Security Strategies in Web Applications and Social Networking

Lesson 8

Securing Web Applications

Learning Objective

- Describe the attributes and qualities of the software development life cycle (SDLC).

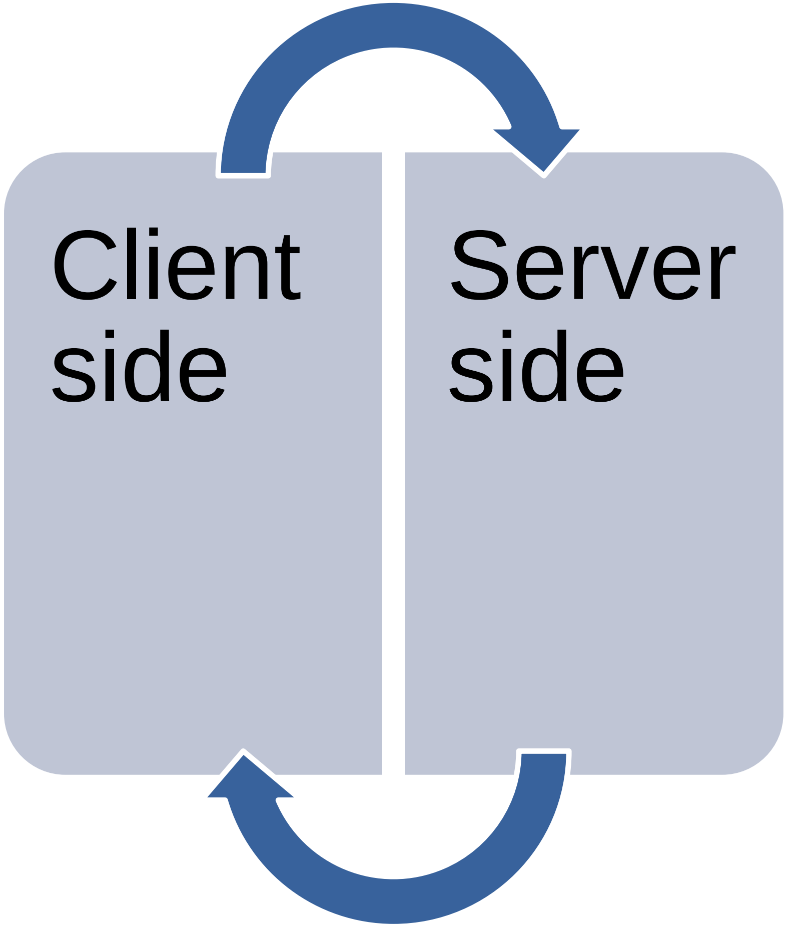
Key Concepts

- Technologies and systems used to make a complete functional Web site
- Secure software development life cycle (SDLC) approaches
- Best practices in securing Web applications

