Background

During my years working in the banking industry, I came to realize the importance of Enterprise Architecture (EA) and its role in implementing a bank’s business strategy while minimizing the overall IT cost for the bank. As a natural consequence of a growing bank, organizations tend to become “silo-ed”, with each business unit having its own dedicated tech & ops functions. While this organizational model is meant to be more agile in terms of time-to-market, the down-side effect is that bank-wide technology and platform standards are harder to enforce, resulting in higher overall IT costs for the bank. EA frameworks, platforms, and best practices are meant to address this problem.

SMU has embarked on a multiyear programme entitled “SMU Teaching Bank for Financial Services Education”, referred to as “SMU Teaching Bank” (or “SMU tBank”). Starting from a clean sheet, we are building a “teaching bank” from the ground up, using today’s architecture best practices.

I am the Architect and Champion of SMU tBank, and my academic research is anchored around the EA best practices demonstrated in SMU tBank’s implementation.

Mission Statement

“The mission of SMU tBank is to become a world class ‘teaching bank’, generating an on-going supply of undergrad and postgrad student projects whereby classroom learning outcomes can be put into practice, leveraging industry leading banking software and enterprise platforms.”

Guiding Principles

1. SMU tBank shall exist for academic purposes only, to support banking related coursework, labs, and student projects.

2. SMU tBank shall align to, and inform, SMU’s Unified Banking Process Framework.

3. SMU tBank shall be assembled using a mixture of vendor products, in order to demonstrate real world large-scale change scenarios, for example:
4. SMU tBank shall be a platform for collaboration with banks and product vendors, for incubating new innovations.

Conceptual Model
SMU has acquired several off-the-shelf banking products over the years from leading software vendors, in order to support hands-on labs for both post-grad and undergrad courses. SMU also uses leading middleware products to support some of its undergrad core technology courses. SMU tBank is then assembled leveraging this mix of vendor software into a flexible service-oriented-architecture, and student projects then develop banking applications that consume reusable services.
Banking channel prototypes developed by undergrad projects are integrated into the SMU tBank architecture, and are in turn used to support hands-on labs for both undergrad and post-grade courses.

Reference Architecture
The solution architecture for each component of SMU tBank shall align to and “refer” back to a common reference architecture. The Enterprise Platform layer of the architecture is key to enabling the on-going flexibility of SMU tBank as it evolves. Briefly the components of the Enterprise Platform layer are described as follows:

- **Enterprise Service Bus** (ESB) decouples systems within an industry best practice **Service Oriented Architecture**. Exposes functionality of banking systems as reuse-able services. Data abstraction is guided by the Banking Industry Architecture Network (BIAN) Service Definitions.

- **Channel Integration Layer** enables a 360° view of and by customers across multiple banking channels. Supports long running state-full business processes, eg; for multi-channel account origination, enabling origination to start on one channel and complete on another.

- **Payment Services Hub** orchestrates end-to-end payment processing, and interbank clearing & settlements. Provides full payment life cycle
support including; payment instruction validation, repair, credit check, FX, liquidity management, advice/statement, etc..

- **Master Data Management** (MDM) cross references customers to accounts across multiple banking systems. Contains customer master data, eg; the CIF record. Contains reference data for: Product codes, Account codes, Branch codes, etc..

- **Business Rules Management System** (BRMS) provides centrally managed business rules "externalized" from the processing logic within channel applications, for; credit decisioning, collections/NPL handling, product pricing (interest, fees), etc..

- **Content Management System** (CMS) provides centrally managed content delivered to customers across various channels, eg; product marketing content delivered via Internet Banking, Mobile Banking, email, post, etc..

- **Operational Data Store** (ODS) holds frequently accessed data within an in-memory data grid, for fast access, eg; customer profile, payment transactions, core banking system cache. Supports real-time analytics, eg; customer next best offer triggered by a customer interaction

*Figure 3 – Reference Architecture*
Deployment – Phase 1

The initial deployment of SMU tBank included all of the Retail Banking Channels, including: Branch Teller, Retail Internet Banking, Retail Mobile Banking (iOS & Android), ATM Network Management, and an ATM Simulation.

In the classroom, students work in groups covering different roles and scenarios. A customer walks into a branch and; opens up a CASA account, applies for a loan, applies for a term deposit, sets up a direct debit (GIRO) arrangement, pay’s their phone bill, etc..

