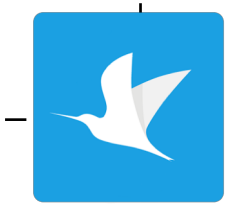
STRIDE: Digging Vulnerability by Threat Modelling

Friday, 9 Nov 2018

Agenda :

1. Security Development Lifecycle (SDL)
2. Threat Modelling Overview
3. STRIDE
4. STRIDE Implementation
5. Q & A



traveloka

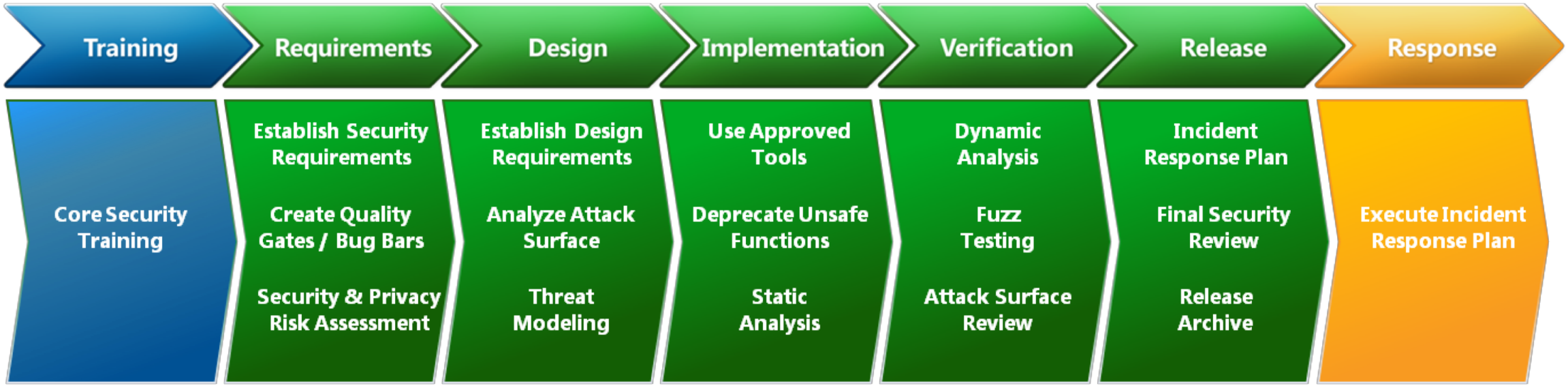
1. What is

Security Development Lifecycle (SDL) ?

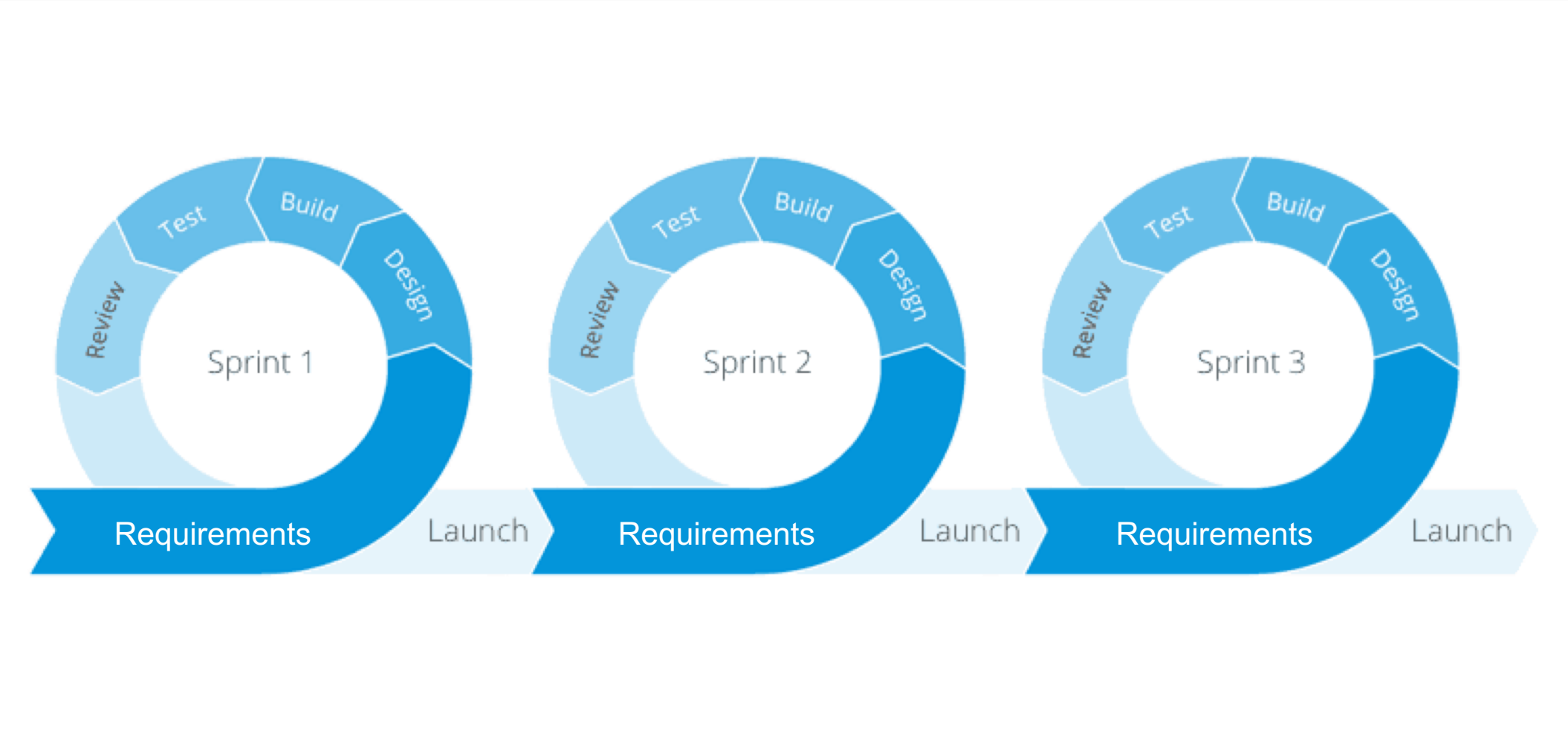
"A software development process that helps developers build more secure software and address security compliance requirements while reducing development cost".

