. This point might sound obvious, but it is important: technical advances in general do not solve all ethical issues, and privacy is not an exception to that. Every data analysis has some ethical issues regarding privacy associated with it, which must be approached as such.

However, there are many crucial privacy issues in data analysis that technology can help to over-come. Two aspects for which we have in principle satisfactory technical solutions are privacy of stored data, i.e., encryption of data at rest (on disk), and privacy of data as it is being transmitted, i.e., encryption of data in transit. Current basic research challenges have to do with preserving privacy even during processing, and generally correspond to two orthogonal but tightly-related aspects: privacy-preserving computation and privacy-preserving disclosure.

Privacy-preserving computing: The result and nothing but the result!

Let's say you upload your data encrypted to the cloud, but still allow for some concrete computations to be performed on it by service providers, such as training machine learning models, or selecting ads tailored for you. This would certainly keep the service providers happy, while protecting your private data from data breaches.

So now how do we execute software on machines owned and maintained by an untrusted party? Or, more generally, how do we compute on private data held by mutually untrusted parties? There are several emerging techniques to do this that could be combined in principle, and come from the areas of hardware security and cryptography.

Secure enclaves.

The idea behind secure enclaves is based on new technology (not so new on the iPhone but new to servers) called a Trusted Execution Environment.[3] Such trusted hardware provides a secure container into which the secure cloud user can upload encrypted private data, securely decrypt it, and compute on it. Both the decryption and the computation are run in a processor, which, in principle, not even its owner can break into. The result is again securely transmitted to the user, together with a proof that it is indeed the result of the intended computation.

It is important to remark that this approach relies on trusted hardware, which is in general hard to patch if vulnerabilities are found. Moreover, there are some limitations to its security guarantees, as it does not protect against cache-timing and physical attacks, as well as limitations in terms of scalability because the amount of available RAM within a container is often limited.

Examples of this technology are Intel's SGX and ARM TrustZone, which are evolving and being adopted quickly. A recent instance of such adoption are the Azure Confidential Computing capabilities.

Homomorphic encryption.

An encryption scheme is said to be homomorphic with respect to a given operation if one can perform that operation on the encrypted data by just manipulating the corresponding ciphertext. For example, if an encryption scheme is homomorphic with respect to addition, two encryptions of arbitrary values, say 23 and 19, can be combined—without prior decryption—to produce the encryption of their sum. Asymmetric key encryption schemes that are homomorphic with respect to either addition, e.g., Paillier, or multiplication, e.g. ElGamal, have been known for a while, but it wasn't until 2009 that Gentry described the first fully homomorphic encryption scheme,[4] namely a scheme that's homomorphic with respect to both addition and multiplication. Note that, if we operate on a binary domain, i.e., mod 2, addition and multiplication is all that one needs to do anything a modern processor can do. This enables secure outsourced computation relying solely on encryption, as opposed to the secure enclave approach, as a user can encrypt all their data and share it with the cloud encrypted, together with the public key of the encryption and a description of the computation. Then the cloud provider can compute on it in encrypted form—as if it was computing blindfolded—and return the encrypted result.

Fully homomorphic encryption is a remarkable breakthrough, as before Gentry's contribution, it was not even clear whether such kind of encryption could even exist. However, although several alternative improved schemes have been proposed since Gentry's, homomorphic encryption is currently far from

scaling to the secure cloud computing application, and in particular, data analysis tasks involving massive input sizes. Nevertheless, several homomorphic encryption libraries are available, and a limited notion of fully homomorphic encryption supporting a fixed number of nested multiplications called somewhat homomorphic encryption might be enough for some data science applications.

Multiparty Computation (MPC).
Another alternative is to use secure Multiparty Computation (MPC), an area of cryptography kicked off by Andrew Yao in the 80s.[5] There are a number of protocols, including the lovely Yao's garbled circuits, that revolve around the idea of sharing secrets without actually giving them away, and then computing on them by transforming their shares, and still keeping them secret. An example is the way to find out who is the richest person in the room, without revealing how much each person actually possesses. These are hard to reason about for the layperson, but can be verified in design, and probably therefore are a promising additional technique. Moreover, MPC techniques are quite efficient and, due to a sequence of theoretical and engineering breakthroughs, have become of practical interest, with many available libraries and applications, and even commercial products.

MPC technologies allow for moving away from the trusted aggregator model for analysis of distributed data. Instead of moving all the data to a single server (where it might be leaked), we can leave data in peoples' devices (smart homes, smart TVs, cars, IoT devices, tablets, etc,) and distribute the programs that do the analytics in a privacy-preserving way. This then moves the results (e.g., market segment statistics) to businesses that wish to exploit them without ever moving the raw personal data anywhere at all. Hence, the parties interested in an aggregated model learn such a model, and nothing but the model. In principle, this permits reversing the business models' direction of value—the subject (user) can now charge for their data! In the distributed approach, since there is no central data center/cloud anymore, there's no need to cover its cost, so the change adds up.

The caveat here is that while MPC techniques allows keeping the data with the parties owning it, such parties must get involved in the computation, hence incurring some computation and communication cost. This is in contrast with secure cloud approaches, where the encrypted data only has to be uploaded once. This motivates architectures that include sets of noncolluding untrusted parties that are used to simulate a secure cloud. It is important to remark that MPC techniques provide high-assurance cryptographic guarantees.

Edge computing.
There are performance advantages to edge computing in some new scenarios, especially in the smart home and Internet-of-Things use cases. At the least, we can remove the burden of sending large amounts of detailed data from very large numbers of edge devices into the cloud. Instead, we retain the data in local hubs (e.g., smart home hubs), and send analytics software to execute there, rather than on the central cloud. This is recognized in the IoT hub work by Microsoft and in several other IoT platforms. We still need to retain all the same approaches to supporting privacy concerning the data, but the very distributedness of the data and computation reduces the risk of a mass-leak of information because the attack-surface of the whole system is now fragmented. Techniques for decentralized analytics work quite well and can adopt many of the techniques used for large-scale analytics in data centers. Moreover, one can in principle enhance such approaches with the cryptographic techniques mentioned above to yield high-assurance guarantees.

Tailored approaches.
For a concrete problem, for example, logistic regression on distributed data, custom "hybrid" protocols that combine several of the techniques above are likely to give the best results by sacrificing generality. Research prototypes that follow this hybrid approach have been proposed for private training and classification in models such as neural networks, ridge and logistic regression, nearest neighbors, text classification, and random forests, among others.

Privacy-preserving disclosure: How much does the result actually disclose?

Although the techniques above can be used to compute a statistical model in a privacy-preserving way,

namely not disclosing any unnecessary information, they do not address the problem of quantifying how much is disclosed by such a model. This (vague) question regarding "how much is disclosed" has many aspects. For example, one might be interested in quantifying to what extent a sensitive feature of the training dataset is disclosed by the model. Alternatively, one could try to address whether the model would allow deanonymizing a public, in principle, unrelated dataset, or whether a given individual can be identified as part of the training dataset. Each of these goals captures something about our intuition regarding privacy and they may be more or less suitable in different contexts. As with issues such as bias and fairness in statistical models, mathematical definitions of privacy are important even if they only capture part of what we intuitively mean by privacy preservation.

One thing is clear among information security experts: simply removing the primary keys (names, birthday, postcode, etc.) of a database and replacing with some pseudo-random numbers, so called "de-identification," won't work in general. There are too many diverse holders of records to prevent trivial re-identification (sometimes called triangulation) by linking data from different sources and inferring who the subject is. Another defense, often referred to as *k*-anonymization, consists on "fuzzing" the dataset so that any allowed query includes data from at least *k* individuals, hence providing some uncertainty that should protect privacy. However this also does not account for the above mentioned linkage attacks. In summary, what makes privacy difficult is dimensionality: a sometimes surprisingly small number of features is enough to make a database record essentially unique. Hence, an attacker with a bit of background knowledge about a given individual can use it to obtain the additional information about that person present in a database.

A particularly successful mathematical definition of privacy, as it is receiving lots of attention from both academia and industry, is Differential Privacy (DP). Intuitively, DP allows us to design analyses in a way that quantifies how much they give away in terms of whether a record was part of the database or not. As Dwork and Roth put it,[6]

> *"Differential privacy describes a promise, made by a data holder, or curator, to a data subject: You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources are available."*

How it is done is a detail that varies across applications, and can involve several approaches to filtering data collected, or fuzzing features of the data, or analyzing and blocking overly intrusive or too-frequent questions. Given that most market research style analytics is concerned with identifying groups (segments or bins) in the data, this may not lose any value at all. Users with really obscure or rare features don't represent significant market opportunities.