Then the customer goes home and logs into the internet banking application use either of two forms of Two-Factor-Authentication, and then proceeds to; update their personal details, check their balance, set up fund transfer beneficiaries, make fund transfers, set up standing instructions, apply for a loan, opt in for marketing promotions, etc..

Then the customer uses the mobile app to do all of the same functionality as the internet banking, and also to perform a QuikPay person-to-person fund transfer using Near-Field-Communication.

ATM network managers (bank staff) plan and deploy ATM machines, monitor cash inventories and project top-up frequencies. Customer perform ATM transactions.

<table>
<thead>
<tr>
<th>Customer</th>
<th>Account Maintenance</th>
<th>Platforms Supported</th>
<th>Transactions</th>
<th>ATM Network Mgmt</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Account Opening</td>
<td>✓ Update Personal Details</td>
<td>✓ iOS</td>
<td>✓ Cash Withdrawal</td>
<td>✓ Setup New ATM</td>
</tr>
<tr>
<td>✓ PIN Creation</td>
<td>✓ Setup Preferences</td>
<td>✓ Android</td>
<td>✓ Fund Transfer</td>
<td>✓ Monitor ATM Network Utilization &amp; Uptime</td>
</tr>
<tr>
<td>✓ Credit Evaluation Accounts</td>
<td>✓ Setup Alerts</td>
<td></td>
<td>✓ Bill Payment</td>
<td>✓ Cash Inventory Management</td>
</tr>
<tr>
<td>✓ CASA</td>
<td>✓ Setup Likes/Interests</td>
<td></td>
<td></td>
<td>✓ Cash Top-up Forecasting</td>
</tr>
<tr>
<td>✓ Term Deposit</td>
<td>✓ Apply for Loan</td>
<td></td>
<td></td>
<td>✓ ATM Location Optimization</td>
</tr>
<tr>
<td>✓ Home Loan</td>
<td>✓ Apply for GIRO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Auto Loan</td>
<td>✓ Account Statement</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Education Loan</td>
<td>✓ Setup Beneficiaries</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(16 products total)</td>
<td>✓ Fund Transfer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transactions</td>
<td>✓ Standing Instruction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Deposit/Withdrawal</td>
<td>✓ Bill Payment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Bill Payment</td>
<td>✓ Transaction History</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Loan Repayment</td>
<td>✓ RM Scheduler</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ GIRO / Direct Debit</td>
<td>✓ Real-time Offers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Transaction History</td>
<td>✓ Merchant Discounts</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓ Bank Products</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4 – Retail Banking Channels
Deployment – Phase 2
One of the goals is to make SMU tBank available to other universities for the purposes of classroom teaching as well as collaborative development. To enable this, the architecture supports multiple instances of SMU tBank, such that each instance may have its own; base currency, reserve ratio, SWIFT routing number, interest rates, fees and charges, etc..

The multi-instance version of SMU tBank is deployed to Amazon Web Services (AWS), and is used in the classroom to teach enterprise architecture and banking concepts.

As of Oct 2014, SMU tBank is licensed by Ngee Ann Polytechnic for use in their Financial Informatics Diploma course.

Deployment of SMU tBank to the AWS cloud adds a whole new level of possible extensions, and since then the following software modules have been developed and deployed and are in use:

- **Automated Clearing House (ACH)** to orchestrate interbank payments between instances of SMU tBank. The ACH executes both real-time as well as batch instructions, perform sorting and routing of payment instructions between participating banks, perform netting on a schedule sending settlement instructions to the Central Bank (another instance of SMU tBank). Other features include; multicurrency, liquidity management, transaction volume and revenue reporting.

- **Corporate Internet Banking** featuring; Cash Management (Account Sweeping), and Payments; Credit Transfer and Direct Debit (GIRO). Possible use cases are, a) manufacturing company (eg; Seagate Technology) sends payment invoices to its suppliers, b) billing organization (eg; SingTel) invokes monthly billing of its customers.

- **Trade Finance** featuring a number of financial instruments related to international trade, such as; Letter of Credit, Bill of Exchange, Bill of Lading, Bank Guarantee, Documentary Collection, Open Account, Export Factoring. The solution includes a Trade Operations BPM process to manage the end-to-end trade process used in the classroom to illustrate the movement of goods and trade documents across the relevant parties, eg; Importer, Exporter, Freight Forwarder, Issuing Bank, Advising Bank.

