RSAConference2019

San Francisco | March 4 – 8 | Moscone Center

BETTER.

SESSION ID: CSV-F01

Securely Deploying Micro Services, Containers & Serverless PaaS Web Apps

Murray Goldschmidt

Chief Operating Officer
Sense of Security
@ITsecurityAU

AGENDA

1	Serverless, Microservices and Container Security
2	Key Implications for Penetration Testing Programs
3	Key Security features for Container Deployments
4	CI/CD Integration for Automated Security
	End to End Vulnerability Management
	Continuous Monitoring, Governance & Compliance Reporting

Are Containers As Good as it Gets?

Cloud containers are designed to virtualize a single application

*** Modified *** https://searchcloudsecurity.techtarget.com/feature/Cloud-containers-what-they-are-and-how-they-work

As Good as it Gets?

e.g., you have a MySQL container and that's all it does, provide a virtual instance of that application.

*** Modified *** https://searchcloudsecurity.techtarget.com/feature/Cloud-containers-what-they-are-and-how-they-work

As Good as it Gets?

Containers ***SHOULD*** create an *isolation boundary* at the application level rather than at the server level.

*** Modified *** https://searchcloudsecurity.techtarget.com/feature/Cloud-containers-what-they-are-and-how-they-work

As Good as it Gets?

This isolation ***SHOULD*** mean that if anything goes wrong in that single container (e.g., excessive consumption of resources by a process) it only affects that individual container and not the whole VM or whole server.

*** Modified *** https://searchcloudsecurity.techtarget.com/feature/Cloud-containers-what-they-are-and-how-they-work

Secure | https://docs.docker.com/engine/security/security/

docker docs | Search the docs | Guides | Product manuals | Glossary | Reference | Samples

- Get Docker
- Get started
- Develop with Docker
- Configure networking
- Manage application data
- Run your app in production

Docker security

Estimated reading time: 10 minutes

There are four major areas to consider when reviewing Docker security:

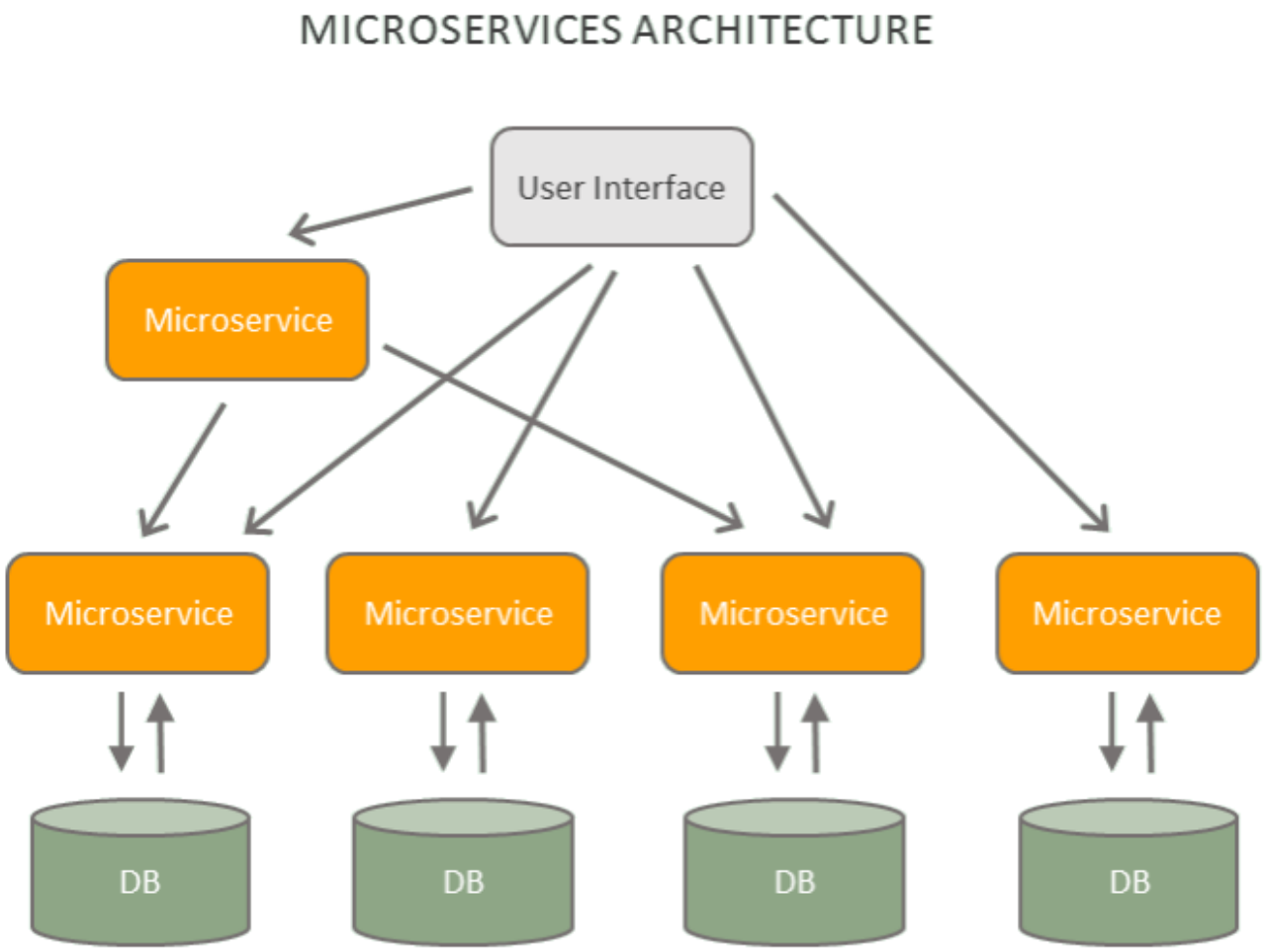
- the intrinsic security of the kernel and its support for namespaces and cgroups;
- the attack surface of the Docker daemon itself;
- loopholes in the container configuration profile, either by default, or when customized by users.
- the "hardening" security features of the kernel and how they interact with containers.

Container Security – Tech Neutral

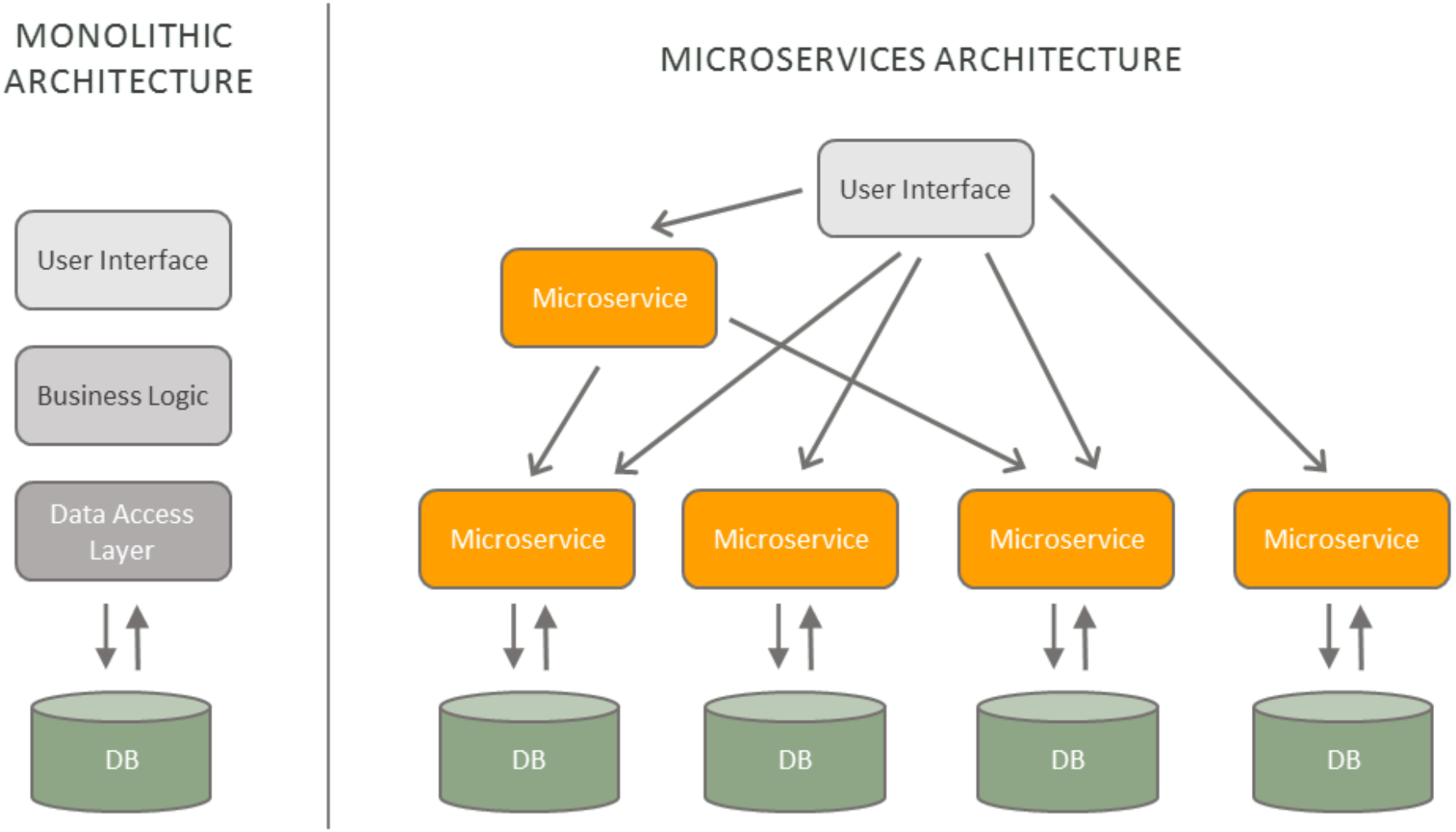
Security Requirements	Addressed By
Intrinsic Security of the Kernel	Supply Chain Risk Mgt/ Vuln Mgt/ CaaS
Attack Surface Reduction	Hardening/Config Mgt/Vuln Mgt
Container Configuration	Configuration Management
Hardening of the Kernel and how it interacts with Containers	Hardening

Monolithic vs Microservices Architecture

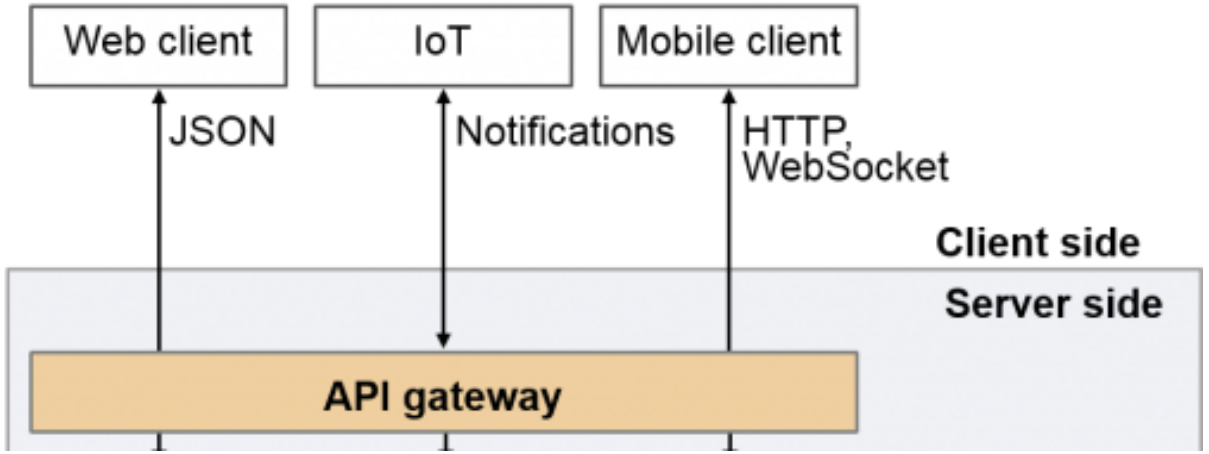
Monolithic vs Microservices Architecture



Monolithic vs Microservices Architecture

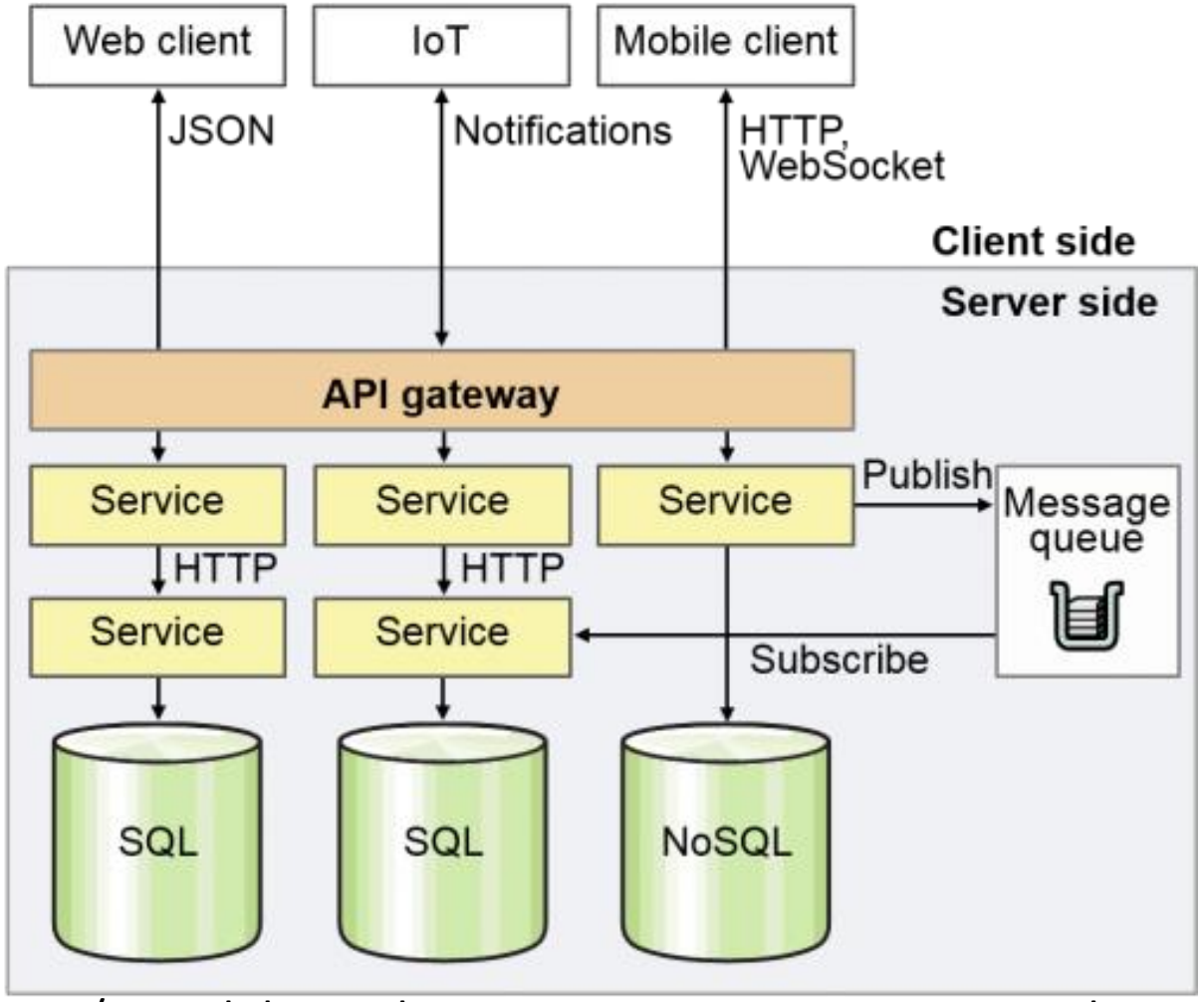


Monolithic vs Micro Services (API Centric)



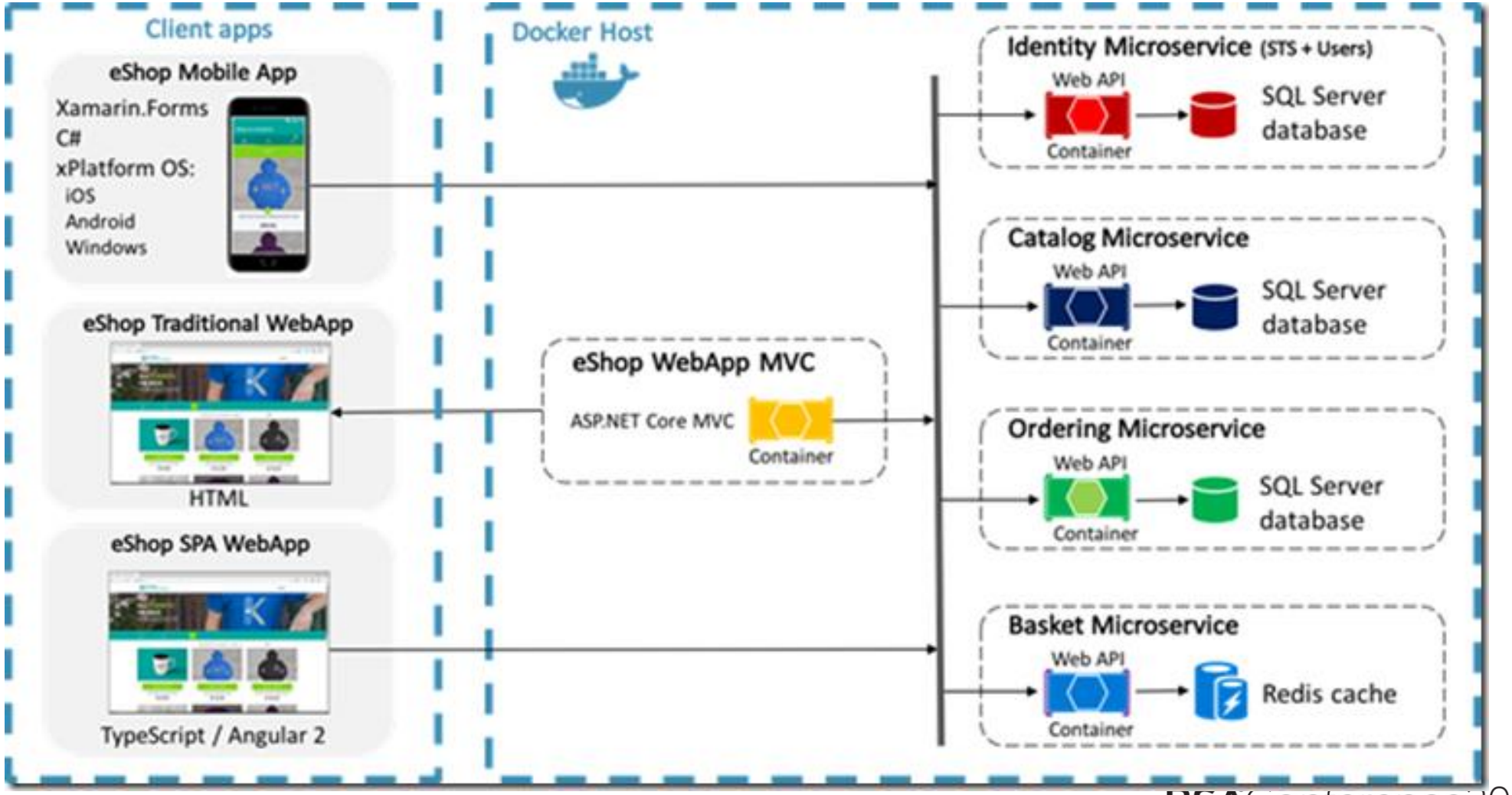
https://developer.ibm.com/courses/monolithic-architecture-versus-microservices-architecture-dwc024/

Monolithic vs Micro Services (API Centric)

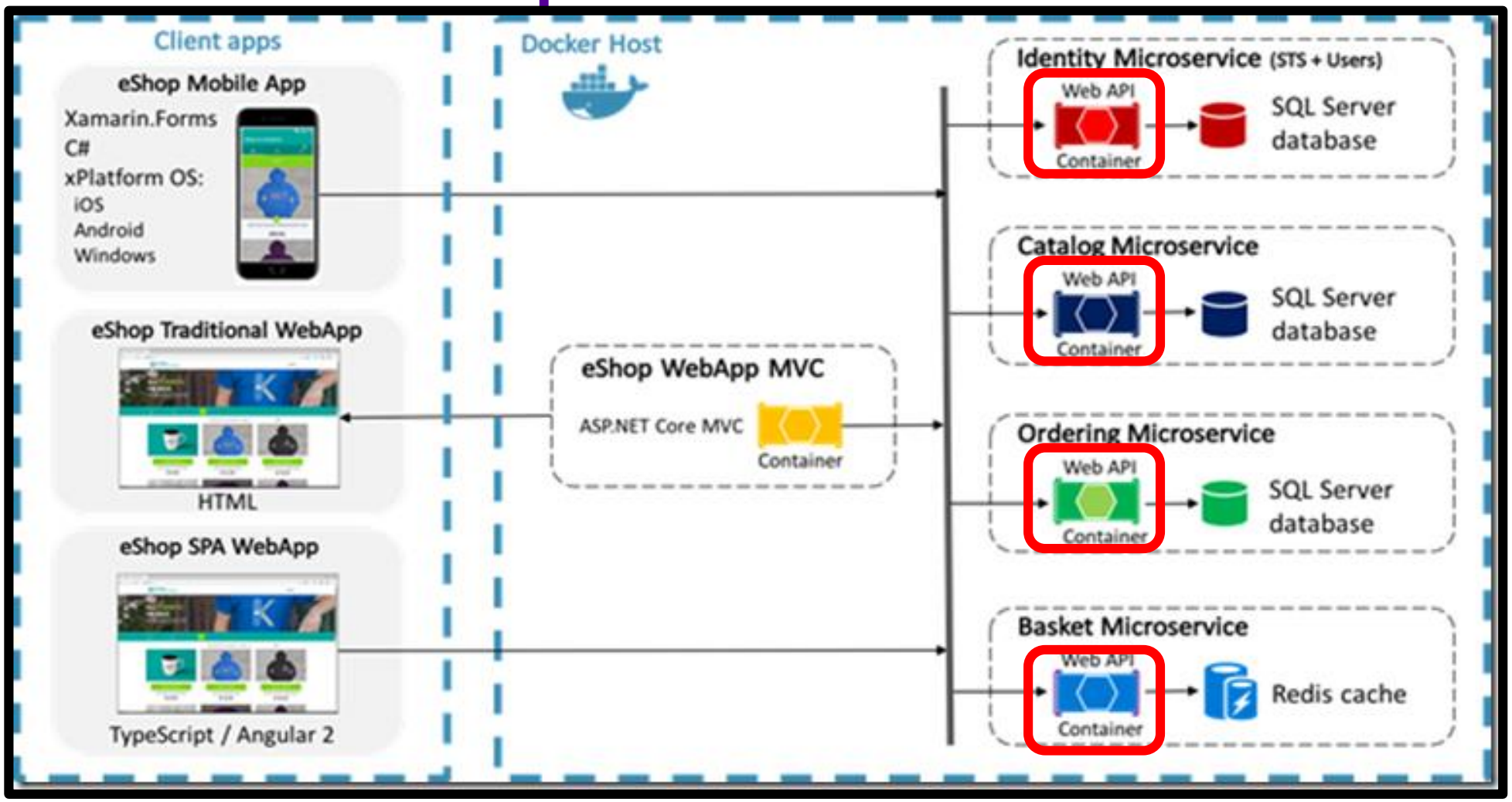


https://developer.ibm.com/courses/monolithic-architecture-versus-microservices-architecture-dwc024/