There are several aspects that make DP an appealing definition. First of all, DP neutralizes linkage attacks, as it is defined as a property of the analysis, not the data on which the analysis is run. For the same reason, DP is also immune to post-processing: running subsequent analysis on the result of a differentially private analysis cannot result in a less private result. DP also has nice composition properties, that allow the building of provably private analysis from simpler building blocks. Finally, one of the main characteristics of DP is that it allows for privacy quantification, as it defines privacy in terms of real-valued parameters ε and δ. Setting these parameters properly is in general an open problem that corresponds to the tension between privacy and utility in data analysis: smaller values of the parameters provide better privacy, but might render the analysis useless.

To illustrate the ideas behind DP to provide privacy by a randomized mechanism, it is useful to consider the very related idea of randomized response. Randomized response is a technique developed in the 60s to collect statistics about illegal or embarrassing behavior. Every participant of the study is instructed to (a) (privately) flip a coin before replying to the question, if the coin comes up heads then (b) answer truthfully, and if the toss comes up tails (c) randomly answer "yes" or "no," using a second coin toss. The surveyor can then correct the result using their knowledge about the surveying mechanism to get an approximate count. Note that privacy for the participants here comes in the form of plausible deniability: they can always claim "I did not vote for Brexit!, the coin tosses made me report yes."

DP has been shown to be a very rich concept with interesting connections to several aspects of information theory and learning. There have also been some applications of it by big data controllers such as Google and Apple, but a clear path to standardization does not yet exist. The main challenges have to do with modelling and parameterization choices, to which DP is very sensitive not only in terms of privacy but also utility.

TOWARD STANDARDS FOR LARGE SCALE DATA ANALYTICS

The technologies mentioned above and more importantly, their interplays, are not mature enough to be completely standardized. Moreover, the complexity of secure data analysis will require several kinds of standards, related not only to the different aspects of privacy-preserving analytics discussed above, but also related issues like personal data management and consent. First, just like there are standard virtualization APIs, we need standard APIs for trusted execution. Moreover, one has to address choices regarding cryptographic protocols, the architectures on which they are deployed (possibly involving semi-trusted parties), and with which security guarantees in terms of key sizes and similar parameters. Even if we agree on which protocols to use and how to instantiate them, there are always a set of services that are required to deploy such protocols in practice, and a reasonable incentive system for parties to provide such services must be in place. This includes tasks such as key distribution, attestation, and verification, which might potentially involve actors focused on these tasks. A major challenge to overcome is that of "how much privacy is enough?" Privacy, unlike secrecy or security, is in some cases not a binary predicate, as it undermines utility in many applications. For example, establishing a "safe" differential privacy modelling and parameters for recurrent analysis on sensitive census data is a major challenge. What one means by "safe" would have to be not only rigorously established, but also effectively communicated by, for example, something like kitemarks for safety, but instead for privacy level.

Privacy-preserving data analysis is an emerging discipline within data science, which posts several challenges currently being simultaneously tackled from several areas such as hardware/systems security, cryptography, statistics, and machine learning. Several privacy-enhancing techniques have evolved significantly in the last decade from being mainly theoretical to becoming academic prototypes and even commercial products and, as recognized by both governments and industry, have the potential to revolutionize the field. These techniques have different tradeoffs, maturity levels, and privacy guarantees, and in some cases solve slightly different problems. A fully fledged approach to privacy-preserving data analysis would still require significant interdisciplinary effort, some of which have to do with issues such as effective personal data management and consent, which we did not address in this paper.

The need for robust privacy-preserving data analysis technologies has been recognized by both regulators and industry. This would not only mitigate the growing risks of privacy failures, but also enable opportunities based on computing on private data. This is analogous to how encryption revolutionized secure communications, enabling a huge economic development, mainly through secure payments. While regulation and standardization would apparently accelerate this process, the technology is not quite there, and more research is needed before the field as a whole is mature enough to yield precisely defined good practices and regulation, capable of, for example, enabling audits to ensure compliance.

REFERENCES

1. R. Serra, Television Delivers People, 1973; https://www.moma.org/collection/works/118185.
2. *Report of the Commission on Evidence-based Policymaking*, September 2017; https://www.cep.gov/content/dam/cep/report/cep-final-report.pdf.
3. V. Costan and S. Devadas, "Intel SGX explained," *IACR Cryptology ePrint Archive*, 2016; https://eprint.iacr.org/2016/086.pdf.
4. C. Gentry, "Fully homomorphic encryption using ideal lattices," *Proceedings of the forty-first annual ACM symposium on Theory of computing (STOC 09)*, 2009, pp. 169–178.
5. A.C. Yao, "How to generate and exchange secrets," *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, 1986, pp. 162–167.
6. C. Dwork and A. Roth, "The algorithmic foundations of

differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, 2014, pp. 211–407.

JON CROWCROFT is a professor at the University of Cambridge and a Fellow at The Alan Turing Institute. He received a PhD in Computer Science from UCL. His research interests include Internet Protocols, Cloud Computing, Social Media Analytics and Privacy Technologies . Contact him at jon .crowcroft@cl.cam.ac.uk.

ADRIÀ GASCÓN a Turing Senior Research Fellow at Warwick University and a Research Fellow at The Alan Turing Institute. He received a PhD in Computer Science from the Polytechnic University of Catalonia (UPC). His research interests include program synthesis, distributed algorithms, applied cryptography, and machine learning. Contact him at agascon@turing.ac.uk.

Contact editor Yong Cui at cuiyong@tsinghua.edu.cn.

This article originally appeared in
IEEE SECURITY & PRIVACY
vol. 17, no. 5, 2019

It Takes a Village to Secure Cellular Networks

Elisa Bertino, *Purdue University*

The 4GLTE technology has tremendously increased the bandwidth available for smartphones, essentially delivering broadband capacity to them. The most recent 5G technology is poised to further enhance transmission capacity and latency through the use of the unused radio spectrum bands (e.g., millimeter waves) and devices, such as massive multiple-input, multiple-output antennas. 5G will enable a host of new applications and permit the massive deployment of the Internet of Things systems.

Cellular networks are and will increasingly be one of the most critical infrastructures, and their security is obviously crucial. However, securing cellular networks is a challenging task. These networks consist of multiple layers: physical, radio resource control, nonaccess stratum, and so on. Each layer, in turn, has its own protocols, such as those for attaching/detaching devices to/from the network and paging devices for notifications of incoming calls and SMS.

Research in recent years has succeeded in identifying a few security and privacy vulnerabilities. However, we are far from having systematic and comprehensive approaches to identifying such vulnerabilities. In some cases, past work has relied on network traces, in other cases, on cryptographic verifiers, and in still other cases, on researchers' intuition. As a result, past analyses are limited.

More recently, systematic methods have emerged, such as those combining cryptographic verifiers and model checkers. However, the use of such methods is far from trivial. The first challenge is that, to use those methods, we must encode the behaviors of the parties involved in the protocol into finite-state machines, which must be at a suitable abstraction level. The second is that the verification of the protocols is often performed with respect to properties of interest, which requires considerable domain knowledge. Extracting finite-state machines from standardization documents is challenging. These documents are written in natural language and, thus, have ambiguities. In addition, they are lengthy, and their structure makes reading them a cumbersome task.

Of course, we cannot expect writers of standardization documents to produce finite-state machines out of their documents. Perhaps a community-based effort involving researchers from academia that aims to formalize those protocols and make these formalizations publicly available would be the way to go. The availability of such formal specifications and relevant properties would enable researchers and developers to systematically analyze cellular network protocols. It would allow the community to test the limits of currently available tools for formal methods and promote new research directions for enhanced approaches and, perhaps, of tools based on tools other than formal methods. These specifications would also be relevant for industries implementing those protocols in that, on the one hand, they would provide a more precise description, and, on the other hand, they would allow the comparison of protocol implementations against the standard.

Let's now talk about vulnerabilities. When we look at those so far identified, it is striking that most of them are due to the lack basic and well-known security practices. Notable examples include lack of integrity verification or lack of replay protection for certain messages and lack of authentication for certain broadcast messages. An important example of the latter is the messages periodically broadcast by base

DOI No. 10.1109/MSEC.2019.2922867
Date of publication: 3 September 2019

 Published by the IEEE Computer Society

stations to advertise their presence in corresponding geographical cells and provide parameters so that devices can connect through them. These messages are broadcast with high frequency, irrespective of any device's presence in the cell area: every 40 ms for the master_info_block message and every 80 ms for the system_info_block message. Because those messages are not digitally signed, the devices have no assurance that they are connecting to a legitimate base station. As a result, it is possible for malicious parties to spoof legitimate base stations.

At first glance, we may think that the deployment of well-known approaches, such as a public-key infrastructure-based authentication mechanism, would easily fix this problem. However, a closer analysis of how such an approach would work to authenticate those two types of broadcast messages shows that there are many different requirements for such a method. For example, sending certificates along with those messages would impose high communication overhead; bandwidth is a precious resource, and network providers may be reluctant to spend bandwidth for sending certificates.

