Editor in Chief: **Ipek Ozkaya**
Carnegie Mellon Software Engineering Institute
ipek.ozkaya@computer.org

Understanding and Building Trust in Software Systems

Ipek Ozkaya

INTEGRATING CROSSCUTTING CONCERNS into the software development and sustainment lifecycle effectively to realize them fully in software systems is an enduring challenge in software engineering. This challenge is exacerbated when the crosscutting concerns have overloaded meaning, and their scope changes as new attributes and technologies emerge. Trust is one such crosscutting concern. Developing software systems with trust today increasingly also implies moving from a trust approach where the philosophy is *trust but verify* to one focused on the continuous assessment of trust across every device, user, application, and data, where the philosophy is *never trust, always verify*, denying access by default, also referred to as *zero trust*. The zero trust concept is mostly referred to in the context of network security, and the zero-trust architecture approach is used to describe the network architecture to enable it.[1]

Digital Object Identifier 10.1109/MS.2021.3134112
Date of current version: 18 April 2022

However, trust as a crosscutting software concern implies many other requirements in addition to security and how to manage communication networks. Building trust in software systems also requires us software engineers to consider how they are built by understating different trust-related attributes; organizational practices that are implicated in the development, operations, and sustainment of systems; users' perception of these systems; and related data and its management. Trust has functional requirement implications and quality attribute requirement implications as well as teaming and organizational implications. On the one hand, such an overload of perspectives creates hard-to-manage complexity. On the other hand, recognizing the multiattribute nature of trust can also be a tool to ensure that a system is developed and sustained from a more holistic perspective.

The Many Facets of Trust

There is existing work that aims to distinguish different aspects of trust. For example, in the context of embedded systems, trustworthiness includes attributes related to trust, such as reliability and dependability, as well as other concerns like maintainability.[2] Ideally, trust should be a computed, quantifiable attribute of the system; therefore, different attributes related to trust need to be decomposed to be allocated as responsibilities to different components within the system.[3] Ability to compute what trust means for a given system and to guarantee trust through its design, protocols, and algorithms demonstrated with computable metrics helps assessing its level of trustworthiness. A system where trust is a computable and demonstrable attribute is more likely to be perceived as trustworthy by its users.

Understanding and building trust in software systems has system-wide implications, including the hardware, the software, the users, and the communication protocols among all. These four elements will decompose to different numbers of components based on their

CONTACT US

AUTHORS

For detailed information on submitting articles, visit the "Write for Us" section at www.computer.org/software

LETTERS TO THE EDITOR

Send letters to software@computer.org

ON THE WEB

www.computer.org/software

SUBSCRIBE

www.computer.org/subscribe

SUBSCRIPTION CHANGE OF ADDRESS

address.change@ieee.org
(please specify *IEEE Software*.)

MEMBERSHIP CHANGE OF ADDRESS

member.services@ieee.org

MISSING OR DAMAGED COPIES

contactcenter@ieee.org

REPRINT PERMISSION

IEEE utilizes Rightslink for permissions requests. For more information, visit www.ieee.org/publications/rights/rights-link.html

particular application domain. Taking a zero-trust approach and focusing on the continuous verification of trust is not at the expense of an effort of building trustworthy components. Software engineers need to embrace a mindset where they develop components that are trustworthy by design yet always recognize that the same components may behave differently in their environments or that their behavior may be altered intentionally or unintentionally.

While there are quite a number of attempts to understand all the different aspects of trust, in particular coming out of the robotics and autonomy communities, there is not one ground truth-accepted categorization of attributes implied by trust. A universally accepted decomposition of trust to all its relevant attributes is neither possible nor useful. Many different terms and concepts are associated with trust. While security and privacy are often commonly included as critical aspects of building trust in systems, there are countless other attributes that are also relevant. These range from safety to dependability to reliability to adaptability to include all the expected runtime quality attributes of a system, including but not limited to availability, usability, and performance. The list can go on. Cho et al.[2] offer one such decomposition on trustworthiness that includes trust, resilience, and agility as its main attributes.

It is important to understand the relationships of all these attributes in building trust, in particular when systems will need to incorporate continued runtime verification to guarantee trust. The continued verification requirement implies improved accuracy and precision in identifying and quantitatively expressing all trust-related attributes. The relevance of some of these attributes evolves in time and new priorities emerge, implying new attributes. For example, with increasing number of business requirements to incorporate machine learning components into systems, engineers need to articulate algorithmic trust and trust in data. More effort, therefore, needs to be spent in employing reliable techniques for accurately and quantitatively expressing key trust attributes from the perspective of users and system components as well as building techniques to ensure their runtime enforcement and verification.