- Microsoft

Microsof SDL



Agile SDL


Review
Test
Build
Design
Sprint 1
Requirements
Launch
Review
Test
Build
Design
Sprint 2
Requirements
Launch
Review
Test
Build
Design
Sprint 3
Requirements
Launch

traveloka

2. Threat Modelling

Overview

- **Threat modeling** is the use of abstractions to aid in thinking about risks [1].

- Threat Model Samples
  1. Trike
  2. P.A.S.T.A
  3. STRIDE
  4. VAST
  5. OCTAVE

[1] Shostack A, *Threat Modeling: Designing for Security*, John Wiley & Sons, Inc., 2014, Indianapolis, Indiana

traveloka

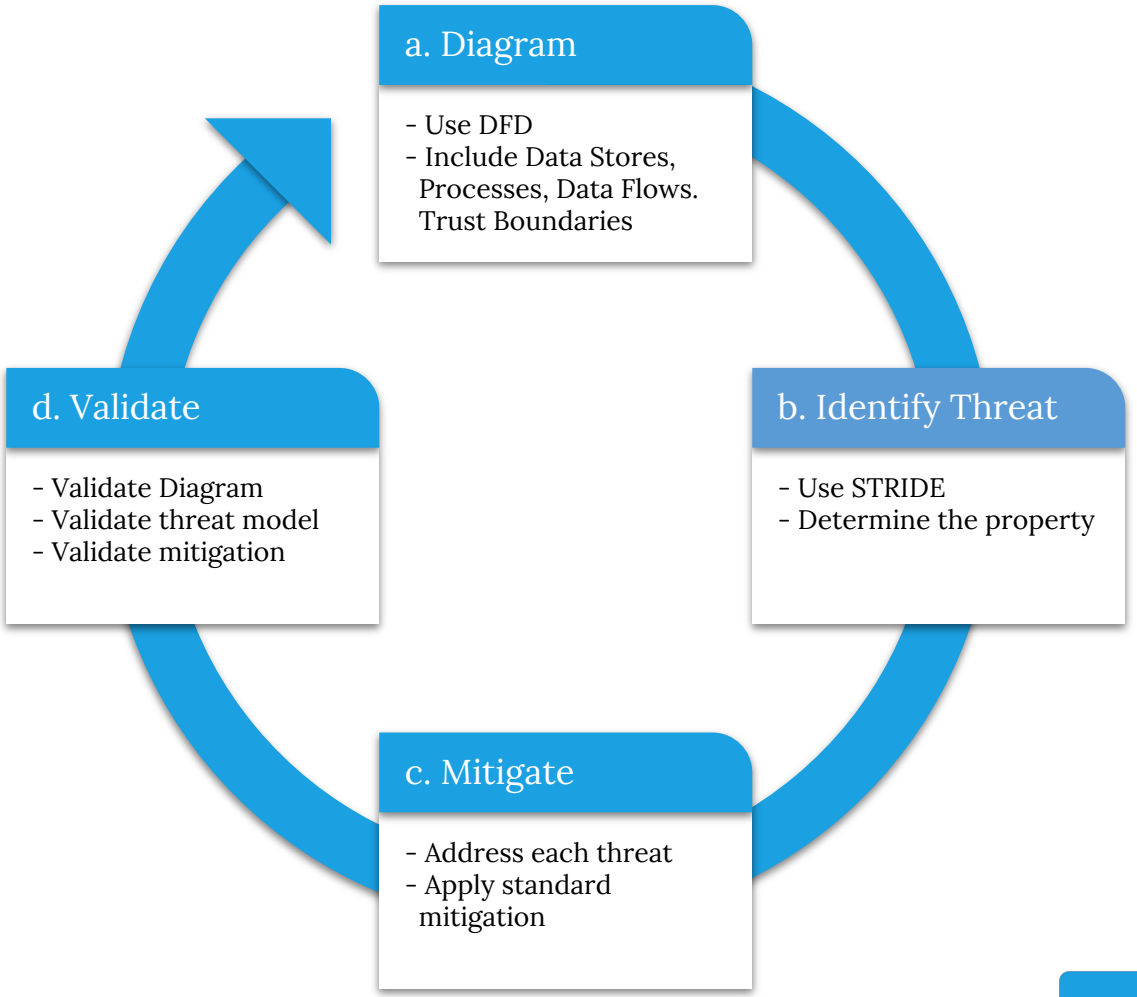
3. STRIDE

STRIDE Overview

Property	Threat	Definition	Example
Authentication	**S**poofing	Impersonating something or someone else.	Pretending to be quiz organizer that want to steal the credentials
Integrity	**T**ampering	Modifying data or code	Modifying a email and phone number
Non-repudiation	**R**epudiation	Claiming to have not performed an action.	“I didn’t reschedule my ticket,” “I didn’t refund that ticket”
Confidentiality	**I**nformation Disclosure	Exposing information to someone not authorized to see it	Allowing someone to read the other user information.
Availability	**D**enial of Service	Deny or degrade service to users	Brute force login attempts
Authorization	**E**levation of Privilege	Gain capabilities without proper authorization	Allowing a remote internet user to run commands in Traveloka Apps or desktop application

Process

- The STRIDE threat modeling was introduced in 1999 at Microsoft, for developers to find 'threats to our products'.