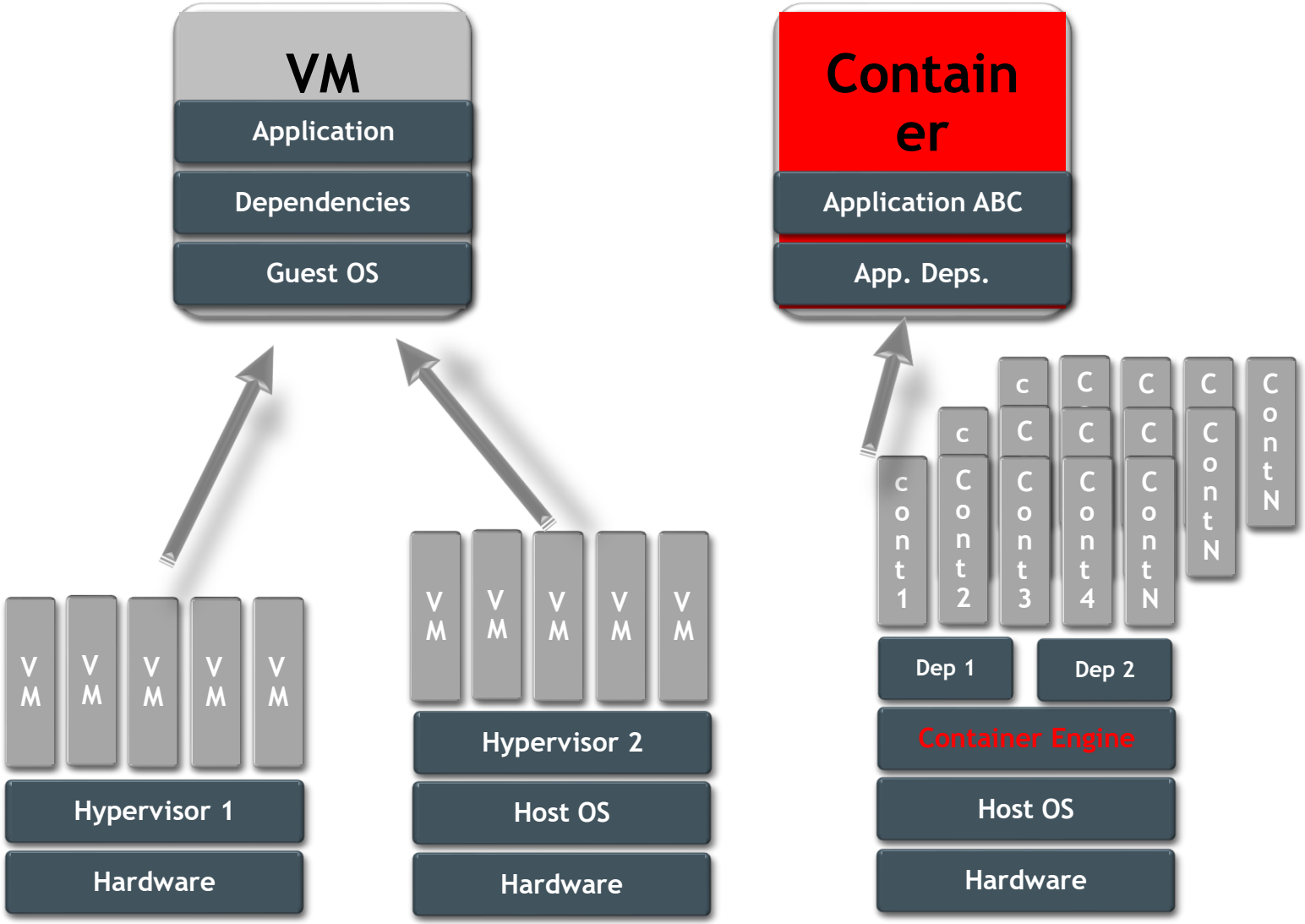
Example: Microsoft eShop Reference Architecture



Example: Microsoft eShop Reference Architecture



VM vs. Containers (where the abstraction occurs)



