SOBEY
SCHOOL OF BUSINESS

MASTER OF TECHNOLOGY
ENTREPRENEURSHIP
AND INNOVATION

PIP: A (Privacy) Injection Pattern for Inserting Privacy Patterns and Services in Software

Naureen Ali, Dawn N. Jutla, Peter Bodorik

Dawn N. Jutla , PhD, Director, Board of OASIS
Scotiabank Professor of Technology Entrepreneurship, Sobey School of Business, Saint Mary's University
co-Chair/co-editor, OASIS PbD-SE with Dr.Ann Cavoukian; co-editor, OASIS PMRM

ANNUAL PRIVACY FORUM
OCT 7-8, 2015
LUXEMBOURG

SOBEY
SCHOOL OF BUSINESS

MASTER OF TECHNOLOGY
ENTREPRENEURSHIP
AND INNOVATION

R&D and EMERGING Standards to make **Privacy-by-Design** **Instinctual** on the **Internet**

FOR EVERY ORGANIZATION AND SOFTWARE ENGINEER – ON PURPOSE, IN A MANAGED WAY

OASIS
Advancing open standards for the information society

GARTNER 2014 PREDICTS:
By 2017, 80% of consumers will
collect, track and barter
their personal data for cost savings,
convenience and customization.

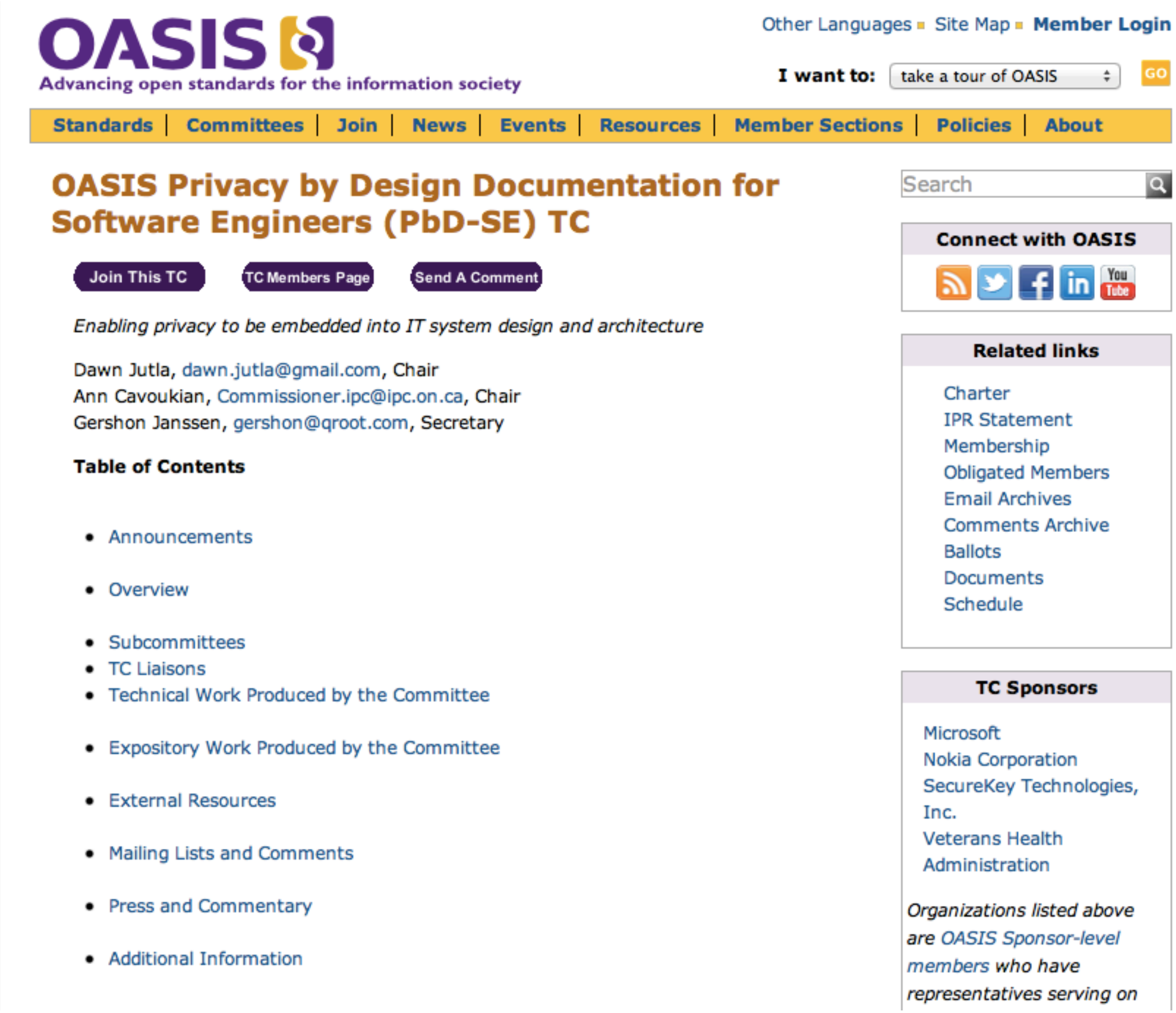
Why should business care … about consumer privacy & empowerment over personal data?

- Loss of customers, customer loyalty, dimishing stock value, brand reputation
- Increased legal costs, class action lawsuits
- Shareholder and board dissatisfaction

Problem Observations

- Increasing leakage and theft of sensitive and private data.
- Software engineers are unaware of privacy requirements at the design phase
- Protect by incorporating privacy requirements during a system’s design phase.
- Engineers in organizations are looking for repeatable ways to embed privacy in their code.