Therefore, certificate size should be minimized as much as possible, with careful selection of which messages to sign. The overhead for signature generation and verification is also critical and needs to be minimized. Base stations need to reduce the time required for generating the signatures because of the high frequencies of those broadcast messages; approaches, such as signature aggregation, should be devised to be deployed at base stations and highly optimized.

Mobile devices, on the other hand, need to minimize the signature verification times, as saving energy is critical. This would require mobile-device manufacturers to come up with energy-efficient implementation of signature verifications. Additional constraints include revocation of certificates, backward compatibility, and certificate management for devices roaming across different providers (for example, when traveling abroad). The research community has developed many interesting and novel solutions to the problem of digital signatures. However, we need to test these solutions and the solutions to other security problems in the complex scenario of cellular networks that, especially in the case of 5G networks, are increasingly complex and have even more stringent real-time constraints. Testing those solutions and devising further requirements will require the engagement of network providers and device manufacturers. It will take a village to secure cellular networks!

ELISA BERTINO is a professor with Purdue University. Contact her at bertino@purdue.edu.

COLUMN: STANDARDS

This article originally appeared in IEEE Internet Computing vol. 23, no. 1, 2019

Improving Performance and Scalability of Next Generation Cellular Networks

Ali Mohammadkhan and K. K. Ramakrishnan, *University of California, Riverside*

Uma Chunduri and Kiran Makhijani, *Future Networks, Huawei Technologies*

The 5G cellular network's packet core architecture has adopted concepts of software-based networking to improve scale and flexibility. In this paper, we investigate potential improvements to the current architecture, the protocols for the 5G control plane and backhaul network to achieve signaling efficiencies, improve user experience, performance, scalability, and support low-latency communications.

5G networks promise to revolutionize cellular communications, with a substantial increase in per-user bandwidth and low latency through improvements in the wireless radio technology. 5G networks are being proposed as an alternative not only for traditional smart-phone based data and telephony applications but also for Internet-of-Things (IoT) and even for residential Internet service. While the use of improved radio technology will help tremendously, challenges remain because of the complexity of the cellular network protocols. Of particular concern is the complexity of the control plane protocol and the use of the GPRS tunneling protocol (GTP). Tunnels carry traffic between the end user equipment (UE) and the cellular packet core network. With the increased use of small cells (potentially more frequent handovers) and the need to support a large number of IoT devices (which switch between idle and active more frequently to save battery power), the need for efficiency of the control plane is even more important.

The 5G Core (5GC) consists of several different components that carry out individual tasks. When an event for a user (e.g., attach, handover, service request) occurs, a large number of messages are exchanged between these components for notification and synchronizing state. Consider, for example, an IoT device that conserves energy by quickly transitioning to an idle state, turning off the radio. A service request event (when the UE transitions from idle to active to exchange packets), requires between 13 and 32 messages (Figure 3).[1] This long sequence of messages introduces undesirable latency in initiating a data transfer after the idle period. The overhead (in messages exchanged) and latency may nullify the purpose and goal of transitioning to an idle state.

We suggest a careful re-examination of the 5G architecture and control plane protocol to improve performance. There are three aspects we explore (a) redesigning the control plane signaling protocol and 5GC system architecture, (b) an optimized traffic engineering (TE) path selection in the backhaul network, and (c) an enhanced programmable data plane. We begin with an overview of 5GC architecture, its control plane protocol and approaches to simplify them, thus reducing latency, improving efficiency, throughput, and scalability. Second, we propose a simplification of the backhaul network, which is usually treated as an opaque entity. We explore alternatives currently being considered in the Internet Engineering Task Force (IETF). Finally, a programmable data plane is discussed to enable additional network level functions necessary for 5G applications that require high reliability or low latency.

BACKGROUND

Conceptually, the 5G architecture follows the principles of the Control and User Plane Separation of

DOI No. 0.1109/MIC.2018.2884882
Date of current version 6 March 2019.

 Published by the IEEE Computer Society

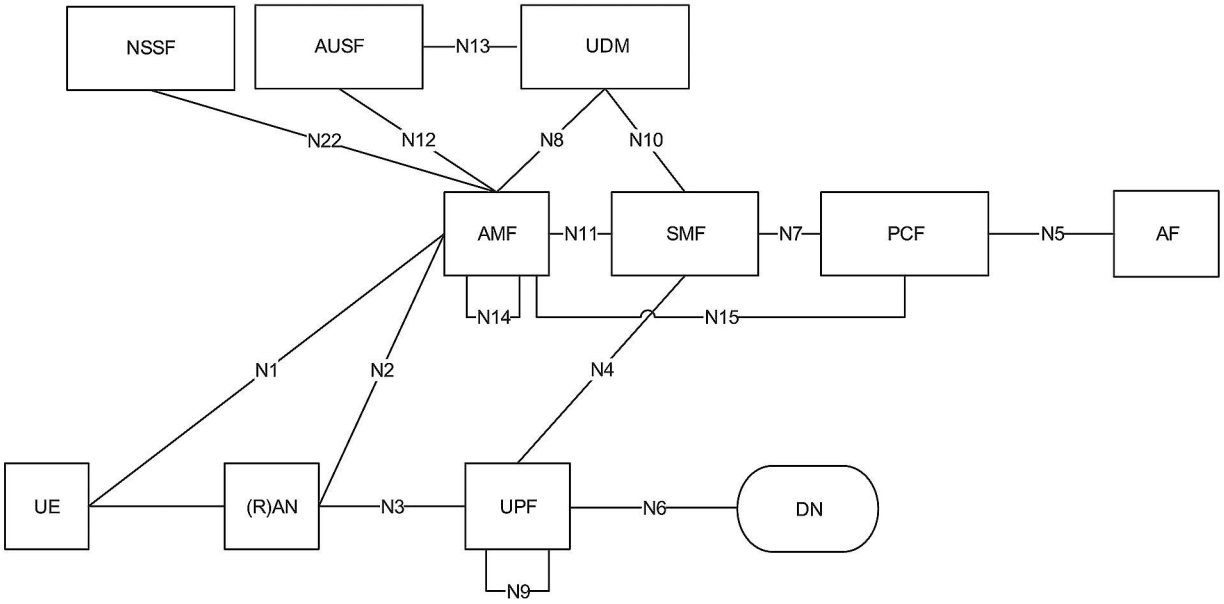


FIGURE 1. 5G system architecture.[3]

cellular core architecture introduced in Release 14 of the 3 GPP specification thus simplifying the functionality needed to be supported by each component.[2] However, the separation between the control and user plane components significantly increases the number of messages needed to coordinate a user session state across these components. There are five main entities, apart from the UE and cellular base station (called the gNB in 5G new radio) as shown in Figure 1. These are access and mobility management function (AMF), Service Management Function (SMF), authentication server function (AUSF), unified data management (UDM) and user plane function (UPF). The UPF is a data plane entity, while the others are control and management plane entities. The AMF is main control plane orchestrator, managing UE mobility, session establishment (through SMF), and handling service requests. Additionally, entities such as the NSSF, PCF, and AF also play important roles, the details of which can be found in 3 GPP TS 23.501.[3]

POTENTIAL ARCHITECTURAL ENHANCEMENTS

Traditionally, hardware components are purpose-built and customized for distinct functions. While the control plane requires the capability to handle complex processing and has more sophisticated capabilities involving compute nodes, the data plane needs to perform high-speed simple forwarding and is built with hardware accelerated forwarding engines. However, with the advent of virtualization, common off-the-shelf server (COTS) systems with a large number of processor cores, software libraries such as the "Data Plane Development Kit" and high-performance network interface cards, this separation of functionality is no longer necessary.[4] For example, a single server running the OpenNetVM platform can process and forward 10 s of millions of packets per second with software-based network functions (NFs) handling both complex control plane functions and high rate data plane workloads.[5] This has led the cellular industry to evolve into a software-based packet core (5GC) system architecture. However, the 5G architecture continues to emphasize the separation between the control plane and data plane as one of the goals even as software-based systems are able to elegantly support multiple classes of functions running on the same system.[6] The main requirement of efficiency and high performance can, in fact, be achieved by having the 5G control and data plane functions coresident on the same COTS system. Coresident NFs can share state information more easily, and where possible take advantage of shared packet processing. Use of software-based NFs should be viewed as an