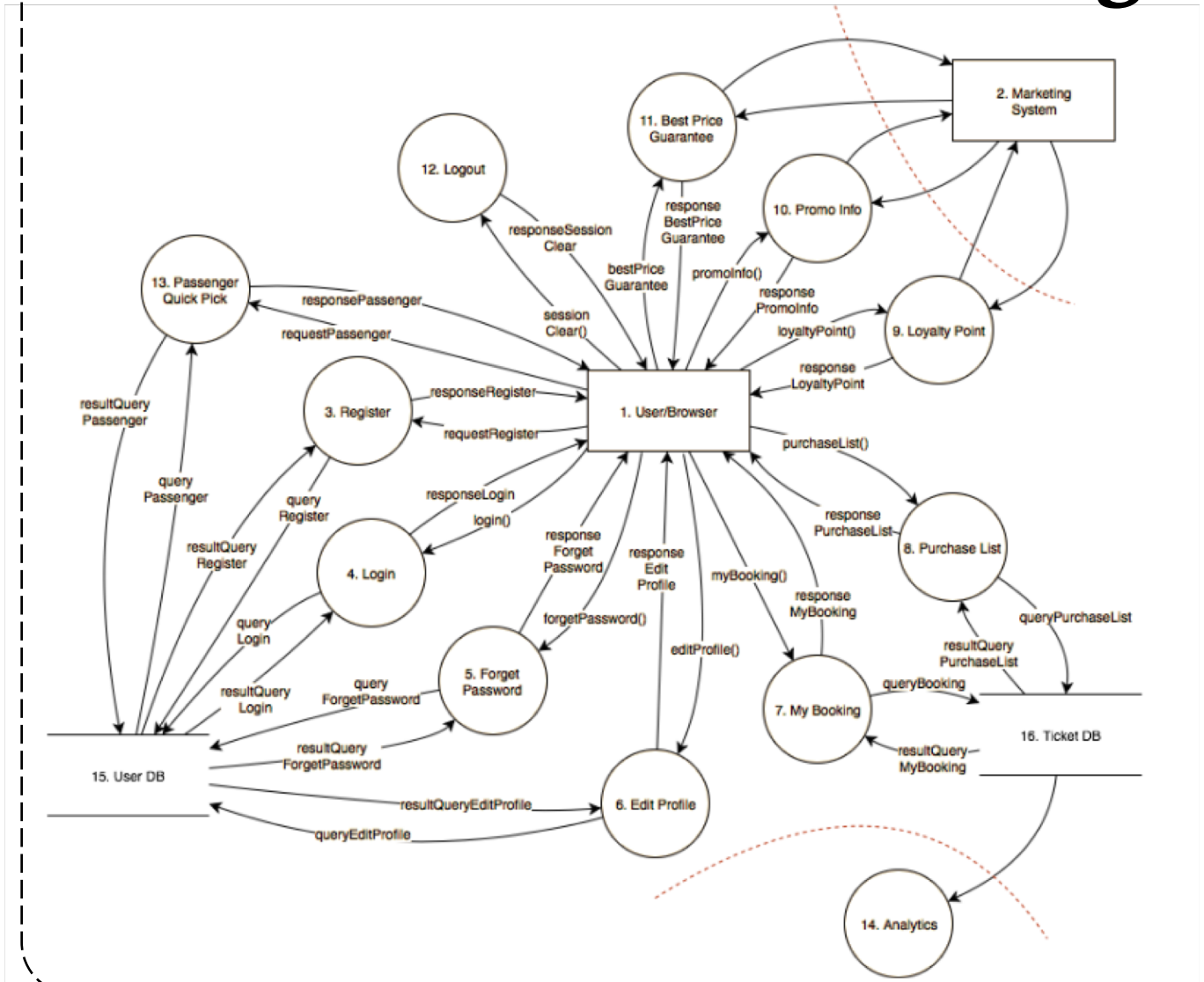


traveloka

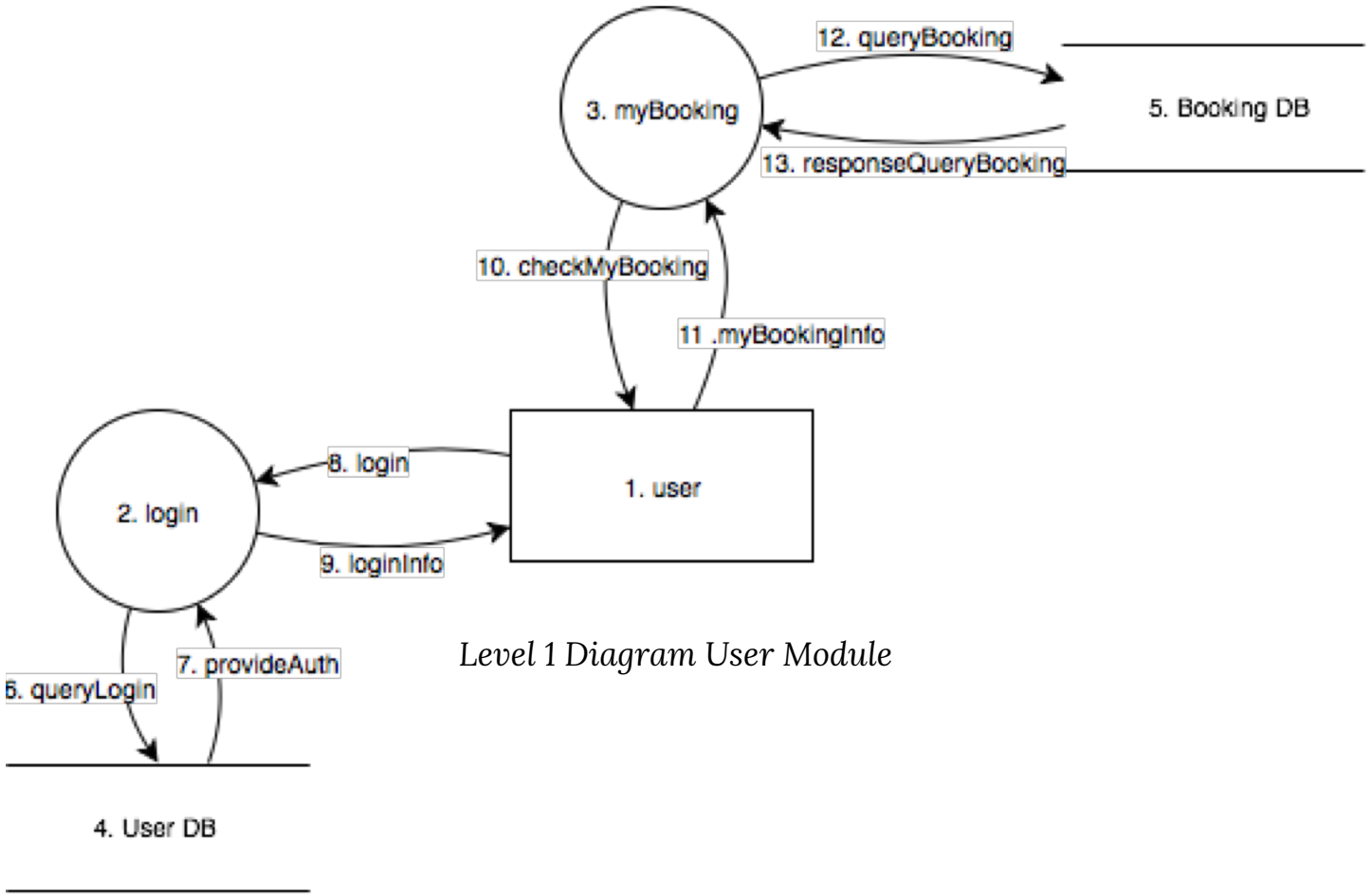
4. STRIDE

Simulation

a. Diagram



External Entity	- People - Other system
Process	- Component - Services - Web Services
Data Store	- Database - File - Registry - Queue / Stack
Data Flow	- Function Call - Remote Call Procedure (RPC) - Network Traffic
Trust Boundaries	- Process Boundary - File System



Level 1 Diagram User Module

- Diagram Layer
  Very high-level; entire component/ product / system
- **Level 1 Diagram**
  High level; single feature / scenario
- Level 2 Diagram
  Low level; detailed sub-components of features
- Level 3 Diagram
  More detailed; need more layers in huge projects

b. Identify Threat

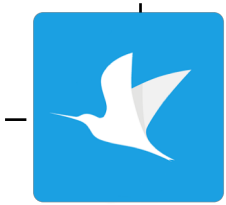
Element	S	T	R	I	D	E	Name
	✓		✓				1. user
	✓		✓	✓	✓	✓	2. login 3. myBooking
 Data Store		✓		✓	✓	✓	4. User DB 5. Booking DB
Data Flow 		✓		✓	✓		6. readQueryLogin 7. provideAuth 8. login . . 13. queryBooking

c. Mitigate

