Secure Software Development Design Principles

Objectives

- List and explain the Secure Design Principles
- Critique a modern software application from a security standpoint
- Explain the concept of an attack surface
- Explain the difference between privacy and security

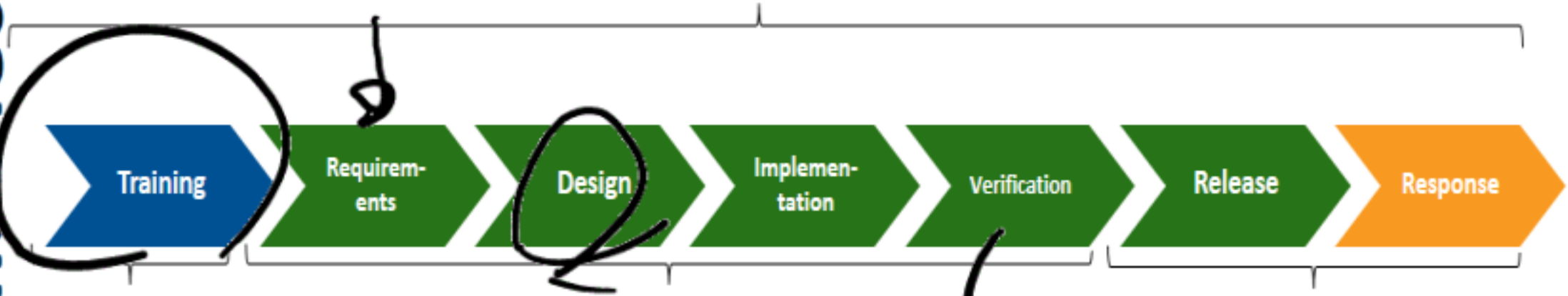
Secure Design Principles

- Principle of Least Privilege
- Separation of Duties
- Defense in Depth
- Fail Secure
- Economy of Mechanisms
- Complete Mediation
- Open design
- Least Common Mechanisms
- Psychological Acceptability
- Leveraging Existing Components

Microsoft Security Development Lifecycle (SDL)

Delivering secure software requires:

Executive commitment → SDL a mandatory policy at Microsoft since 2004



Education | Technology and Process | Accountability

This image cannot currently be displayed.

Testing

Ongoing Process Improvements → 6 month cycle

SDL Secure Design Principles

- Safer applications begin with secure design
- Core SDL secure design principles:
  - Attack Surface Reduction
  - Basic Privacy
  - Threat Modeling — Done
  - Defense in Depth
  - Least Privilege — Done
  - Secure Defaults — Deployment

SDL Core Principle: Attack Surface Reduction

- **Attack Surface:** Any part of an application that is accessible by a human or another program
  - Each one of these can be potentially exploited by a malicious user
- **Attack Surface Reduction:** Minimize the number of exposed attack surface points a malicious user can discover and attempt to exploit

Interfaces

Firewalling

Attack Surface Example

Attack Surface Analysis

Step 1: Look at Your Entry Points

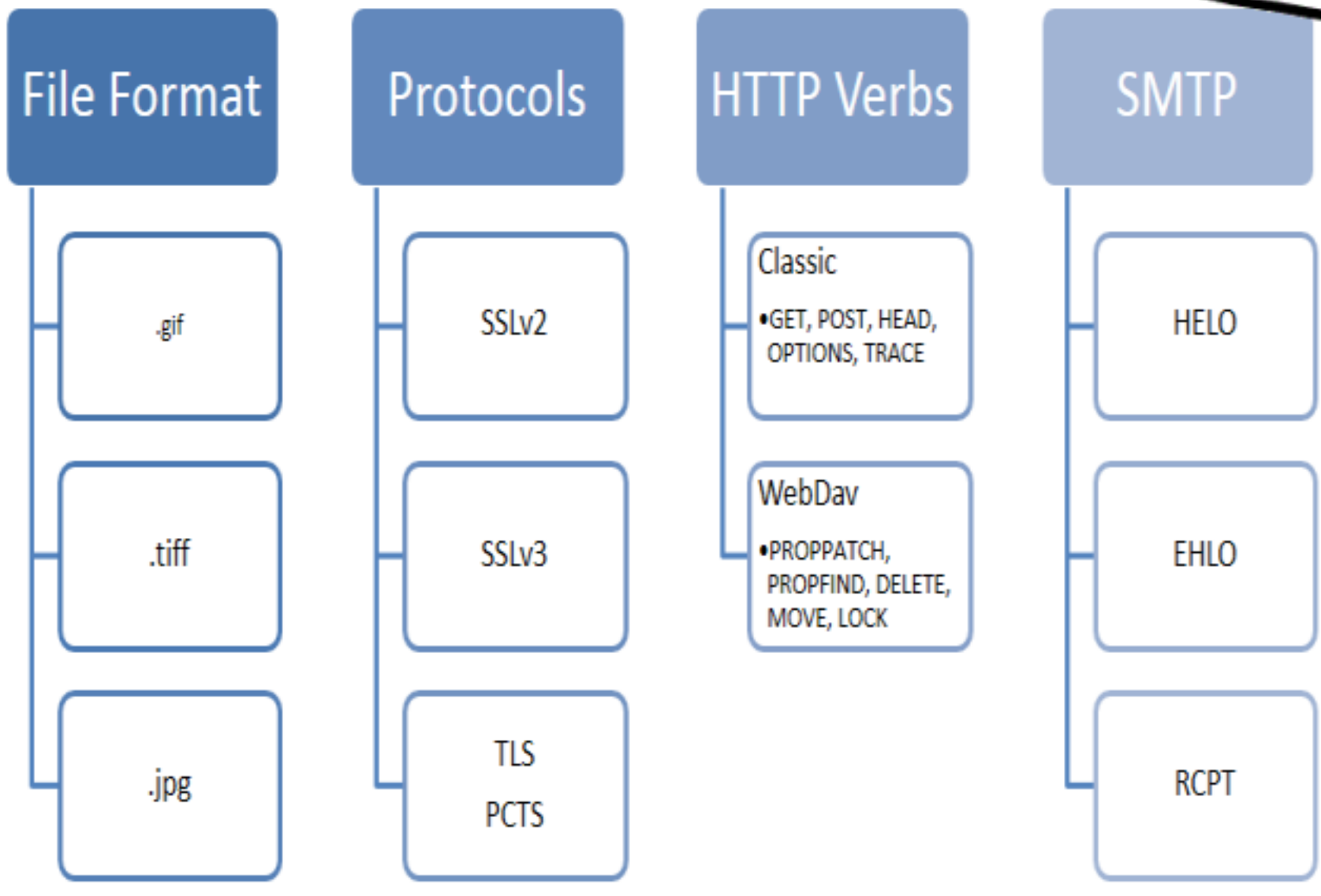
- Network inputs/outputs
- File inputs/ outputs
- Etc.