- **General Ledger & Financial Reporting** features; Chart of Accounts setup and maintenance, GL posting rules management, Balance Sheet and Income Statement reporting. For interbank payments, dual-entry accounting is ensured in both sending and receiving bank ledgers.
Figure 5 – Interbank Payments Involving Multiple Instances of SMU tBank

![Diagram of interbank payments involving multiple instances of SMU tBank]

Figure 6 – Interbank Payments; Supported Payment Modes

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Settlement</th>
<th>Function</th>
<th>Sender</th>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Credit Transfer</td>
<td>Realtime</td>
<td>Fund Transfer</td>
<td>Retail</td>
<td>Corp</td>
</tr>
<tr>
<td>2 Credit Transfer</td>
<td>Realtime</td>
<td>Bill Payment</td>
<td>Retail</td>
<td>Corp</td>
</tr>
<tr>
<td>3 Credit Transfer</td>
<td>Realtime</td>
<td>Invoice Payment</td>
<td>Corp</td>
<td>Retail</td>
</tr>
<tr>
<td>4 Credit Transfer</td>
<td>Batch</td>
<td>Invoice Payment</td>
<td>Retail</td>
<td>Corp</td>
</tr>
<tr>
<td>5 Credit Transfer</td>
<td>Batch</td>
<td>Direct Deposit (Payroll)</td>
<td>Corp</td>
<td>Retail</td>
</tr>
<tr>
<td>6 Direct Debit</td>
<td>Batch</td>
<td>GIRO - Retail Account</td>
<td>Corp</td>
<td>Retail</td>
</tr>
<tr>
<td>7 Direct Debit</td>
<td>Batch</td>
<td>GIRO - Corp Account</td>
<td>Corp</td>
<td>Retail</td>
</tr>
<tr>
<td>8 *Cheque</td>
<td>Future</td>
<td>Cheque Payment - Retail</td>
<td>Retail</td>
<td>Corp</td>
</tr>
<tr>
<td>9 *Cheque</td>
<td>Future</td>
<td>Cheque Payment - Corp</td>
<td>Corp</td>
<td>Retail</td>
</tr>
<tr>
<td>10 Bill of Exchange</td>
<td>Future</td>
<td>Trade Finance Payment</td>
<td>Corp</td>
<td>Retail</td>
</tr>
</tbody>
</table>

*Cheque truncation: Sender captures image, Sender’s bank scans image & processes payment

Figure 6 – Interbank Payments; Supported Payment Modes
Benefits

Benefits of **Building** SMU tBank

- SMU tBank uses banking software that is developed by students. Students that develop the SMU tBank gain a deep technical understanding of how a bank works. As well as gaining banking domain knowledge, students will benefit from implementing industry best practices in enterprise architecture. Note: We believe that we are the only University in the world that is going to the extent of actually building a bank, for teaching purposes.

Benefits of **Using** SMU tBank

In the classroom, the labs are useful for both undergrad and post grad.

- **IS419 Retail Banking** – Students use SMU tBank to learn banking processes such as; account opening, credit evaluation, loan repayments, fund transfers, foreign exchange, standing instructions, GIRO, mobile payments, Two-Factor-Authentication, ATM network management, real-time customer specific promotion offers. Lab questions assess the students understanding of both bank processes as well as financial accounting.

- **IS430 ePayments** – Students use SMU tBank to understand how interbank payments works through an Automated Clearing House (ACH), from different perspectives, a) corporate and retail customers, b) participating banks, and c) central bank. Lab exercises will include; payment initiation from corporate customers for both credit transfer and direct debit (GIRO), and bank liquidity management demonstrating scenarios whereby a participating bank has insufficient funds during net settlement with the central bank.

- **IS301 Enterprise Integration** – Students use SMU tBank to learn integration technologies such as; Message-Oriented-Middleware, and Service-Oriented-Architecture. Labs exercises include; building integration components that allow different applications in the bank to communicate, and drill-down visualizations of what is actually happen in the integration layer when a fund transfer is executed, for example.