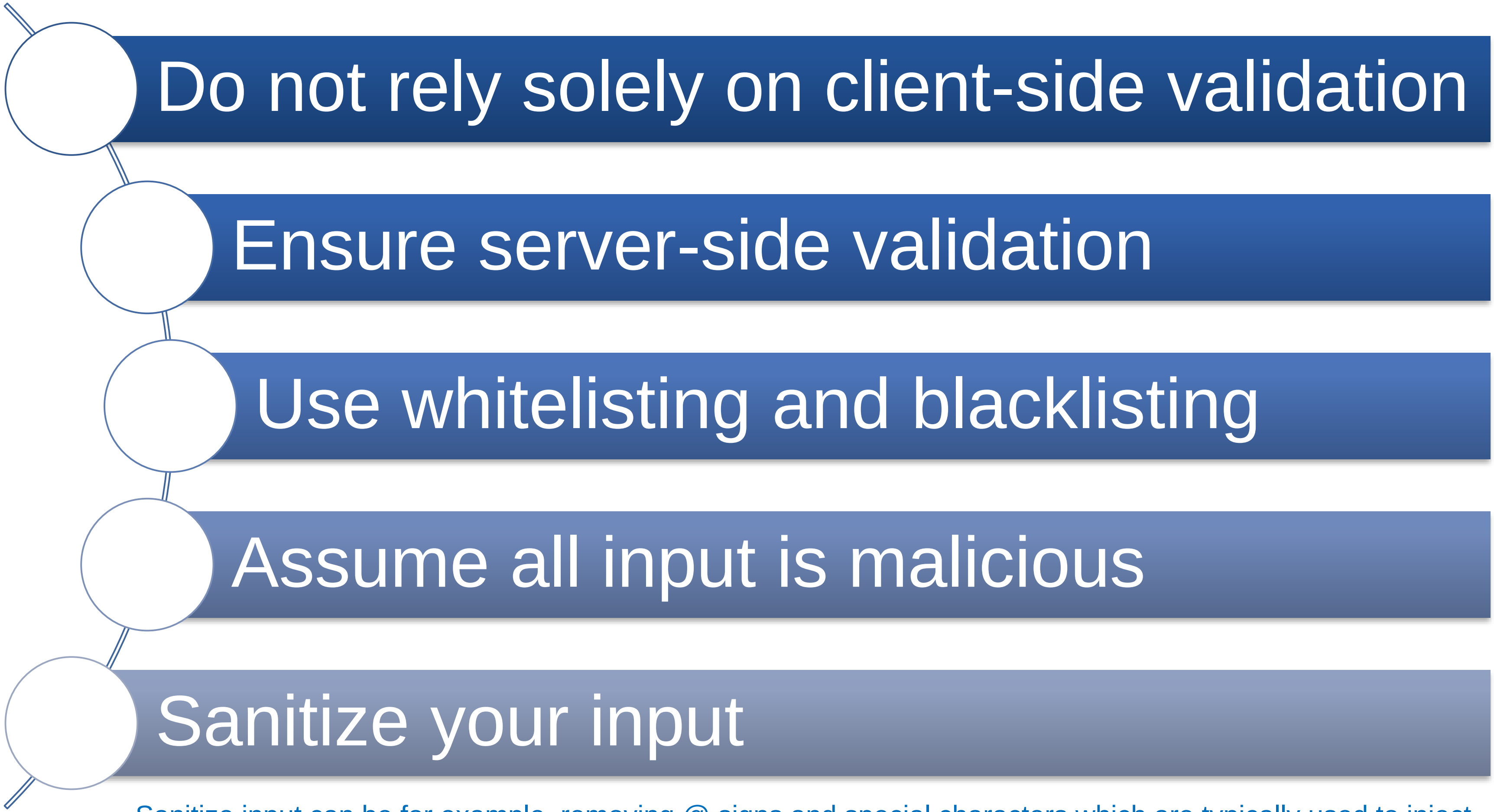
Data Input Validation

Previously, validation was only done at one side: server,
Now we have input coming from client side as well, so need to do validation on both ends

better to do client side as well because it saves a validation trip to server
Don’t need to go to server, determine it is invalid data.
Can just invalidate on client side, don’t make trip to server



Data Input Validation (Continued)



- Do not rely solely on client-side validation
- Ensure server-side validation
- Use whitelisting and blacklisting
- Assume all input is malicious
- Sanitize your input

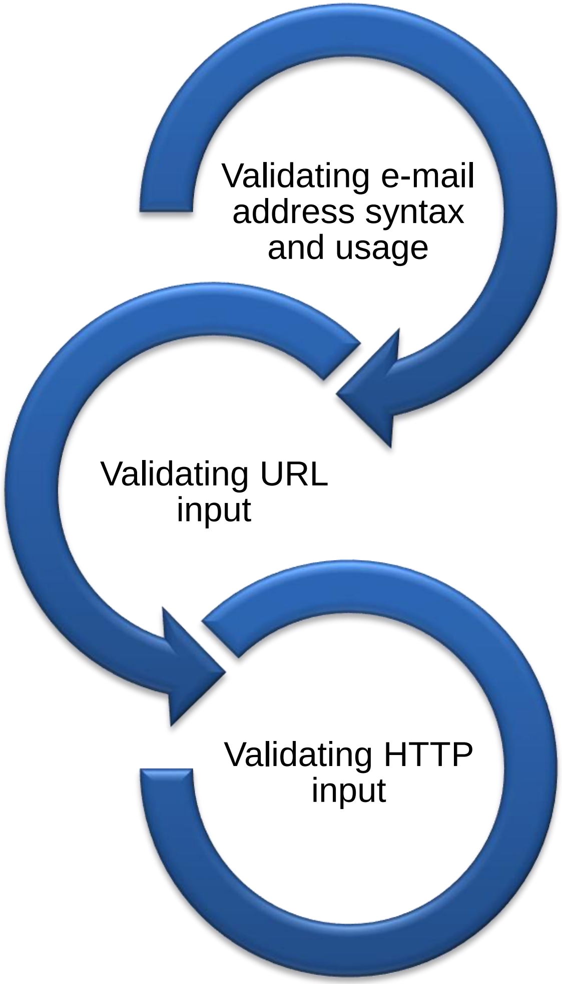
Sanitize input can be for example, removing @ signs and special characters which are typically used to inject commands for example