Step 2: Rank Your Entry Points

- Authenticated or non-authenticated
- Administrative or user-level access
- Network or local
- UDP or TCP
- Etc.

Attack Surface Analysis Tips

- Iterative process, for all features you need to also analyze their sub-features
- Restrict access to features as much as possible



It's Not Just About Turning Things Off

More Risk

Higher Attack Surface	Lower Attack Surface
On by default	Off by default
Open socket	Closed socket
UDP	TCP
Anonymous access	Authenticated user access
Constantly on	On as needed
Administrative access	User access
Internet accessible	Local subnet accessible
Running as SYSTEM	Running as user, network service or local service account
Uniform defaults	User defined settings
Large code	Small code
Weak access controls	Strong access controls

Attack Surface Reduction Examples

Microsoft Product	Attack Surface Reduction
Windows	• Authenticated Remote Procedure Call (RPC) • Firewall on by default
Internet Information Services 6.0 and 7.0	• Off by default • Running as network service by default • Static files by default
SQL Server 2005 and 2008	• xp_cmdshell stored procedure off by default • CLR and COM off by default • Remote connections off by default
Visual Studio 2005 and 2008	• Web server localhost only • SQL Server Express localhost only

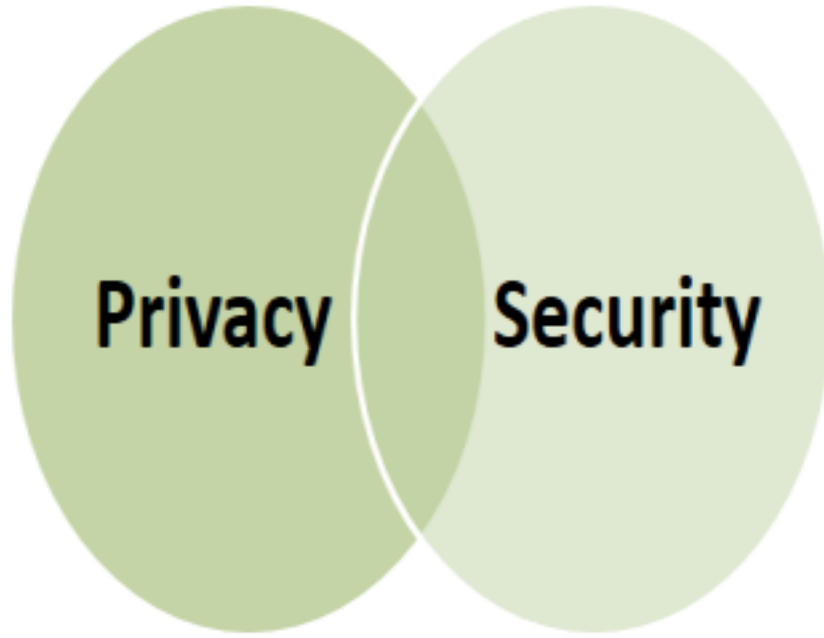
settings

Privacy Versus security

- What is the difference?

SDL Core Principle: Basic Privacy

- Privacy versus Security
  - **Privacy:** Empowering users to control the use, collection and distribution of their personal information
  - **Security:** Establishing protective measures that defend against hostile acts or influences and protects the confidentiality of personal information
- Privacy AND Security together are key factors for building trusted applications