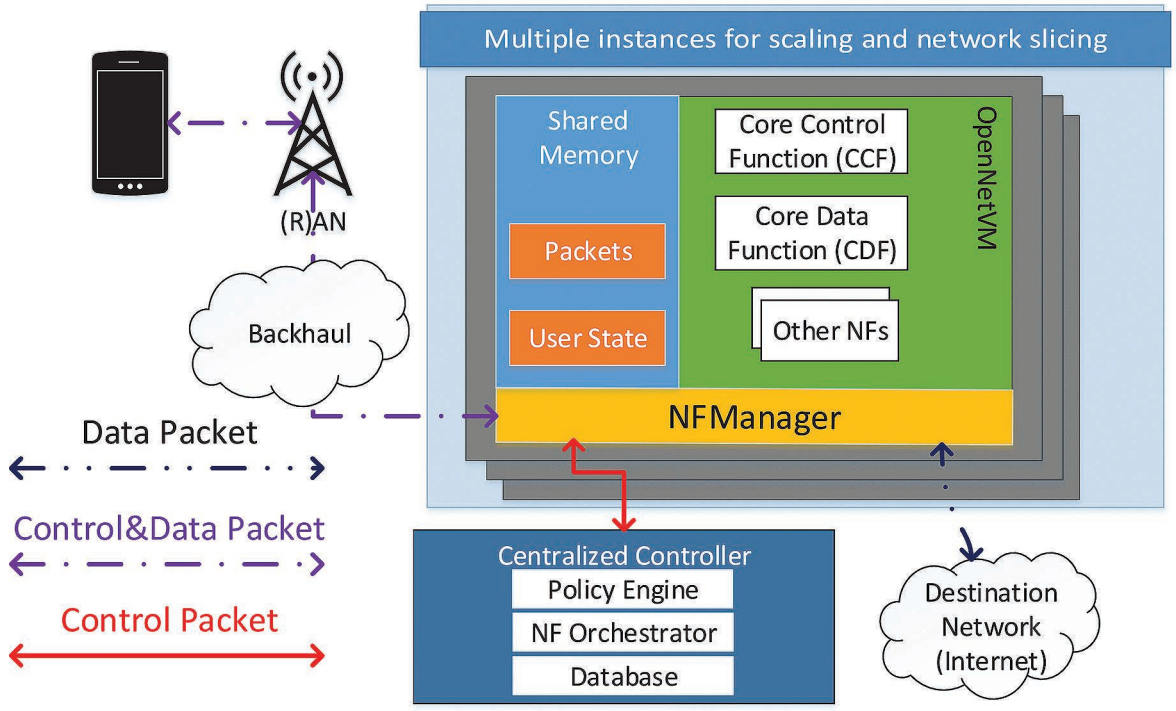


FIGURE 2. Example architecture using software-based NFV for implementing core control and data functions.

opportunity to carefully take stock of how the functional architecture of the 5GC should be implemented. By doing so, we see the potential for reidentifying the functional components, but not require separation into different physical entities. Instead, these components can be implemented as submodules or NFs in a service chain (or multiple service chains) on a single system.

We propose an architecture called CleanG,[7] where the data plane and control plane are supported by distinct NFs collocated on the same physical system as depicted in Figure 2 (i.e., a single component with two submodules). For example, supporting them on OpenNetVM platform, each submodule can be assigned resources (CPU and buffering) dynamically as needed based on the control and data plane workloads. CleanG remains true to 5G system architecture (release 16), both based on NFV. In CleanG 5GC functions remain logically and functionally decoupled. However, by virtue of being coresident with other 5GC user and control functions, several messages between functions are unnecessary and communications overheads are reduced. There is no need to distribute and synchronize state information. These are major contributors to performance improvements in CleanG. Additionally, the NFV-based 5GC platform allows inherent scale-out of functions on-demand. CleanG can also conveniently support slicing of the 5G network into multiple logical slices (whether it is for having different logical planes for distinct services or for different virtual network operators) by having distinct instances of the core NF for each slice.

A direction currently being pursued in the industry is to have the 5G control and data planes separated by having an SDN controller as an intermediary.[8] Unlike IP networks where the timescales for control plane updates (infrequent, of the order of seconds or more) are very different from data plane operations (frequent, of the order of microseconds or less), the cellular control plane and data plane are much more tightly coupled (e.g., when a UE transitions from idle to active, data packets can only flow after control plane operations for processing the service request are completed).[9] Having a controller to mediate the updates between the control and user plane adds substantial delay. Additionally, the controller may become a bottleneck under heavy control traffic. Because of the need to minimize the delays between control and user plane operations in the cellular environment, we

Entity	Dir	UE		RAN		AMF		SMF		UPF		UDM		PCF		UDR		N3IWF		SEAF		AUSF		EIR		DN		Total	
Message Type		B	O	B	O	B	O	B	O	B	O	B	O	B	O	B	O	B	O	B	O	B	O	B	O	B	O	B	O
Register	S	2	1	1	0	11	2	0	0	0	0	3	1	0	2	0	0	1	0	0	3	0	2	1	0	0	0	19	11
	R	1	2	1	0	10	2	1	0	0	0	4	0	0	2	0	0	1	0	0	3	0	2	1	0	0	0		
PDU Session	S	2	0	1	0	6	0	13	2	6	0	2	0	0	3	1	1	0	0	0	0	0	0	0	0	2	0	33	6
	R	1	0	1	0	9	0	9	1	7	1	2	0	0	3	1	1	0	0	0	0	0	0	0	0	3	0		
User Service request	S	1	1	3	0	3	0	4	6	2	5	0	1	0	1	0	0	0	0	0	3	0	2	0	0	0	0	13	19
	R	1	2	2	0	4	0	4	6	2	5	0	0	0	1	0	0	0	0	0	3	0	2	0	0	0	0		
Network Service Request	S	1	1	4	0	6	0	6	6	3	5	0	1	0	1	0	0	0	0	0	3	0	2	0	0	0	0	20	19
	R	3	2	3	0	5	0	6	6	3	5	0	0	0	1	0	0	0	0	0	3	0	2	0	0	0	0		
Deregister	S	1	0	0	0	3	1	3	1	1	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	10	4
	R	1	0	0	0	3	1	3	1	1	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0		
Handover	S	1	0	5	0	6	4	6	5	3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	13
	R	1	0	4	0	4	3	6	6	3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

TABLE 1. Approximate number of control plane messages received (R) and sent (S) for different events in 5G. (B = baseline, O = optional messages).

believe it is highly desirable to have them coresident on the same node wherever possible.

Supporting software-based NF offers additional opportunities for simplifying complex functions such as roaming. With the ability to copy over the state of a user session, an NF can be initiated in the visiting network in a short time (less than a second). This enables the user to be served by an NF in the visiting network while maintaining information for the home network, avoiding extra packet exchanges with the home network. This approach is more efficient than normal roaming or local breakout approaches as both data and control plane components are closer to the roaming user.

Our proposed CleanG architecture for the 5G cellular core exploits logically separate but physically consolidated core control (CCF) and core data plane functions (CDF), running on the OpenNetVM platform is shown below.[7] The CCF supports functionality provided by the SMF and AMF, while CDF implements the functionality of the 5G network's UPF. A primary goal is minimizing delay for an update from the cellular control plane resulting in changes to the data plane. This is achieved by the CDF and CCF sharing data and state using the OpenNetVM shared memory. Finally, multiple instances of this NFV-based 5GC may be created to scale-out based on traffic, and to dynamically adapt to the ratio of control to data plane traffic.

IMPROVING CELLULAR CONTROL PLANE PROTOCOL

As the number of components in 5G increased compared to LTE, additional messages are needed to keep state synchronized among them. Table 1 shows the number of messages exchanged for the 5G for each user event.

Most of these messages are exchanged sequentially, or on occasion, after a timer expiration. The completion time for control plane actions for an event is the cumulative time for exchanging these messages, thus contributing to high delays. This delay includes time to process each control message and the propagation and queuing delays for sending packets between different components, especially if the control plane and data plane components are far apart. Based on the CleanG NFV-based architecture, multiple core network processing components may be consolidated into one or more NFs running on the same node. This facilitates reducing the number of control plane messages, lowering completion time.

A second major 5G overhead is in using GTP-U tunneling to carry data packets between different user plane components. The latency consuming task is the setting up of the tunnel. The receiving end assigns a tunnel ID (TEID) to a flow and notifies the sender. Because the control plane components (AMF and SMF) are involved in initiating and mediating the

tunnel setup, a number of messages are exchanged, which is time-consuming. In the CleanG architecture, we use simple Generic Routing Encapsulation tunneling that does not require explicit setup or exchanging TEIDs. Different classes of services can be used to meet simple application requirements using the differentiated service code point (DSCP) field in the outer IP header.

