10-2020

Digital banking accelerator: A Service-Oriented Architecture starter kit for banks

Alan @ Ali MADJELISI MEGARGEL
Singapore Management University, ALANMEGARGEL@SMU.EDU.SG

SHANKARARAMAN, Venky
Singapore Management University, venks@smu.edu.sg

Citation

1

Digital Banking Accelerator: A Service-Oriented Architecture Starter Kit for Banks

Alan Megargel, *Member, IEEE*, Venky Shankararaman, *Member, IEEE*,

***Abstract*—Digital banking refers to the delivery of interactive financial services through online mechanisms which include web and mobile apps. The main barrier to digital banking for traditional banks is the presence of legacy core banking systems. Service-Oriented Architecture (SOA) is a key enabler to overcome this barrier, and a bank's level of SOA maturity influences its time-to-market capability of delivering new innovative digital banking solutions. However, most traditional banks struggle with implementing an SOA due to a number of technology and organizational challenges, and the overall steep learning curve. This paper proposes a Digital Banking Accelerator, a "starter kit" for helping traditional banks, having little or no SOA competencies, to rapidly implement an effective SOA in order to accelerate their digital banking capability.**

***Index Terms*—Digital banking, financial technology, legacy systems, microservices, service-oriented architecture.**

I. INTRODUCTION

Incumbent traditional banks are urgently pursing "digital" strategies in order to protect their market share from FinTech substitutes. "Digital Banking" is the latest buzz word for what was previously referred to as e-banking or online banking. However in today's context, the focus is more on new interactive methods of service delivery and enhancing customer experience. Technology-centric capabilities are now supported by a dynamic and accessible open banking system.

Multiple-case study analysis of eight banks reveals that legacy systems are the greatest inhibitor of digital banking capability, however banks with a high level of service-oriented architecture (SOA) maturity can overcome this barrier [11]. SOA is a key enabler for digital banking [3,5,6,13,15]. However, banks with no prior SOA experience have a steep learning curve, and can take several years to fully implement an effective SOA [5,10]. There are technical challenges in understanding SOA technologies, and organizational challenges in governing SOA decisions [9,10].

This paper proposes a Digital Banking Accelerator, which is essentially a "starter kit" for helping traditional banks, having little or no SOA competencies, to rapidly implement an effective SOA, which in effect will accelerate their digital banking capability. Such an innovation is expected to reduce the SOA implementation cycle time by at least one calendar year in banks with no prior SOA experience. The benefit of this "starter kit" as a digital banking accelerator has already been demonstrated in a large Taiwanese bank.

The Taiwanese bank fit the typical profile of a traditional bank which can benefit from this "starter kit". Such a bank can be characterized as having the following technology limitations: a) a legacy commercial-off-the-self Core Banking System (CBS), b) a batch mode style of data distribution using a file transfer technology, c) a point-to-point style of Enterprise Application Integration (EAI) using a message transformation middleware, and d) little or no prior experience with SOA. The "starter kit" is meant to help such a bank decouple its legacy CBS from its channels (i.e. user interfaces), rather than to replace its CBS with an equivalent collection of microservices. However, if the bank does decide to replace its CBS in the future, the framework provided by the "starter kit" enables a systematic migration from the bank's monolithic CBS to a microservices-based (coreless) banking system [12].

A. About SMU tBank

Singapore Management University (SMU) has developed a full-featured cloud-based digital bank called SMU Teaching Bank (or SMU tBank), for teaching and research purposes. The "starter kit" presented in this paper is a byproduct of that effort.

B. About LTB Bank

A Large Taiwanese Bank (LTB Bank), named anonymously, was established around 2020. From its inception through 2016, they operated as a corporate bank and were not allowed to take deposits from retail customers. In 2015, the Taiwan banking regulator mandated that all Taiwanese banks were to offer retail banking services by a deadline of January 2017. At the time, LTB Bank had only 6 branch offices which was a factor in their strategic decision to open up a retail bank as a completely branchless digital bank.

International Integrated Systems Inc. (IISI) is a System Integrator (SI) and outsource software provider based in Taiwan. IISI took the SMU tBank baseline code, and uplifted the prototype into a commercial-grade "starter kit" product which they implemented at LTB Bank starting in January 2016.

Leveraging this Digital Banking Accelerator ("starter kit"), LTB Bank went from zero to fully functioning digital bank

This paper was submitted for review on 16 August 2020.