- **IS303 Architectural Analysis** – Students will use a “lite” version of SMU tBank which is deployed on their laptops, to demonstrate their understanding of 3 main architecture principles; “resiliency” (ability to failover to a standby system), “concurrency” (handling large number of users), and “performance” (response time of the application).
Research Areas

Anchored around the implementation of SMU tBank, I have in mind to spin off a few areas of research, described briefly as follows:


2. “In-Memory Data Grid Use Cases in Banking” – Covers performance improvements of data caching, eg; characterization of response times with and without a data cache in front of the core banking system, and the resulting impact on customer satisfaction. Also covers the economics of caching data in front of the core banking system, eg; the cost saving in MIPS incurred on mainframe systems. Covers how massive-scale in-memory data grid technology is used to enable real-time cross-sell to banking customers, eg; next best offers pending in memory, triggered on the next customer interaction. Covers real-time fraud detection.

3. “Core Banking System Replacement” – Covers the scenario where a core banking system is replaced, eg; Oracle Flexcube is replaced with Infosys Finacle. The transition from one system to the other can be done with minimum impact to banking channel applications, by using a flexible service oriented architecture. Using SMU tBank as a test bed, specific scenarios can be trialed in conjunction with actual banks in Singapore that want to participate in the study.

4. “Bank Mergers: Technology Migration or Coexistence” – Covers the scenario whereby one bank acquires another, and the combined bank needs to make decisions about which technology to keep or discard, and which technologies can coexist. The coexistence of different technologies across the two banks can be achieved with minimal impact to banking channels, by using a flexible enterprise platforms such as; SOA, BPM, BRMS, and MDM. Using SMU tBank as a test bed, specific scenarios can be trialed in conjunction with actual banks in Singapore that want to participate in the study.

5. “Banking Industry Information Model” – Covers the adoption of the Banking Industry Architecture Network (BIAN) Service Landscape as the enterprise data model for SMU tBank. The study will demonstrate, in actual practice, how an industry model can inform and optimize the decomposition of banking processes into reusable services.
Selected Publications and Outputs

1. Chapter Title: Enterprise Integration: Architectural Approaches (co-authored with Venky Shankararaman). The chapter traces the evolution of Enterprise Integration Architectures and will be of great value to anyone working in the area of integration. We also intend to use this content in the Enterprise Integration course. The chapter is part of an edited book entitled: Service-driven Approaches to Architecture and Enterprise Integration/Raja Ramanathan. Published by: IGI Global (2013)

Doctoral Dissertation Synopsis
I am currently taking the Doctor of Innovation programme offered by SMU School of Business. It is a 4 year programme, and I expect to complete the programme in 2019. Wherever possible, I compose my written assignments within the context of main research area which is Enterprise Architecture in Banking. The dissertation synopsis is as follows:

Digital Banking: Overcoming Barriers to Entry

“Digital Banking – a new concept in the area of electronic banking, which aims to enrich standard online and mobile banking services by integrating digital technologies, for example strategic analytics tools, social media interactions, innovative payment solutions, mobile technology and a focus on user experience” (Moeckel, 2013, March).

There is no standard definition of Digital Banking, however the above definition encompasses most of the concepts discussed in the literature. “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 much more on customer experience.

Industry Structure. Digital Banking, which mostly falls within the boundary of retail banking as an industry, is shaped by “five competitive forces” (Porter, 2008); 1) Customers are becoming increasingly more sophisticated, expecting personalized banking services to be delivered to them at any time of the day across any channel, 2) Substitutes such as FinTech and other IT companies are providing non-bank alternative financial services such as payments, peer-to-peer lending, and crowdfunding, 3) Entrants along with substitutes are more agile than traditional banks, have lower barriers to entry in terms of regulatory controls and legacy infrastructures, and are in fact defining the standards for Digital Banking, 4) Incumbents who are traditional banks are urgently pursuing digital strategies in order to salvage their diminishing market shares, but are severely inhibited due to their existing inflexible monolithic legacy systems, and 5) Suppliers such as technology vendors have relatively limited banking domain
knowledge, and their further offerings of commercial-off-the-shelf applications will not help traditional banks to unravel their legacy architectures.

**Barriers to Entry.** Ironically, it is the incumbent traditional banks which have the highest barriers to entry into the Digital Banking market. Legacy core banking systems which are inflexible to change are at the “heart” of the problem, and replacing an existing core banking system on a live bank would be analogous to performing a heart transplant on a runner during a race. Bank management is essentially about managing risk, and the impact of a failed core banking system “transplant” is too high for most bank managers to consider as a viable option. Therefore, core banking system replacements are rare, as many banks are still using legacy mainframe technologies built in the 1970s. These legacy systems, while still high performing and reliable, are inflexible to change and are not easily integrated to the modern application systems needed for delivering Digital Banking services across multiple online banking channels. The literature reveals universally that the solution for overcoming this obstacle is to implement a Service-Oriented Architecture (SOA) whereby the functionality of underlying legacy systems can be exposed as reusable services which are easily consumed by Digital Banking channels.