Type1 - Bare Metal

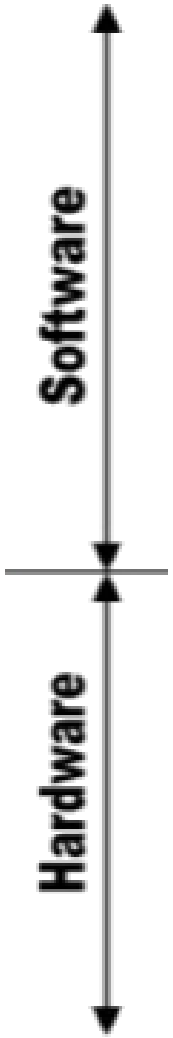
Type 2

Virtualisation

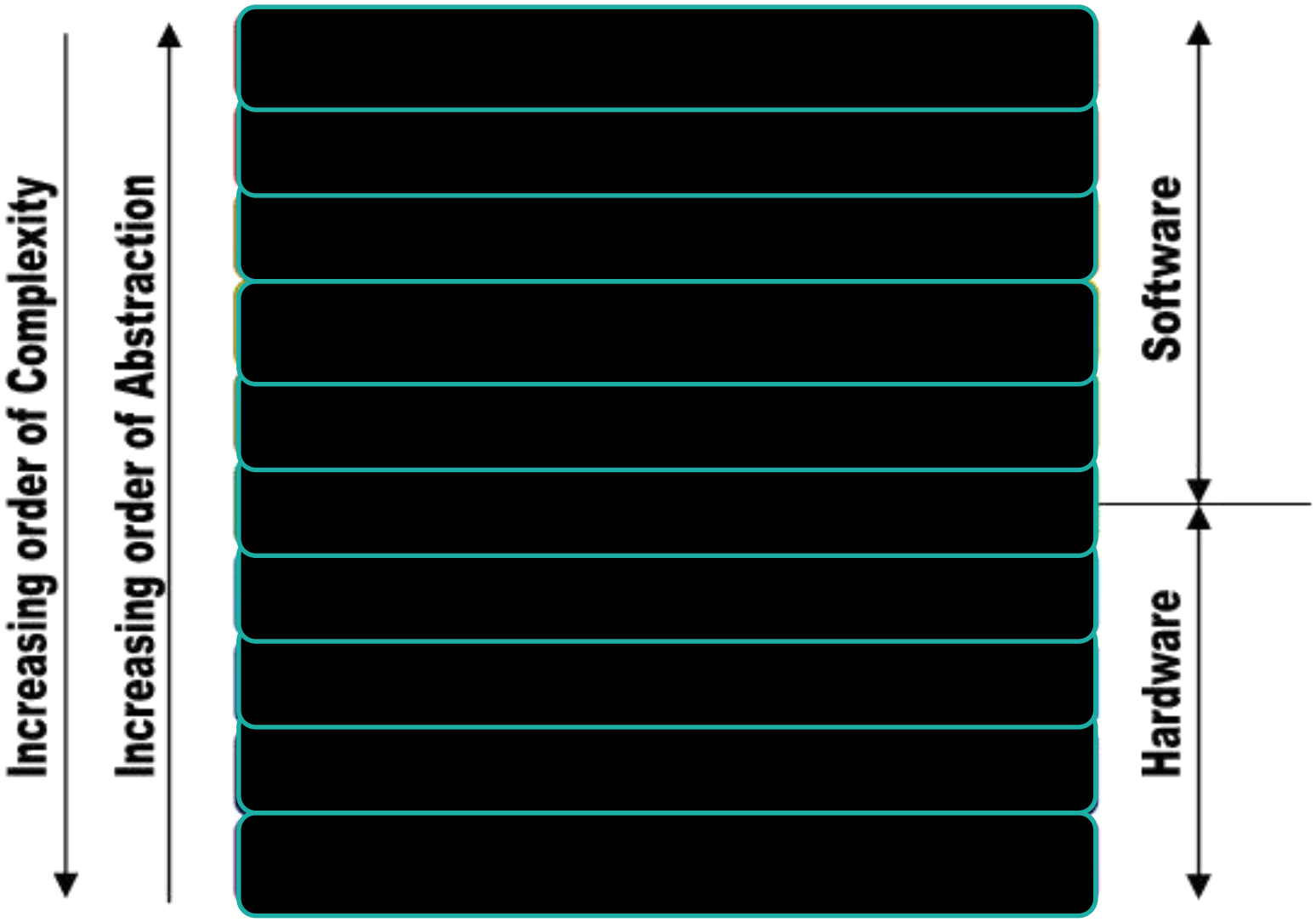
Containerisation

Increasing order of Complexity

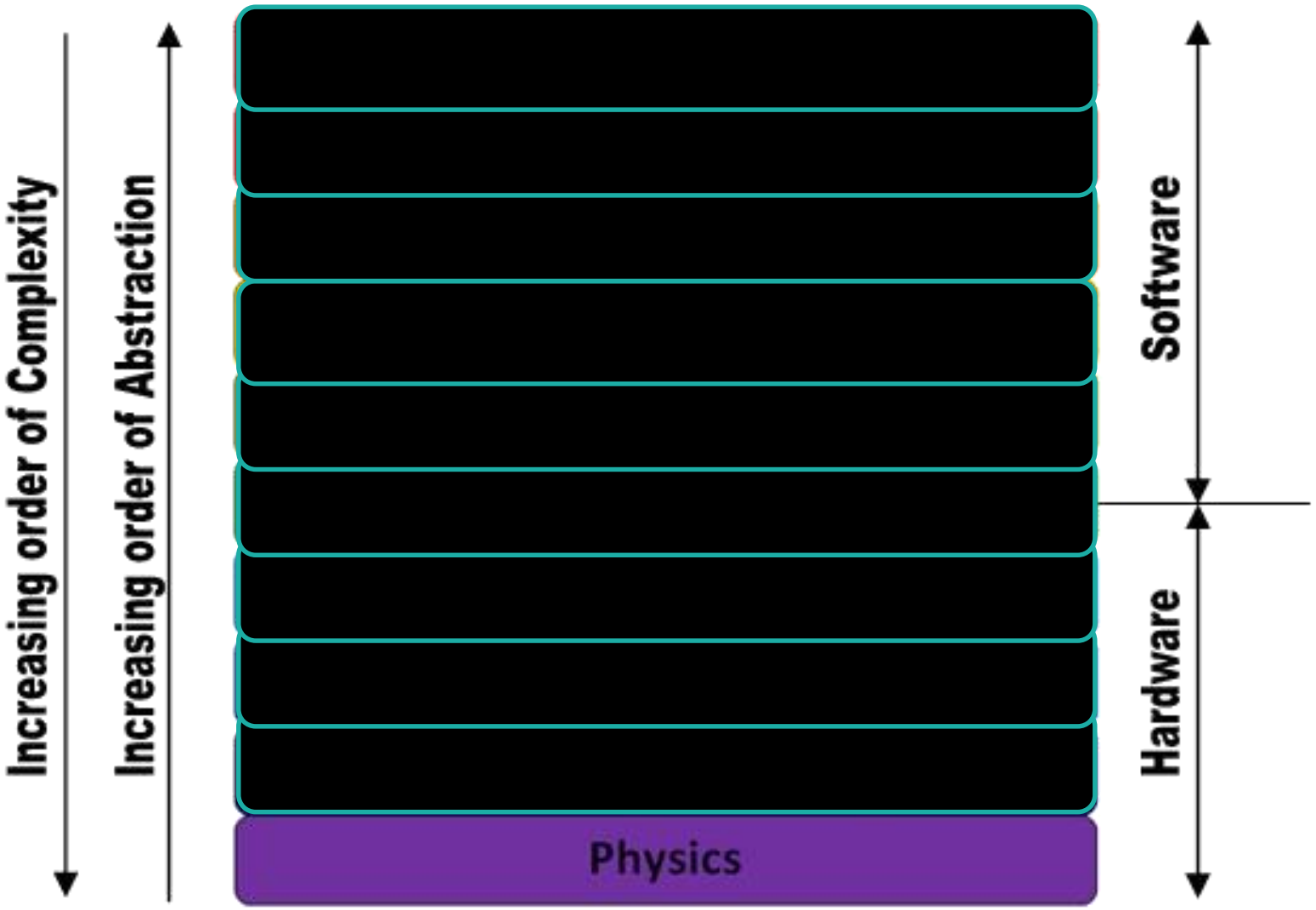
Increasing order of Abstraction



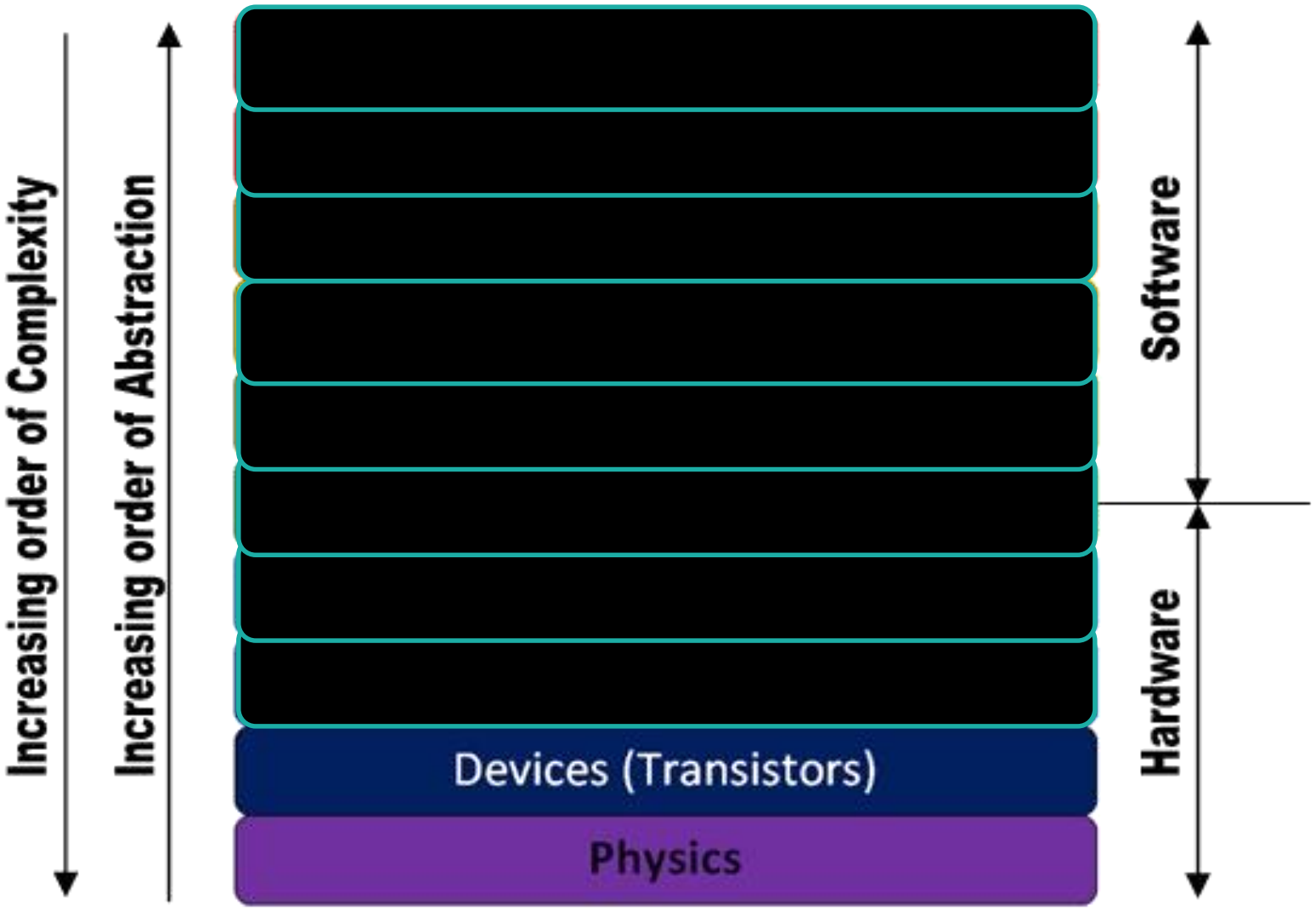
Layers of Abstraction



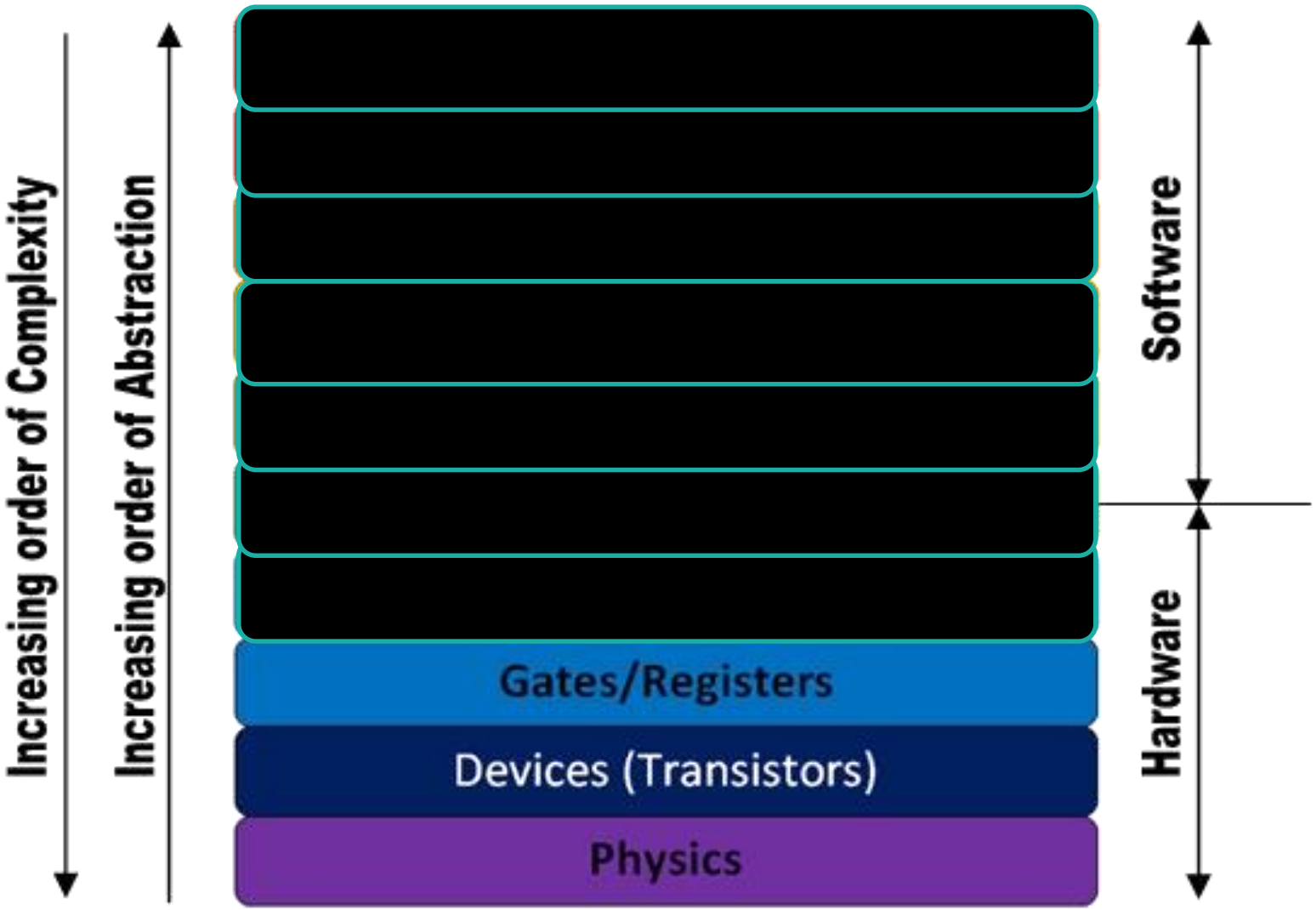
Layers of Abstraction



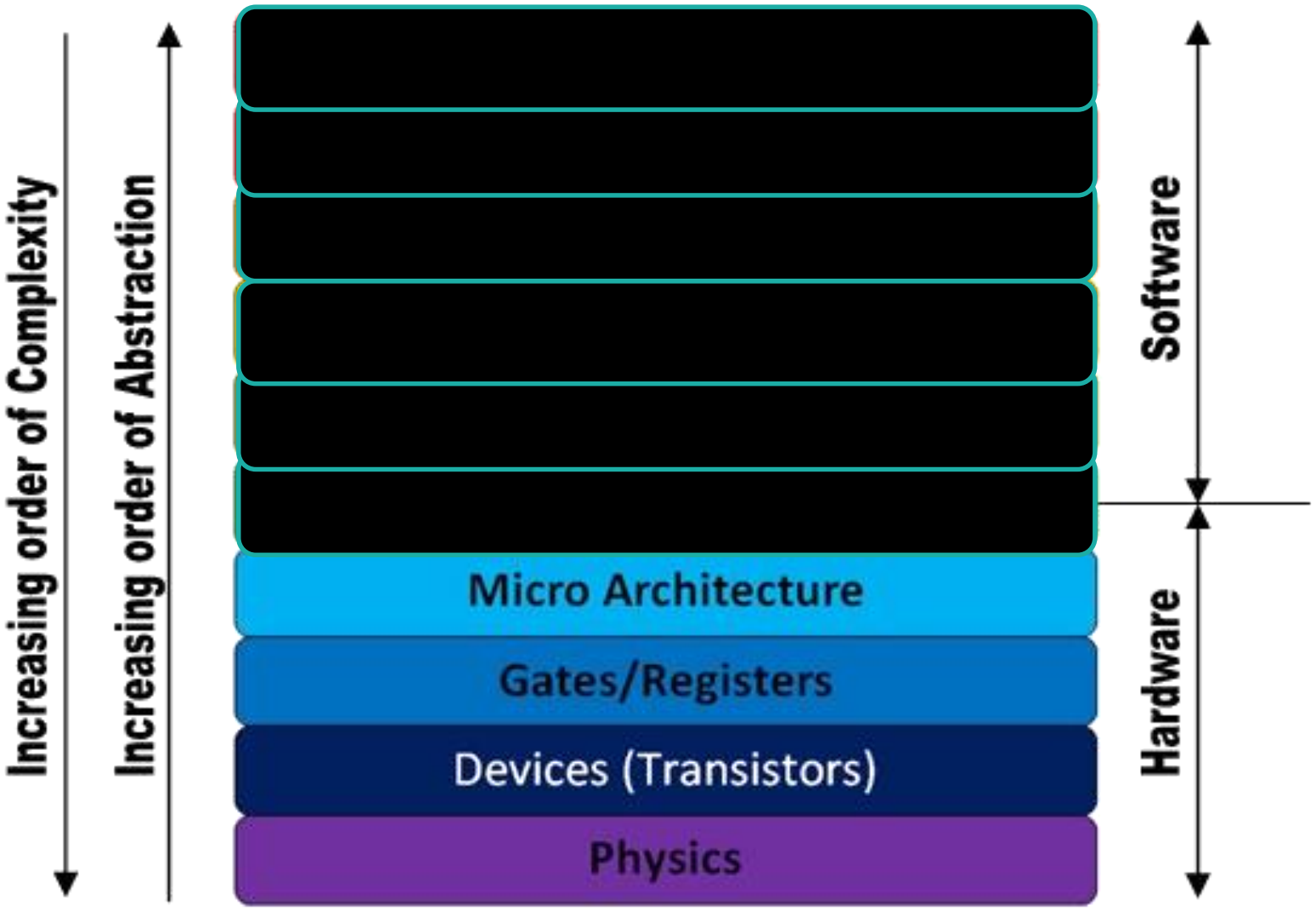
Layers of Abstraction



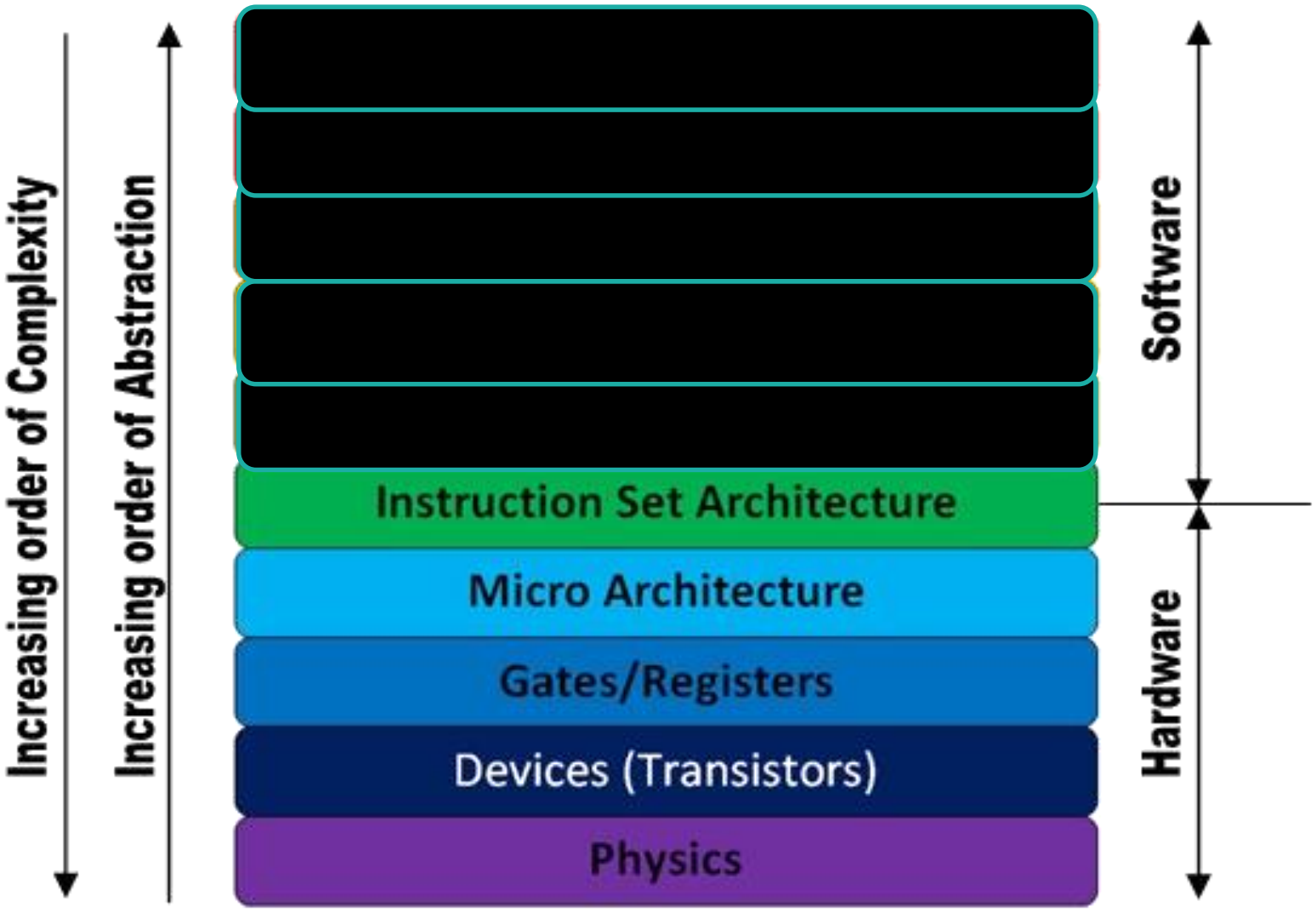
Layers of Abstraction



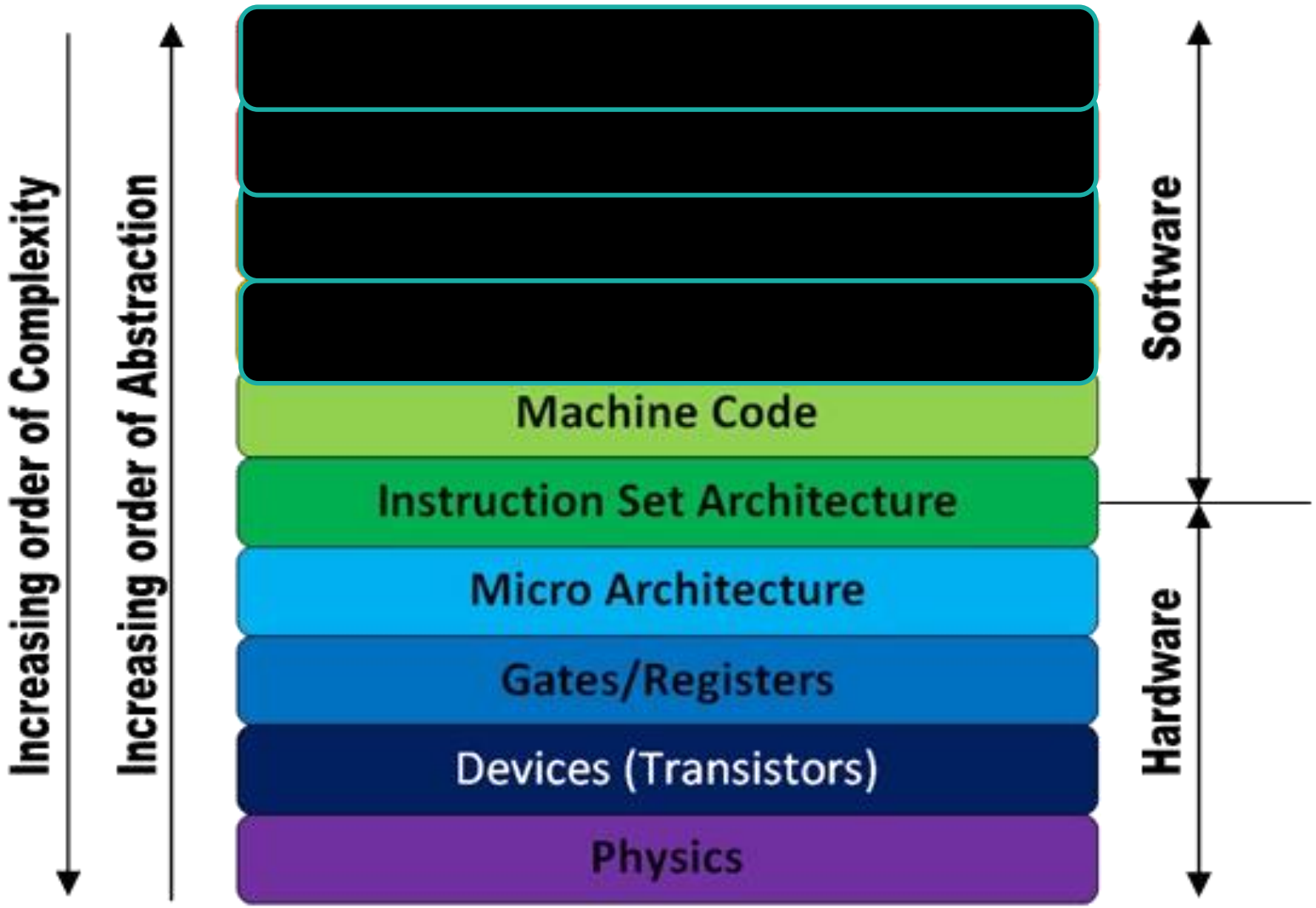
Layers of Abstraction



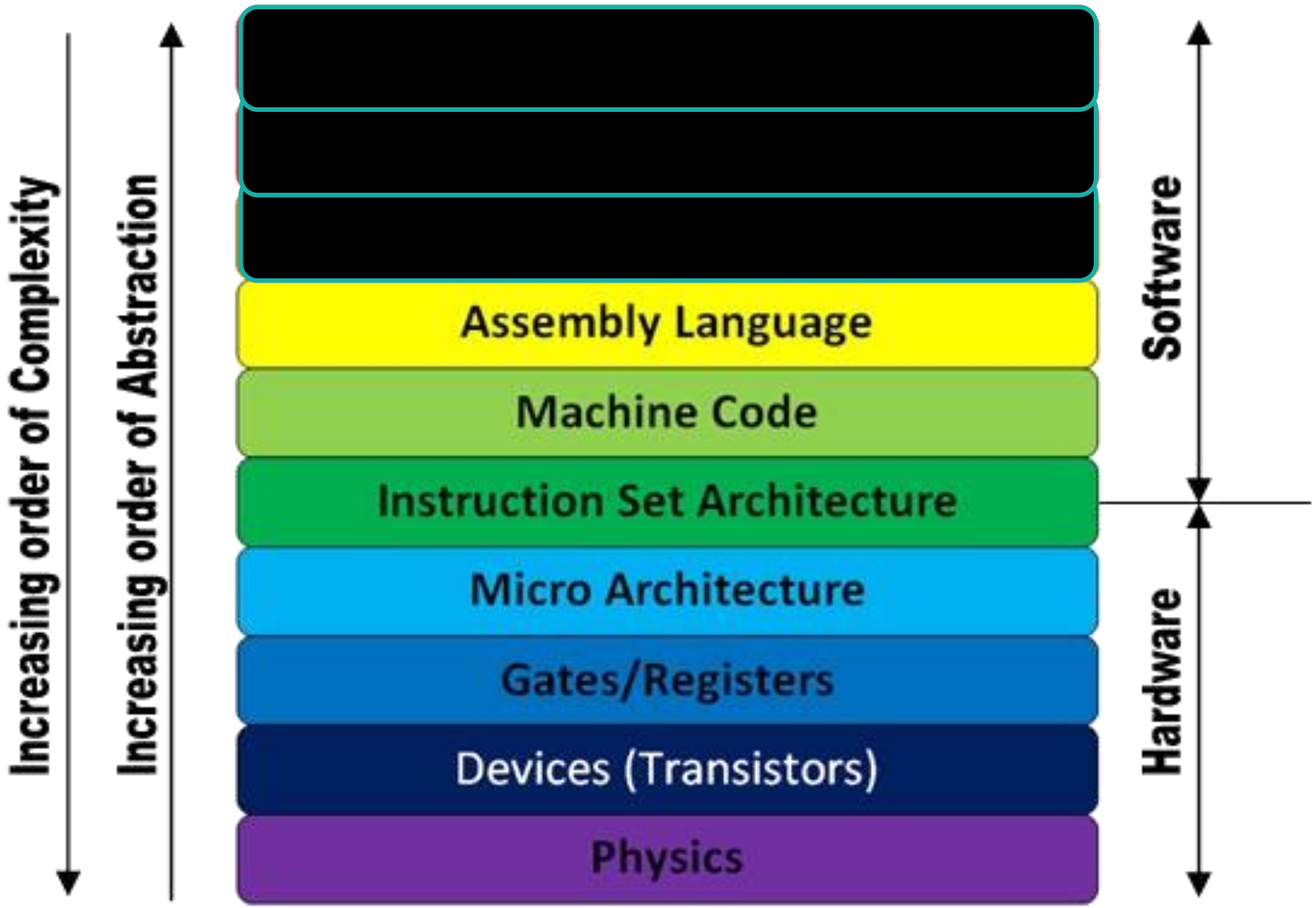
Layers of Abstraction



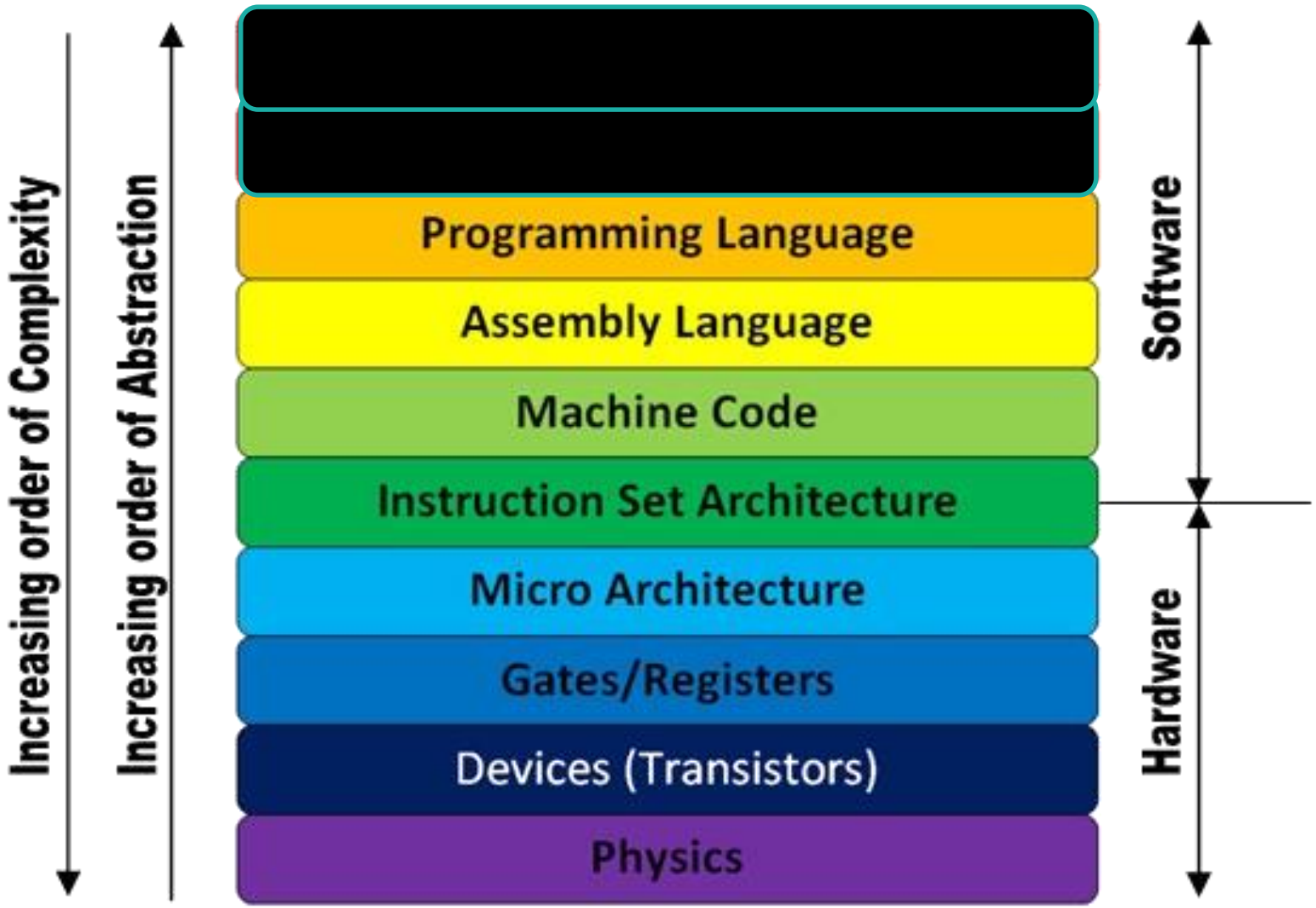
Layers of Abstraction



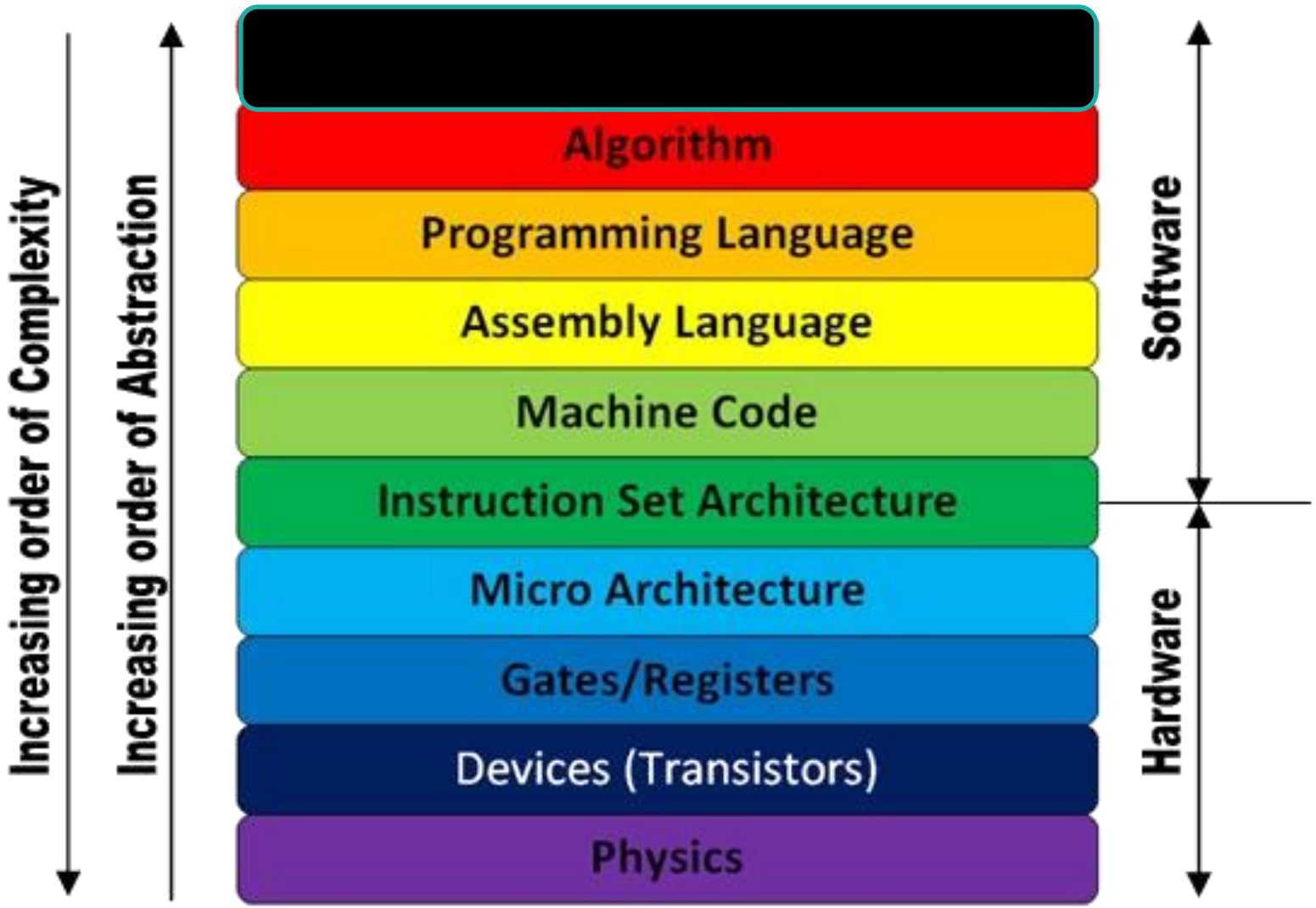
Layers of Abstraction



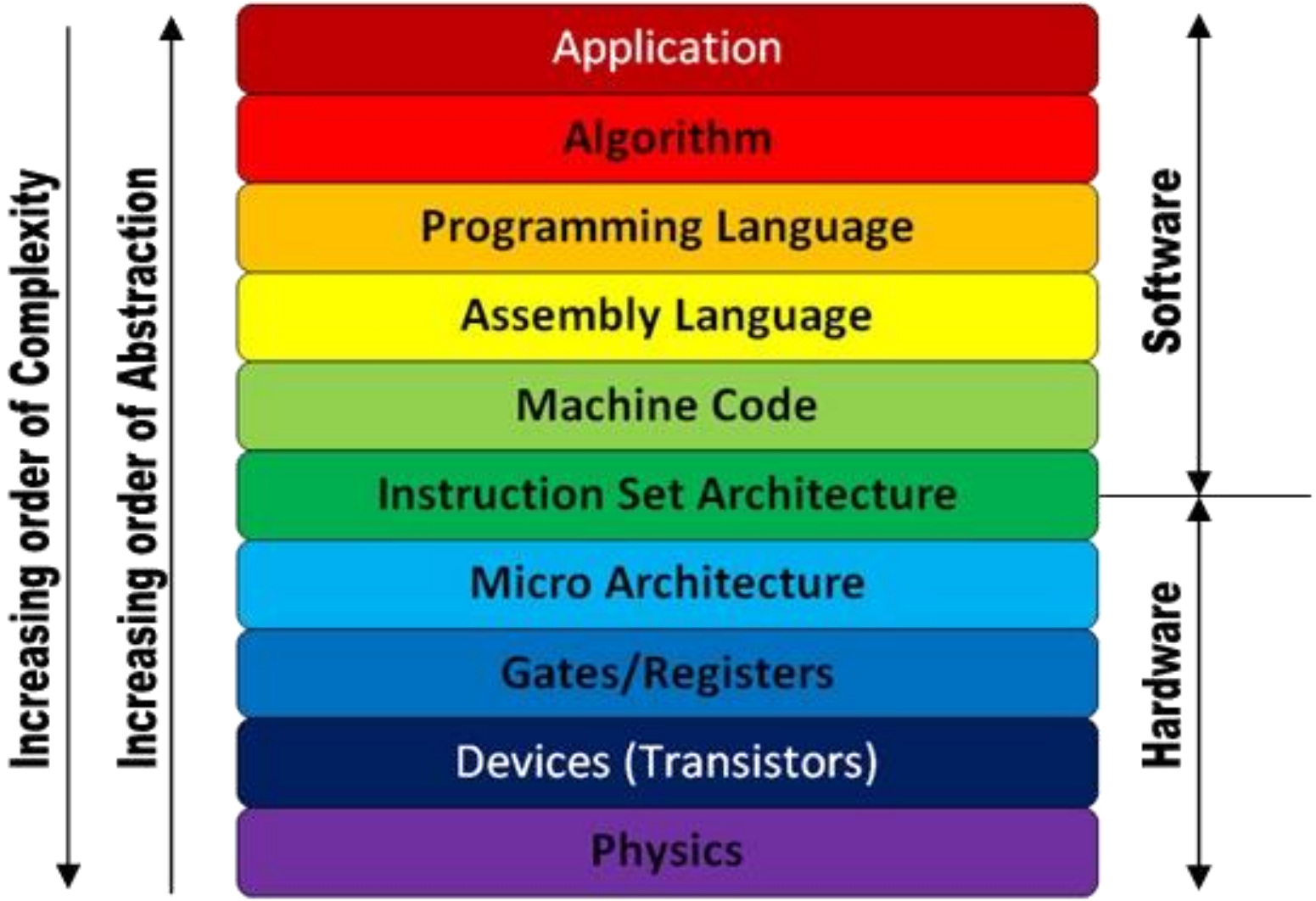
Layers of Abstraction



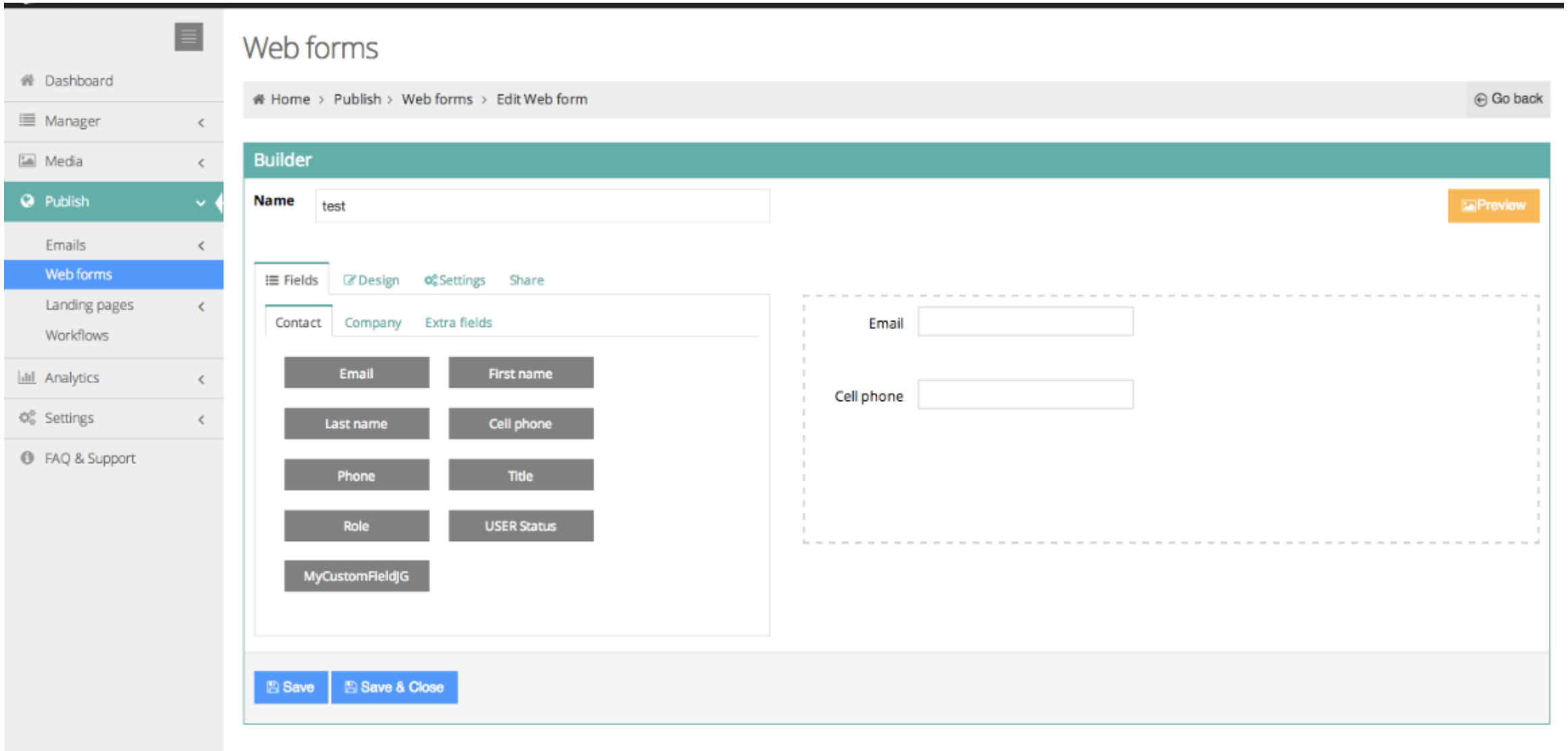
Layers of Abstraction



Layers of Abstraction



Layers of Abstraction

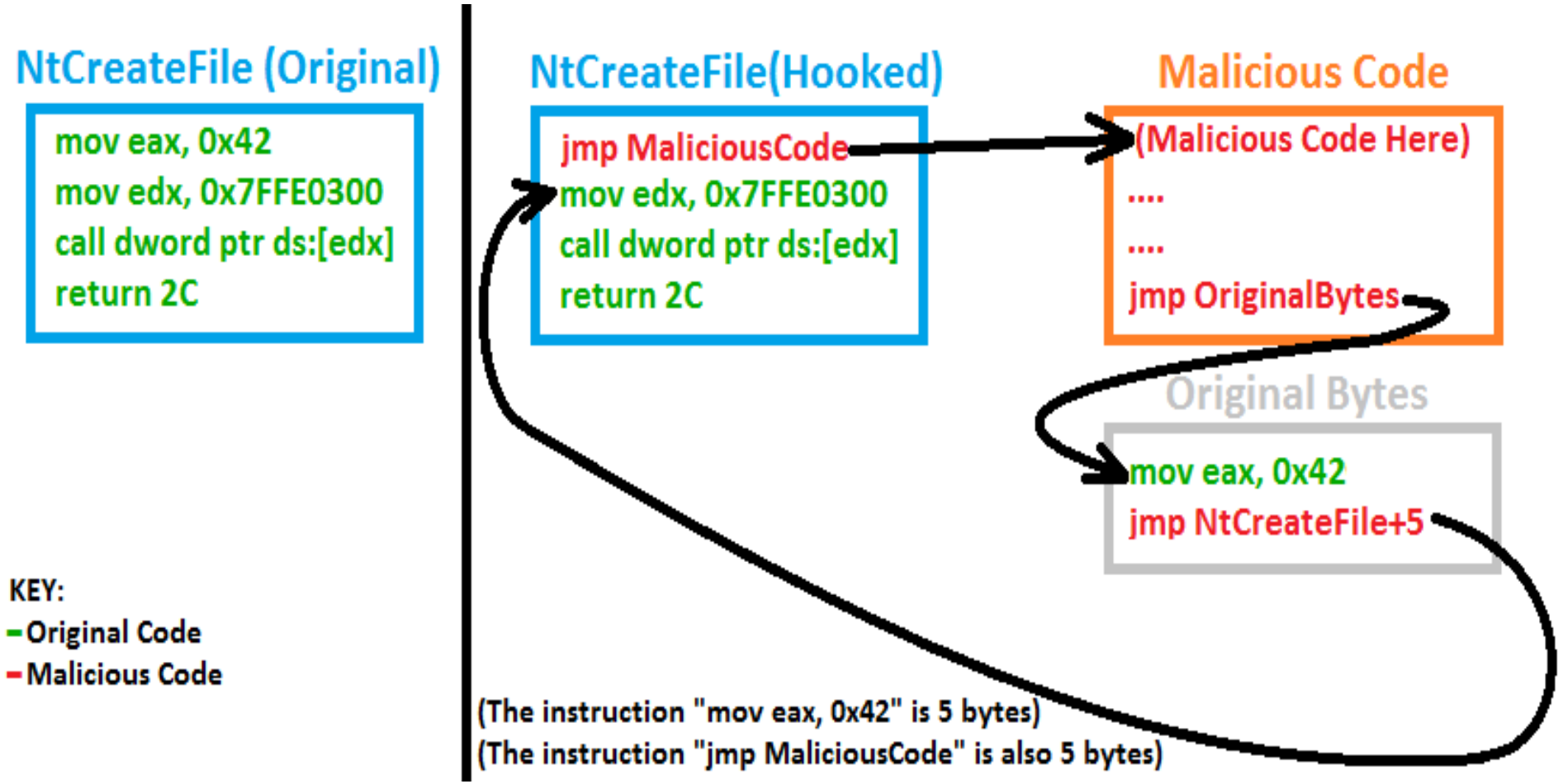

Web forms
Dashboard
Manager
Media
Publish
Emails
Web forms
Landing pages
Workflows
Analytics
Settings
FAQ & Support
Home > Publish > Web forms > Edit Web form
Go back
Builder
Name
test
Preview
Fields
Design
Settings
Share
Contact
Company
Extra fields
Email
First name
Last name
Cell phone
Phone
Title
Role
USER Status
MyCustomFieldJG
Email
Cell phone
Save
Save & Close

Developers

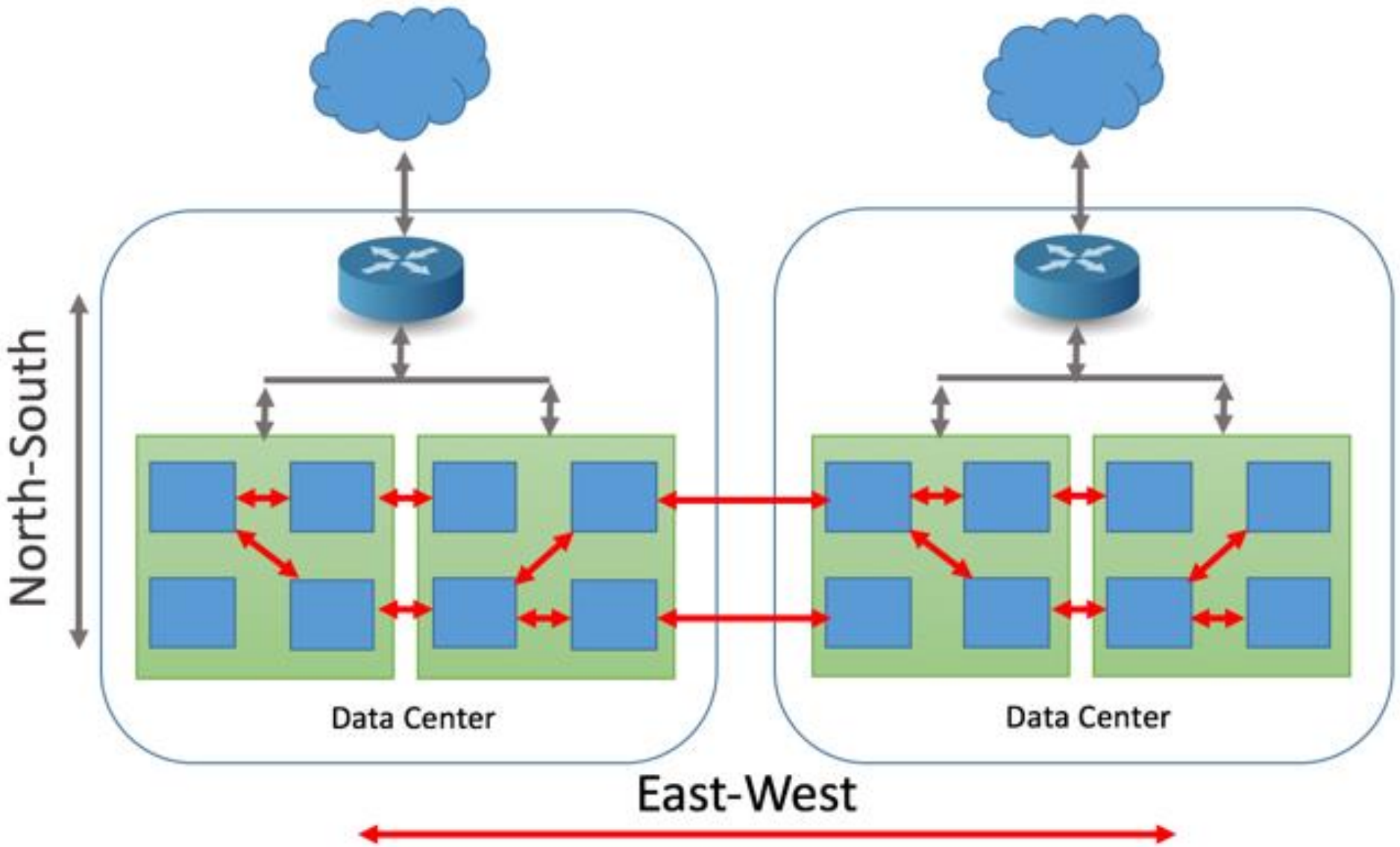
Hackers

RACE TO THE BOTTOM

Hooking Lowest Wins



North-South & East-West Attacks and Pivots



https://neuvector.com/network-security/securing-east-west-traffic-in-container-based-data-center/

Break-In

Entry Point is usually a “Pin Hole” issue

For example a known application issue