Articles in This Issue

This issue features the multifaceted nature of trust with four articles. Each article studies trust from a different perspective, including the implications of trust in COVID-19 apps as well as enforcing trust in emerging technologies, such as blockchains. Three of the four articles focus on the users' perceptions of trust from the perspective of privacy.

In no time, the global COVID-19 pandemic resulted in software applications to help manage the pandemic. Contact tracing was among the top perceived needs. While many countries advocated and attempted the use of contract tracing apps, their adoption had been varied; among many reasons, trust in privacy and security concerns is at the top. Garousi, Cutting, and Felderer,[4] in their article "What Users Think of COVID-19 Contact-Tracing Apps: An Analysis of Eight European Apps," present results studying the features offered in these apps and the challenges in their development and adoption.

Malavolta and colleagues[5] also study COVID-19 applications, taking a broader look, in their article "Engineering Mobile Apps for Disaster Management: The Case of COVID-19 Apps in the Google Play Store." They report on the consequences, including user trust issues, resulting from these apps being developed under tight schedules, as well as social and political pressures. Their results reveal that the lack of trust in implemented privacy and security measures inevitably creates barriers for user adoption.

The article "New Privacy Practices for Blockchain Software," by Bellés-Muñoz et al.,[6] focuses on enabling complex privacy technologies while achieving the public transparency of blockchain software. They focus on the at-odds nature of transparency and guaranteeing trust in privacy zero-knowledge proof for implementing applications with verifiable privacy requirements.

And finally, in their article "Digital Age of Consent and Age Verification: Can They Protect Children?" Pasquale and colleagues[7] study how the mechanisms adopted by apps to verify the age of users can be easily bypassed and expose children to privacy and safety threats. While there are existing regulations, such as the Children's Online Data Protection Act in the United States and the General Data Protection Regulation in Europe, to protect data collection and protect privacy, how they are implemented in applications, especially for those that are used for different age groups, determines the trust level of an application. In this context, *trust* refers to how a software application determines and implements safeguards by not enabling access to content while also not overstepping privacy measures in doing so.

This issue barely scratches the surface, if at all, of the intricate challenges involved in understanding, designing for, and guaranteeing trust in systems. There are a number of challenges around runtime verification of attributes related to trust, modeling the competing priorities of the attributes involved as well as ensuring the completeness and correctness of these attributes identified.

Our goal with this issue is to highlight that concerns around trust evolve as the needs from software applications evolve and as technologies that compromise software system elements evolve. Identifying the crosscutting as well as competing attributes related to trust will always have application- and domain-specific aspects in addition to the common security, privacy, and reliability concerns. Achieving coverage in the elicitation of all such attributes quantitatively and implementing runtime verification approaches for these attributes will continue to be open challenges in software engineering for the next decade. Understanding and building trust and what constitutes trust in software systems will always be a moving target.

References

1. S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," National Institute of Standards and Technology, Gaithersburg, MD, USA, NIST Special Publication 800-207, Aug. 2020. [Online]. Available: https://nvlpubs.nist.

Digital Object Identifier 10.1109/MS.2021.3134111

gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf
2. J.-H. Cho, P. M. Hurley, and S. Xu, "Metrics and measurement of trustworthy systems," in *Proc. 2016 IEEE Military Commun. Conf. (MILCOM 2016)*, pp. 1237–1242, doi: 10.1109/MILCOM.2016.7795500.
3. E. Cioroaica, S. Chren, B. Buhnova, T. Kuhn, and D. Dimitrov, "Towards creation of a reference architecture for trust-based digital ecosystems," in *Proc. 13th Eur. Conf. Softw. Architecture (ECSA '19)*, New York, NY, USA: ACM, 2019, pp. 273–276, doi: 10.1145/3344948.3344973.
4. V. Garousi, D. Cutting, and M. Felderer, "What users think of COVID-19 contact-tracing apps: An analysis of eight European apps," *IEEE Softw.*, vol. 39, no. 3, pp. 22–30, 2022, doi: 10.1109/MS.2021.3097284.
5. I. Malavolta, T. A. Ghaleb, I. David, J. van Rooijen, and M. Stoelinga, "Engineering mobile apps for disaster management: The case of COVID-19 apps in the Google Play Store," *IEEE Softw.*, vol. 39, no. 3, pp. 31–42, 2022, doi: 10.1109/MS.2021.3129978.
6. M. Bellés-Muñoz, J. Baylina, V. Daza, and J. L. Muñoz-Tapia, "New privacy practices for blockchain software," *IEEE Softw.*, vol. 39, no. 3, pp. 43–49, 2022, doi: 10.1109/MS.2021.3086718.
7. L. Pasquale, P. Zippo, C. Curley, B. O'Neill, and M. Mongiello, "Digital age of consent and age verification: Can they protect children?" *IEEE Softw.*, vol. 39, no. 3, pp. 50–57, 2022, doi: 10.1109/MS.2020.3044872.