Another challenge for future cellular networks is from new types of workload, e.g., from IoT devices. Exchanging a large number of control messages for an idle-active transition not only adds delay but results in the overhead on 5GC control components (e.g., AMF, SMF). One option proposed in 3GPP standardization is to piggyback data packets with the first control message to an AMF, to reduce delay. However, this can cause the AMF to become the bottleneck (excessive load from large numbers of IoT devices), contributing to the additional delay. The consolidation of control and data plane components in CleanG enables immediate notification of the control plane while avoiding it having to process data packets.

Consider, for example, a service request user event in the current 3GPP specification (see Figure 3) for 5G networks. The AMF updates the SMF about user sessions and receives responses. The SMF then updates the UPF, enabling forwarding of packets by the data plane. In an NFV environment where the control and data plane components are coresident on the same system and can share state, the need for 6a,b, 7a,b, 18a,b, 21a,b can be eliminated. Consolidation of control entities in the architecture can eliminate messages 4, 11, 15, and 19. For the common cases when an intermediate UPF is not used and dynamic policy is not enforced, 13 out of 15 core message exchanges are not essential.

CARRYING TRAFFIC EFFICIENTLY OVER BACKHAUL NETWORKS

With the higher speeds of new radio technologies, the capacity demands and Traffic Engineering requirements on backhaul networks between gNB and 5GC increase rapidly. Efficient use of the network, its resources and proper use of available paths are essential for both cost and performance reasons. We examine current technologies trends for path selection, routing, and TE in the backhaul network.

Background on Current TE Mechanisms

In the current cellular architecture, traditional approaches like MPLS with RSVP-TE determine the label switched path and manage resources along the path.[10,11] However, they are not dynamic and require provisioning steps along the path, involving out-of-band signaling. Another mechanism being considered is to use segment routing (SR) which is a source routing technology. SR (RFC 8402), uses path segments computed offline by a controller for a particular flow or a service.[12] The path is then decomposed into a sequence of network segments along which packets of a flow are routed. A sequence of segments is carried in the packet, essentially using source routing, with either MPLS labels or IPv6 address formats.[13,14] While SR allows packet steering on a specified path, it does not have any notion of QoS or resources being reserved along the path. Furthermore, SR also has the well-known overhead of increasing the packet header by encoding the segments into packets for routing purposes.[15]

A recent alternative being proposed, called Preferred Path Routing (PPR),[15,16] seeks to overcome some of the challenges in above-mentioned approaches. PPR is an innovative path routing architecture to signal explicit paths from sources and per-hop processing including QoS awareness from the controller to network nodes. PPR builds on the SDN paradigm and utilizes various central path computation engine features to create the TE paths/graphs) with a certain bandwidth,[17,18] latency and/or allowable jitter constraints. When deployed in 5G backhaul, PPR can combine TE and QoS guarantees of the path without significant packet overheads.

Figure 4 shows a typical backhaul network with cell site routers (CSRs) at gNB and virtualized UPF. At CSR, PPR encapsulates user packet with destination IP address as PPR-ID, which is a preprogrammed forwarding identifier along with QoS attributes through the underlying routing protocol/IGP. The packet is forwarded along the preferred path associated with PPR-ID from gNB to UPF. PPR can support new services that require low and/or bounded latencies, along with traditional bandwidth guarantees.

PPR flexibly supports a wide range of existing forwarding planes including native IP (IPv4/IPv6) user

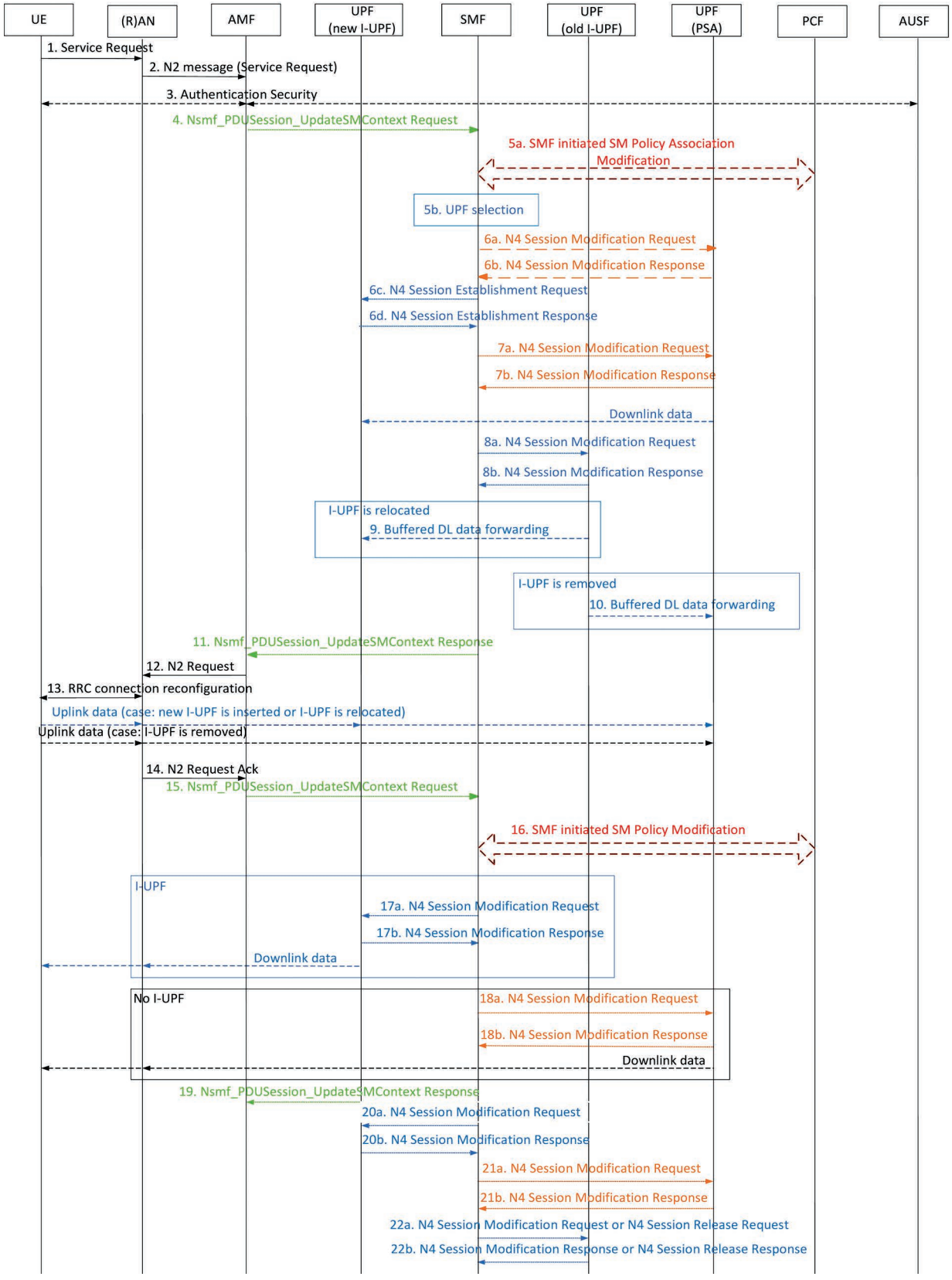


FIGURE 3. Messages exchanged for service request event in 5G among 5GC components and UE & (R)AN.

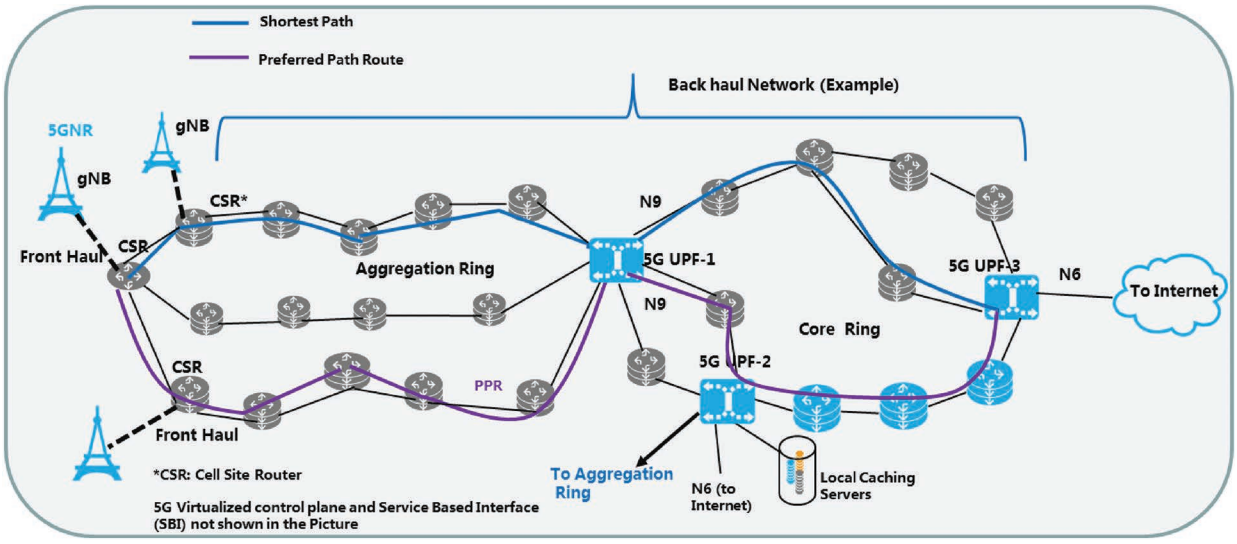


FIGURE 4. Backhaul transport with PPR.

planes. PPR also brings much-needed optimizations to SR defined user planes (SR-MPLS, SRv6). TE is complex and 5G applications need these paths to be available or withdrawn on-demand. The PPR solution simplifies this in the 5G backhaul by enabling different types of TE paths in a scalable manner and allows for rapid traffic switchover to backup paths.