VAULT DOOR
WEIGHT: 22 1/2 Tons
THICKNESS: 22 Inches
STEEL: 11 Layers of Special Cutting and Drill Resistant
LOCKS: 4 Hamilton Watch Movements for Time Locks

Containers – The "Contained" Challenge

IF you can Break-In

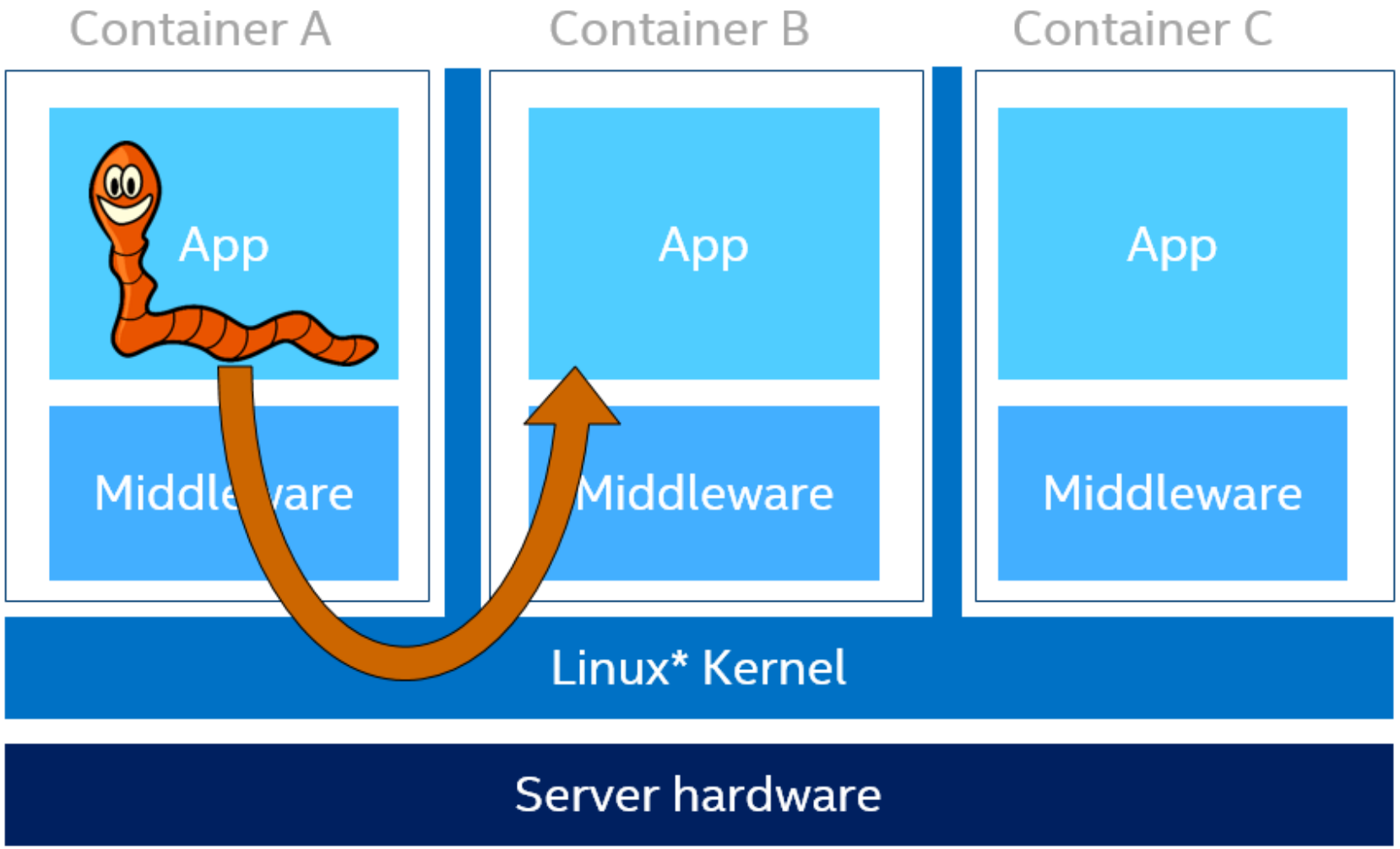
You then Need to Break-Out

Break-Out

<goWest

goEast>

Either Find a Container Vuln & Exploit



Recent Container Vulnerabilities

Introduction (CVE-2019-5736)

Today, Monday, 2019-02-11, 14:00:00 CET CVE-2019-5736 was released:

> *The vulnerability allows a malicious container to (with minimal user interaction) overwrite the host runc binary and thus gain root-level code execution on the host. The level of user interaction is being able to run any command (it doesn't matter if the command is not attacker-controlled) as root within a container in either of these contexts:*
>
> - *Creating a new container using an attacker-controlled image.*
> - *Attaching (docker exec) into an existing container which the attacker had previous write access to.*

- https://brauner.github.io/2019/02/12/privileged-containers.html

Recent Container Vulnerabilities

CVE-2019-5736 Is a Very Very Very Bad Privilege Escalation to Host Root

CVE-2019-5736 is an excellent illustration of such an attack. Think about it: a process running **inside** a privileged container can rather trivially corrupt the binary that is used to attach to the container. This allows an attacker to create a custom ELF binary on the host. That binary could do anything it wants:

- could just be a binary that calls `poweroff`
- could be a binary that spawns a root shell
- could be a binary that kills other containers when called again to attach
- could be `suid` `cat`
- .
- .
- .