OASIS PbD-SE



OASIS Privacy by Design Documentation for Software Engineers (PbD-SE) TC

Enabling privacy to be embedded into IT system design and architecture

Dawn Jutla, dawn.jutla@gmail.com, Chair
Ann Cavoukian, Commissioner.ipc@ipc.on.ca, Chair
Gershon Janssen, gershon@qroot.com, Secretary

Table of Contents

Related links

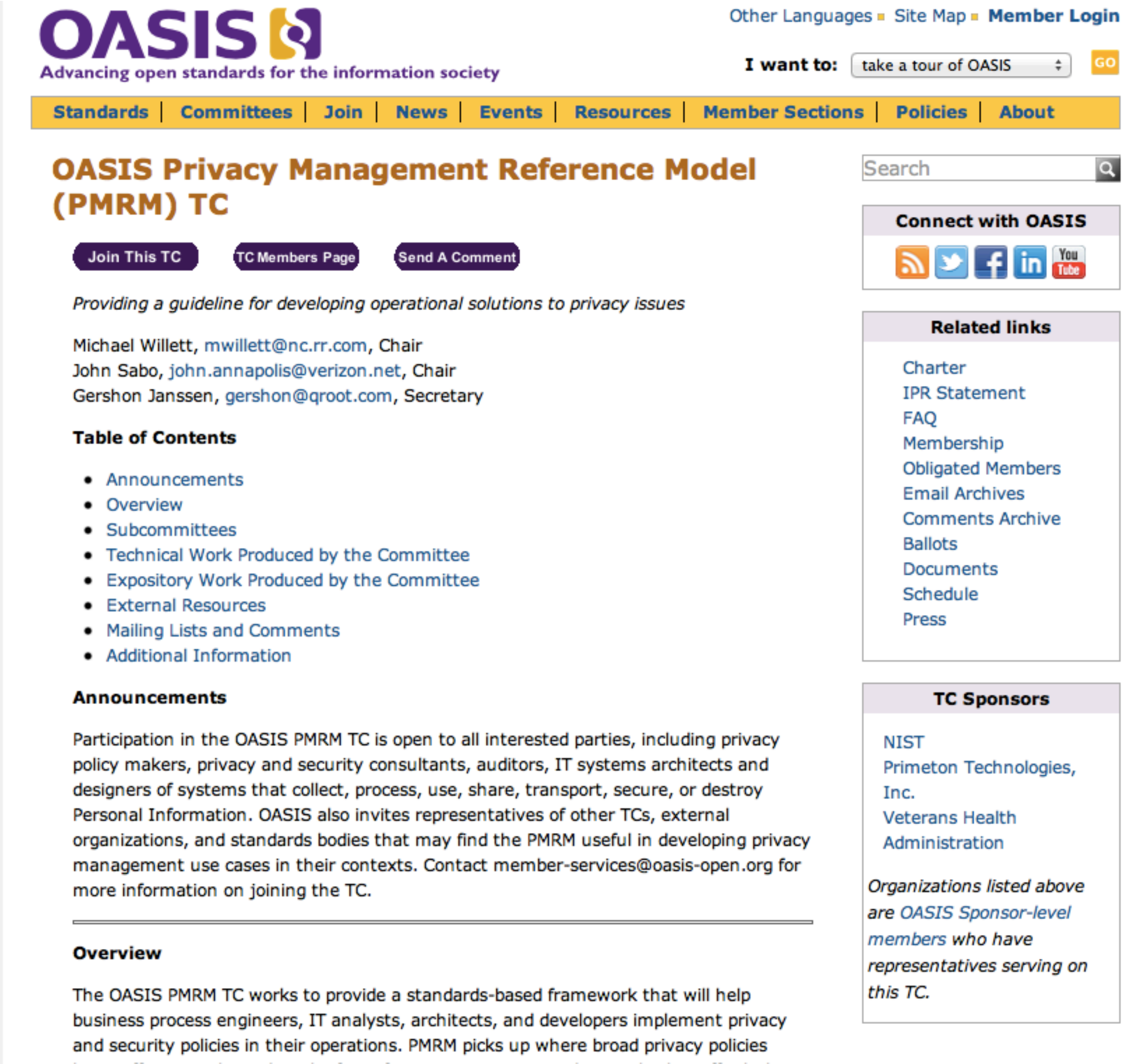
Charter
IPR Statement
Membership
Obligated Members
Email Archives
Comments Archive
Ballots
Documents
Schedule

TC Sponsors

Microsoft
Nokia Corporation
SecureKey Technologies, Inc.
Veterans Health Administration

Organizations listed above are OASIS Sponsor-level members who have representatives serving on

OASIS PMRM



OASIS Privacy Management Reference Model (PMRM) TC

Providing a guideline for developing operational solutions to privacy issues

Michael Willett, mwillett@nc.rr.com, Chair
John Sabo, john.annapolis@verizon.net, Chair
Gershon Janssen, gershon@qroot.com, Secretary

Table of Contents

Announcements

Participation in the OASIS PMRM TC is open to all interested parties, including privacy policy makers, privacy and security consultants, auditors, IT systems architects and designers of systems that collect, process, use, share, transport, secure, or destroy Personal Information. OASIS also invites representatives of other TCs, external organizations, and standards bodies that may find the PMRM useful in developing privacy management use cases in their contexts. Contact member-services@oasis-open.org for more information on joining the TC.