A. Megargel is with Singapore Management University, 80 Stanford Road, Singapore 178902 (e-mail: alanmegargel@smu.edu.sg).

V. Shankararaman is with Singapore Management University, 80 Stanford Road, Singapore 178902 (e-mail: venks@smu.edu.sg).

within one year. During their first year of operation, after going live in January 2017, LTB Bank also implemented: an eLoans product, an online Wealth Management product, an FX Portal, and they opened up their API to third party FinTech payment providers. In October 2018, LTB Bank was awarded as the "Best Digital Bank in Taiwan".

II. Related Work

Traditional banks aiming to implement digital banking capabilities are faced with a seemingly unsurmountable barrier to overcome; legacy CBS's are inflexible to change and inhibit time-to-market of new digital products and services. This section covers related work done around overcoming the legacy system barrier.

Legacy system modernization strategies can be grouped into four categories: a) "replacement strategies" which involve replacing the legacy system with a commercial-off-the-self system, b) "wrapping strategies" which involve exposing the functionality of the underlying legacy system as reusable services in an SOA, c) "redevelopment strategies" which involve re-engineering and re-implementation of legacy system components to be more service-oriented, and d) "migration strategies" which combine redevelopment and wrapping strategies in a phased approach to eventually achieve service-orientation [2].

A survey of 17 papers on SOA migration strategies revealed that "service-wrapping" of legacy system functionality is by far the most ubiquitous technique used, as this approach was found to be more cost effective, less risky, easier, and faster to implement [7]. Service-wrapping based SOA migration strategies were found to have a higher business value and a higher technical value, as compared to the alternatives [7].

For a large European bank, it took several hundred people more than 5 years to complete their legacy CBS migration [5]. A large Netherlands bank spent 600 million Euros over 5 years to migrate their legacy systems to an SOA [8]. The State Bank of India spent over 7 years on their legacy systems migration [1]. It took Credit Suisse Bank more than 10 years to completely decouple their legacy CBS with a services layer [14].

Beyond decoupling legacy CBS's using service-wrapping techniques, i.e. the outcome of using the "starter kit" presented in this paper, there are approaches for migrating away from a monolithic CBS architecture to a cloud-based microservices architecture [12]. On a smaller scale, Danske Bank migrated their monolithic FX currency conversion system to a microservices-based equivalent [4].

III. Digital Banking Accelerator

As established above, legacy systems are an inhibitor of digital banking capability [11], and SOA is a key enabler to overcome this barrier [13,15]. Adding a layer of abstraction over legacy systems through "service-wrapping" [2,5,7], is an effective approach to achieving the flexibility and agility needed for digital banking. However, it takes several years for banks to achieve an effective level of SOA maturity [5,8,14].

This section describes the Digital Banking Accelerator (the "starter kit") prototyped by SMU and commercialized by IISI, in the context of LTB Bank. Components of the "starter kit" are illustrated in Fig. 1 below, and are described in sub-sections A-H which follow.

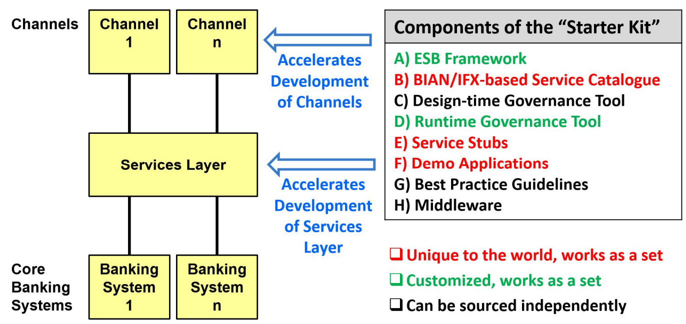


Fig. 1. Components of the Digital Banking Accelerator ("Starter Kit")

For the rest of this section we will refer to the "starter kit", commercialized by IISI, as "SOA-Kit".

A. *ESB Framework*

ESB Frameworks are typically not purchased off-the-shelf, rather they are architected and custom built for each bank. LTB Bank had the option of architecting a framework themselves, or purchasing professional services from their middleware vendor. Either way, implementing an ESB framework from scratch would have taken them more than a year, and they would have missed their regulatory deadline for going live. Hence, they decided to use the ready-made ESB Framework bundled with SOA-Kit. An SOA layered architecture is illustrated in Fig. 2 below, with the ESB Framework positioned in the middle layer.