- https://brauner.github.io/2019/02/12/privileged-containers.html

Recent Container Vulnerabilities

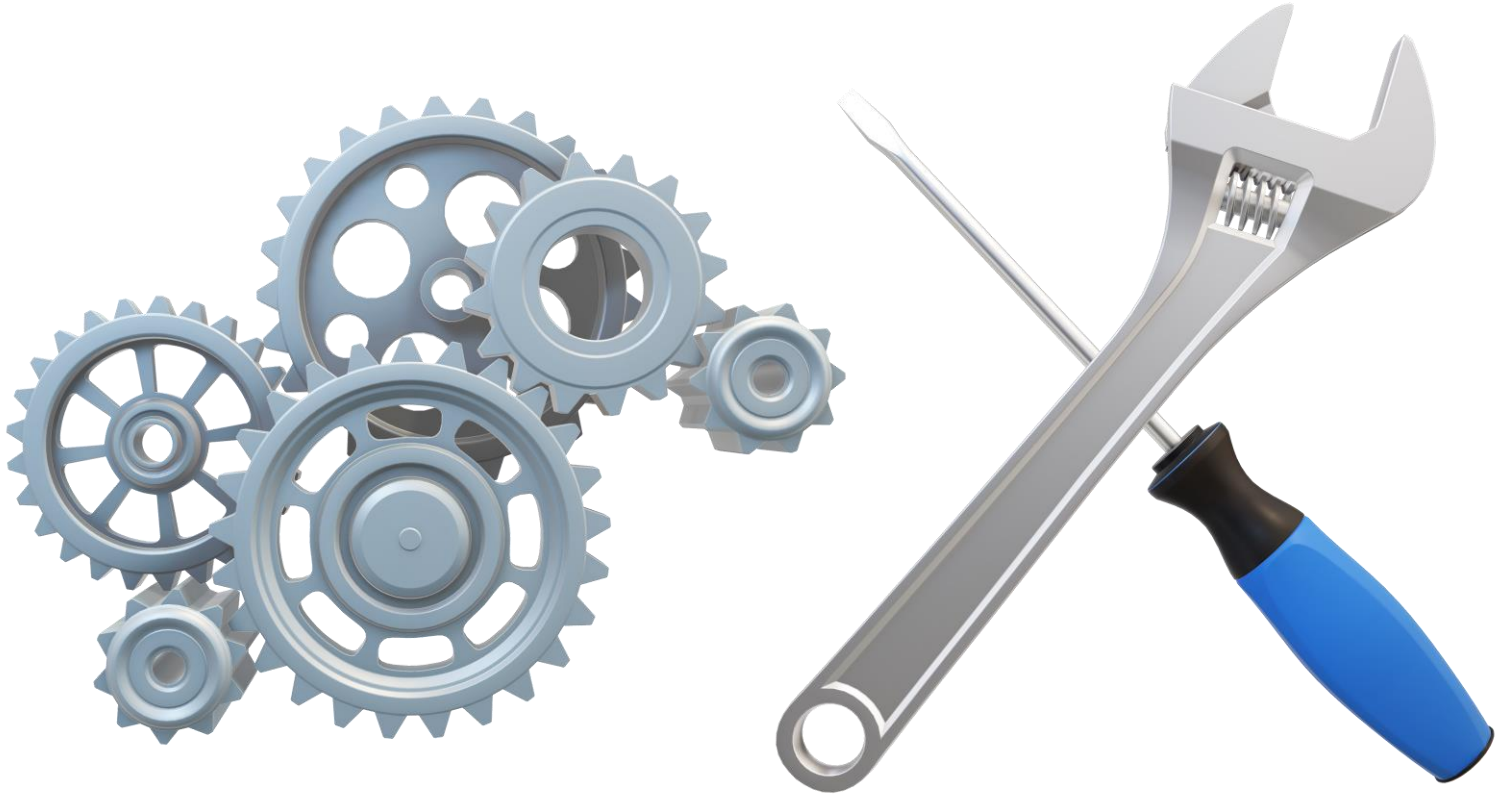
December 06, 2018

Severe Privilege Escalation Vulnerability in Kubernetes (CVE-2018-1002105)

Earlier this week, a severe vulnerability in Kubernetes (CVE-2018-1002105) was disclosed that allows an unauthenticated user to perform privilege escalation and gain full admin privileges on a cluster. The CVE was given the high severity score of 9.8 (out of 10) and it affects all Kubernetes versions from 1.0 onwards, but fixes are available for recent versions.

The first to discover this vulnerability was Darren Shepherd from Rancher, and it was quickly fixed by the community with updates to the major upstream K8s releases v1.10,

Or - Living off the Land



Low TTL Bi-Product
Hacked container may very soon be pulled down.
Much harder for hacker persistence.
Ability to refresh environment quickly – Vuln Mgt improvements e.g. Secure @ Source

Low TTL Challenge
Hard for Forensics and Monitoring
Vuln Mgt – environment constantly changing
Config Mgt – environment constantly changing

Container TTL

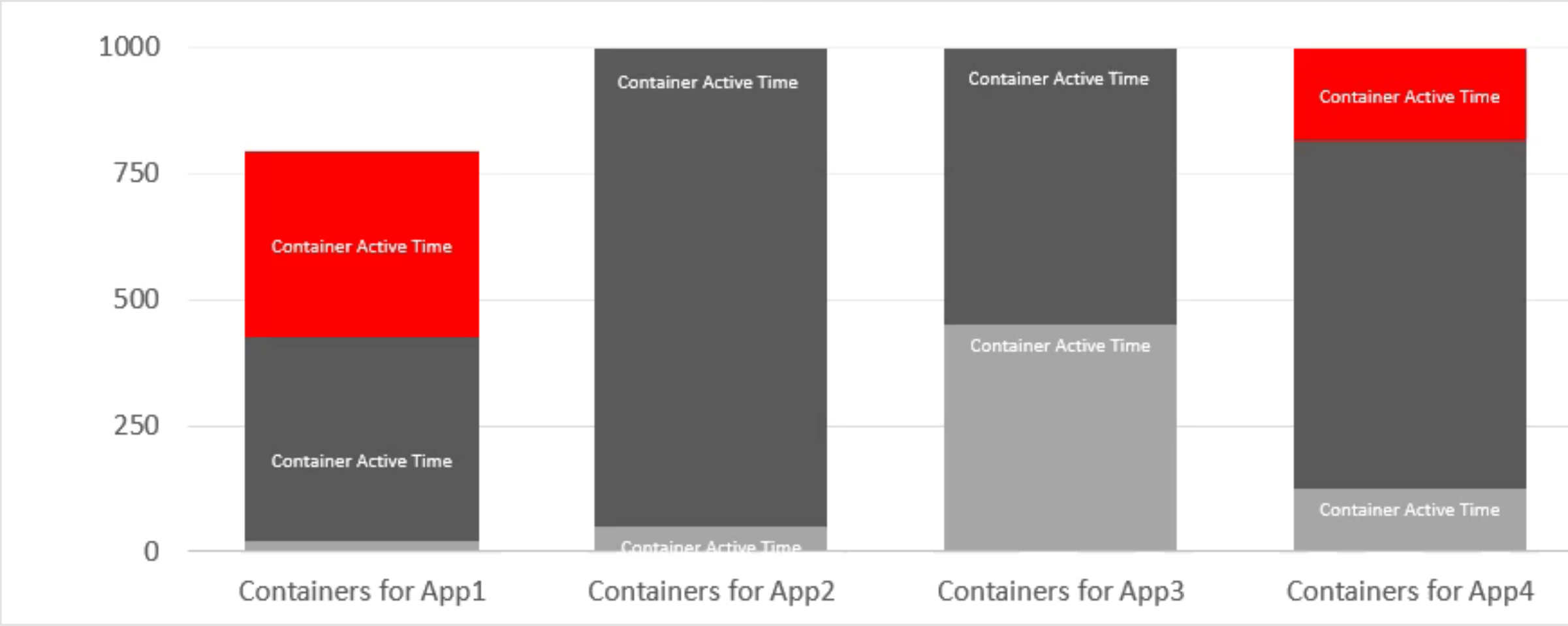
Low TTL Bi-Product	Low TTL Challenge
Hacked container may very soon be pulled down.	Hard for Forensics and Monitoring
Much harder for hacker persistence.	Vuln Mgt – environment constantly changing
Ability to refresh environment quickly – Vuln Mgt improvements e.g. Secure @ Source	Config Mgt – environment constantly changing

Content Slide Layout

Play

App6

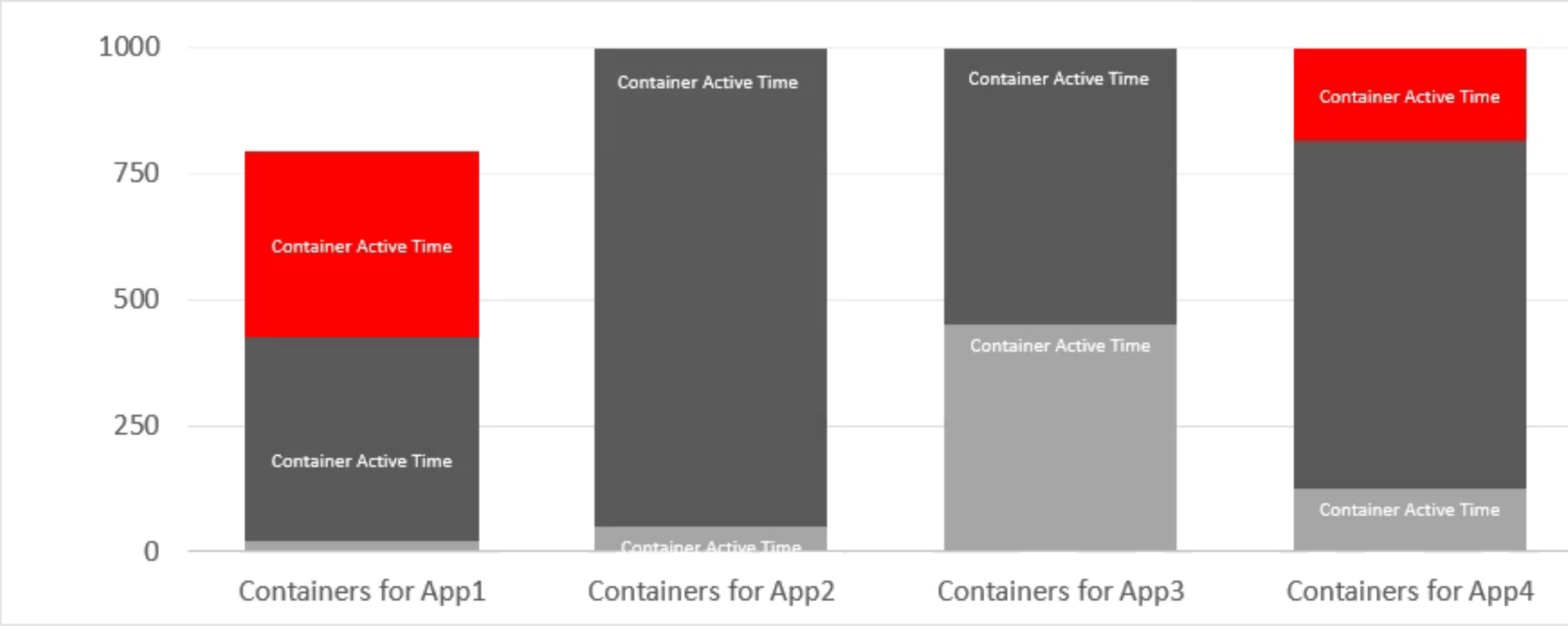
Time Interval 6



Play

App6

Time Interval 6

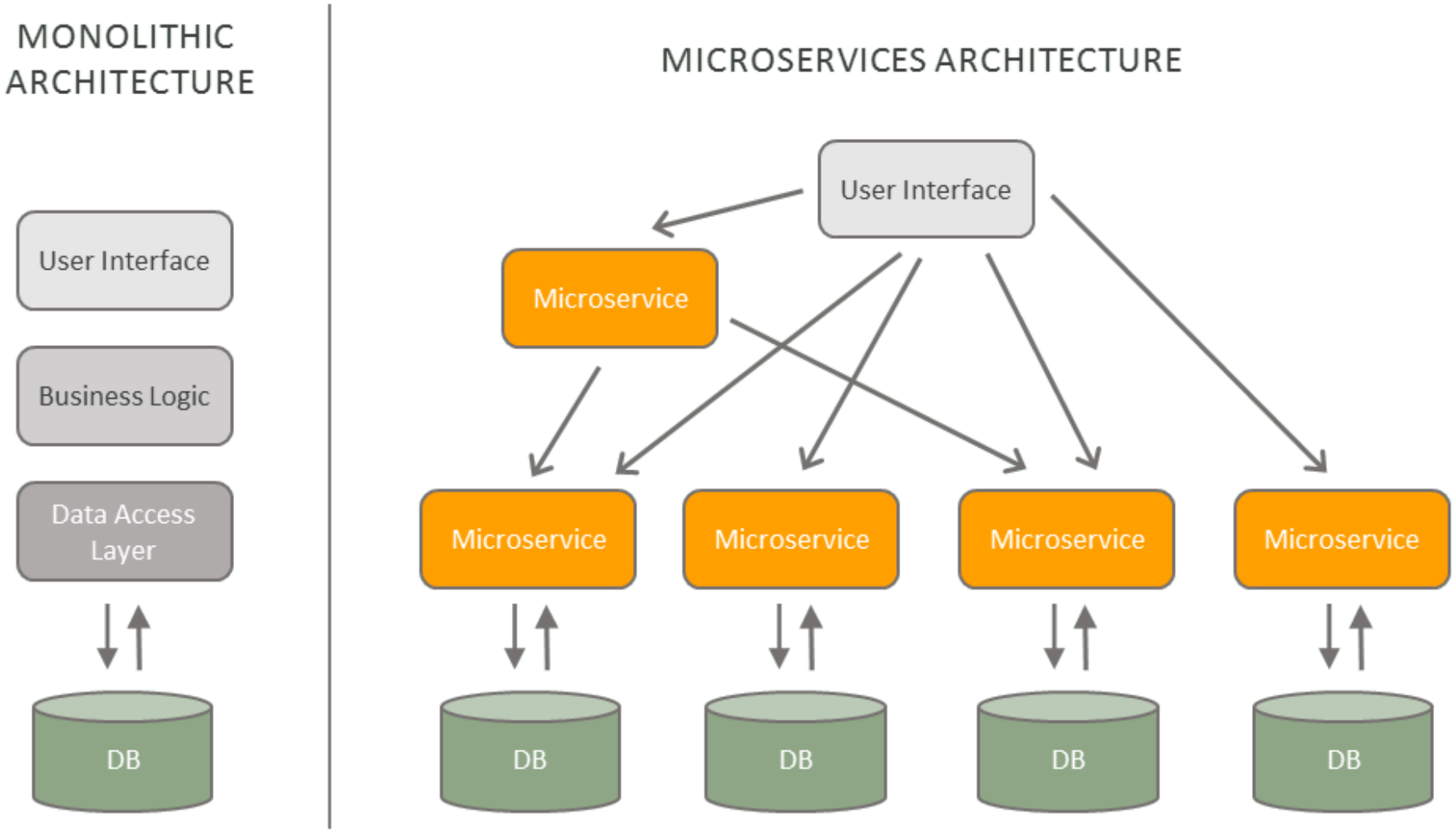


How to Upgrade your Vuln Mgt Program

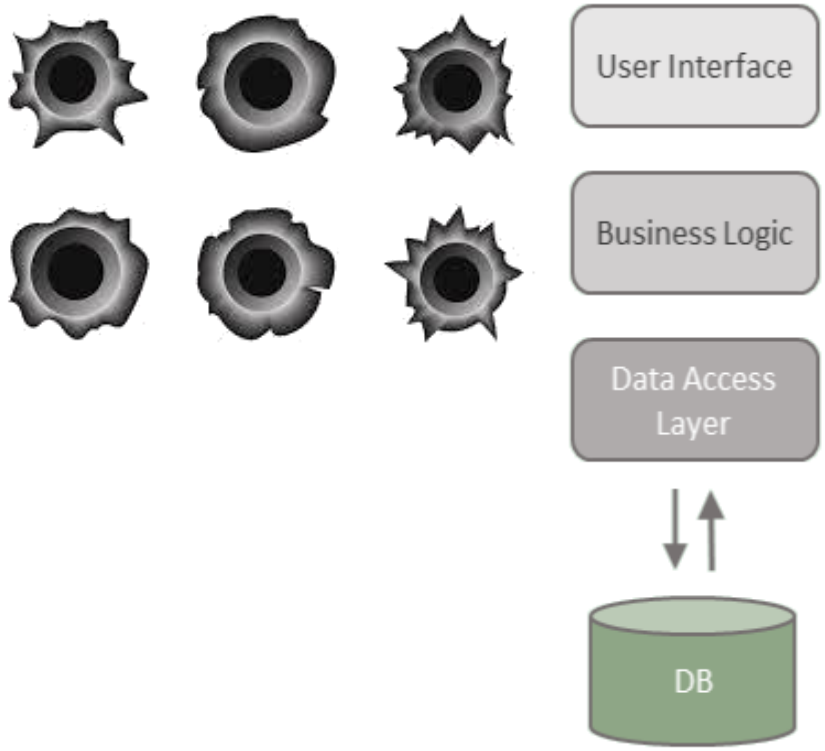
What to expect from a Pen Test	Implications for CaaS
Supply Chain Risk	DevSecOps

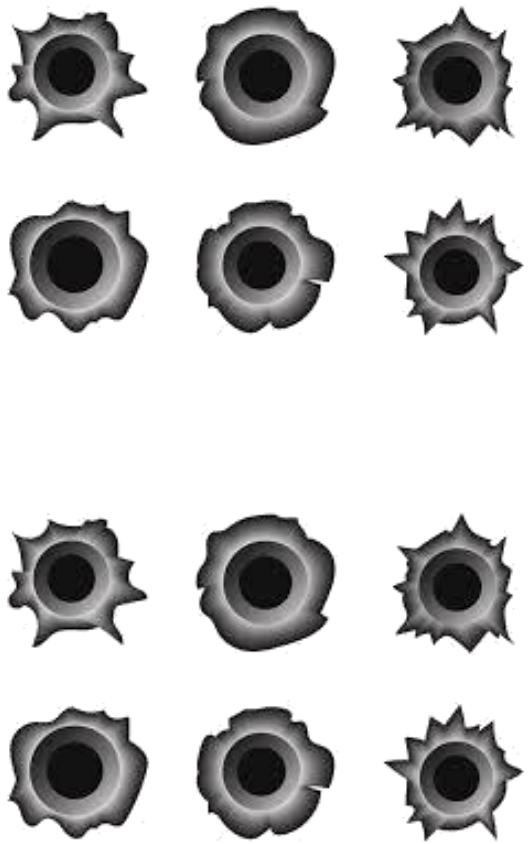
Pen Test – Mechanical Attack vs Knowledge & Finesse