Overview

The OASIS PMRM TC works to provide a standards-based framework that will help business process engineers, IT analysts, architects, and developers implement privacy and security policies in their operations. PMRM picks up where broad privacy policies

Related links

- Charter
- IPR Statement
- FAQ
- Membership
- Obligated Members
- Email Archives
- Comments Archive
- Ballots
- Documents
- Schedule
- Press

TC Sponsors

NIST
Primeton Technologies, Inc.
Veterans Health Administration

Organizations listed above are OASIS Sponsor-level members who have representatives serving on this TC.

OASIS Privacy by Design Documentation for Software Engineers (PbD-SE) TC

1 PbD principles are internationally recognized with mappings/ alignment to FIPPs, GAPPs and NIST 800-53 Appendix J controls.

2 Help stakeholders to **visualize** privacy requirements and design from software conception to retirement

3 A specification of mappings, conformance maturity levels, and guidance to help software engineers to :

- Model and translate Privacy by Design (PbD) principles to conformance requirements within software engineering tasks
- Produce privacy-aware software, and document artifacts as evidence of PbD-principle compliance.
- Collaborate with management and auditors to *simplify* demonstration of compliance/audits.

OASIS Privacy Management Reference Model and Methodology (PMRM) Emerging Standard

TC Chair: John T. Sabo, Retired TC co-Chair: Michael Willett

1 PMRM provides a model and methodology for translating & mapping privacy requirements,, as the basis for a PRIVACY SERVICE ARCHITECTURE: http://j.mp/oasisPMRM

2 KEY STRENGTH: Gets at how personal data flow among data platforms... 360 stakeholder view of privacy requirements.

3 Major elements of this emerging standard's methodology and the PbD-SE methodology align with the state-of-the-art

OASIS

Advancing open standards for the information society

Standards applicable to ….

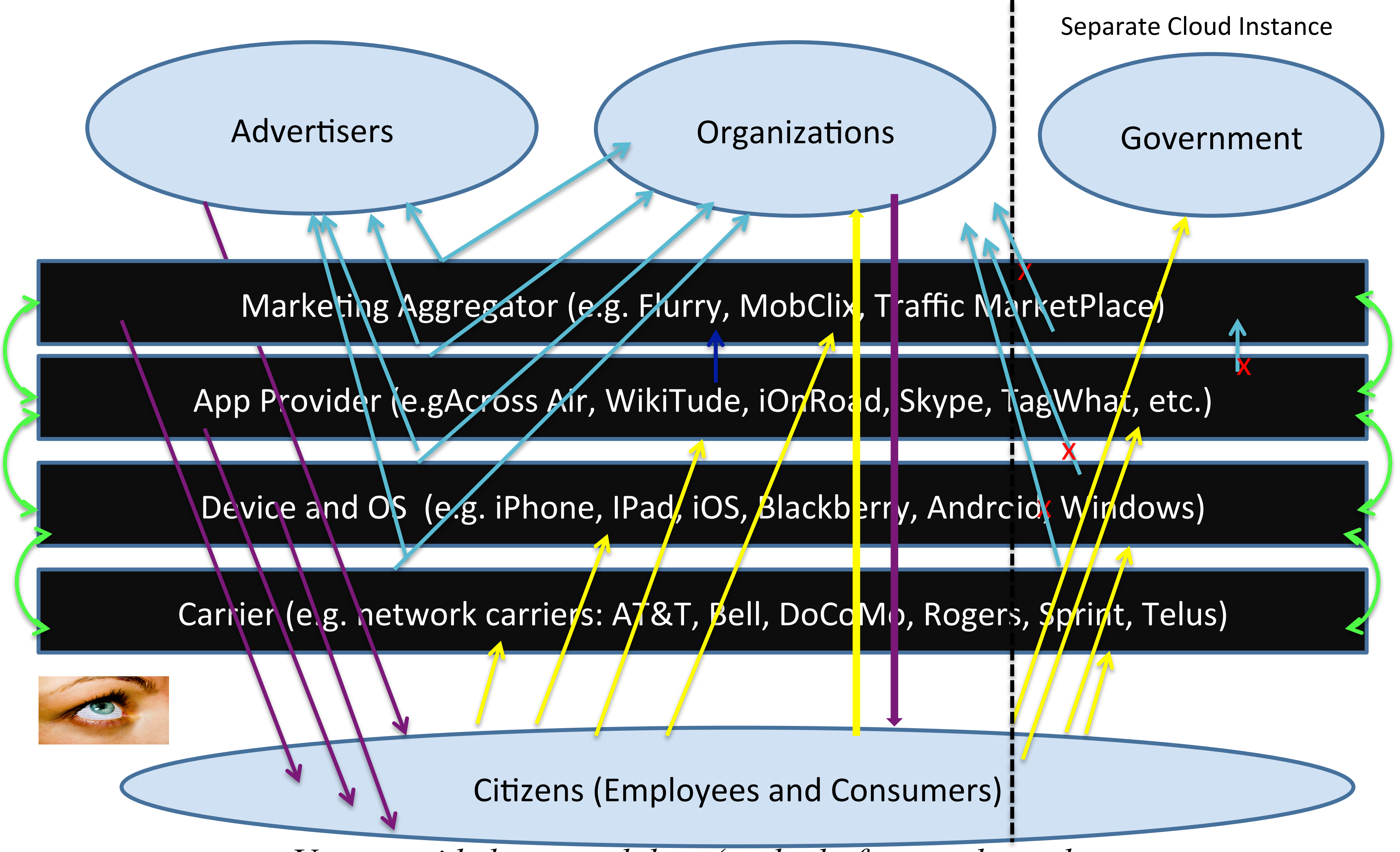
Applicable to all organizations and individuals producing Information Technology Products and Services

Software Engineer: A person that adopts engineering approaches, such as established methodologies, processes, architectures, measurement tools, standards, organization methods, management methods, quality assurance systems and the like, in the development of large scale software, seeking to result in high productivity, low cost, controllable quality, and measurable development schedule.