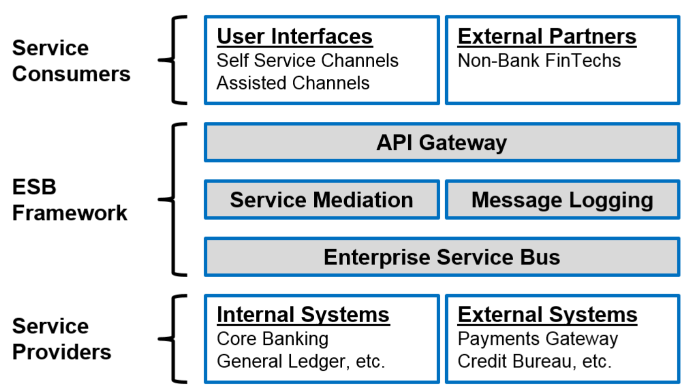


Fig. 2. SOA Layered Architecture of a Digital Bank

SOA-Kit includes a fully developed implementation-ready ESB Framework including the following sub-components:

1) *Enterprise Service Bus*

A collection of reusable services. The ESB exposes the functionality of the underlying banking systems as distinct services which are built once and reused by multiple channels. The ESB adds a layer of abstraction such that service consumers are decoupled from service providers.

2) *Service Mediation*
Provides runtime control of service usage. Implements a channel-to-service mapping such that specific channels can be configured to access specific versions of a service, at runtime. This enables multiple versions of a service to coexist such that specific requirements of various channels can be satisfied concurrently.

3) *Message Logging*
A logging framework which captures the service request messages from channels, at multiple configurable logging points. The message logs enable the reporting of runtime service usage statistics.

4) *API Gateway*
A gateway which routes RESTful JSON (Javascript Object Notation) formatted service request messages (API commands) to the Service Mediation component. Enables cloud-based access from internal banking channels, as well as from external non-bank FinTech firms.

B. BIAN/IFX-based Service Catalogue

Prior to this project, LTB Bank had no experience with SOA, and had no idea how to define and design services for a digital bank. Fortunately, SOA-Kit includes a complete catalogue of service definitions. The service catalogue is a standards-based library of WSDL's (Web Services Description Language) which are BIAN (Banking Industry Architecture Network) compliant at the service domain level, and IFX (Interactive Financial Exchange) compliant at the message level.

The Service Catalogue includes a set of API specifications. API commands, specified as RESTful web services using the JSON data format, expose the internal BIAN/IFX-based SOAP (Simple Object Access Protocol) web services. LTB Bank's channels (user interfaces) also invoke these API commands.

The Service Catalogue component of SOA-Kit is a new innovation. No other source is known where a bank can obtain a comprehensive set of BIAN/IFX-based WSDLs which are implementation-ready. LTB Bank benefited significantly from this pre-defined set of services which accelerated their SOA implementation.

C. Design-Time Governance Tool

SOA-Kit includes a design-time governance tool preloaded with the entire BIAN/IFX-based service catalogue described above. This saved significant effort for LTB Bank's architects which would have otherwise needed to define services from scratch. The design-time tool features a service repository which may be extended to include bank-specific services as required, using a maker-checker workflow whereby service designers can promote services along a sequence of lifecycle states: proposed, tested, commissioned (deployed), modified (versioned), and decommissioned (un-deployed).

This design-time tool enabled LTB Bank's developers to search the repository for existing services which could be reused and assembled into different solutions. Developers could download the machine-readable WSDL file, which then could be imported into a standards-compliant software development tool capable of automatically generating the service request invocation code. Alternatively, developers could download the related API specification, and develop the JSON request/reply code by hand.

D. Runtime Governance Tool

LTB Bank needed a way to control and monitor the usage of services at runtime. SOA-Kit includes a runtime governance tool which accesses the Service Mediation and Message Logging components of the ESB Framework described above. The governance tool updates the service mediation rules at runtime. The message logging component captures the service request messages, and the governance tool reads the message logs at runtime to generate service usage statistics and service performance statistics.

The Service Endpoint Management feature provides a means to configure service endpoints uniquely identified by; service name, service operation, service version, and the system ID of the underlying banking system. Each service endpoint configuration includes a JMS endpoint binding and an HTTP endpoint binding, as well as the SOAP actions which are automatically generated and specified in the service WSDL file.