Request for Comments (RFC) Syntax

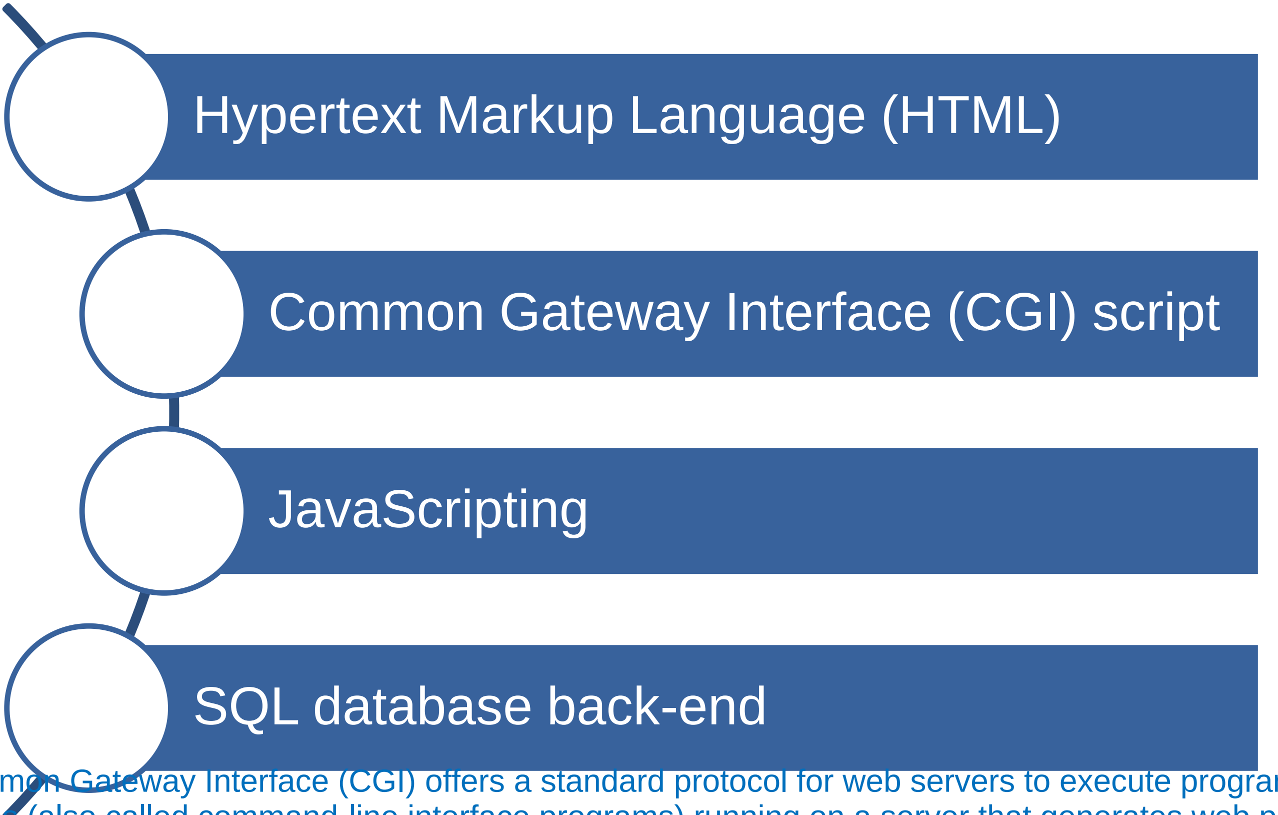
Request for Comments (RFC): A formal document from the IETF, which is the result of committee drafting and revisions to a technical document.