Source: Adapted from Y. Wang, Senior Member of the IEEE and ACM. Theoretical Foundations of Software Engineering, Schulich School of Engineering, University of Calgary, 2011.

Large scale software extends to include apps that scale to millions of users

Organizations and individuals adopting design processes, privacy methodologies and standards to obtain better user privacy going forward.



Separate Cloud Instance

Advertisers

Organizations

Government

Marketing Aggregator (e.g. Flurry, MobClix, Traffic MarketPlace)

App Provider (e.gAcross Air, WikiTude, iOnRoad, Skype, TagWhat, etc.)

Device and OS (e.g. iPhone, IPad, iOS, Blackberry, Android, Windows)

Carrier (e.g. network carriers: AT&T, Bell, DoCoMo, Rogers, Sprint, Telus)

Citizens (Employees and Consumers)

→ *User-provided personal data (each platform and merchant may get different data attributes) in a single service*

→ *User profiles sent to advertiser networks, aggregators, and to merchants*

→ *Ads, offers, deals etc.*

→ *Personal data flows between platforms.*

© Dawn N. Jutla

Observations in a Narrower Research Context

- Software engineers' productivity improve with the recognition and use of repeatable patterns.
- Challenging to implement privacy services in existing and/or legacy applications without affecting other modules as compared to incorporating privacy in new applications.
- Numerous privacy patterns exist.
- Once a privacy pattern is identified, the pattern or its service implementation may be called or automatically "injected" into existing or new software.

Privacy patterns…

- Porekar et al (2008) classify organizational privacy patterns as:
  - Obtaining explicit consent”
  - “Access control to sensitive data based on purpose”;
  - ”Time limited personal data keeping”
  - “Maintaining privacy audit trails”,
  - “Creating privacy policy, Maintaining (versions of) privacy policies”,;
  - Privacy negotiation.
- Doty and Gupta (2013) discuss a privacy policy as a pattern and reference Hoepman’s work (2012, 2014) on privacy strategies and categorization of privacy patterns.
- Others (e.g. Hafiz (2006)) discuss collections of privacy patterns.
- Romanosky et al(2006)] specify three privacy patterns
  - informed consent for web-based transactions
  - masked online traffic and
  - minimal information asymmetry

NIST 800-53 Appendix J Control Families

- Authority and Purpose (AP)
- Accountability, Audit, and Risk Management (AR)
- Data Quality and Integrity (DI)
- Data Minimization and Retention (DM)
- Individual Participation and Redress (IP)
- Security (SE)
- Transparency (TR) Use Limitation (UL)
- NIST EQUATION: Vulnerability * Threat = Likelihood; Security Risk – Likelihood * Impact.

Implementing Privacy Patterns with AOP

- Used individually in the past to implement security and access control method extensions (e.g. Sharma et al, and Win and Piessens 2005)
- Chen, K., & Wang, D.-W. (2007). An Aspect-Oriented Approach to Privacy-Aware Access Control. Proceedings of the Sixth International Conference on Machine Learning and Cybernetics (págs. 3016 - 3021). Hong Kong: IEEE
  - Chen and Wang use AOP as a mechanism to implement privacy-aware access control. In their work, application-level access control is extended to enforce privacy policies on personal data using AOP.
- Sharma, N., Batra, U., & Mukherjee, S. (2014). Enhancing Security in Service Oriented Architecture driven EAI using Aspect Oriented Programming in healthcare IT. International Journal Of Scientific & Engineering Research, Volume 5, Issue 3 , 50-5
  - Sharma, et al, propose using AOP for the secure transfer of data over the Internet. They implement privacy patterns for encrypting/decrypting data and key generation using hashing as aspects performed by security agent.
- Win, B. D., Joosen, W., & Piessens, F. (2005). Developing secure applications through aspect-oriented programming. En Aspect-Oriented Software Development (págs. 633-650). Addison-Wesley.
  - Win et al [4] also use AOP for security and transmission privacy.