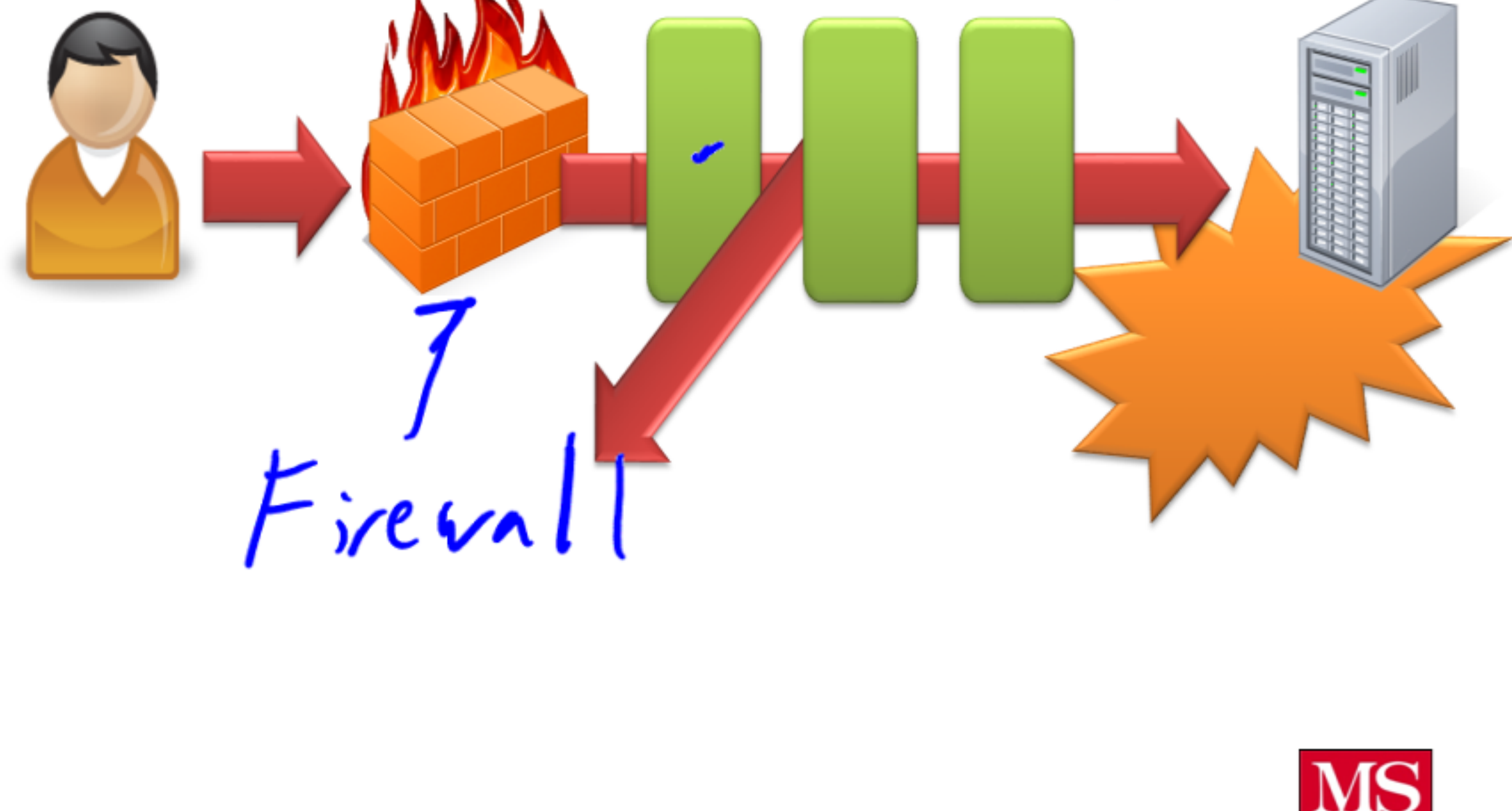
Important Note: Security Does Not Always Guarantee Privacy

It is possible to have a secure system that does not preserve users' privacy.

SDL Core Principle: Defense In Depth

- Assume that software and hardware will fail at some point → Failure mode & effects Analysis (FMEA)
  - Trusted applications: security and privacy features and mechanisms
- Most applications today can be compromised when single, and often only, layer of defense is breached (firewall)
- **Defense in Depth:** If one defense layer is breached, what other defense layers (if any) provide additional protection to the application?

Defense in Depth Example



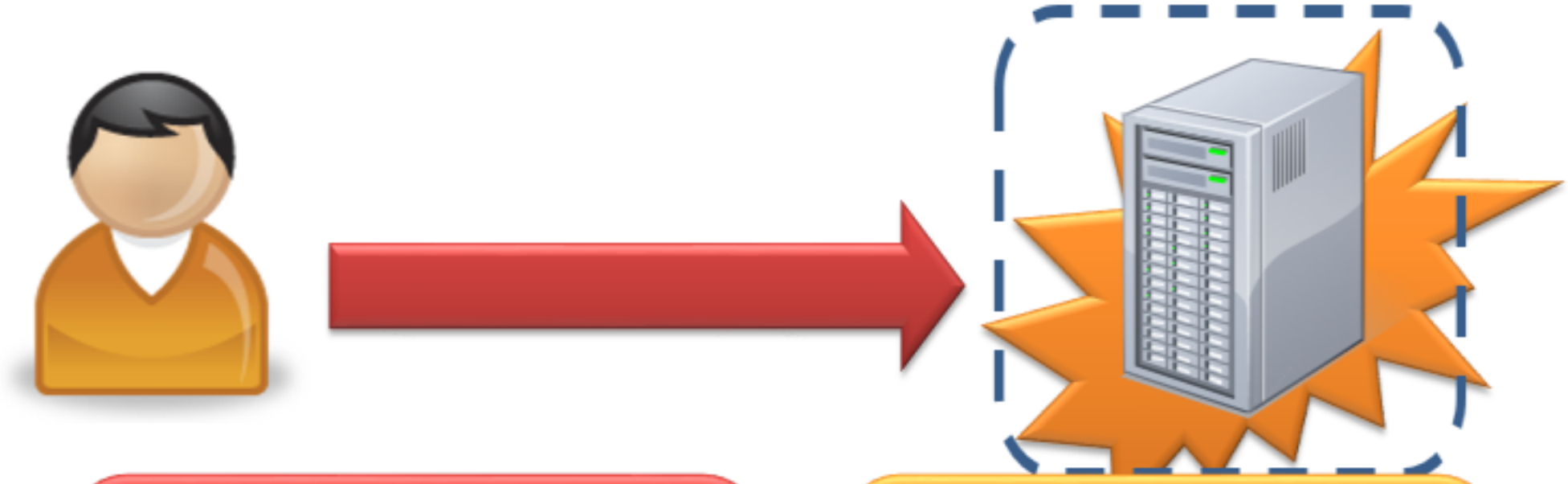
SDL Core Principle: Least Privilege

- Assume that all applications can and will be compromised
- **Least Privilege:** If an application is compromised, then the potential damage that the malicious person can inflict is contained and minimized accordingly