The Channel to Service Endpoint Mapping feature provided a means for LTB Bank to control service access at runtime. This governance tool updates mapping rules at runtime such that specific channels are configured to access specific versions of a service, which exposes the functionality of an underlying banking system identified by a specific system ID. For example, version 1 of a customer information service mapped to system ID 1 might be integrating to an Oracle Flexcube CBS for the requested customer information, and the same service version however mapped to system ID 2 might be integrating to an Infosys Finacle CBS. For LTB Bank, their services were initially mapped and integrated to a Temenos (T24) CBS.

The Service Usage Monitoring feature provided LTB Bank a graphical dashboard for monitoring runtime service usage over a given timeframe. The dashboard provides statistics on service usage by service consumers which the bank used as a basis for cost recovery from business units whom were consuming services via their respective channel applications. More importantly, LTB Bank used the dashboard to track service reuse rates, and published (internally) their cost avoidance due to service reuse in order to sustain a positive perception of their SOA implementation.

The Service Performance Monitoring feature provided LTB Bank a dashboard for monitoring the runtime performance of services over a given timeframe, in terms of response time in milliseconds. The Message Logging component of the ESB Framework supports multiple configurable logging points such that request/reply interaction with the underlying banking system can by separately measured. This monitoring capability enabled LTB Bank to identify slow performing services which were subsequently tuned for better performance.

E. Service Stubs

SOA-Kit includes an ESB which implements a collection of BIAN/IFX-based services as defined above in the Service Catalogue section. A conceptual view of the ESB is illustrated in Fig. 3 below.

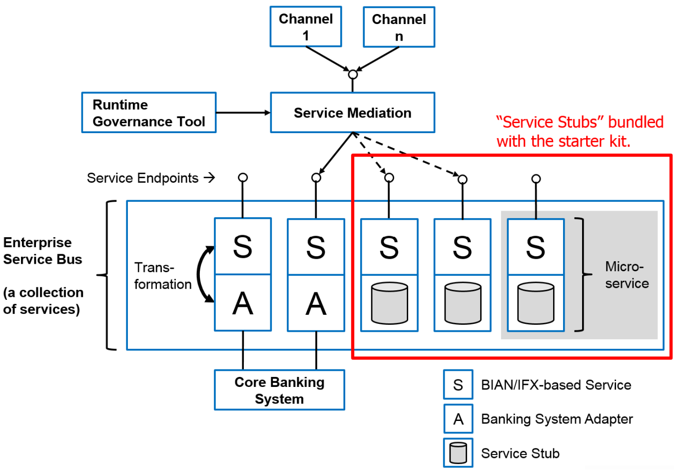


Fig. 3. Enterprise Service Bus (A Collection of Services)

What comes out-of-the-box is a set of service interfaces, labelled as "S" in Fig. 3 above, each invoking an underlying "Service Stub". The service interface and the Service Stub encapsulate the complete functionality of the service, including access to a database table that is owned by the service. The service architecture is such that the only access to a service's data is via the service's interface.

Out-of-the-box, all of the deployed Service Stubs are fully functioning, and LTB Bank used them as stand-ins for the functionality of their T24 CBS which they integrated to on the backend (behind the ESB). Having industry-standard BIAN/IFX-based services which are fully functioning out-of-the-box enabled LTB Bank to develop and test their frontend channel applications using Service Stubs as stand-ins, while integration to their T24 CBS on the backend of the ESB was developed concurrently.

Backend integration involves developing an adapter, labelled as "A" in Fig. 3 above, which exposes the functionality of the underlying banking system, and implements transformation between the data structure of the underlying banking system and the data structure of the BIAN/IFX-based services. LTB Bank developed and deployed these adapters using a phased approach, such that the frontend channel applications did not need to change any code during the transition.

Through the runtime control of the service mediation layer, as illustrated in Fig. 3 above, LTB Bank was able to switch one service consumer at a time from using the Service Stub to using their T24 CBS, without any impact to the service consumer during the transition. Taking this concept further, it is technically possible for LTB Bank to swing an entire service consumer (e.g. Internet Banking) from using their existing T24 CBS to using any new CBS, without impact to the service consumer. Taking this concept to an extreme, the ESB framework provided by SOA-Kit makes it technically possible for LTB Bank to completely replace their existing T24 CBS with any new CBS, without impacting any frontend channels. This concept has been proven by SMU tBank, where we replaced our legacy Oracle Flexcube CBS with an equivalent set of microservices [12].