Monolithic vs Microservices Architecture



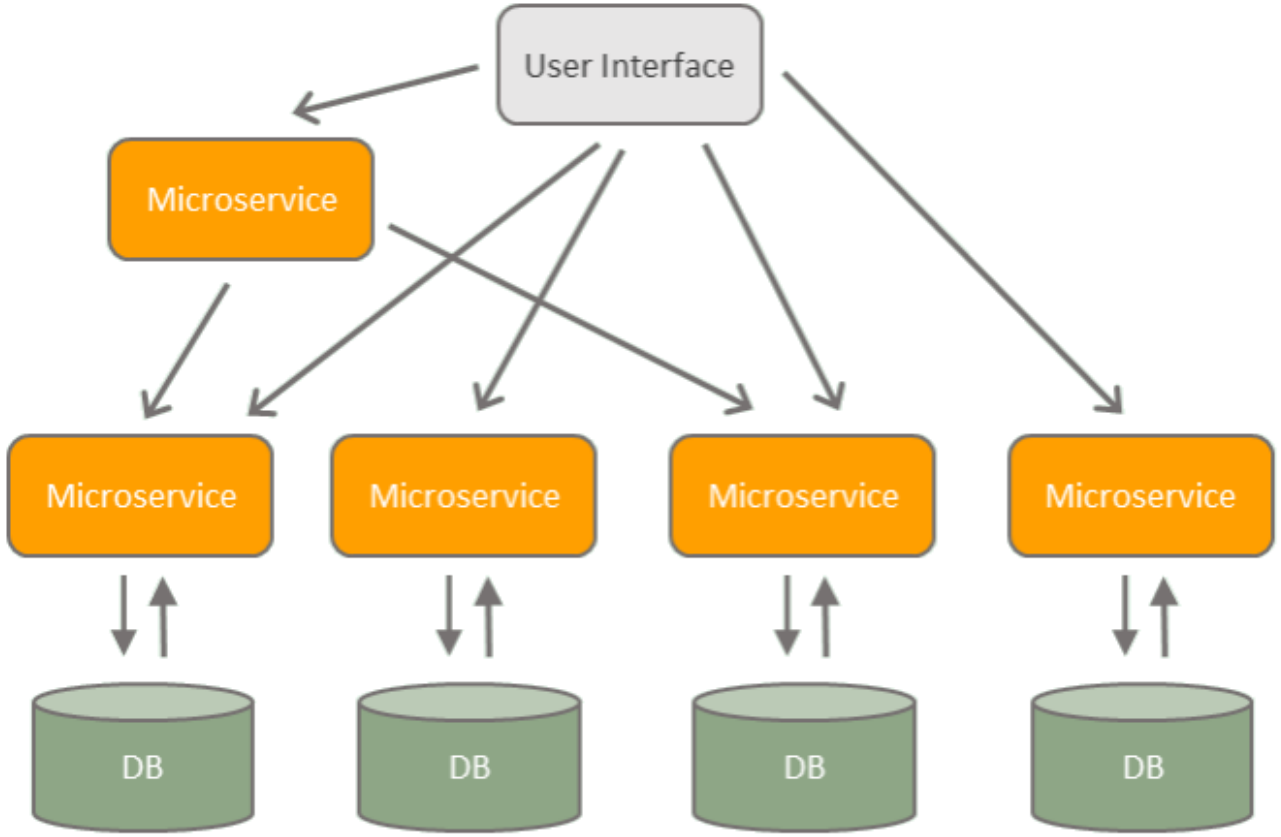
MONOLITHIC ARCHITECTURE

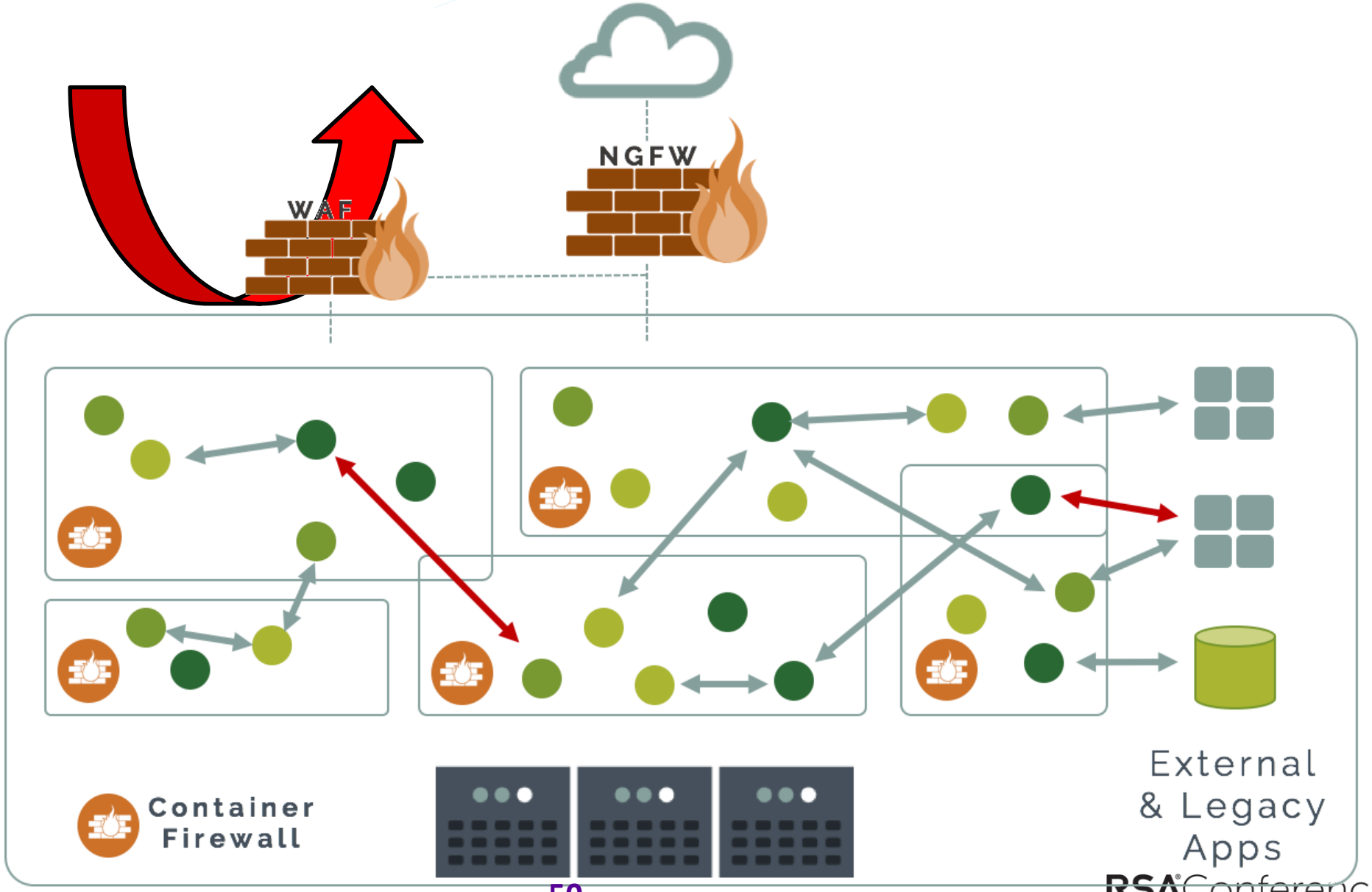




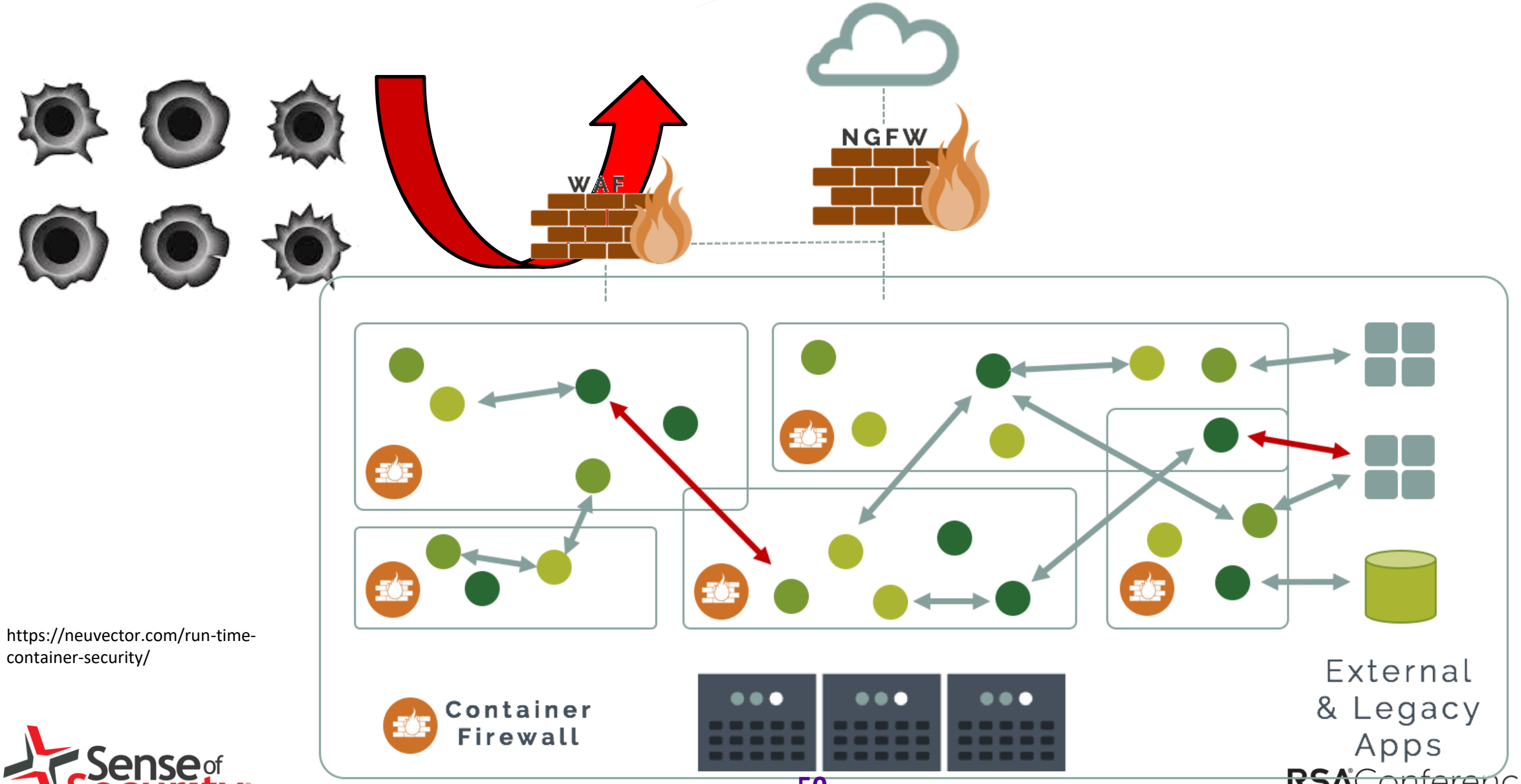
MONOLITHIC ARCHITECTURE

MICROSERVICES ARCHITECTURE

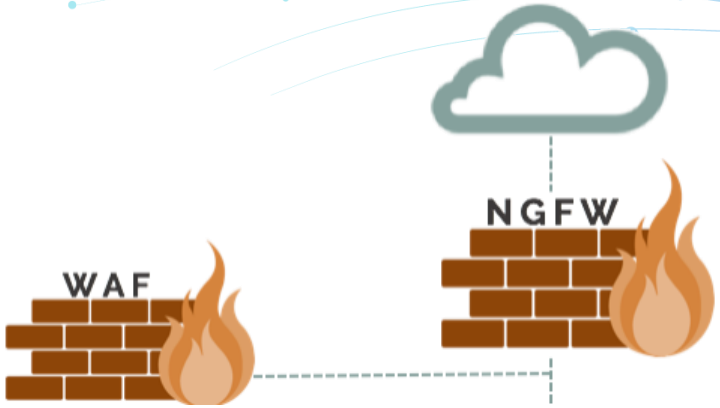


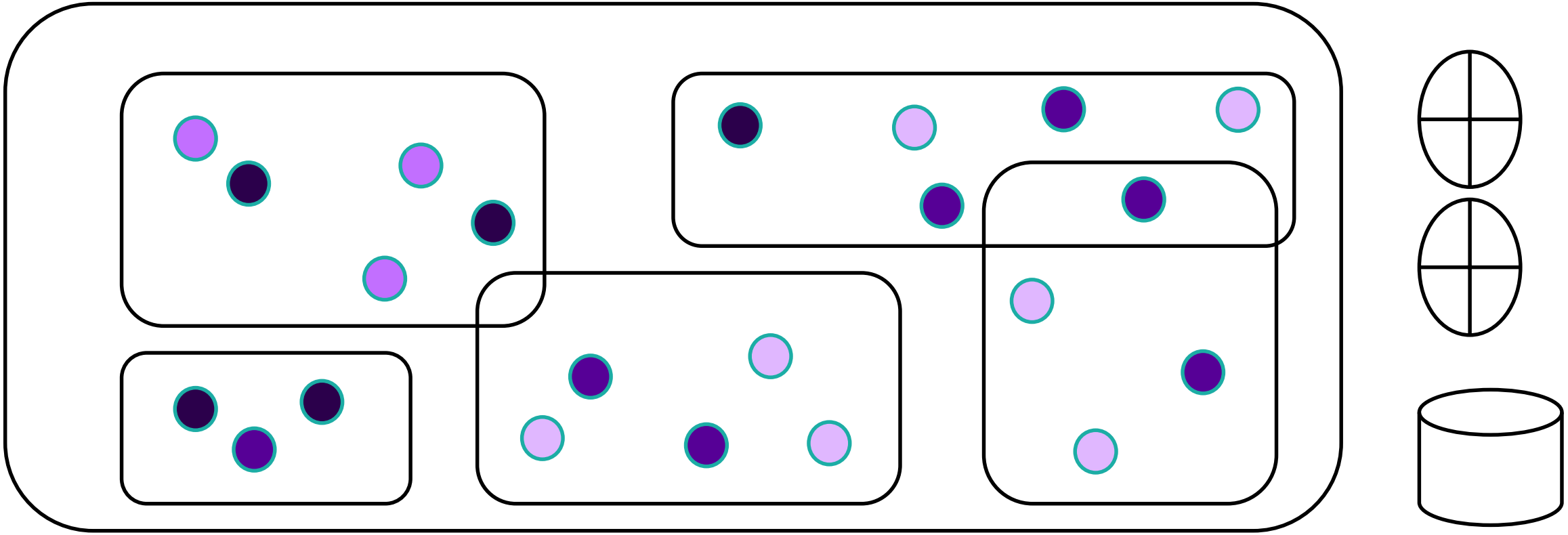


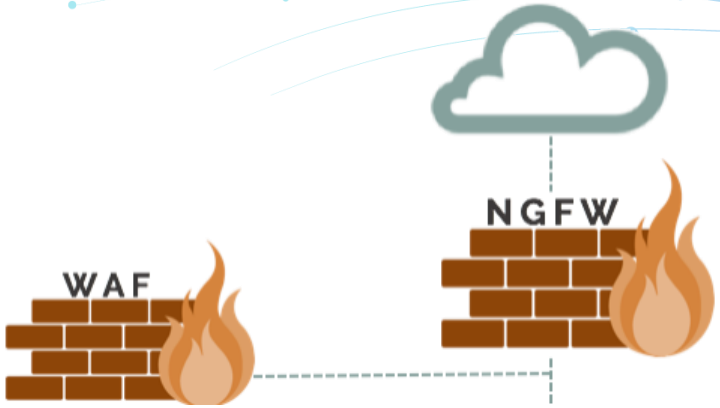
https://neuvector.com/run-time-container-security/

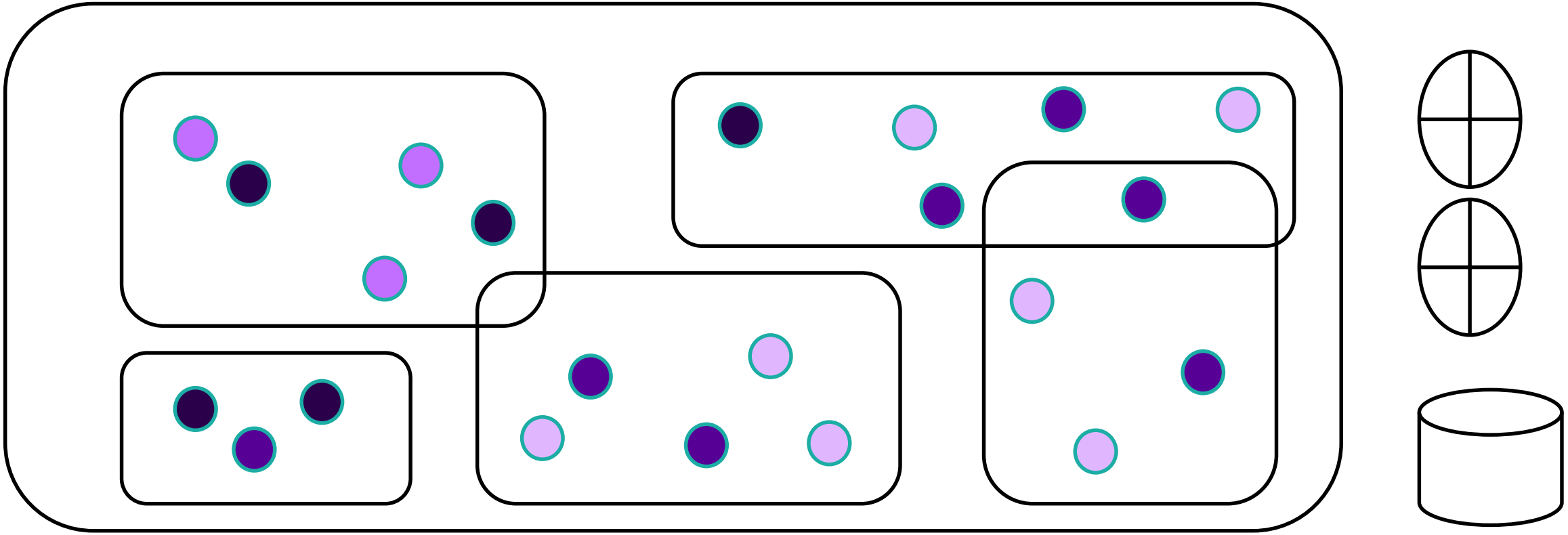


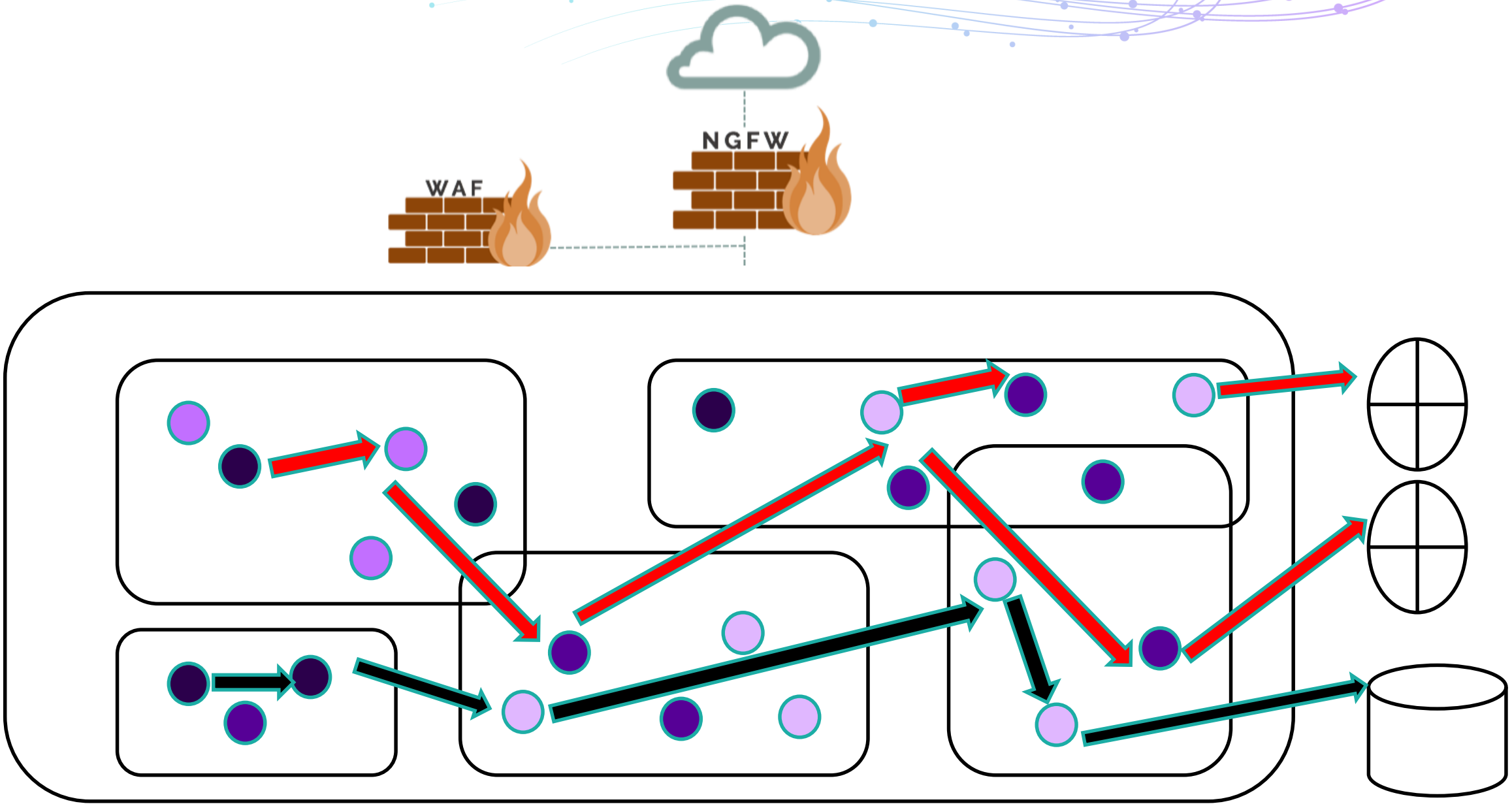
https://neuvector.com/run-time-container-security/