Least Privilege Example

NON-ADMIN

LOCAL SYSTEM



ADMIN / SYSTEM LEVEL

- Read user files
- Change system passwords
- Download malicious files
- Anything

NON-ADMIN

- ~~Read user files~~
- ~~Change system passwords~~
- ~~Download malicious files~~
- Limited capabilities

MSOE

SDL Core Principle: Secure Defaults

- **Secure Defaults:** Deploy applications in more secure configurations by default.
- Helps to better ensure that customers get safer experience with your application out of the box, not after extensive configuration
- It is up to the user to reduce security and privacy levels

Secure Defaults Examples

Application Component	Secure Defaults Principle
Firewall	Firewall ON by default
SSL Socket	Requires last latest SSL version (v3, TLS, etc.) by default
User can access application anonymous or authenticated	Application requires authenticated user sessions by default
Password complexity can be enforced	Password complexity is required by default
Store user passwords as hashes or clear text	Store user passwords as hashes by default

Separation of Duties

- Design is compartmentalized
  - Split keys for cryptographic functions
- Development roles
  - Programmer does not review his own code
  - Programmer does not deploy code onto production system

Defense in Depth

- Layered defense towards security
  - The breach of a single vulnerability does not result in complete or total system compromise
- Deters curious hacker / nondetermined hacker

Security Trust – Defense in Depth

- Layering protections so that the compromise of one is mitigated
- Running services and daemons as low privileged accounts
- Isolating different functions to different pieces of hardware
- Demilitarized zones
- Stack and heap guards

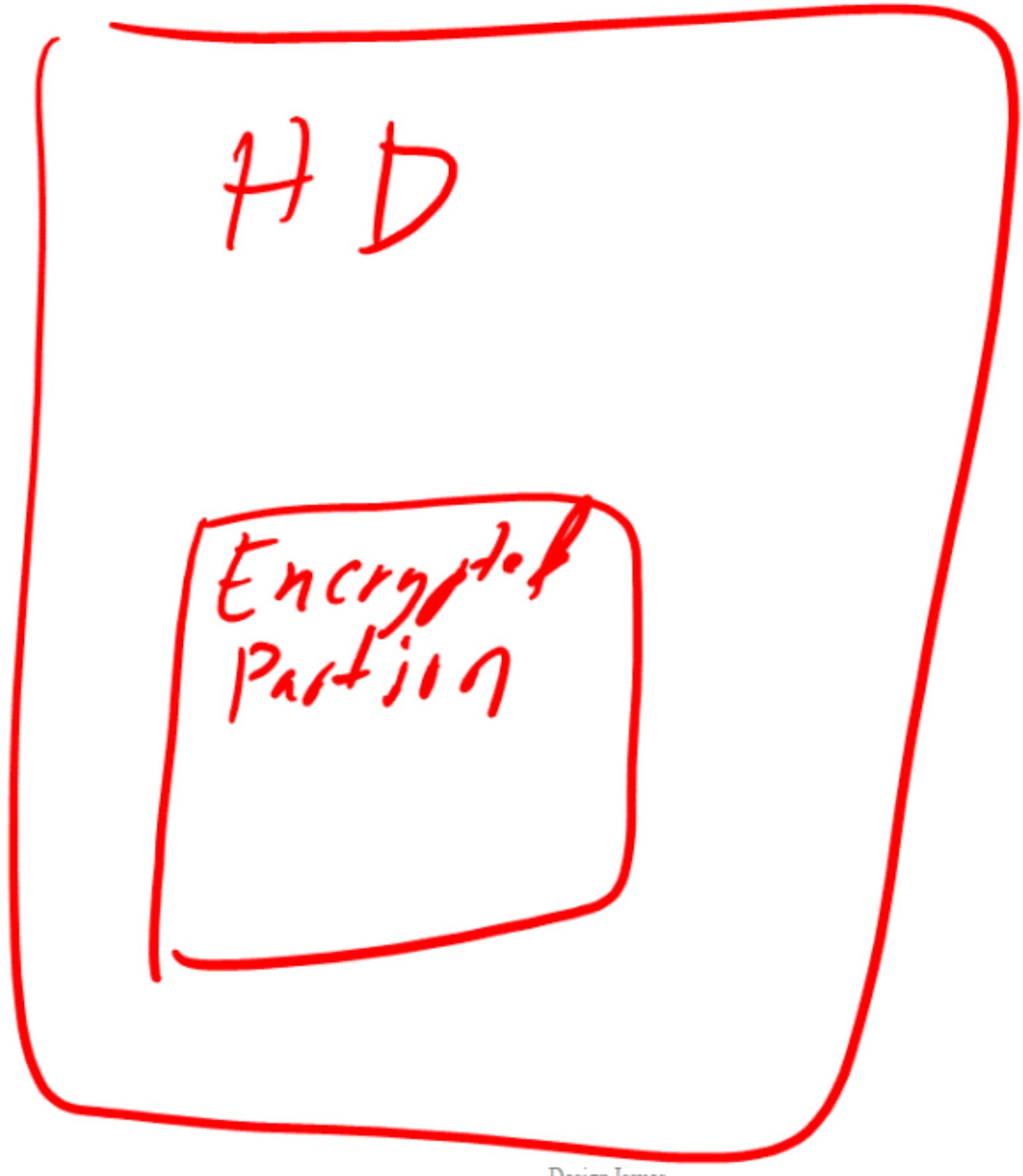
- Software reliably functions when attacked
- Is rapidly recoverable in the event of a failure
- Fails to a secure state if a failure occurs