DI & Security & Architecture

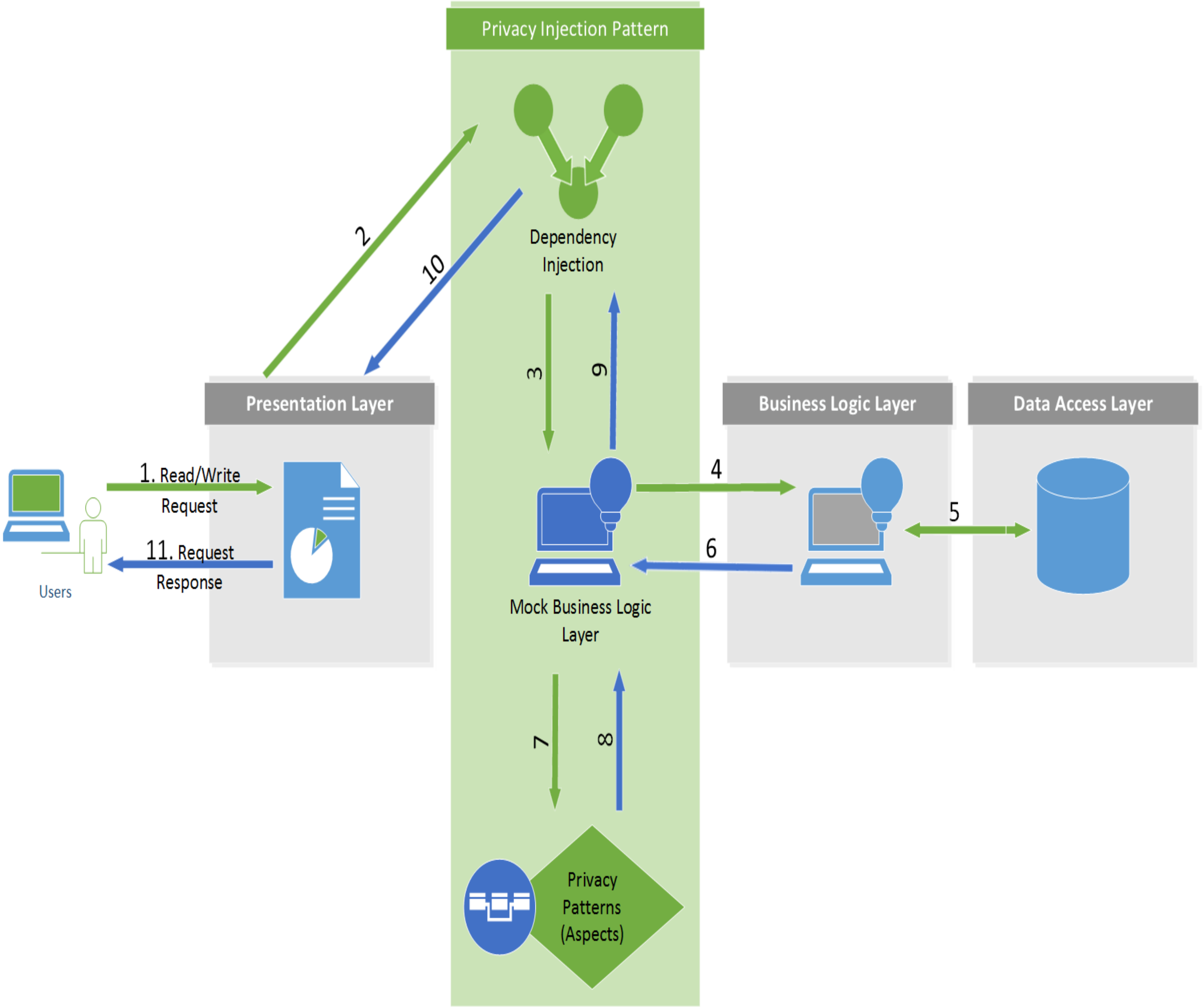
- Benenson et al. (2006) propose a smart card based framework for Secure Multiparty Computation (SMC). Their model consists of multiple processes having a security module that securely interacts with the security modules of other processes. The authors use DI to configure the component that selects the actual algorithm at runtime without recompiling the code.
- Livne et al (2011) present a health care architecture using dependency injection, AOP, and XML configurations to make the architecture flexible, reusable, loosely coupled and service-oriented.
- Almorsy et al. (2012) propose a novel service called VAM-aaS (Vulnerability Analysis and Mitigation as-a-service) to mitigate the security vulnerabilities in the cloud environment.
  – It analyzes the online services. and in case of vulnerabilities injects security handler classes at runtime based on the required mitigation actions of that vulnerability.

Prior Work - Mock

- Mocking has also been used to provide a simple fake-data pattern to applications to preserve privacy.
- Beresford et al (2011) propose a modified version of the Android operating system called MockDroid to mock resources accessed by an application. For example, in an application that requests IP connectivity, location data, read-write access to calendar data, the user may provide mock data instead of actual data to the application.
- Hornyack et al (2011) and Zhou et al (2011), also propose to provide fake or empty data to software applications that require access to users' personal data. A user may view all the permissions that an application requests at the time of installation of the application and then select one of the four modes (trusted, anonymous, bogus, or empty) for each of the permissions.

Summary of Prior Work

- Our Privacy Architecture for Web Services (PAWS – 2009-now)
  - semi-dynamically injects privacy web services for notice and consent in existing web pages built on ROA architectures.
- Software patterns that *fully automate injecting* privacy patterns are not found in the literature.