F. Demo Applications

SOA-Kit includes a set of demo applications which invoke services via the API Gateway or directly via the Service Mediation component, as illustrated in Fig. 2 above. These applications can be categorized into 3 groups; applications which bank staff use, applications which bank customers use, and external non-bank FinTech applications which invoke services via the API Gateway.

LTB bank benefited by having these demo applications, including the source code, as part of SOA-Kit because: a) the bank's developers had working examples of how to invoke services, and b) the bank's developers leveraged the demo application source code as a starting point for developing their own applications. Alternatively, the bank's developers could have fully leveraged the demo applications by extending their features, security-hardening them, adding the bank's logo, and deploying them to production.

G. Best Practice Guidelines

The technical challenges of SOA notwithstanding, what are more difficult to overcome are the organizational and cross-functional challenges of managing SOA as a bank-wide asset [9,10]. SOA-Kit includes a set of best practice guidelines which helped LTB Bank to establish: who makes decisions, who provides funding, who owns the assets, how implementation is to be managed, and how success is to be measured.

LTB Bank benefited by having these best practice guidelines as part of SOA-Kit because: a) it accelerated their learning curve, having no prior experience with SOA, b) it reduced their dependency on high-priced vendor professional services consulting, and c) it reduced the time it took them to reach the point where they could quickly assemble new innovative digital banking solutions using their existing reusable services.

H. Middleware

If SOA-Kit is to be delivered as a complete solution to banks, then the SI offering this solution should be a licensed distributor of a specific middleware vendor not named in this paper. This is required because the ESB Framework and all of the Service Stubs mentioned above are developed using this specific middleware vendor's tools. What is distinctive about the specific middleware vendor selected for the SOA-Kit is the ease-of-use of their GUI-Driven Designer tool which supports nearly 100% codeless, and thereby rapid, development of services. In the case of LTB Bank, IISI was a licensed distributor of the specific middleware vendor.

IV. Discussion

A. Key Benefits

The key benefits of the Digital Banking Accelerator are listed as follows:

1) Accelerated learning curve on SOA implementation and governance, leveraging best practice guidelines.
2) Accelerated service design, leveraging a banking industry standards-based catalogue of service definitions.
3) Accelerated service implementation, leveraging a ready-made ESB framework and supporting tool set.

4) Accelerated banking channel development, leveraging a set of demo banking applications and service stubs.

B. Management Implications

Based on the LTB Bank implementation, the proposed Digital Banking Accelerator is expected to reduce the SOA implementation cycle time by at least one calendar year. Assuming, for example, a typical implementation team size of 50 at a man-day rate of $600 times 20 days per man-month, a one calendar year (50 man-year) reduction in effort would amount to a $7,200,000 cost avoidance for the bank. Beyond cost avoidance, the business agility and improved time-to-market capability enabled by a more flexible architecture would provide significant upside revenue potential, and overall improved profitability for the bank [10].

It is important to note that middleware vendors would not offer such a "starter kit" to digital banks for two reasons: 1) they lack the necessary banking domain knowledge, and 2) such an offering would cannibalize their after-sales professional services business. Therefore, SI's are in the best position to offer such a kit. As banks across Asia are urgently pursing their digital strategies, many of them having not yet invested in an SOA, there is considerable business potential for any SI that can offer such a "starter kit".

V. Conclusion

In this paper we proposed a Digital Banking Accelerator, an SOA "starter kit" for banks, which is comprised of eight components including; an ESB Framework, an industry standards-based Service Catalogue, a Design-time Governance Tool, a Runtime Governance Tool, a set of fully functioning Service Stubs, a set of Demo Applications, Best Practice Guidelines, and infrastructure Middleware. In collaboration with an SI, this accelerator has been successfully used to help digitize a large traditional Taiwanese bank. With the emergence of cloud platforms and native cloud applications, future work will be directed at adapting our "starter kit" to accelerate a microservices-based (coreless) banking system for new digital bank startups.