Fail Secure

No stack traces to user

No open files

True Crypt and fail secure



Economy of Mechanism

- The more complex the design of the software, the more likelihood for a security failure there is
  - Unnecessary functionality or unneeded security mechanisms should be avoided
  - Strive for operational ease of use

KISS

Complete Mediation

- Every access to every object must be checked for authority every time the object is accessed.
- **Example 1**
  - When a UNIX process tries to read a file, the operating system determines if the process is allowed to read the file. If so, the process receives a file descriptor encoding the allowed access. Whenever the process wants to read the file, it presents the file descriptor to the kernel. The kernel then allows the access. If the owner of the file disallows the process permission to read the file after the file descriptor is issued, the kernel still allows access. This scheme violates the principle of complete mediation, because the second access is not checked. The cached value is used, resulting in the denial of access being ineffective.

Index of /

https://myweb.msoe.edu/?user=sebern&path=msoe/Winter2011/ce2800/ce2800.shtml

Suggested Sites | Web Slice Gallery | Church ... | LXR linux/include/li... | Professional Audio ... | Other bookmarks

Index of /

Index of /

https://myweb.msoe.edu/?user=schilling&path=msoe/Winter2011/ce2800/ce2800.shtml

Suggested Sites | Web Slice Gallery | Church ... | LXR linux/include/li... | Professional Audio ... | Other bookmarks

Index of /

Web server

URL

Name	Last modified	Size	Description
favicon.ico	08-Jun-2004 17:12	766	
graphics/	01-Nov-2004 10:37	-	
index.html	18-Nov-2010 10:17	1.9K	
local/	08-Apr-2009 13:56	-	
msoe.ico	08-Jun-2004 17:12	766	
robots	06-Jul-2007 16:12	0	
test.cgi	25-Feb-2008 14:22	65	
test.py	25-Feb-2008 14:22	65	

Apache/2.2.8 (Ubuntu) mod_auth_kerb/5.3 DAV/2 SVN/1.4.6 mod_jk/1.2.25 mod_ldap_userdir/1.1.12-20070601 PHP/5.2.4-2ubuntu5.12 with Suhosin-Patch mod_ssl/2.2.8 OpenSSL/0.9.8g mod_perl/2.0.3 Perl/v5.8.8 Server at myweb.msoe.edu Port 443

Complete Mediation Bad Design

Credit Card's Billing Name & Address:

First Name:

Last Name:

Address:

City:

State/Province:

Zip/Postal Code:

Country:

Process Now

(do not click more than once)

Stopped here

Open Design

- All information about crypto systems is public knowledge except the key, and the security of the system against cryptanalysis attacks is dependent on the secrecy of the key

- Not Security through obscurity

Least common mechanisms

- Mechanisms common to more than one user or process should not be shared
  - Design should compartmentalize or isolate the functions by user roles

Psychological Acceptability

- The security principle should be designed to maximize usage, adoption, and automatic application

- Discuss strong passwords as an example

Leverage Existing Components

- Use existing components when possible

Trust Relationship

- Every communication between parties must have some degree of trust associated with it
  - Trust relationship
- For simply communications systems, each system has full trust and allows the other complete access to its communication facilities
  - Not very secure

Trust Boundaries

- Distinguishes between regions of shared trust
  - Region of shared trust is a trust domain