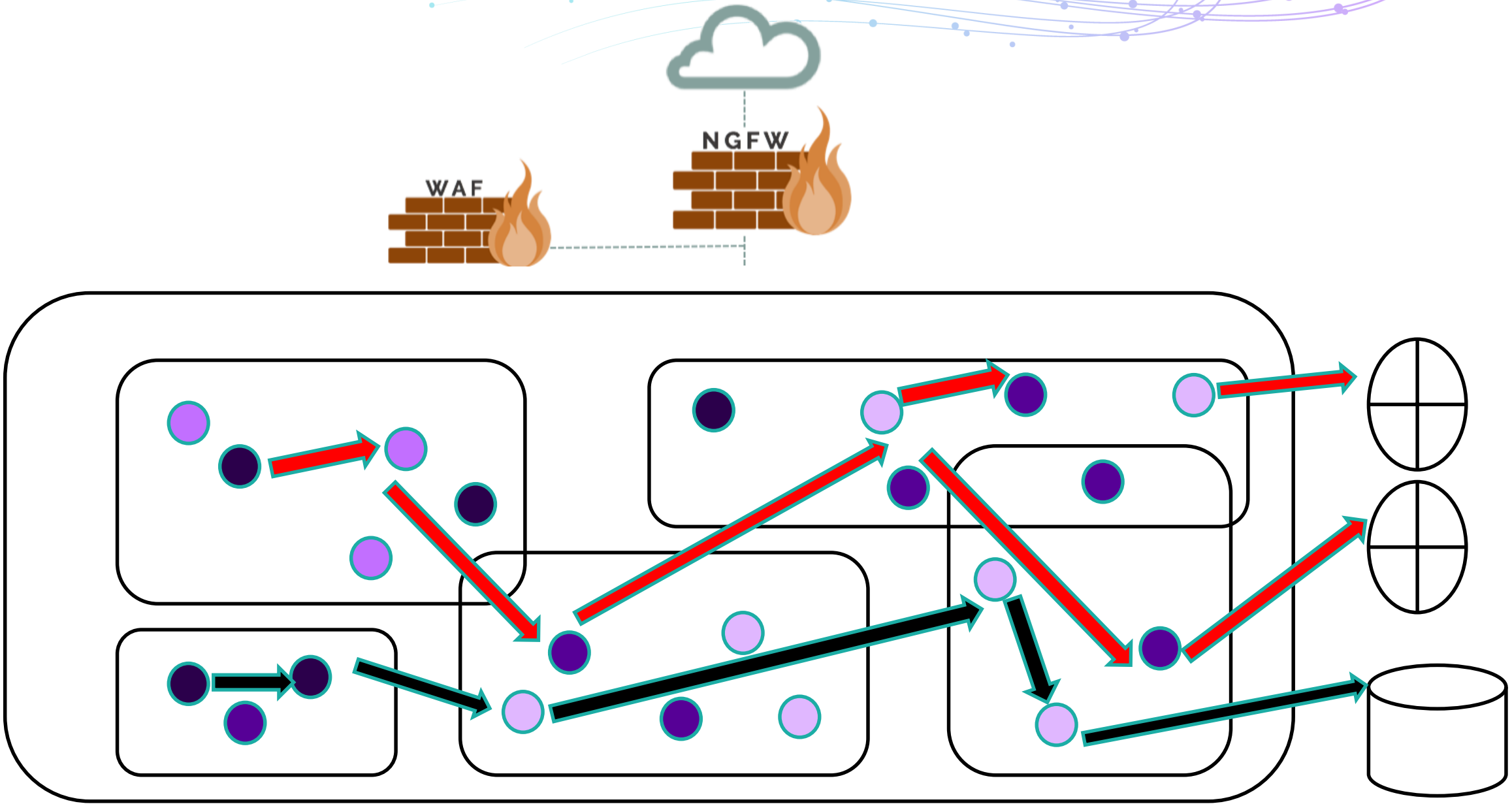

NGFW
WAF

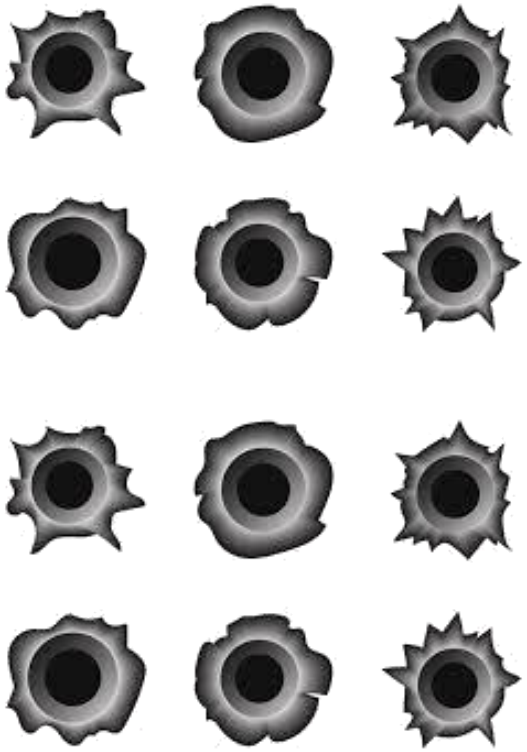



NGFW
WAF



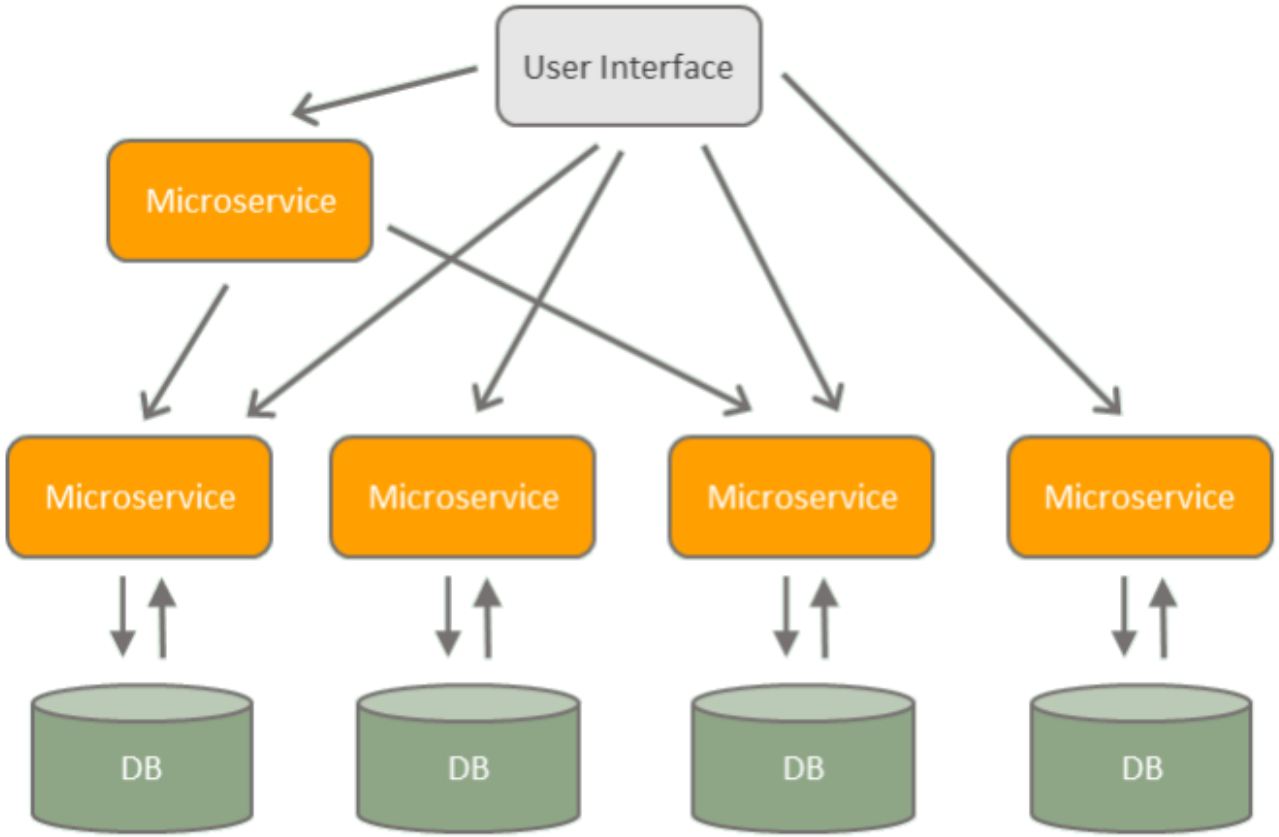

NGFW
WAF

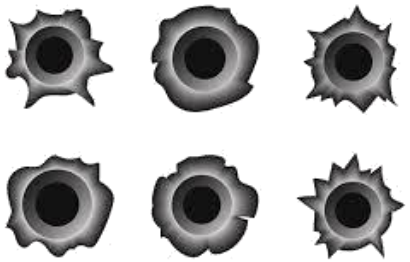

NGFW
WAF

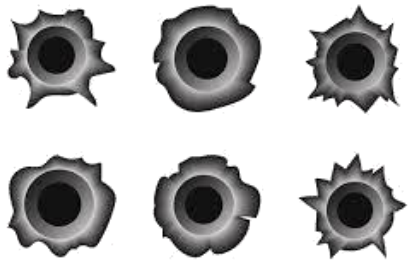


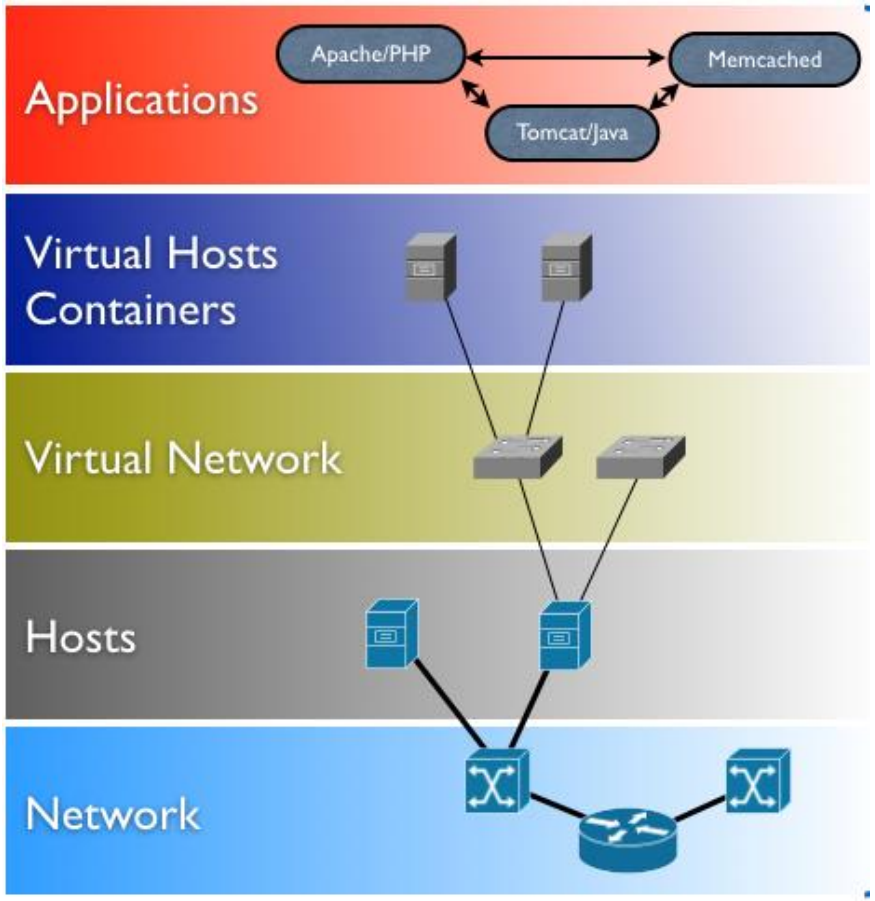
Load Balancing

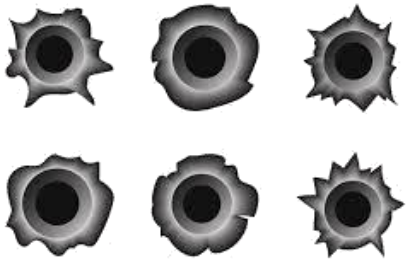
Perimeter Public Functions

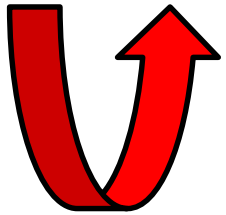


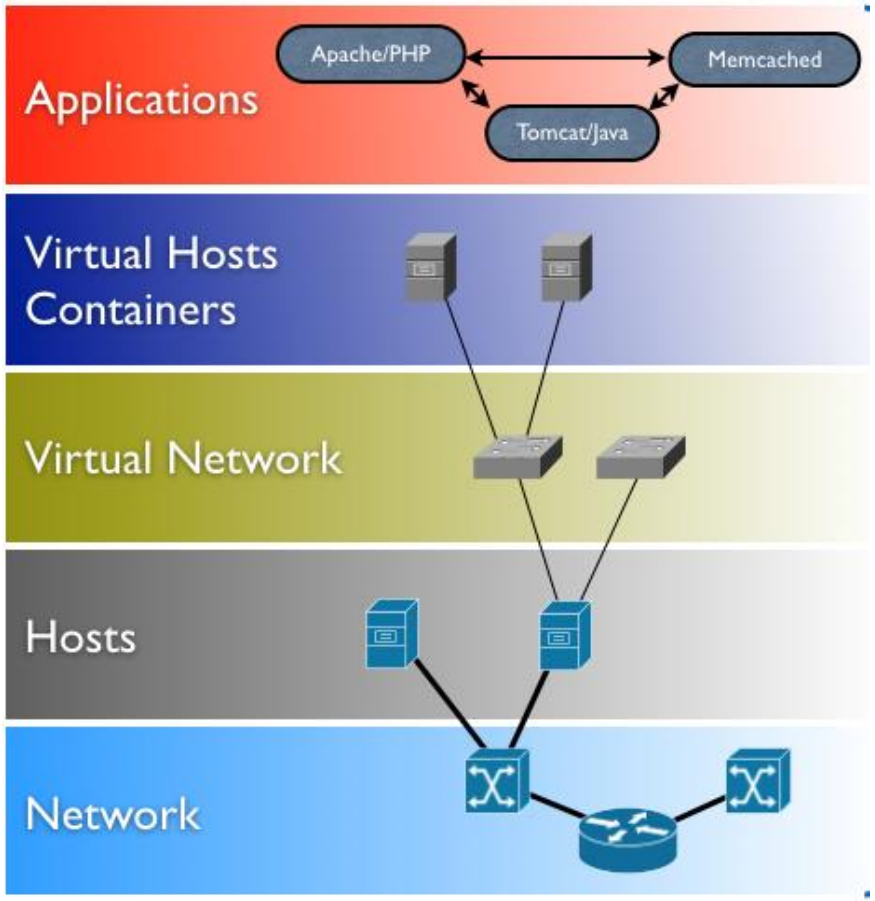


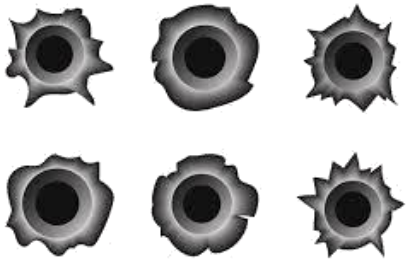


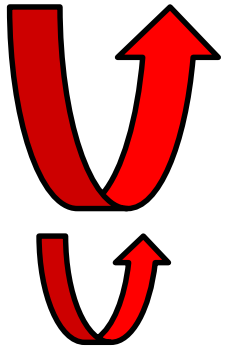

Applications
Apache/PHP
Memcached
Tomcat/Java
Virtual Hosts
Containers
Virtual Network
Hosts
Network

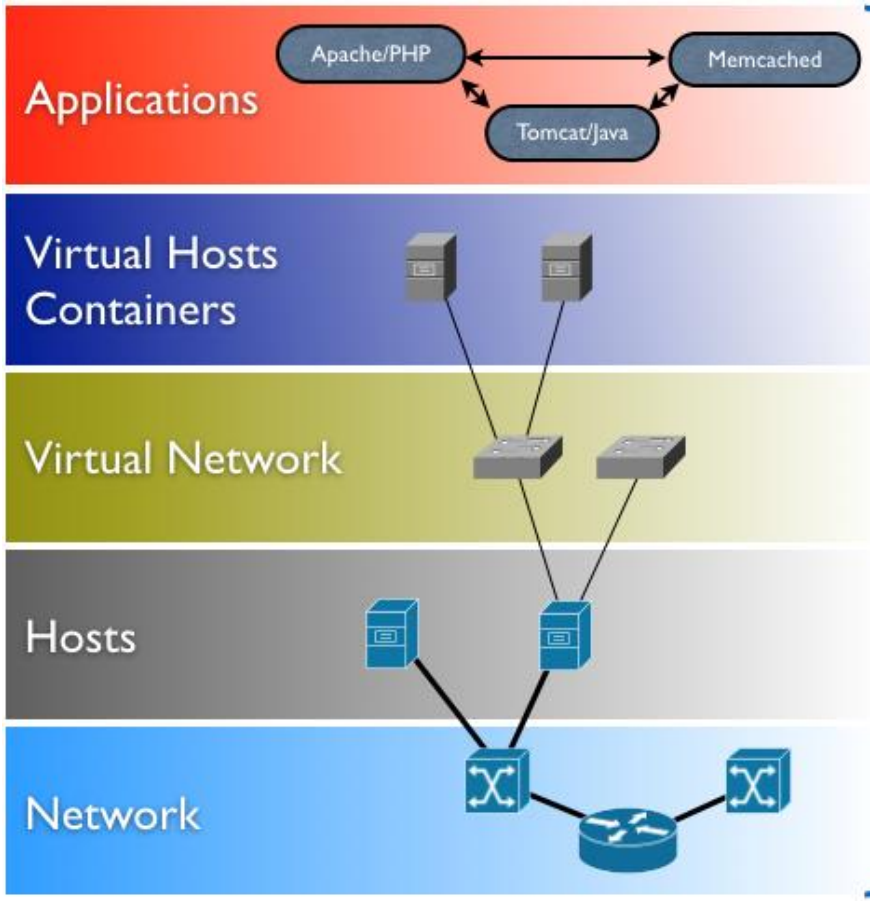





Applications
Apache/PHP
Memcached
Tomcat/Java
Virtual Hosts
Containers
Virtual Network
Hosts
Network

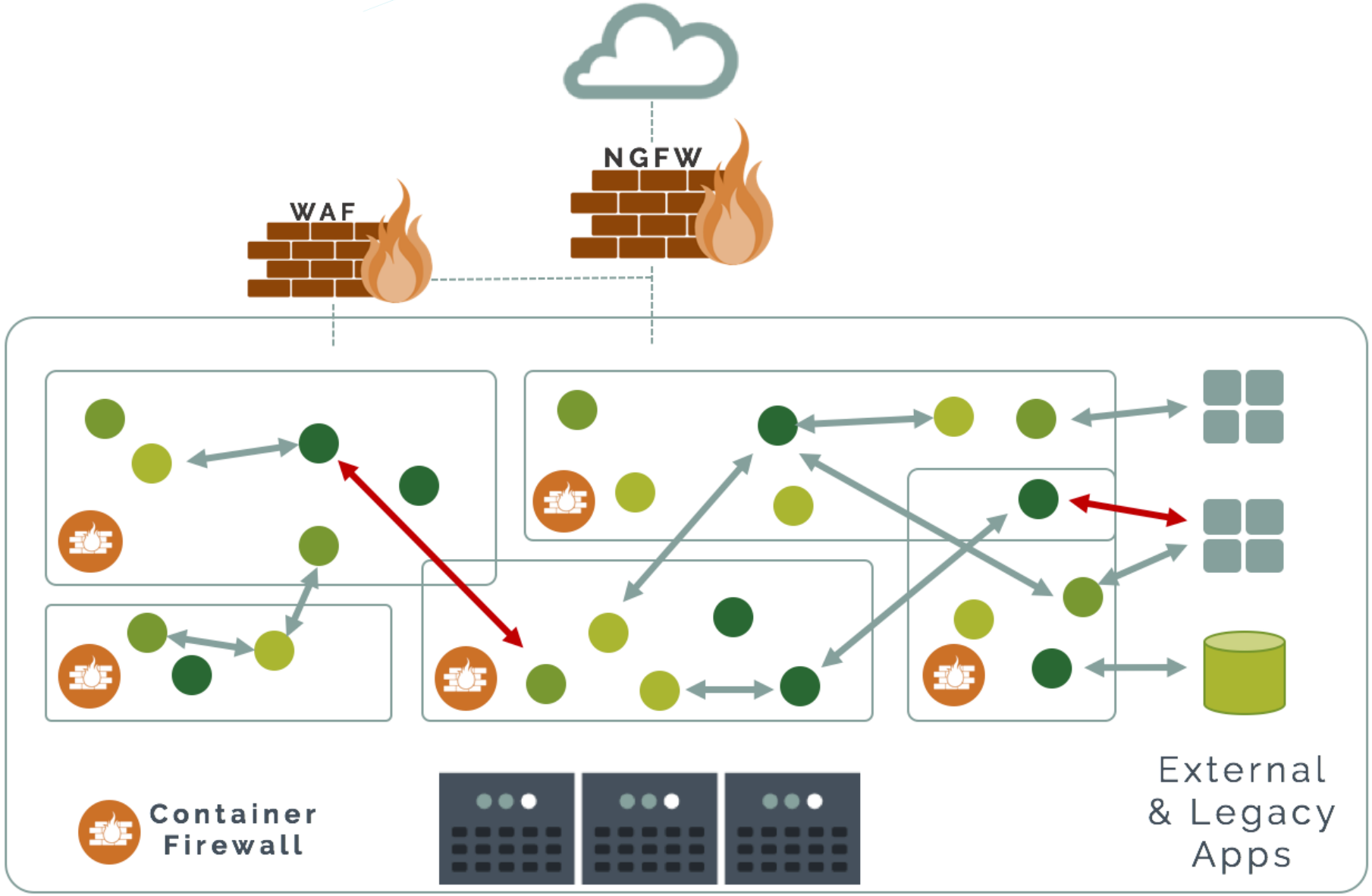





Applications
Apache/PHP
Memcached
Tomcat/Java
Virtual Hosts
Containers
Virtual Network
Hosts
Network