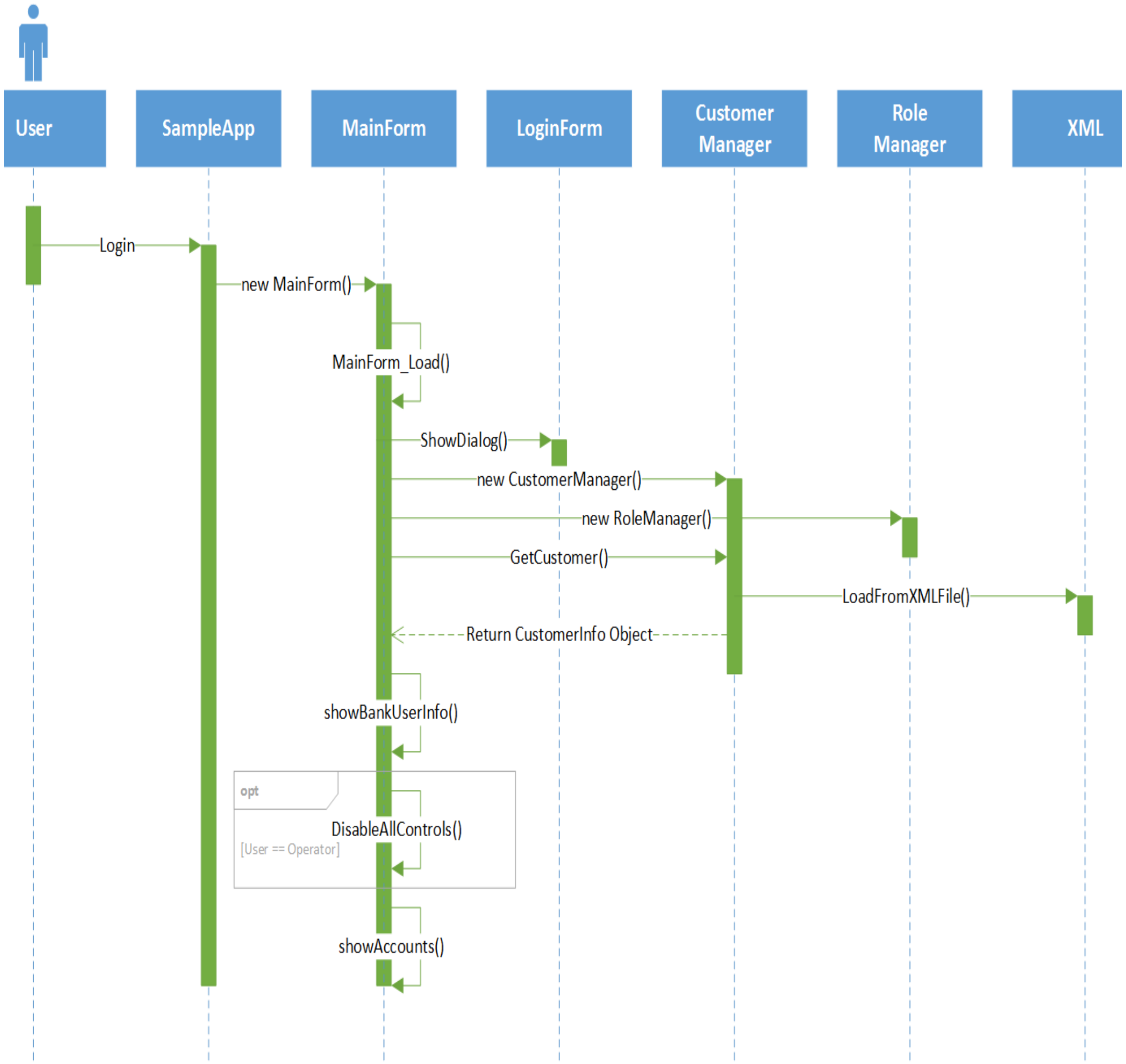

Privacy Injection Pattern
Dependency Injection
2
10
3
9
Presentation Layer
Business Logic Layer
Data Access Layer
1. Read/Write Request
11. Request Response
Users
4
5
6
Mock Business Logic Layer
7
8
Privacy Patterns (Aspects)

Implementing De-identification Patterns

- Use case scenario from a banking application that uses de-identification patterns.
- Data de-identification is a privacy-preserving technique.
  - substitution, shuffling, nulling out, character masking and cryptographic techniques.
- We implement the nulling out and character masking privacy patterns for illustration using AOP in our example.
- The mocking and the dependency injection techniques automatically inject the AOP instance of the de-identification service.