IMPROVING THE BACKHAUL DATA PLANE

GTP-U is the legacy 4G user plane protocol that is also used in 5G. The new N9 interface between UPFs and its encapsulation is currently under study in both 3 GPP and IETF in order to determine whether retaining GTP-U or using other IP approaches over the backhaul network would address concerns about supporting a diverse set of services, such as having low-latency applications as well as supporting high bandwidth flows in 5G networks.

In this section, we discuss challenges in supporting such a wide range of applications with traditional IP and recommend further refinements to the 5G user plane.

Supporting deterministic service guarantees for 5G applications, such as a vehicle to infrastructure communication, and industrial automation and control is a nontrivial task in modern data planes. IP offers coarse-grained control over how packets of such flows are scheduled and processed as they transit through multihop networks. Specifically, when faced with congestion, packets of a service offering ultra-reliability may be arbitrarily dropped because the data plane has little knowledge about the need to provide ultra-reliable service to a flow. To keep data planes more aware of service constraints, SDN frameworks and programmable switches may be used to provide dynamic treatment of flows within a switch or router via the control plane; nevertheless, such out of band programmability alone is insufficient to respond to changing network conditions in a timely, service-aware manner. A contextual data plane is desired that can discriminate packets based on their service guarantees at runtime. In this regard, the "big packet protocol" (BPP) data plane solution providing a higher degree of customization of flows across a network has been suggested.

The BPP data plane is a generic framework that enables carrying different service-specific parameters and constraints along with the packets and processing them on each hop. With this additional information, routers know precisely what to do for each packet, going beyond just the coarse-grained indication from the DSCP bits of an IP packet. A typical BPP packet on the wire is shown in Figure 5(a). The BPP block provides "commands," for example, basic forwarding actions such as next hop, redirect, inspect, drop, etc., as well as new scheduling or shaping actions such as bounded end-to-end latency or

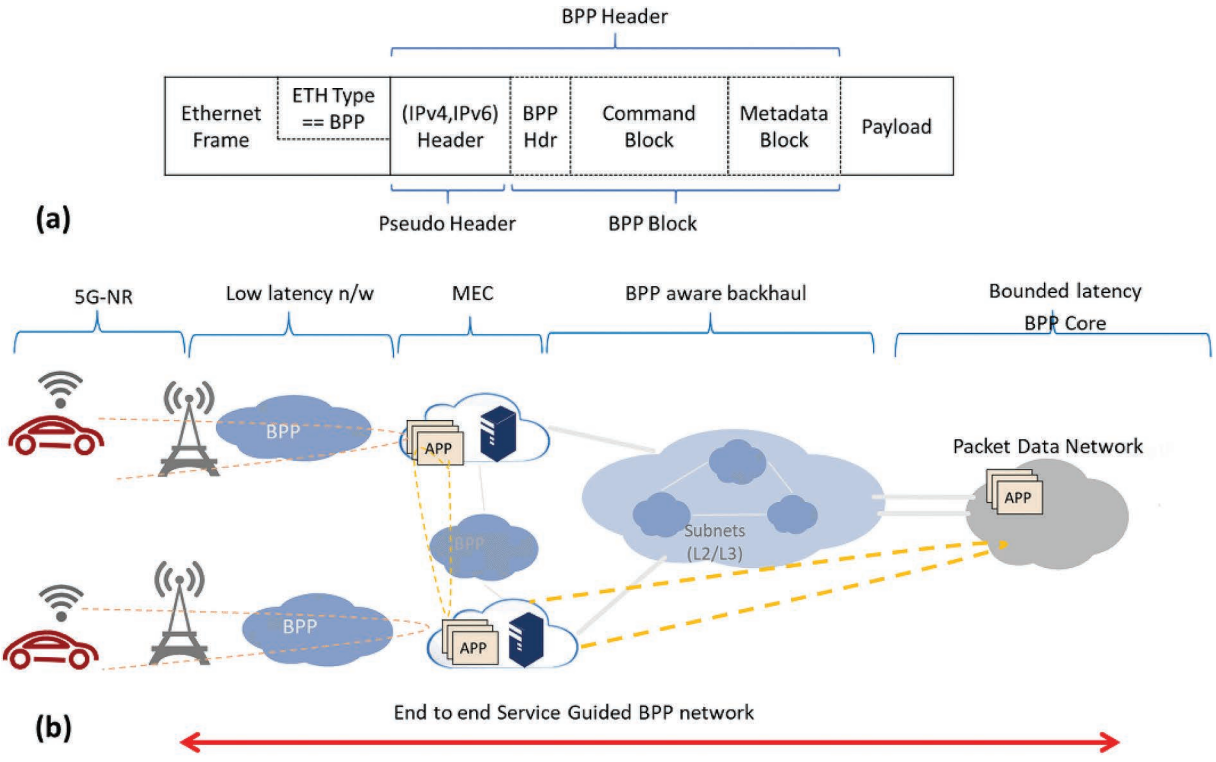


FIGURE 5. (a) BPP Packet format. (b) BPP aware packet network to enable low-latency networking.

acceptable jitter. Additionally, the meta-data block may carry information about the packet such as what service it is part of, and/or mappings to the path that packet may use. Although the protocol itself does not preclude inserting the BPP Block by a host where the packet originates, it is best if inserted at the edge of the service network thereby remaining under network operator's control. We discuss below how BPP may be useful for providing very low latency, and highly reliable delivery (the so-called "ultra-reliable, low-latency" class of traffic, URLLC) over the cellular backhaul next.

Supporting Low-Latency Traffic Over the Cellular Backhaul

The example used to demonstrate BPP capabilities is that of the vehicle to infrastructure (V2X) communication in a 5G network. In V2X applications, receivers do not wish to receive traffic that is already delayed beyond a latency bound. For example, a vehicle that already crossed intersection does not benefit from receiving traffic information for that intersection.

In Figure 5(b), two latency sensitive paths are shown, a) from NR to edge compute node shown as green dotted path, which needs low latency guarantees of the order of 10 ms, b) from edge compute to data center in core network with bounded latency shown as yellow dotted path that should support bounded latency in the order of 30 ms. For simplicity, assume these are BPP enabled IP networks, capable of processing data packets per BPP instructions and appropriate scheduling.

A BPP packet carries "residual latency" instruction as it traverses the BPP aware backhaul network, the value of which gets adjusted at each hop. When a packet arrives on a particular hop, the forwarding pipeline on the node reads the instruction, schedules the packet based on the residual latency for the packet but also updates the value of latency for next hop. If the latency bound (specified in latency instruction metadata) cannot be met because of the current queueing at the router, the packet will be dropped early as there is no value in late arrival, thus avoiding wasted network bandwidth further downstream.

Similar instructions can be incorporated for reliability (backup path) but are not discussed here due lack of space.

Admittedly, such data planes would require new capabilities in the forwarding engines, which are well served with programmable software-based NFV platforms like OpenNetVM.

5G cellular networks promise to provide low latency and high bandwidth to meet emerging, demanding, performance-sensitive applications. A key enabler is the use of NFV that offers flexibility from being software based. We note that the use of NFV allows data plane and control plane functionality to be supported on the same platform, unlike the traditional packet core network of multiple distributed components. Scalability is enabled by the dynamic instantiation of the NFV platform supporting the CDF and CCF. However, architectural and implementation changes alone with NFV squander the opportunity for truly improving the cellular network's performance if the protocols do not properly take advantage of the ability to consolidate the tightly interdependent cellular control and data plane. We rethink the design of the control protocol to achieve low latency and high throughput by simplification and using fewer messages. Complementing this, the backhaul network and protocols must also be designed to judiciously utilize capacity and achieve low latency. The PPR protocol is a key enhancement for the cellular backhaul. Low-latency applications are enabled by having a more flexible plane, using the ideas of BPP.