Many RFCs are intended to become Internet standards.

Review the RFC to verify syntax used when reviewing acceptable syntax for e-mail addresses, URL input, XML input, and so forth.



Common Web Elements

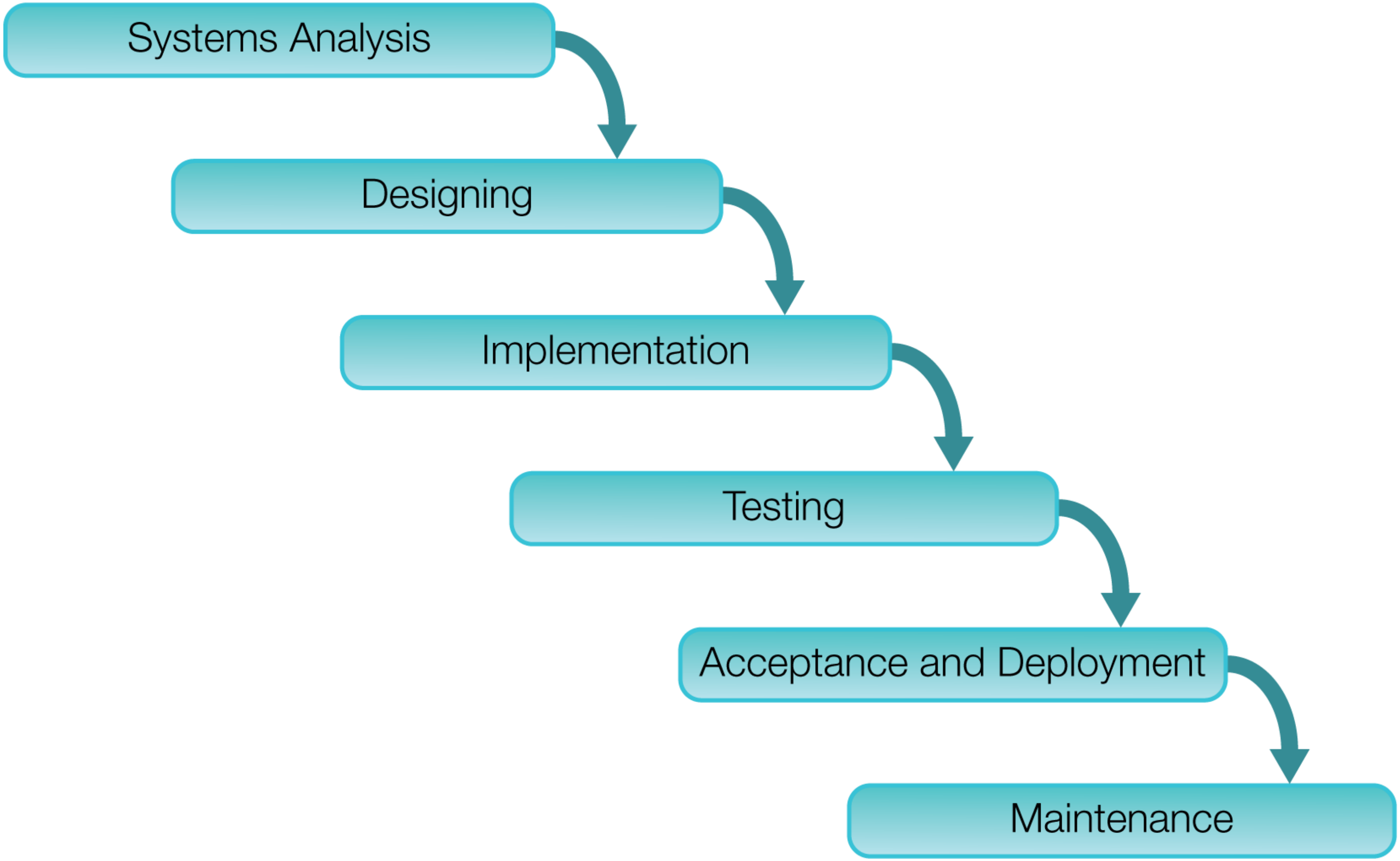


- Hypertext Markup Language (HTML)
- Common Gateway Interface (CGI) script
- JavaScripting
- SQL database back-end