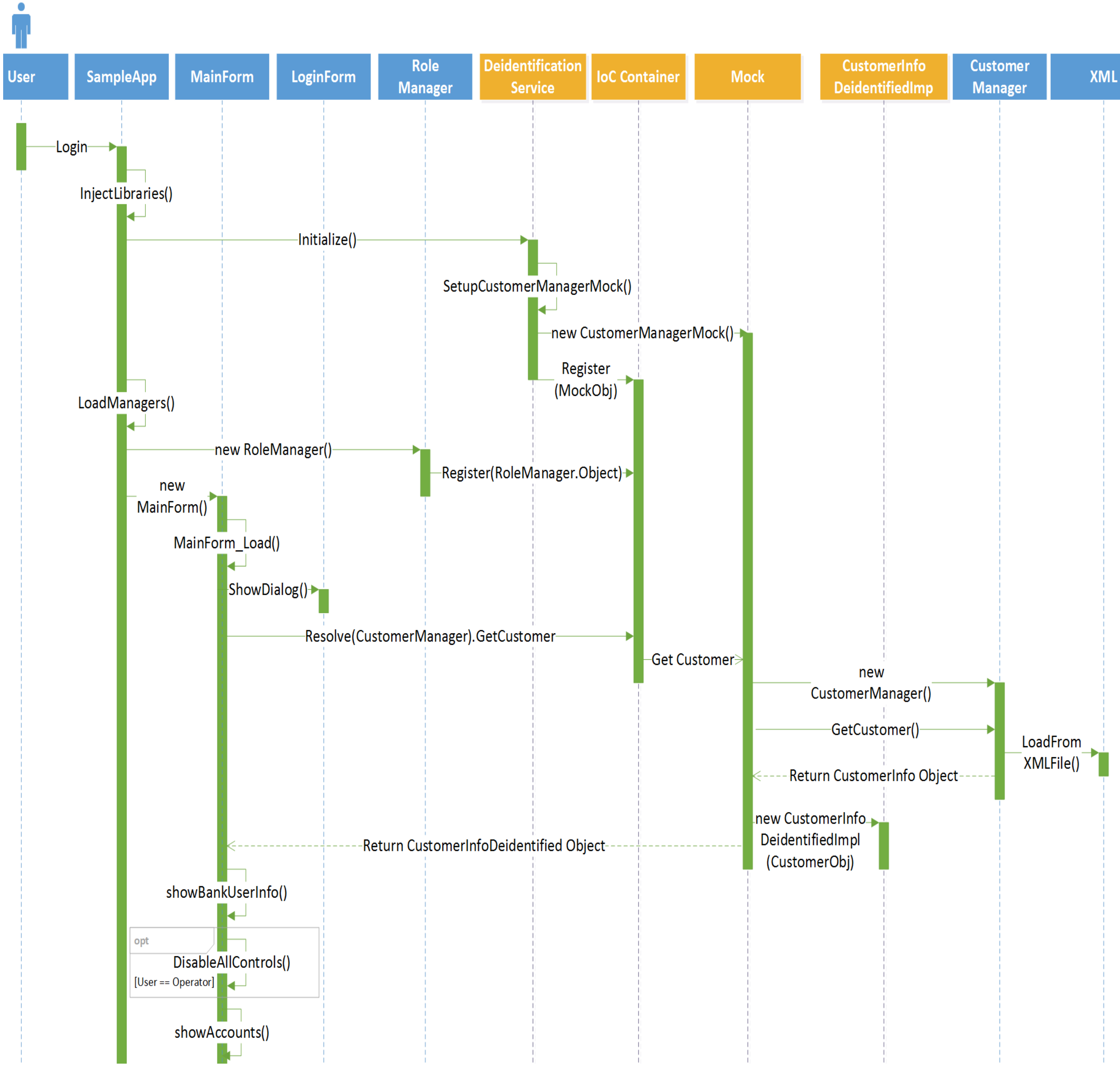
Implementation

Before Applying PIP

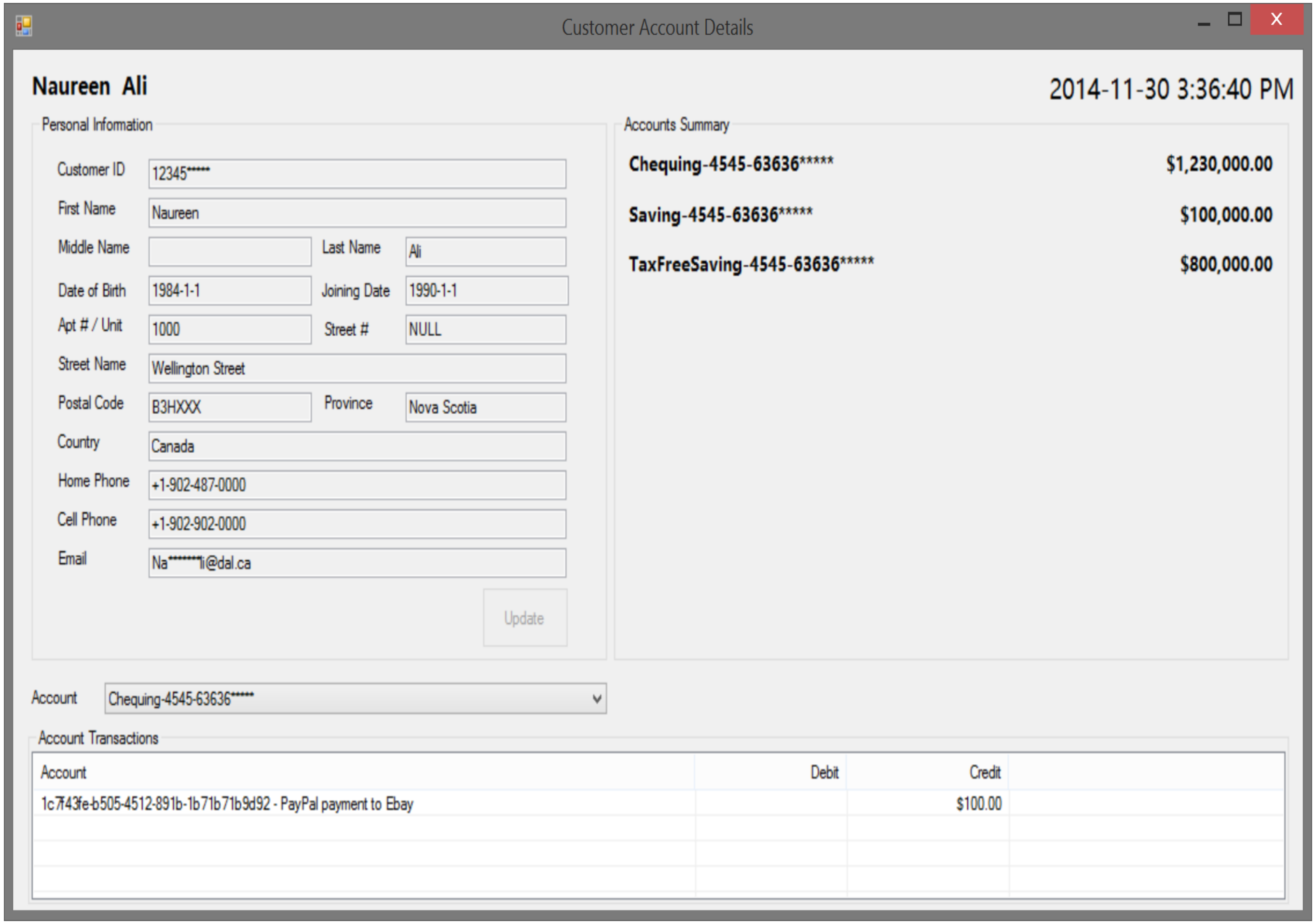


Implementation

After Applying PIP



Implementation



Customer Account Details

Naureen Ali

2014-11-30 3:36:40 PM

Personal Information

Customer ID	12345*****		
First Name	Naureen		
Middle Name		Last Name	Ali
Date of Birth	1984-1-1	Joining Date	1990-1-1
Apt # / Unit	1000	Street #	NULL
Street Name	Wellington Street		
Postal Code	B3HXXX	Province	Nova Scotia
Country	Canada		
Home Phone	+1-902-487-0000		
Cell Phone	+1-902-902-0000		
Email	Na*******li@dal.ca		