In this paper, we sought to analyze several of the complexities of cellular networks without completely disrupting the 5G system architecture, and proposed carefully thought-out approaches to enhance the architecture and protocols of 5GC, the 5G backhaul, and using a new backhaul transport data plane.

ACKNOWLEDGMENTS

This work was supported in part by the NSF under Grant CNS-1618344 and in part by a grant from Futurewei Inc.

REFERENCES

1. 3 GPP TS 23.502, "Procedures for the 5G system," 2018.
2. P. Schmitt et al., "Control and user plane separation of EPC nodes (CUPS)," 2017. Available at: http://www.3gpp.org/cups
3. 3 GPP TS 23.501, "System architecture for the 5G system," 2018.
4. [Online]. Available at: https://www.dpdk.org
5. W. Zhang et al., "OpenNetVM: A platform for high performance network service chains," in *Proc. Workshop Hot Topics Middleboxes Netw. Funct. Virtualization*, pp. 26–31.
6. 3 GPP TR 23.799, "Study on architecture for next generation system (Release 14)," V14.0.0, 2016.
7. A. Mohammadkhan et al., "CleanG: A clean-slate EPC architecture and control plane protocol for next generation cellular networks," in *Proc. ACM Workshop Cloud-Assisted Netw.*, 2016, pp. 31–36.
8. R. Shah et al., "Cuttlefish: Hierarchical SDN controllers with adaptive offload," in *Proc. Int. Conf. Netw. Protocols*, 2018, pp. 198–208.
9. A. Mohammadkhan et al., "Considerations for re-designing the cellular infrastructure exploiting software-based networks," in *Proc. 24th Int. Conf. Netw. Protocols*, 2016, pp. 1–6.
10. D. Berger et al., "RSVP-TE: Extensions to RSVP for LSP tunnels," IETF RFC 3209, Dec. 2001.
11. RFC 3031, MPLS "Multiprotocol label switching architecture," 2001.
12. C. Filsfils et al., "The segment routing architecture," in *Proc. IEEE Global Commun. Conf.*, 2015, pp. 1–6.
13. A. Bashandy et al., "Segment routing with MPLS user plane," draft-ietf-spring-segment-routing-mpls-14, IETF, Jun. 2018.
14. C. Filsfils et al., "IPv6 segment routing header (SRH)," draft-ietf-6man-segment-routing-header-14, IETF, Jun. 2018.
15. U. Chunduri et al., "Preferred path routing – A next-generation routing framework beyond segment routing," in *Proc. IEEE Global Commun. Conf.*, 2018.
16. T. Eckert, Y. Qu, and U. Chunduri, "Preferred path routing (PPR) graphs beyond signaling of paths to networks," in *IEEE Hi-Precis. Netw.*, 2018.
17. U. Chunduri et al., "Preferred path routing (PPR) in IS-IS," Available at: https://tools.ietf.org/html/draft-chunduri-lsr-isis-preferred-path-routing-01, 2018.
18. U. Chundur et al., "Transport network aware mobility for 5G," Available at: https://tools.ietf.org/html/draft-clt-dmm-tn-aware-mobility-01, 2018.

19. "Optimized Mobile User Plane Solutions for 5G, draft -bogineni-dmm-optimized-mobile-user-plane-01," Available at: https://www.ietf.org/id/draft-bogineni-dmm-optimized-mobile-user-plane-01.txt, 2018.
20. R. Li, A. Clemm, U. Chunduri, L. Dong, and K. Makhijani, "A new framework and protocol for future networking applications," in *Proc. Workshop Netw. Emerg. Appl. Technol.*, 2018, pp. 21–26.

ALI MOHAMMADKHAN is currently working toward the Ph.D. degree in computer science with the University of California, Riverside, Riverside, CA, USA. Contact him at amoha006@ucr.edu.

K. K. RAMAKRISHNAN is a Professor of Computer Science with the University of California, Riverside, Riverside, CA, USA. Contact him at kk@cs.ucr.edu.

UMA CHUNDURI is a System Architect, with Future Networks team at America Research Center, Huawei Technologies, Santa Clara, CA, USA. Contact him at uma.chunduri@huawei.com.

KIRAN MAKHIJANI is a Principal Engineer with Future Networks, America Research Center, Huawei Technologies, Santa Clara, CA, USA. Contact her at Kiran.Makhijani@huawei.com.

SUBMIT TODAY

IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING

SCOPE

The *IEEE Transactions on Sustainable Computing* (*T-SUSC*) is a peer-reviewed journal devoted to publishing high-quality papers that explore the different aspects of sustainable computing. The notion of sustainability is one of the core areas in computing today and can cover a wide range of problem domains and technologies ranging from software to hardware designs to application domains. Sustainability (e.g., energy efficiency, natural resources preservation, using multiple energy sources) is needed in computing devices and infrastructure and has grown to be a major limitation to usability and performance.

Contributions to *T-SUSC* must address sustainability problems in different computing and information processing environments and technologies, and at different levels of the computational process. These problems can be related to information processing, integration, utilization, aggregation, and generation. Solutions for these problems can call upon a wide range of algorithmic and computational frameworks, such as optimization, machine learning, dynamical systems, prediction and control, decision support systems, meta-heuristics, and game-theory to name a few.

T-SUSC covers pure research and applications within novel scope related to sustainable computing, such as computational devices, storage organization, data transfer, software and information processing, and efficient algorithmic information distribution/processing. Articles dealing with hardware/software implementations, new architectures, modeling and simulation, mathematical models and designs that target sustainable computing problems are encouraged.

SUBSCRIBE AND SUBMIT

For more information on paper submission, featured articles, calls for papers, and subscription links visit:

www.computer.org/tsusc

IEEE COMPUTER SOCIETY

IEEE COMMUNICATIONS SOCIETY

CEDA IEEE Council on Electronic Design Automation

IEEE

IEEE TRANSACTIONS ON

COMPUTERS

Call for Papers: *IEEE Transactions on Computers*

Publish your work in the IEEE Computer Society's flagship journal, *IEEE Transactions on Computers* (*TC*). *TC* is a monthly publication with a wide distribution to researchers, industry professionals, and educators in the computing field.

TC seeks original research contributions on areas of current computing interest, including the following topics:

- Computer architecture
- Software systems
- Mobile and embedded systems
- Security and reliability
- Machine learning
- Quantum computing

All accepted manuscripts are automatically considered for the monthly featured paper and annual Best Paper Award.

Learn about calls for papers and submission details at
www.computer.org/tc.

IEEE COMPUTER SOCIETY

IEEE

Get Published in the New *IEEE Open Journal of the Computer Society*

Submit a paper today to the premier new open access journal in computing and information technology.

Your research will benefit from the IEEE marketing launch and 5 million unique monthly users of the IEEE *Xplore*® Digital Library. Plus, this journal is fully open and compliant with funder mandates, including Plan S.

Submit your paper today!
Visit www.computer.org/oj to learn more.

IEEE COMPUTER SOCIETY | IEEE

Conference Calendar

IEEE Computer Society conferences are valuable forums for learning on broad and dynamically shifting topics from within the computing profession. With over 200 conferences featuring leading experts and thought leaders, we have an event that is right for you. Questions? Contact conferences@computer.org.

MAY

3 May
- FCCM (IEEE Int'l Symposium on Field-Programmable Custom Computing Machines), Fayetteville, USA

18 May
- SP (IEEE Symposium on Security and Privacy), San Francisco, USA
- IPDPS (IEEE Int'l Parallel and Distributed Processing Symposium), New Orleans, USA

30 May
- ISCA (ACM/IEEE Int'l Symposium on Computer Architecture), Valencia, Spain

JUNE

7 June
- ARITH (IEEE Int'l Symposium on Computer Arithmetic), Portland, USA

14 June
- CVPR (IEEE Conf. on Computer Vision and Pattern Analysis), Seattle, USA

15 June
- ICHI (IEEE Int'l Conf. on Healthcare Informatics), Oldenburg, Germany

22 June
- CSF (IEEE Computer Security Foundations Symposium), Boston, USA

29 June
- DSN (IEEE/IFIP Int'l Conf. on Dependable Systems and Networks), Valencia, Spain

30 June
- MDM (IEEE Int'l Conf. on Mobile Data Management), Versailles, France

JULY

6 July
- ICME (IEEE Int'l Conf. on Multimedia and Expo), London, UK

8 July
- ICDCS (IEEE Int'l Conf. on Distributed Computing Systems), Singapore

13 July
- COMPSAC (IEEE Computer Software and Applications Conf.), Madrid, Spain

28 July
- CBMS (IEEE Int'l Symposium on Computer-Based Medical Systems), Rochester, USA

AUGUST

1 August
- JCDL (ACM/IEEE Joint Conf. on Digital Libraries), Xi'an, China

14 August
- SmartIoT (IEEE Int'l Conf. on Smart Internet of Things), Beijing, China

16 August
- Hot Chips, Palo Alto, USA

17 August
- ACSOS (IEEE Int'l Conf. on Autonomic Computing and Self-Organizing Systems), Washington, DC, USA

19 August
- RTCSA (IEEE Int'l Conf. on Embedded and Real-Time Computing Systems and Applications), Gangnueng, South Korea