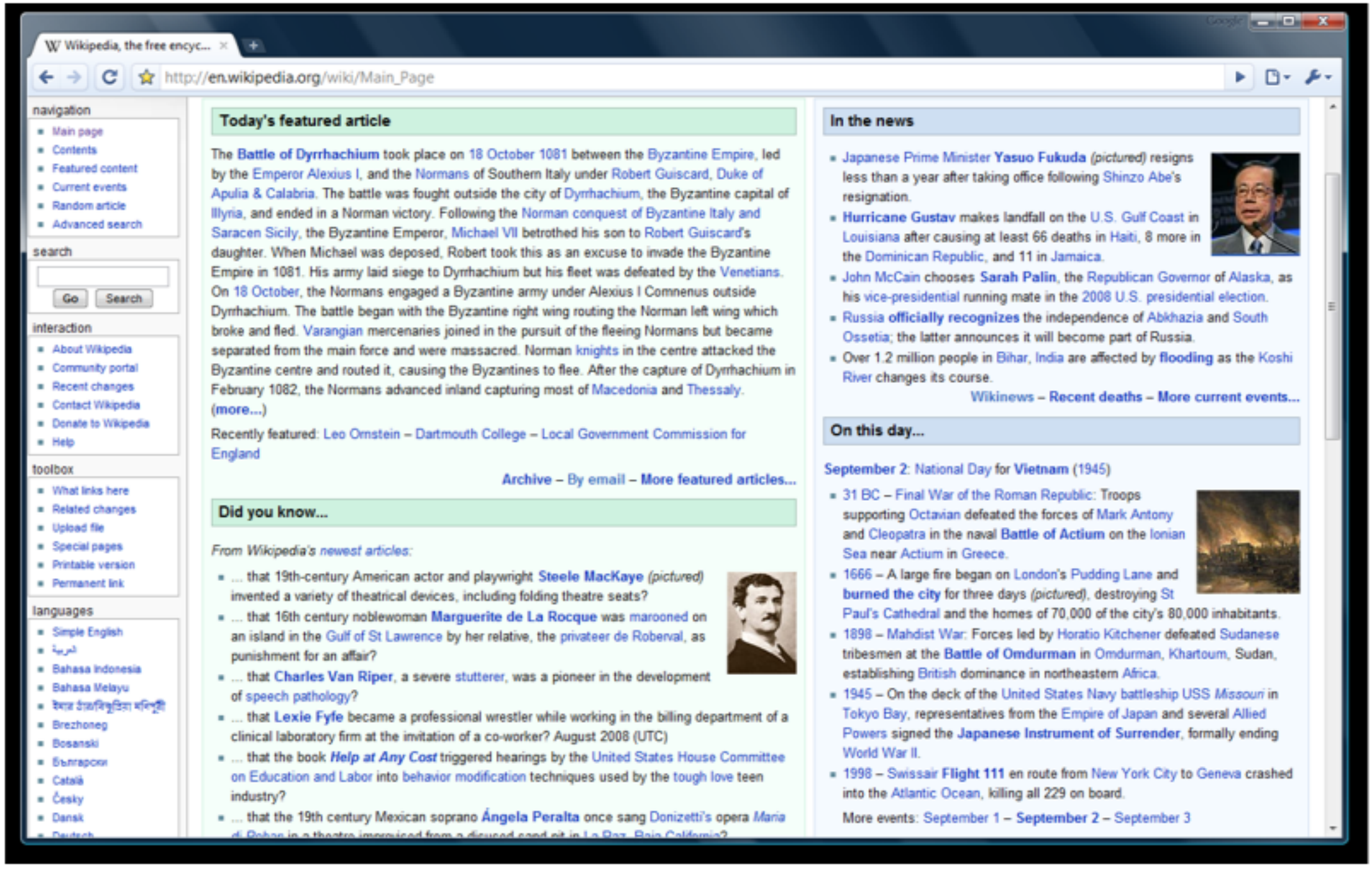
Design Gotcha's

- Strong coupling
  - Strong coupling indicates a high level of trust amongst components
  - High exposure of internal interfaces
  - High risk of problems
  - Data validation error prone and difficult
- Strong cohesion
  - Strong cohesion indicates module handles only one specific task

Design Gotcha's

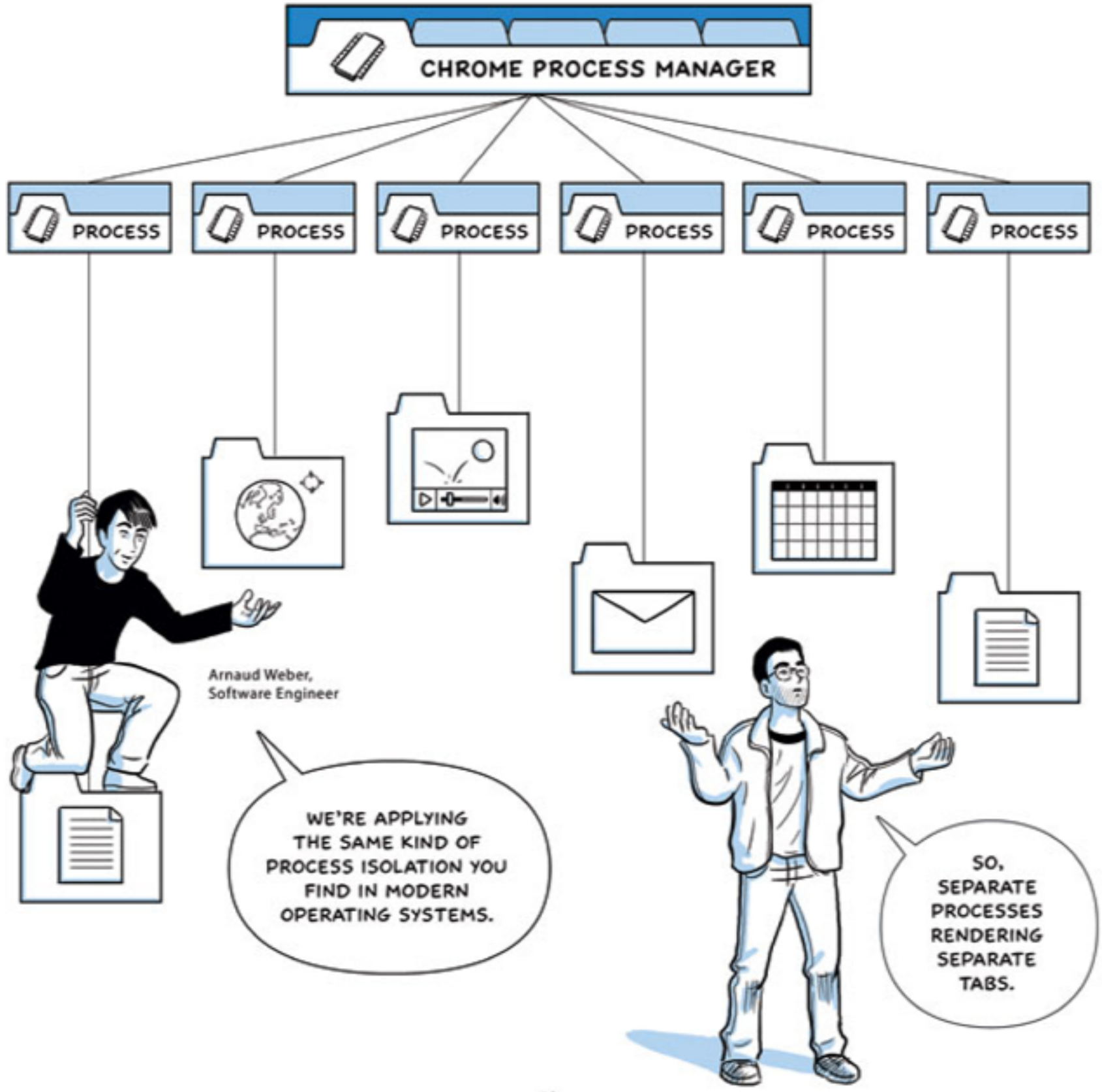
- Modules which cross trust boundaries
  - Design decomposition which fail to decompose modules along trust boundaries