Accounts Summary

Chequing-4545-63636*****	**$1,230,000.00**
Saving-4545-63636*****	**$100,000.00**
TaxFreeSaving-4545-63636*****	**$800,000.00**

Account: Chequing-4545-63636*****

Account Transactions

Account	Debit	Credit
1c7f43fe-b505-4512-891b-1b71b71b9d92 - PayPal payment to Ebay		$100.00

Bank Example Use Case

- Our technical implementation uses Visual Studio .Net (IDE), PostSharp (AOP), the Unity Container (Dependency Injection) and the Mock library to realize an example injection of our de-identification service into a banking application.
- This example's implementation code may be downloaded from https://web.cs.dal.ca/~naureen/BankExample.
- The banking application's use case scenario contains account information that shows individual and account details. We use two roles, manager and operator, to study the behavior of the system before

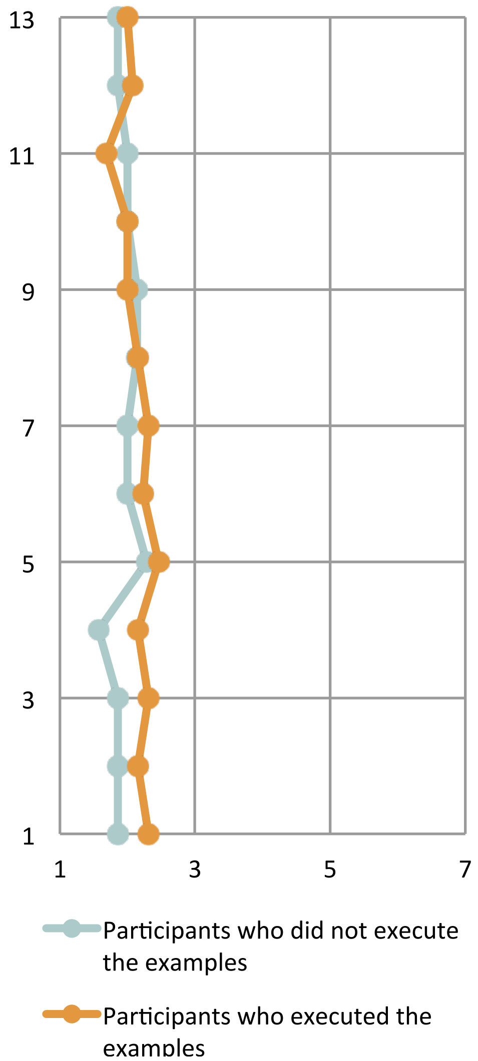
Evaluation Methodology

- Technology Acceptance Model (TAM)
- Software engineers consider embedding (1) a simple privacy pattern, and (2) a complex privacy pattern in legacy software.
- Software engineers from software multinationals such as IBM, Intel, Dell, and end user companies with software engineers (e.g. AT&T) to small software engineering companies, such as Canada's Newpace.

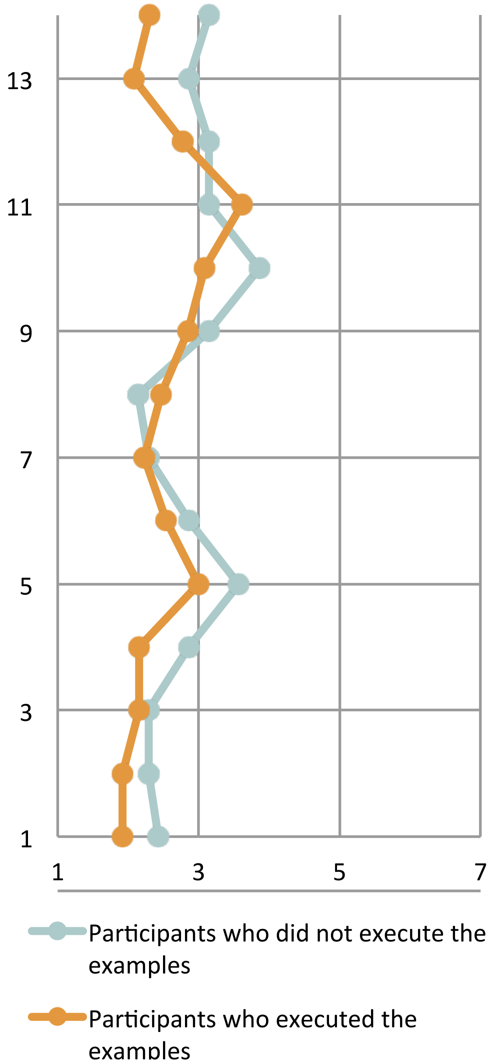
Early Evaluation

- Received 25 completed and usable responses.
- Sample of practicing software engineers evaluate the PIP pattern as useful. However, responses were mixed with respect to its perceived ease of use.
- Respondents indicated across the board that the pattern would improve their productivity when embedding privacy controls and it is useful in general.
- Respondents did not provide us with outlines or descriptions of any or better alternatives.

Perceived Usefulness



Perceived Ease of Use



Call to action: Knowledge mobilization and new tools

- Experience is that patterns such as k-anonymity is too difficult to implement for the average developer – simplify and disseminate
- Our abstraction-unifying, higher-level Privacy Injection Pattern helps remove fragmentation from the software engineering landscape for privacy.
- Patterns for privacy, its constructs, and desirable properties (e.g. unlinkability and unobservability at the data level, and the 7Cs at the user level such as comprehension, choice, consent, consciousness, consistency, confinement, and context [14]) will become increasingly available,
- Policy levers, such as Privacy by Design in regulations, begin to work.
- Take the survey at https://surveys.dal.ca/opinio/s?s=29098