24 August
- FiCloud (Int'l Conf. on Future Internet of Things and Cloud), Rome, Italy

31 August
- RE (IEEE Int'l Requirements Eng. Conf.), Zurich, Switzerland

SEPTEMBER

7 September
- EuroS&P (IEEE European Symposium on Security & Privacy), Genova, Italy

14 September
- CLUSTER (IEEE Int'l Conf. on Cluster Computing), Kobe, Japan

21 September
- ASE (IEEE/ACM Int'l Conf. on

Automated Software Eng.), Melbourne, Australia

24 September

- BigMM (IEEE Int'l Conf. on Multimedia Big Data), New Delhi, India

27 September

- ICSME (IEEE Int'l Conf. on Software Maintenance and Evolution), Adelaide, Australia

28 September

- SecDev (IEEE Secure Development), Atlanta, USA

OCTOBER

3 October

- PACT (Int'l Conf. on Parallel Architectures and Compilation Techniques), Atlanta, USA

5 October

- EDOC (IEEE Int'l Enterprise Distributed Object Computing Conf.), Eindhoven, the Netherlands
- ICSE (IEEE/ACM Int'l Conf. on Software Eng.), Seoul, South Korea

17 October

- MICRO (IEEE/ACM Int'l Symposium on Microarchitecture), Athens, Greece

18 October

- MODELS (ACM/IEEE Int'l Conf. on Model-Driven Eng. Languages and Systems), Montreal, Canada

21 October

- FIE (IEEE Frontiers in Education Conf.), Uppsala, Sweden

25 October

- VIS (IEEE Visualization Conf.), Salt Lake City, USA

NOVEMBER

2 November

- ICFEC (IEEE Int'l Conf. on Fog and Edge Computing), Melbourne, Australia

6 November

- SmartCloud (IEEE Int'l Conf. on Smart Cloud), Washington, DC, USA

8 November

- NAS (IEEE Int'l Conf. on Networking, Architecture, and Storage), Riverside, USA

9 November

- ICTAI (IEEE Int'l Conf. on Tools with Artificial Intelligence), Baltimore, USA
- ISMVL (IEEE Int'l Symposium on Multiple-Valued Logic), Miyazaki, Japan

15 November

- SC, Atlanta, USA

16 November

- FOCS (IEEE Symposium on Foundations of Computer Science), Durham, USA
- LCN (IEEE Conf. on Local Computer Networks), Sydney, Australia

DECEMBER

9 December

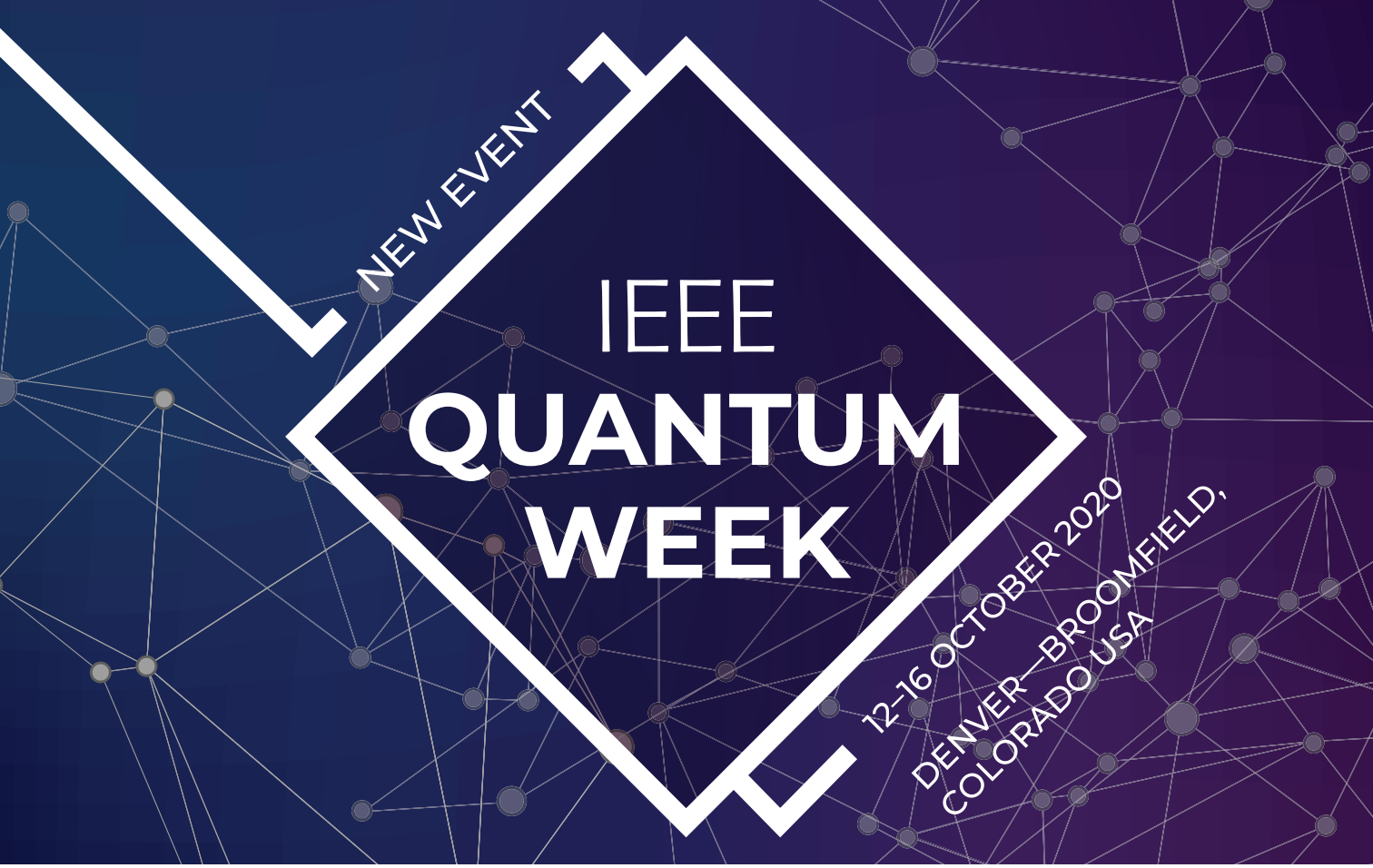
- CC (IEEE Int'l Conf. on Conversational Computing), Irvine, USA
- AIKE (IEEE Int'l Conf. on Artificial Intelligence and Knowledge Eng.), Irvine, USA

14 December

- BCD (IEEE/ACIS Int'l Conf. on Big Data, Cloud Computing, and Data Science Eng.), Macao
- CloudCom (IEEE Int'l Conf. on Cloud Computing Technology and Science), Bangkok, Thailand

16 December

- BIBM (IEEE Int'l Conf. on Bioinformatics and Biomedicine), Seoul, South Korea



IEEE Quantum Week 2020 Is Open for Submissions

Participation opportunities are available for the inaugural IEEE International Conference on Quantum Computing and Engineering (QCE 2020) to be held 12–16 October 2020, in Denver—Broomfield, CO.

IEEE Quantum Week aims to be a leading venue for presenting high-quality original research, ground-breaking innovations, and compelling insights in quantum computing, engineering, and technologies.

Authors are invited to submit proposals for technical papers, posters, tutorials, workshops, and panels. Submission schedules are available at qce.quantum.ieee.org/important-dates.

IEEE Quantum Week includes the following technical paper tracks:

- Quantum Communications, Sensing, Cryptography
- Quantum Photonics and Optics
- Quantum Computing
- Quantum Algorithms & Information
- Quantum Applications and Simulating Nature
- Quantum Engineering
- Quantum Benchmarks & Measurements
- Quantum Education

Papers accepted by IEEE QCE will be submitted to the IEEE Xplore Digital Library. The best papers will be invited to the journals *IEEE Transactions on Quantum Engineering* (*TQE*) and *ACM Transactions on Quantum Computing* (*TQC*).

Submission instructions and details:
qce.quantum.ieee.org/callforcontributions

IEEE QUANTUM · IEEE COMPUTER SOCIETY · IEEE ComSoc IEEE Communications Society · IEEE CSC Council on Superconductivity · IEEE Photonics Society · IEEE TEMS Technology & Engineering Management Society · IEEE ELECTRONICS PACKAGING SOCIETY · IEEE