Google Chrome

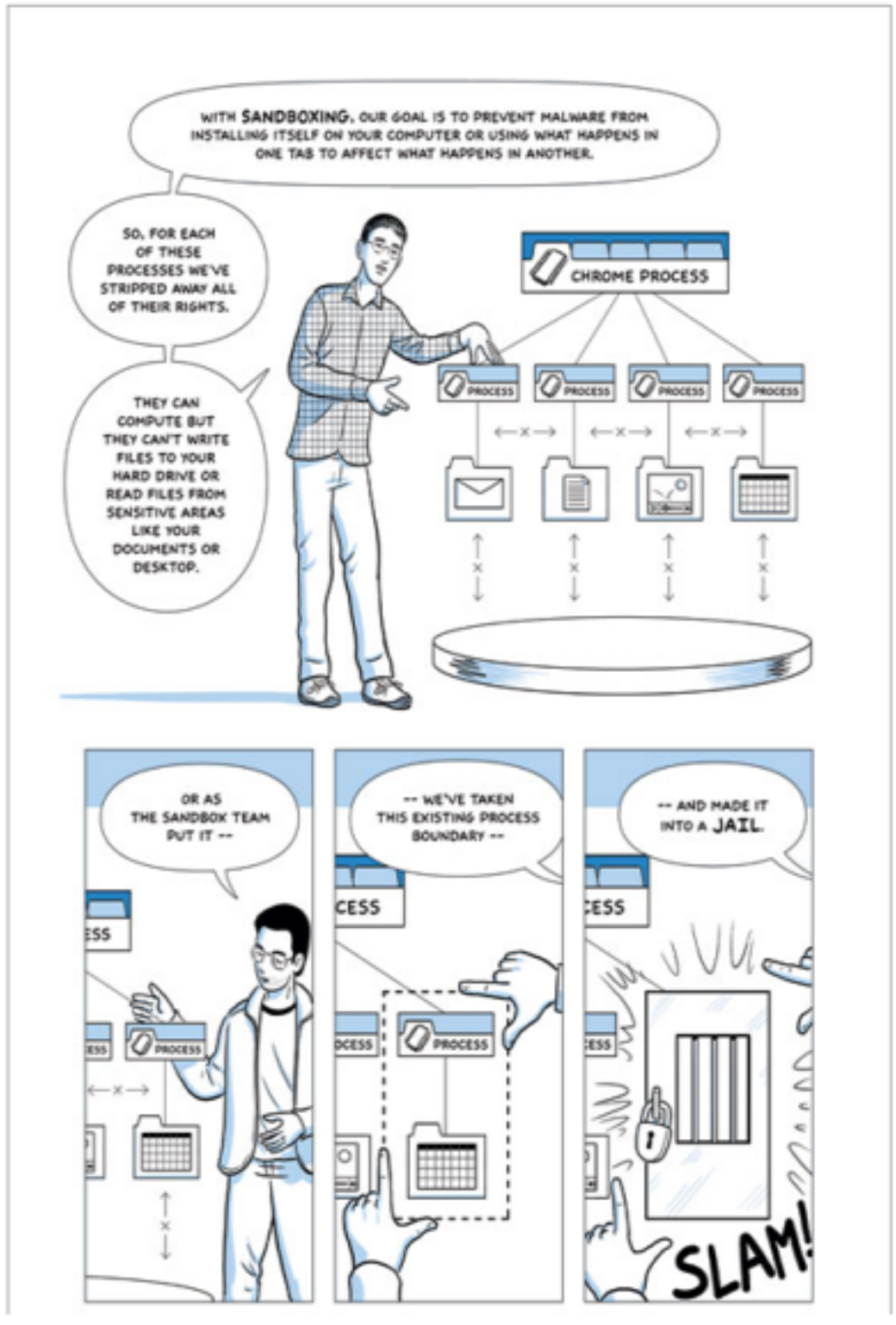


Google Chrome

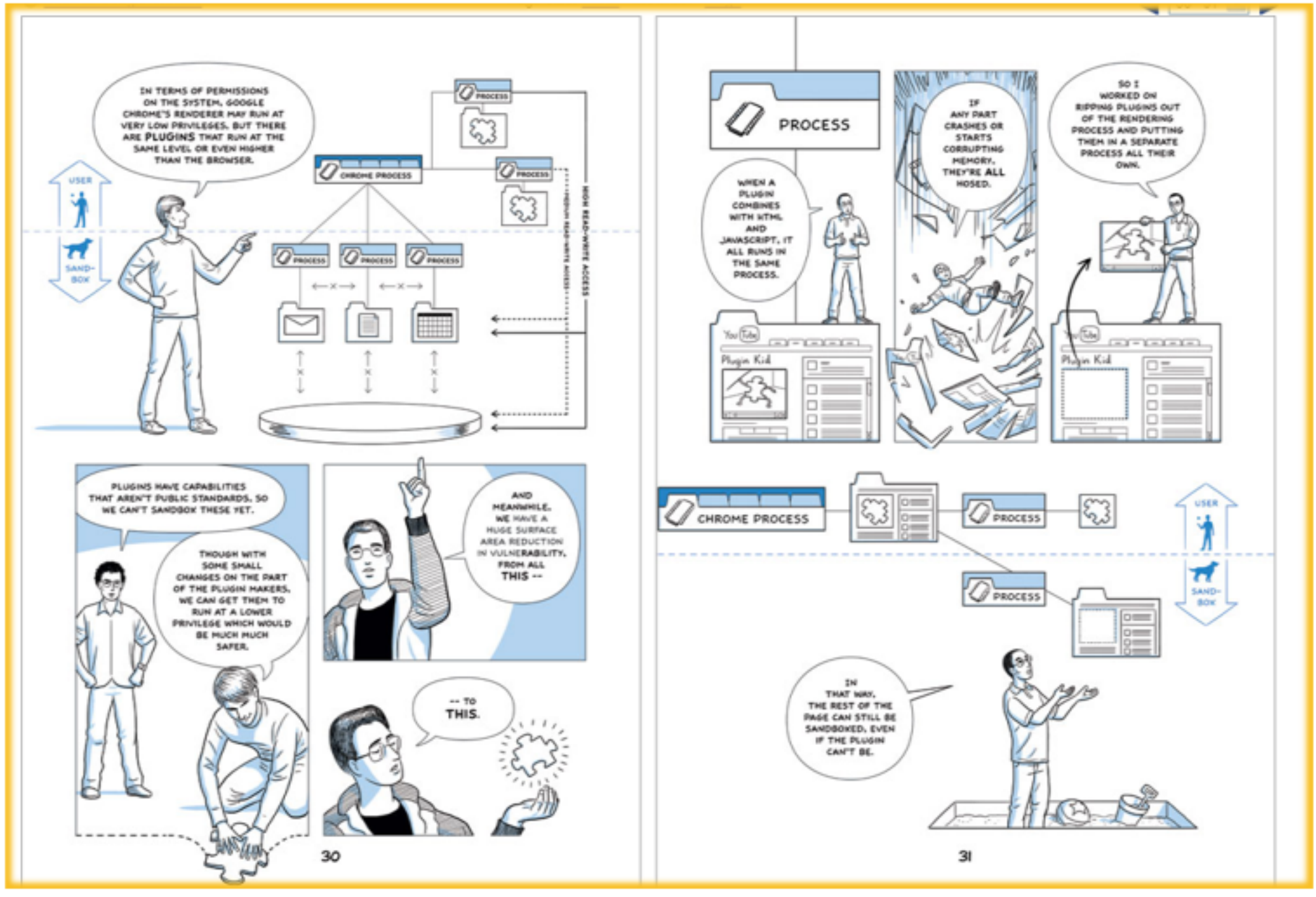
Google Chrome



Google Chrome



Google Chrome




Bank of America | Onli...
2009 Sports Illustrated...
Google Chrome - Wiki...
https://sitekey.bankofamerica.com/sas/signonScreen.do?state=WI
Bank of America
Online Banking
En Español
Sign In
Enter Online ID:
somebody
(6 - 32 characters)
Save this Online ID (How does this work?)
Sign In
Where do I enter my Passcode
Forgot or need help with your ID?
Not using Online Banking?
Enroll now
for Online Banking
Learn more
about Online Banking
Service Agreement
Go to Online Banking for
a state other than Wisconsin
Secure Area
Home . Locations . Contact Us . Help . Sign in . Site Map
Personal Finance . Small Business . Corporate & Institutional
About the Bank . In the Community . Finance Tools & Planning . Privacy & Security
Bank of America, N.A. Member FDIC. Equal Housing Lender
©2009 Bank of America Corporation. All rights reserved.
start
2 Firefox
Windows Task...
Presentation1
Lecture6Archit...
Jasc Paint Sho...
Bank of Ameri...
9:59 PM


MODELS
NBA DANCERS
TENNIS STARS
ON LOCATION
VIDEO
SWIMSUIT GOODIES
VIDEO LINEUP
ALL VIDEO
ARIEL MEREDITH
CHENEY LARSCHIED
MELISSA HARO
ALISON PRESTON
DANIELA HANTUCHOVA
Brooklyn
DECKER
VIDEO
PHOTOS
ALL
MODELS
SWEEPSTAKES
start

Chrome

- Each tab is its own process
  - Not thread
  - "prevent malware from installing itself" or "using what happens in one tab to affect what happens in another",
- Can not write files or read from sensitive areas (e.g. documents, desktop)
- two levels of security, user and sandbox
  - *sandbox* can only respond to communication requests initiated by the *user*.[34]
- Plugins are run in separate processes
  - communicate with the renderer in dedicated per-tab processes.[1]
- *Incognito* mode prevents the browser from storing any history information or cookies
  - Referred to as a porn mode

Google