Element	S	T	R	I	D	E	Name	Mitigation
	✓		✓				1. user	- Strong authentication, 2FA - Secure logging and auditing
	✓		✓	✓	✓	✓	2. login 3. editProfile	- Strong authentication, 2FA - Secure logging and auditing - Token, encryption - Packet filtering - Input validation and privileges control
		✓		✓	✓	✓	4. User DB 5. Booking DB	- Authentication - Token, encryption - Packet filtering - Input validation and privileges control
		✓		✓	✓		6. readQueryLogin 7. provideAuth 8. login . . 13. queryBooking	- Authentication - Token, encryption - Packet filtering - Input validation and privileges control

Standard Mitigation

Threat	Property	Mitigation
Spoofing	Authentication	- Cookie authentication - PKI systems such as SSL/TLS and certificates - Digital signatures
Tampering	Integrity	- ACLs - Digital Signatures
Repudiation	Non-repudiation	- Secure login and monitoring - Digital Signatures
Information Disclosure	Confidentiality	- Encryption - Token
Denial of Service	Availability	- ACLs - Filtering
Elevation of Privilege	Authorization	- ACLs - Group or role membership - Privilege ownership - Input validation



4. Validate

- ❑ **Validate the whole threat model:**
  - ❑ Does diagram match final code?
  - ❑ Is each threat mitigated?
  - ❑ Are mitigations done right
  - ❑ Check before Final Security Review
- ❑ **Validate Quality of Threats and Mitigations**
  - ❑ Describe the attack, context, impact
  - ❑ Mitigations: Associate with the threat, describe the mitigations

traveloka

Question ???

Summary

1. SDL helps developers to **build more secure** software and **address security compliance** requirements while **reducing development cost**
2. Threat modeling is the use of abstractions to **aid in thinking about risks**.
3. STRIDE: **Spoofing**, **Tampering**, **Repudiation**, **Information Disclosure**, **Elevation of Privileges**
4. The Process start from: **Diagram**, **Threat Model**, **Mitigate**, **Validation**

Thank you