In computing, Common Gateway Interface (CGI) offers a standard protocol for web servers to execute programs that execute like console applications (also called command-line interface programs) running on a server that generates web pages dynamically. Such programs are known as CGI scripts or simply as CGIs.

SQL can have physical and logical attacks
Physical harming server storing the db
Logical being sql injection attacks

Traditional SDLC



Systems Analysis

Designing

Implementation

Testing

Acceptance and Deployment

Maintenance

Common SDLC Models

- Waterfall
- Iterative and Agile Scrum
- Rapid Application Development
- Rational Unified Process (RUP)
- Spiral Model and V-Model

Web Site and Web Application Security

Perimeter Security

Host-based Security Mechanisms

End-User Validation

Authentication and Access Management

Input Validation

Vulnerability Management

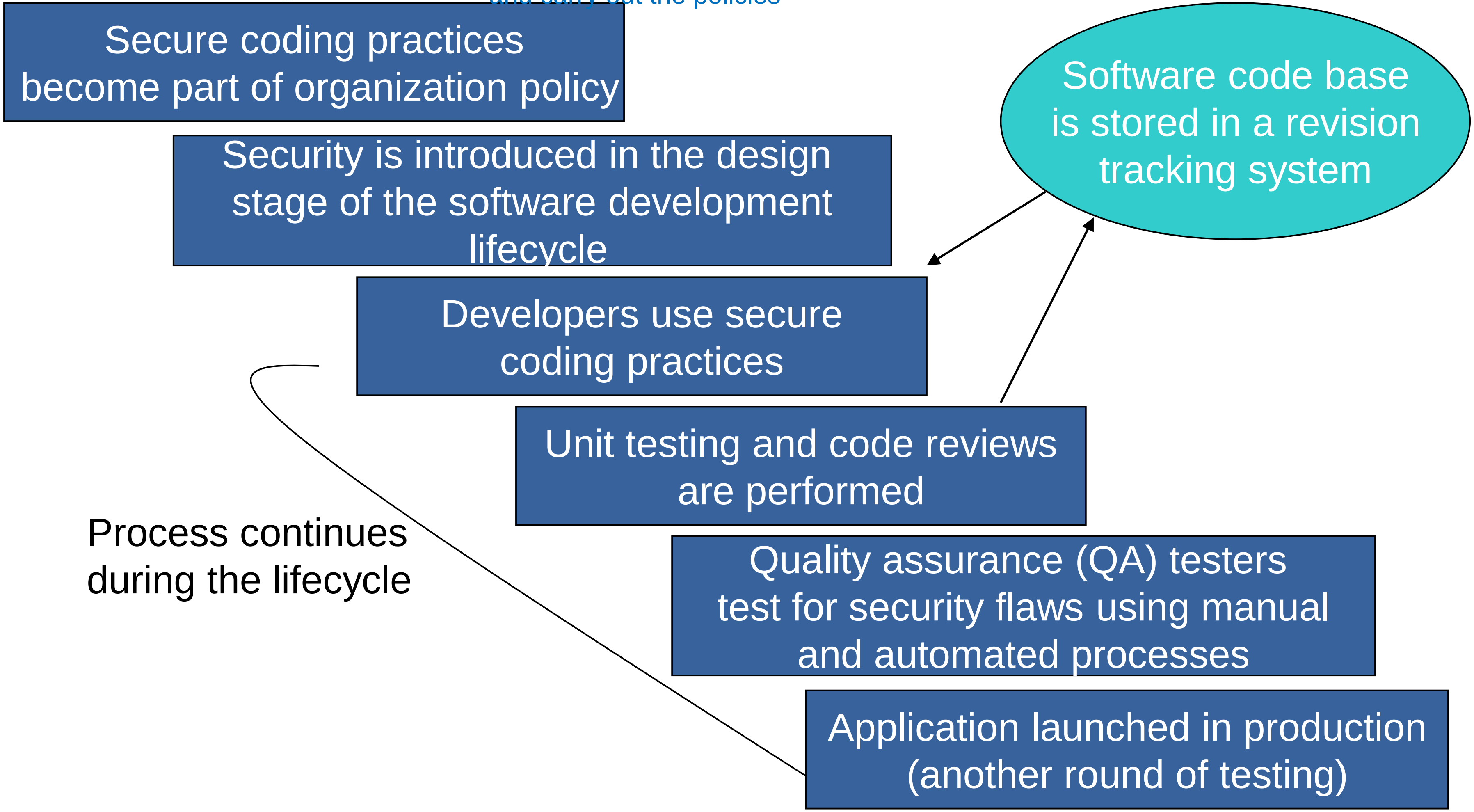
A.I.C – Security Triad
Availability
Integrity
Confidentiality