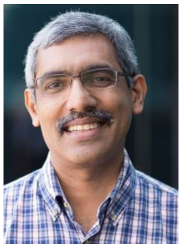
References

[1] Aggarwal, N., "Management report: Roadmap to successful core banking system replacement, critical success factors and best practices," *The Asian Banker, Tanjong Pagar, Singapore, ISBN,* pp. 981-05, 2006.
[2] Almonaies, A.A., Cordy, J.R., and Dean, T.R., "Legacy system evolution towards service-oriented architecture," in *International Workshop on SOA Migration and Evolution*, 2010, pp. 53-62.
[3] Basias, N., Themistocleous, M., and Morabito, V., "A Decision making framework for SOA adoption in e-Banking: A case study approach," *Journal of Economics, Business and Management,* vol. 3, no. 1, pp. 48-53, 2015.
[4] Bucchiarone, A., Dragoni, N., Dustdar, S., Larsen, S.T., and Mazzara, M., "From monolithic to microservices: An experience report from the banking domain," *Ieee Software,* vol. 35, no. 3, pp. 50-55, 2018.
[5] Galinium, M. and Shahbaz, N., "Factors affecting success in migration of legacy systems to service-oriented architecture (SOA)," *School of Economic and Management, Lund University,* 2009.
[6] Janssen, M., "Exploring the service-oriented enterprise: Drawing lessons from a case study," in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, 2008, pp. 101-101: IEEE.
[7] Khadka, R., Saeidi, A., Jansen, S., and Hage, J., "A structured legacy to SOA migration process and its evaluation in practice," in *Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), 2013 IEEE 7th International Symposium on the*, 2013, pp. 2-11: IEEE.
[8] Khadka, R., Saeidi, A., Jansen, S., Hage, J., and Haas, G.P., "Migrating a large scale legacy application to SOA: Challenges and lessons learned," in *Reverse Engineering (WCRE), 2013 20th Working Conference on*, 2013, pp. 425-432: IEEE.
[9] Kontogiannis, K., Lewis, G.A., and Smith, D.B., "A research agenda for service-oriented architecture," in *Proceedings of the 2nd international workshop on Systems development in SOA environments*, 2008, pp. 1-6: ACM.
[10] Kryvinska, N., Strauss, C., Auer, L., and Ivanochko, I., "Assessment of business value from services re-use on SOA-based e-business platform," in *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services*, 2010, pp. 493-500: ACM.
[11] Megargel, A., Shankararaman, V., and FAN, T.P., "SOA maturity influence on digital banking transformation," *IDRBT Journal of Banking Technology,* vol. 2, no. 2, p. 1, 2018.
[12] Megargel, A., Shankararaman, V., and Walker, D.K., "Migrating from Monoliths to Cloud-Based Microservices: A Banking Industry Example," in *Software Engineering in the Era of Cloud Computing*: Springer, 2020, pp. 85-108.
[13] Mueller, B., Viering, G., Ahlemann, F., and Riempp, G., "Towards Understanding the Sources of the Economic Potential of Service-Oriented Architecture: Findings from the Automotive and Banking Industry," in *ECIS*, 2007, pp. 1608-1619.
[14] Murer, S. and Hagen, C., "Fifteen years of service-oriented architecture at Credit Suisse," *IEEE Software,* vol. 31, no. 6, pp. 9-15, 2014.
[15] Schelp, J. and Aier, S., "SOA and EA-sustainable contributions for increasing corporate agility," in *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*, 2009, pp. 1-8: IEEE.

Alan Megargel (M'19) received a Doctor of Innovation degree from Singapore Management University, Singapore in 2018, and a Master of Science degree in software, systems, and information engineering from Sheffield University, Sheffield, England in 1996.

He is currently an Assistant Professor of Information Systems (Practice) at Singapore Management University (SMU), Singapore, where he serves as the faculty advisor and coordinator for their undergraduate financial technology track / major. Prior to joining SMU, he held various roles in industry including; Chief Technology Officer at TIBCO Software Asia, Vice President and Head of Service-Oriented Architecture at OCBC Bank, and Senior Enterprise Architect at ANZ Bank. His current research interests include; microservices architecture, digital banking, and financial technology.

Dr. Megargel is a member of IEEE, and a member of Singapore Computer Society (SCS).



Venky Shankararaman (M'12) received his PhD in Engineering from the University of Strathclyde, Glasgow, UK in 1992.

He is a Professor of Information Systems (Education) and Deputy Dean (Practice & Education) at the School of Information Systems, Singapore Management University. He has over 27 years of experience in the IT industry in various capacities as a researcher, academic faculty member, IT professional and industry consultant. He has published over 75 papers in academic journals and conferences. His current areas of specialization include digital business technologies and transformation, enterprise systems and integration, and education pedagogy.

Dr. Shankararaman is a member of IEEE and IEEE Education Society. He is the current President of the Special Interest Group in Education (SIGED), Association of Information Systems. He is a Senior Member of Singapore Computer Society (SCS).