The SOA style of architecture is proven to be more flexible, enabling banks to become more agile, with improved time-to-market delivery of new products and services across multiple channels. There is much written about the benefits of SOA in banking, as well as the lessons learned from failed SOA implementations. Banks that implement SOA well are in the best position to compete in the Digital Banking market. However, many banks have either failed to realize the benefits of an SOA, or have not yet invested in an SOA. The case studies discuss the benefits analysis and business case development that led to technology investments into SOA. However, there is little written about why banks decide not to invest in an SOA. Only one research paper could be found that reveals banks’ rationale for not investing in SOA. Schulte, Repp, Eckert, Berbner, Blanckenburg, Schaarschmidt, and Steinmetz (2008) revealed the top three inhibitors to be; “Lack of experience”, “Organizational restrictions”, and “Technologies are not standardized”. The proposed innovation central to this dissertation addresses these three inhibitors.

**Proposed Innovation.** “The primary integration design pattern that implements an SOA is the Enterprise Service Bus (ESB)*, whereby “services are developed once and then used by multiple consumers” (Shankararaman & Megargel, 2013). Implementing an ESB in a bank, which typically has several hundred applications systems, is not easy. This author can attest to that first hand. Banks having no prior experience have a steep learning curve, and can take several years to fully implement an ESB. There are technical challenges in understanding integration technologies, and organisational challenges in governing decisions that cut across functional boundaries.
The proposed innovation is an “ESB Starter Kit for Banks” which will essentially de-risk banks’ adoption of SOA and accelerate banks’ entry into the Digital Banking market. The ESB Starter Kit is expected to reduce the implementation cycle time by at least one calendar year. Assuming 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. Pricing of the ESB Starter Kit might then be in the range of $5,000,000.

Components of the ESB Starter Kit are proposed as follows:

- **Service Catalogue** – A complete library of WSDLs (Web Services Description Language) which are BIAN (Banking Industry Architecture Network) complaint at the service domain level, and IFX (Interactive Financial Exchange) compliant at the message level.
- **Design Time Governance Tool** – Allows channel developers to search for and download WSDLs that are uploaded by service developers. Includes workflow control.
- **Service Stubs** – A complete set of stubs implemented behind each WSDL which simulate core banking functionality. Used for testing service interfaces by channels.
- **Demo Applications** – A set of Digital Banking channel applications which consume the complete set of services exposed via the ESB. Includes Mobile Banking, Retail Internet Banking, and Corporate Internet Banking: Trade Finance and Payments.
- **Best Practice Guidelines** – Documentation including; a) Architecture Principles and Patterns, b) Architecture Governance Framework, c) Integration Competency Center Best Practices.
- **Middleware** – Infrastructure software used to integrate enterprise applications. Used to implement the ESB Framework and Service Stubs.

*Note 1: A standards compliant library of WSDLs for banking, as described, is new to the world.
Note 2: An “ESB Starter Kit for Banks” with all of the above components is new to the world.

The ESB Starter Kit would be offered by a System Integrator (SI) such as International Integrated Systems Inc. (IISI) of Taiwan, a partner of this author. Middleware vendors would not offer such a kit 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, SIs are in the best position to offer such a kit. IISI has access to several banks in Northern Asia.
which have not invested in an SOA, and they are actively marketing the kit while it is still under development. Indonesia has many banks that have not invested in SOA. One Indonesian bank in particular has spun off a Digital Bank entity, while still using the parent bank’s legacy core system, and are currently looking for an SI that can help them with their SOA implementation. As banks across Asia are urgently pursing their digital strategies, many of them not yet invested in an SOA, there is considerable business potential for any SI that can offer such an ESB Starter Kit.

**Contribution.** Contribution to research will be in the area of Digital Banking Inhibitors. Contribution to practice, partnering with an SI, will be; a) a standards compliant library of WSDLs for banking, as described, which will be new to the world, and b) an “ESB Starter Kit for Banks”, as described, which will be new to the world. Following this dissertation we plan to write a practice based book, “a how to guide”, detailing architecture of the innovation, targeting both academics and industry.

**References**