Triple A
Authentication
Authorization
Accounting

Secure Software Development Life Cycle

Code review policies developed with consideration of laws surrounding environment and application

Policies are imposed upon developers entering the organization, whom must follow and carry out the policies



Secure coding practices become part of organization policy

Security is introduced in the design stage of the software development lifecycle

Developers use secure coding practices

Unit testing and code reviews are performed

Quality assurance (QA) testers test for security flaws using manual and automated processes

Application launched in production (another round of testing)

Software code base is stored in a revision tracking system

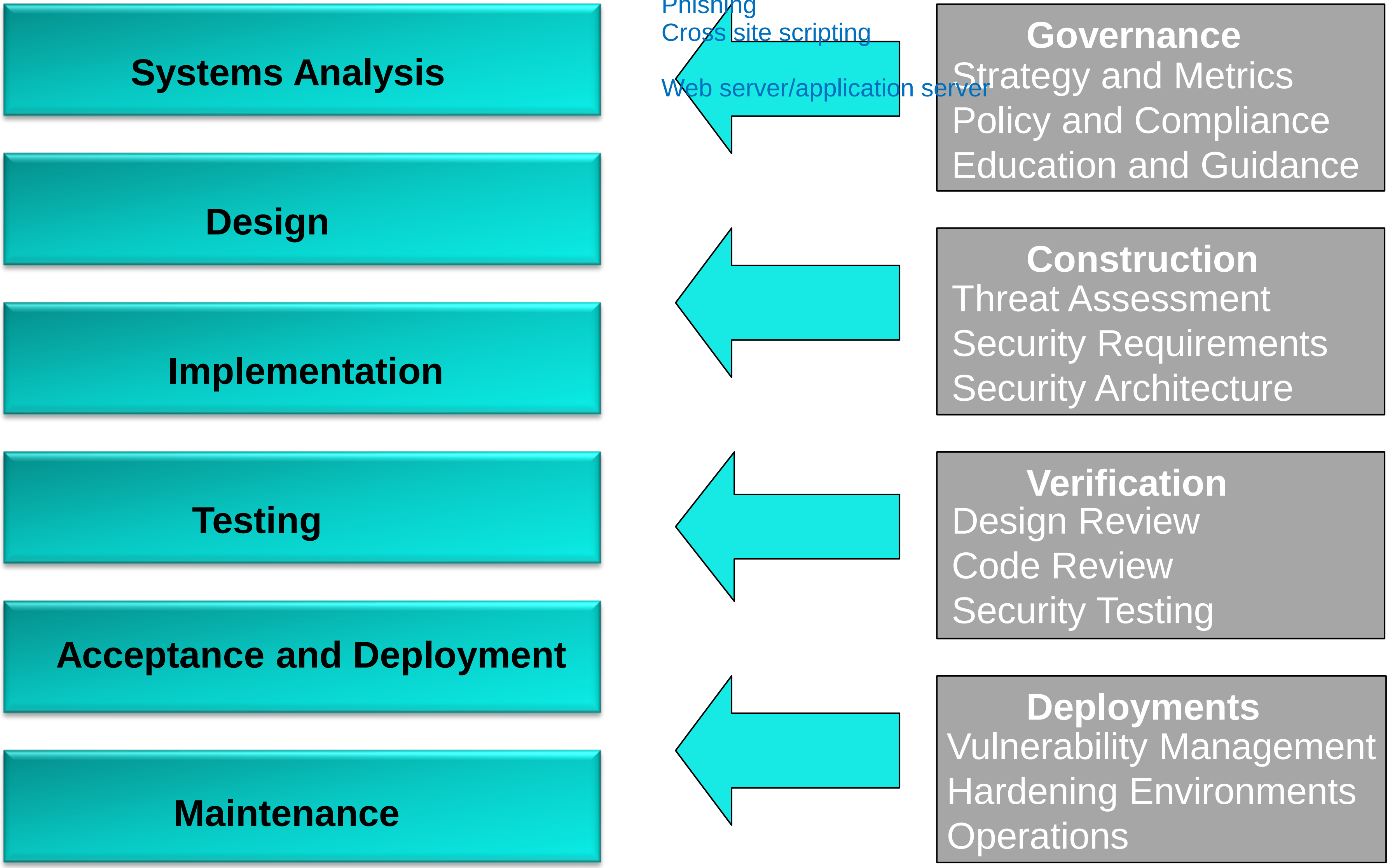
Process continues during the lifecycle

Securing SDLC

You can secure SDLC by using Software Assurance Maturity Model (SAMM)

Client side atk
Social engineering
Phishing
Cross site scripting

Web server/application server



Systems Analysis

Design

Implementation

Testing

Acceptance and Deployment

Maintenance

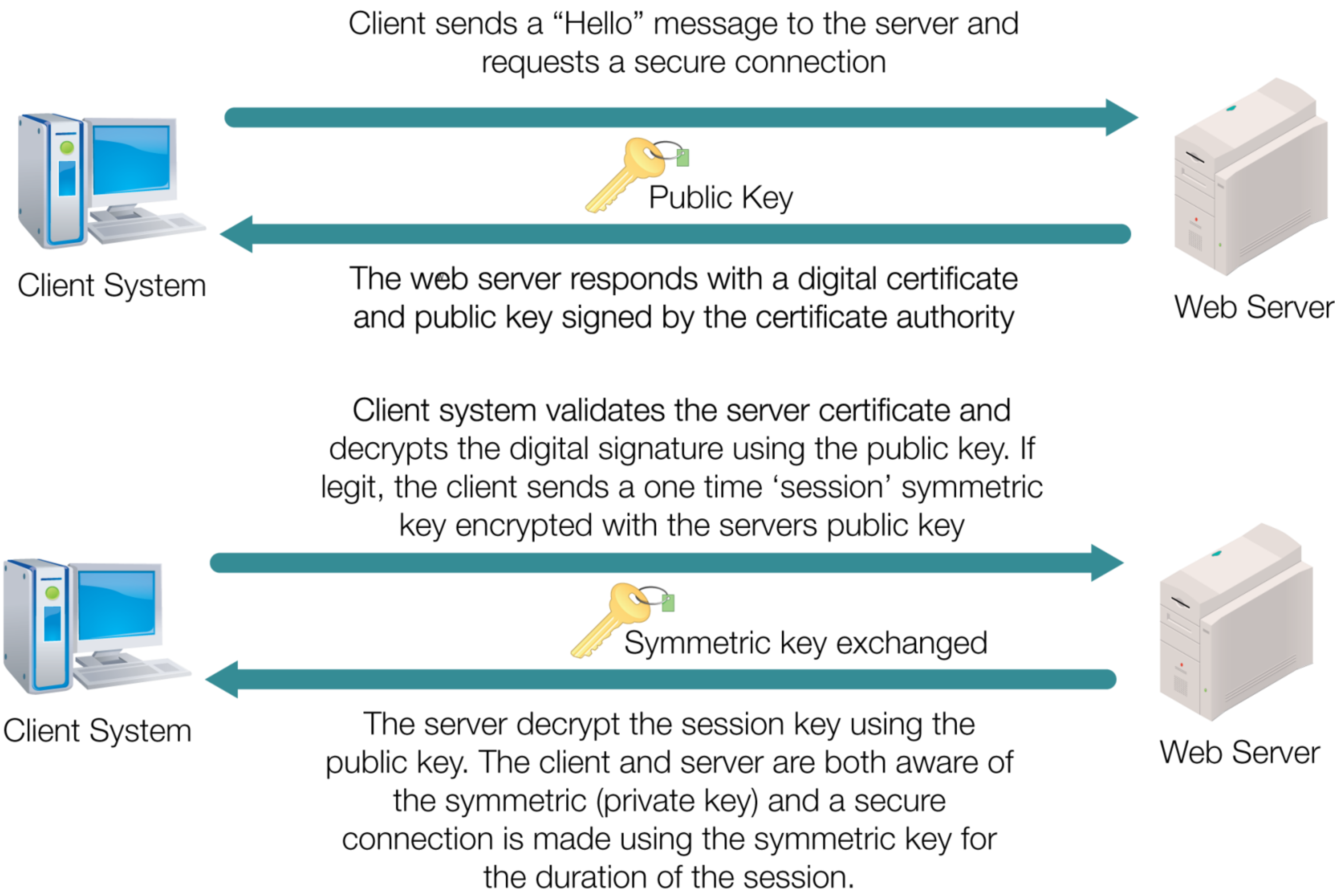
Governance
Strategy and Metrics
Policy and Compliance
Education and Guidance

Construction
Threat Assessment
Security Requirements
Security Architecture

Verification
Design Review
Code Review
Security Testing

Deployments
Vulnerability Management
Hardening Environments
Operations

SSL for Secure Web Sites



Client sends a "Hello" message to the server and requests a secure connection

The web server responds with a digital certificate and public key signed by the certificate authority

Client system validates the server certificate and decrypts the digital signature using the public key. If legit, the client sends a one time 'session' symmetric key encrypted with the servers public key

The server decrypt the session key using the public key. The client and server are both aware of the symmetric (private key) and a secure connection is made using the symmetric key for the duration of the session.

Types of Access Control Methods

The last two are part of MAC,
So it is really just two types here:
DAC and MAC

DAC is ctrl in hands of owner of file/folder
Can choose to just give read, r/w, or rwx for example

MAC is ctrl in hands of administrator who grants the access permissions

Rule-based access ctrl for example may check a list of addresses
If u are accessing from a certain ip addr that is in the list, u will (or will not, depending on the rule) have access

- Discretionary Access Control (DAC)

Role-based
Ex. Ur account part of student group, teacher grp, etc.

- Mandatory Access Control (MAC)
- Rule-Based Access Control

like the modsec rules and exceptions
dan sheng was working on for
Frontier

- Role-Based Access Control

Secure Web Application Development

This will be on the test

- Do not trust data input from users or external services
- Validate data input on the server side using a variety of techniques
- Use well tested and established authentication, authorization, and session management mechanisms

Secure Web Application Development (Continued)

- Establish a user “time out” period
- Do not allow concurrent sessions with the same user ID
- Enforce strong password policies
- Implement encryption for all confidential data
- Provide generic error messages back to the user

Secure Web Application Development (Continued)

- Know the programming language and avoid the use of known vulnerable functions
- Know the database an application is using and utilize secure functions for the database layer
- Never reveal any internal file paths or directories

Secure Web Application Development (Continued)

- Do not allow uploaded files to have execute permissions
- Perform peer code reviews

Best Practices for Maintaining Secure Software

- Incorporate training and awareness programs for developers
- Perform frequent application assessments
- Determine the security requirements early
- Implement secure development practices

Best Practices for Maintaining Secure Software (Continued)

- Formalize vulnerability remediation processes
- Define metrics and monitoring processes
- Establish operational security guidelines

Summary

- Tiers of a typical Web infrastructure
- Secure SDLC
- Practices for and impact of developing secure applications