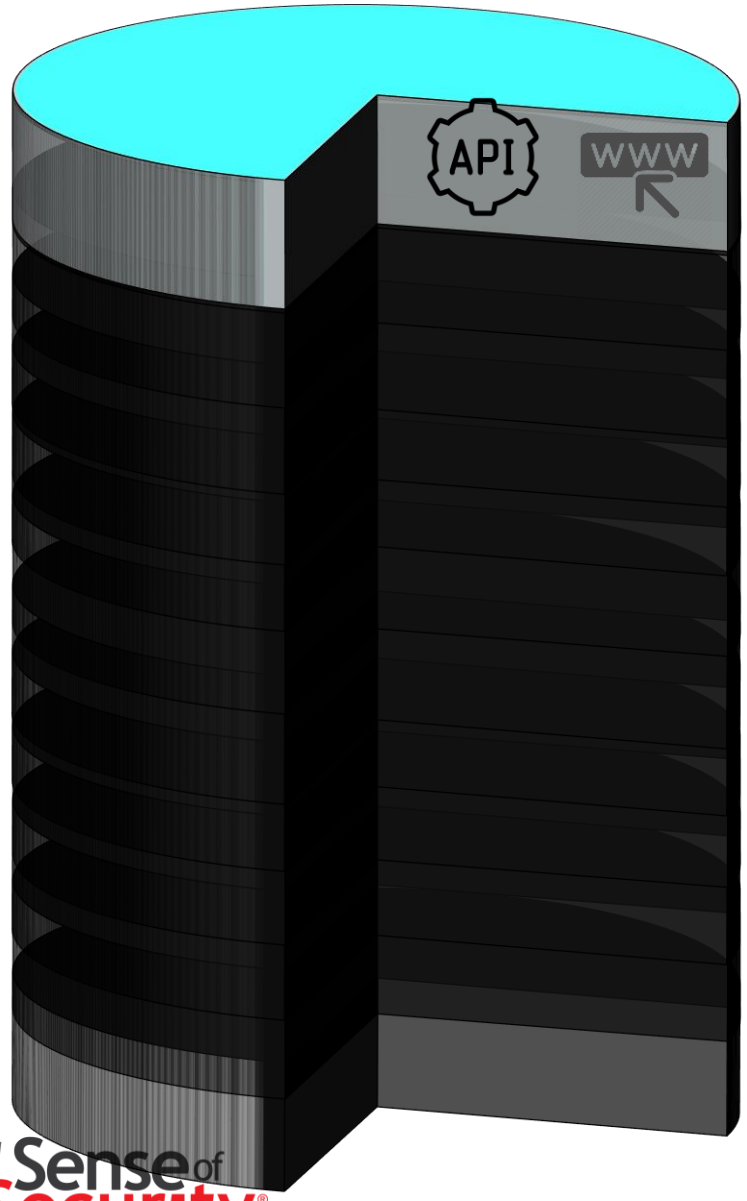
Hack Transformation

Hack Transformation



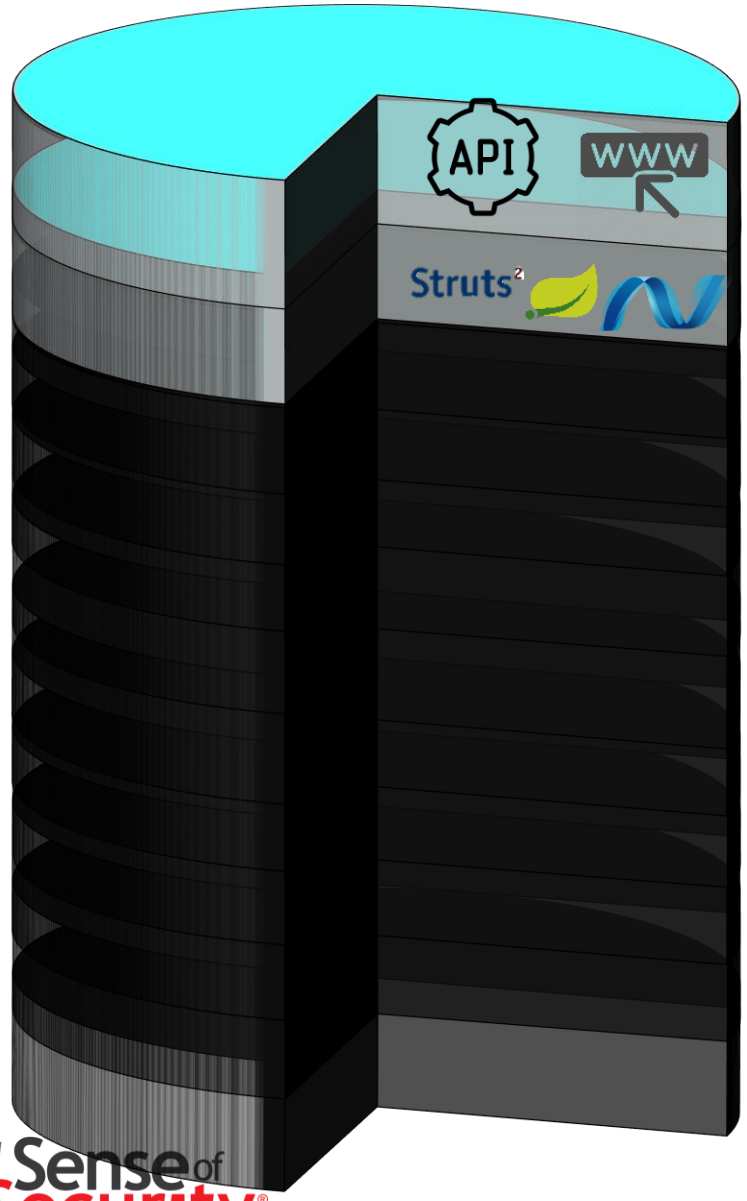
https://neuvector.com/network-security/next-generation-firewall-vs-container-firewall/

Security Testing Needs to Go Down The Stack



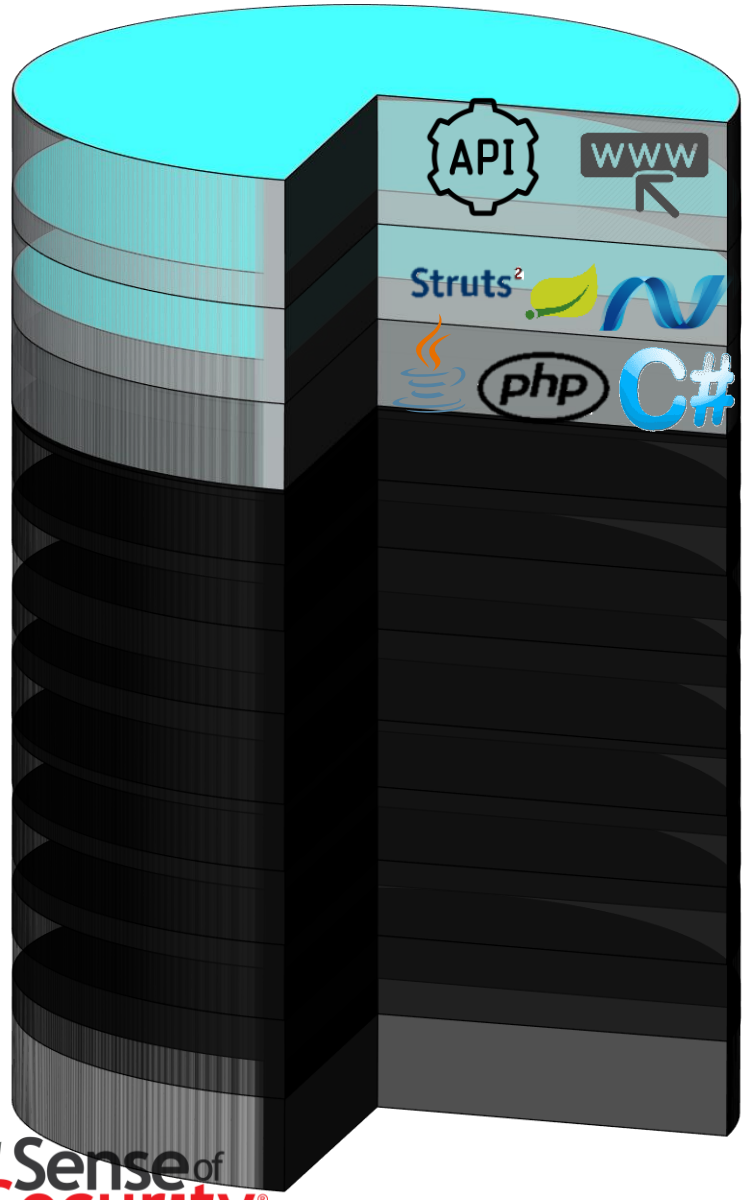
Security Testing Needs to Go Down The Stack

User Interface (WebApps, forms, logons, API's)



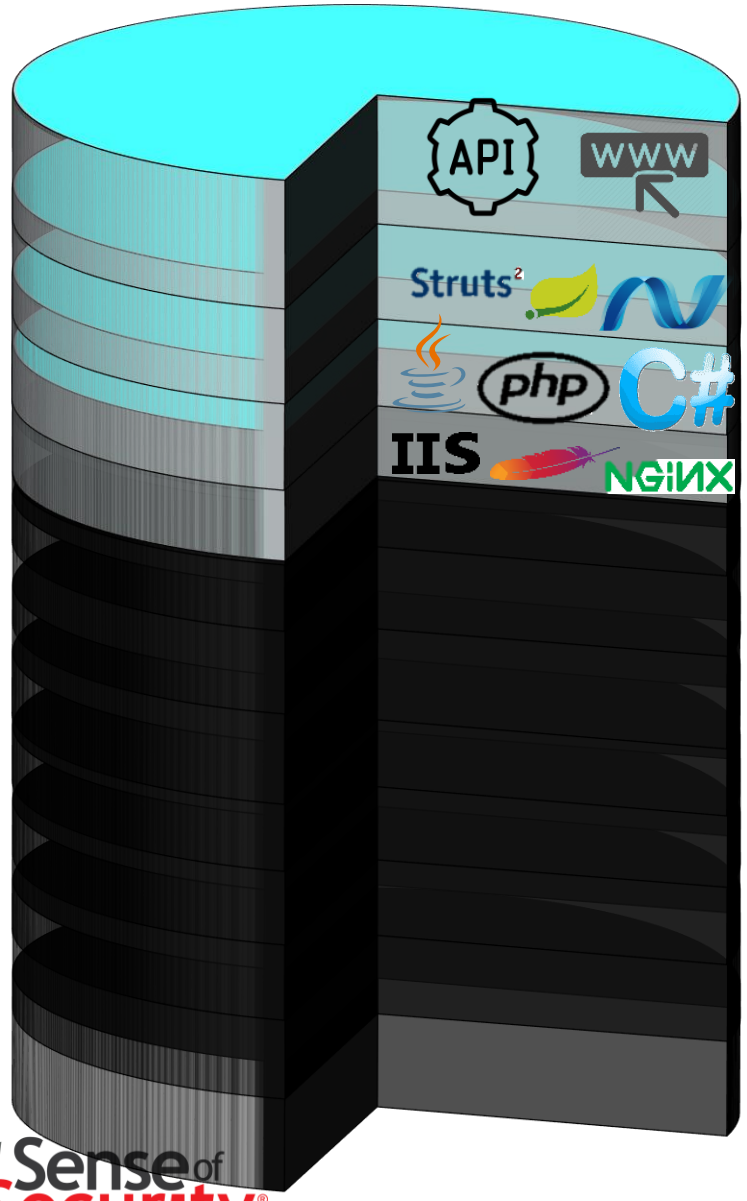
Security Testing Needs to Go Down The Stack

User Interface (WebApps, forms, logons, API's)
Framework (Struts, Spring, .NET)



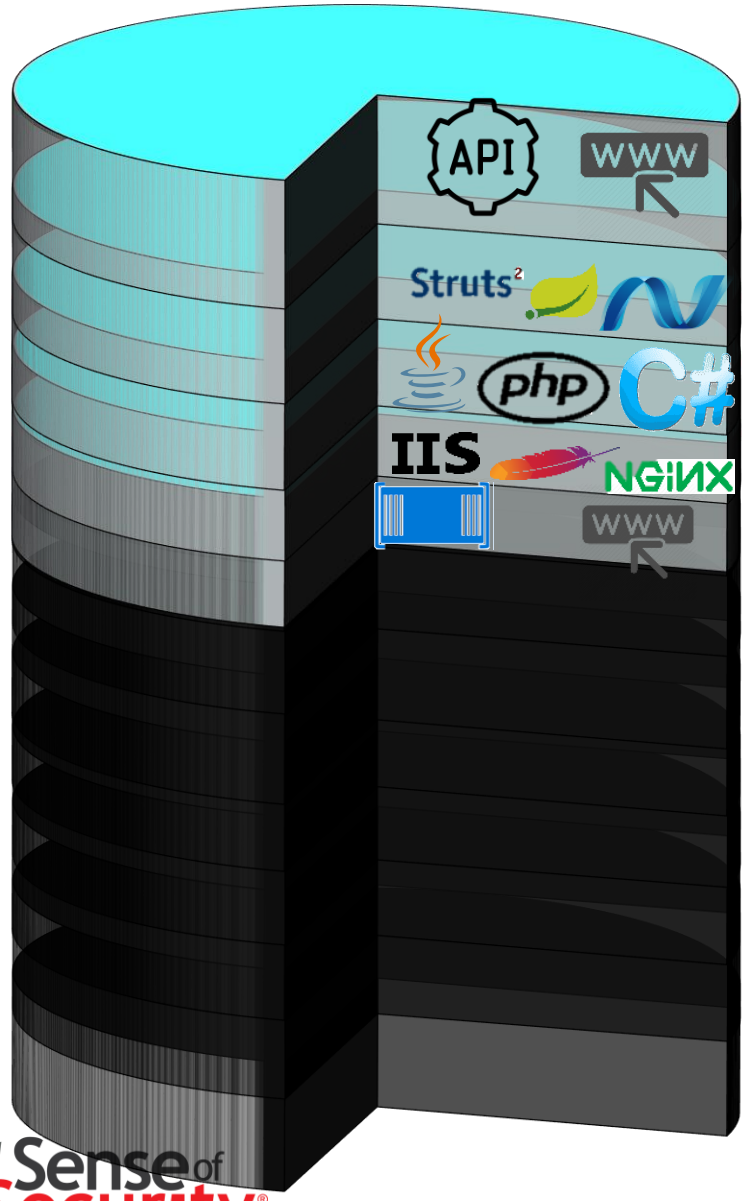
Security Testing Needs to Go Down The Stack

User Interface (WebApps, forms, logons, API's)
Framework (Struts, Spring, .NET)
Language (Java, PHP, .NET)



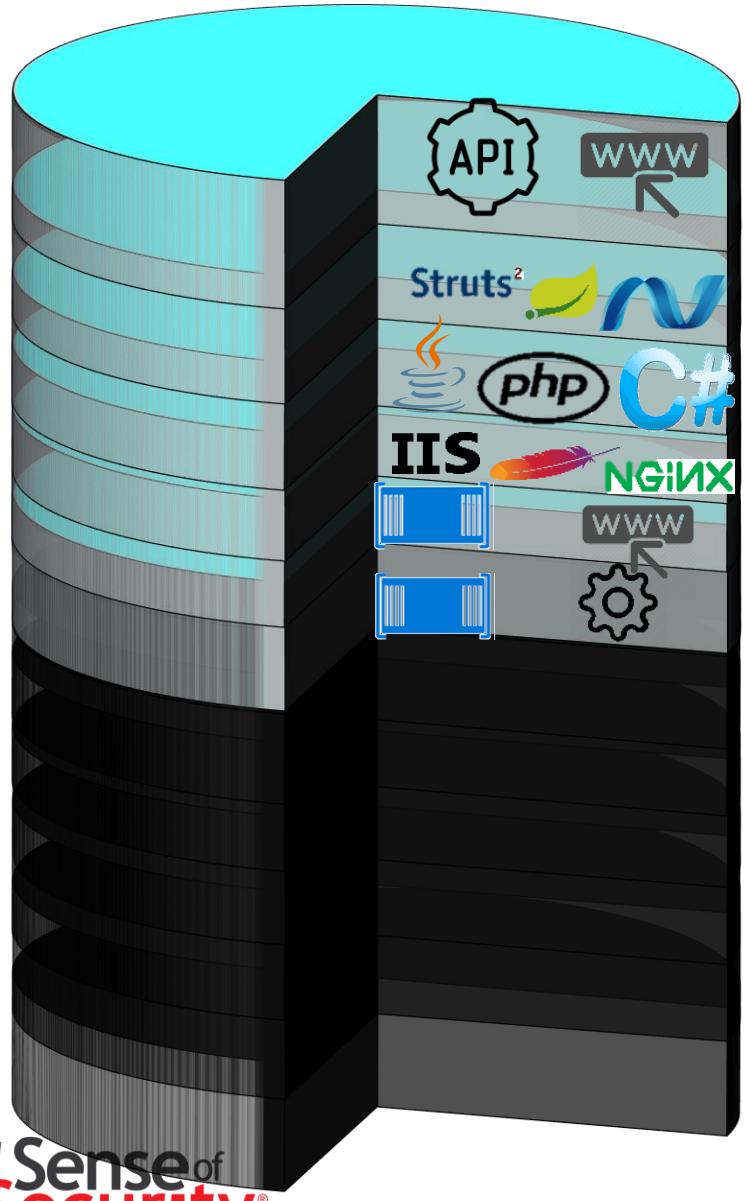
Security Testing Needs to Go Down The Stack

User Interface (WebApps, forms, logons, API's)
Framework (Struts, Spring, .NET)
Language (Java, PHP, .NET)
AppServer (IIS, Apache, Nginx)



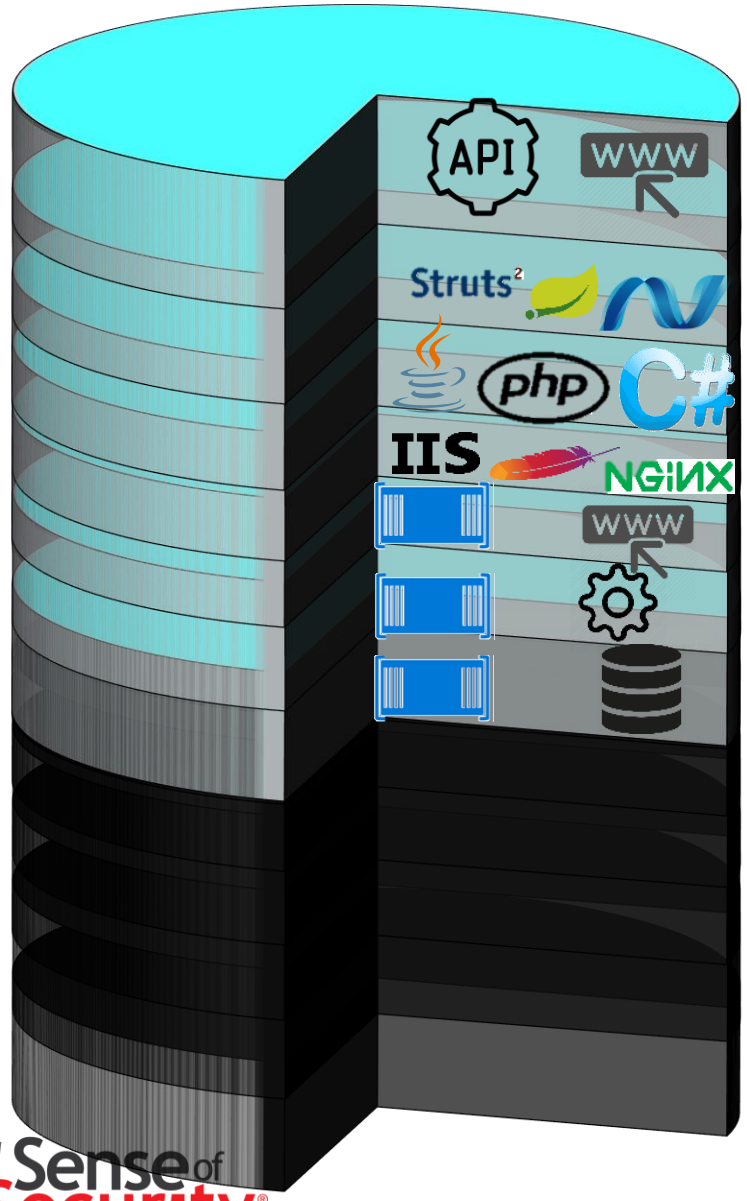
Security Testing Needs to Go Down The Stack

User Interface (WebApps, forms, logons, API's)
Framework (Struts, Spring, .NET)
Language (Java, PHP, .NET)
AppServer (IIS, Apache, Nginx)
Process UI (Container, presentation layer)



Security Testing Needs to Go Down The Stack

User Interface (WebApps, forms, logons, API's)
Framework (Struts, Spring, .NET)
Language (Java, PHP, .NET)
AppServer (IIS, Apache, Nginx)
Process UI (Container, presentation layer)
Process App (Container, application processing)



Security Testing Needs to Go Down The Stack

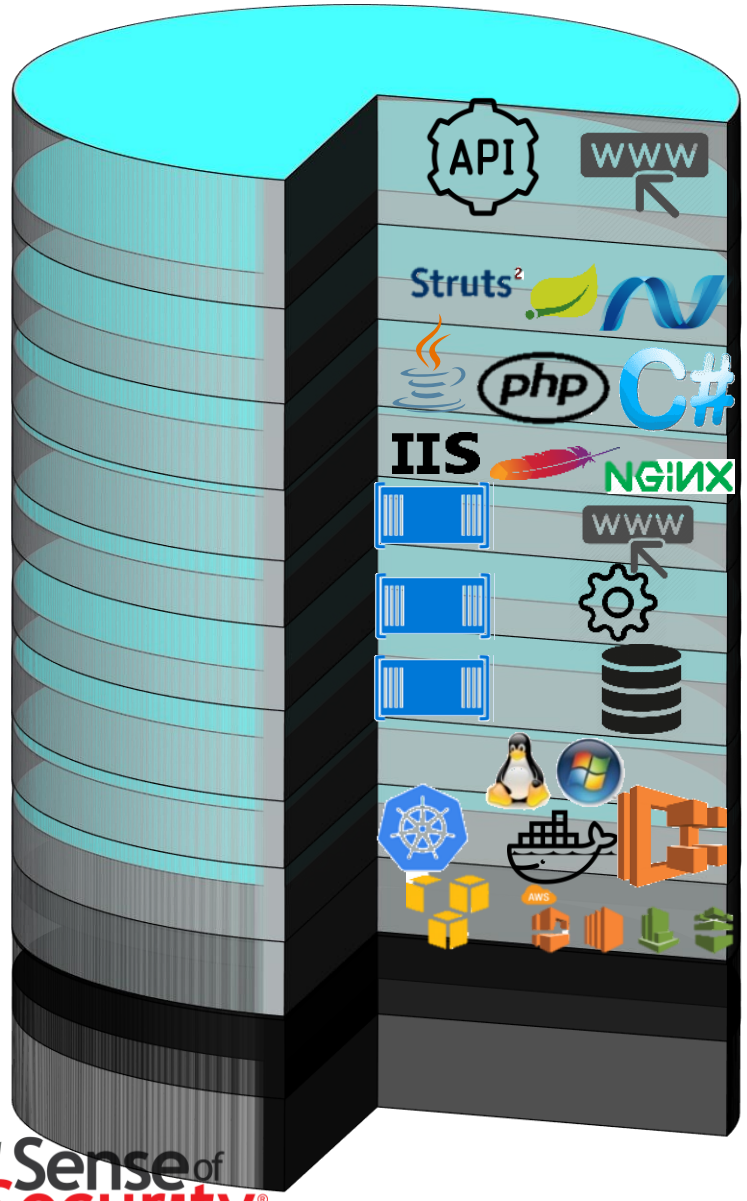
User Interface (WebApps, forms, logons, API's)
Framework (Struts, Spring, .NET)
Language (Java, PHP, .NET)
AppServer (IIS, Apache, Nginx)
Process UI (Container, presentation layer)
Process App (Container, application processing)
Process BackEnd (Container, database)

Security Testing Needs to Go Down The Stack

User Interface (WebApps, forms, logons, API's)
Framework (Struts, Spring, .NET)
Language (Java, PHP, .NET)
AppServer (IIS, Apache, Nginx)
Process UI (Container, presentation layer)
Process App (Container, application processing)
Process BackEnd (Container, database)
Operating System (Linux, Windows)

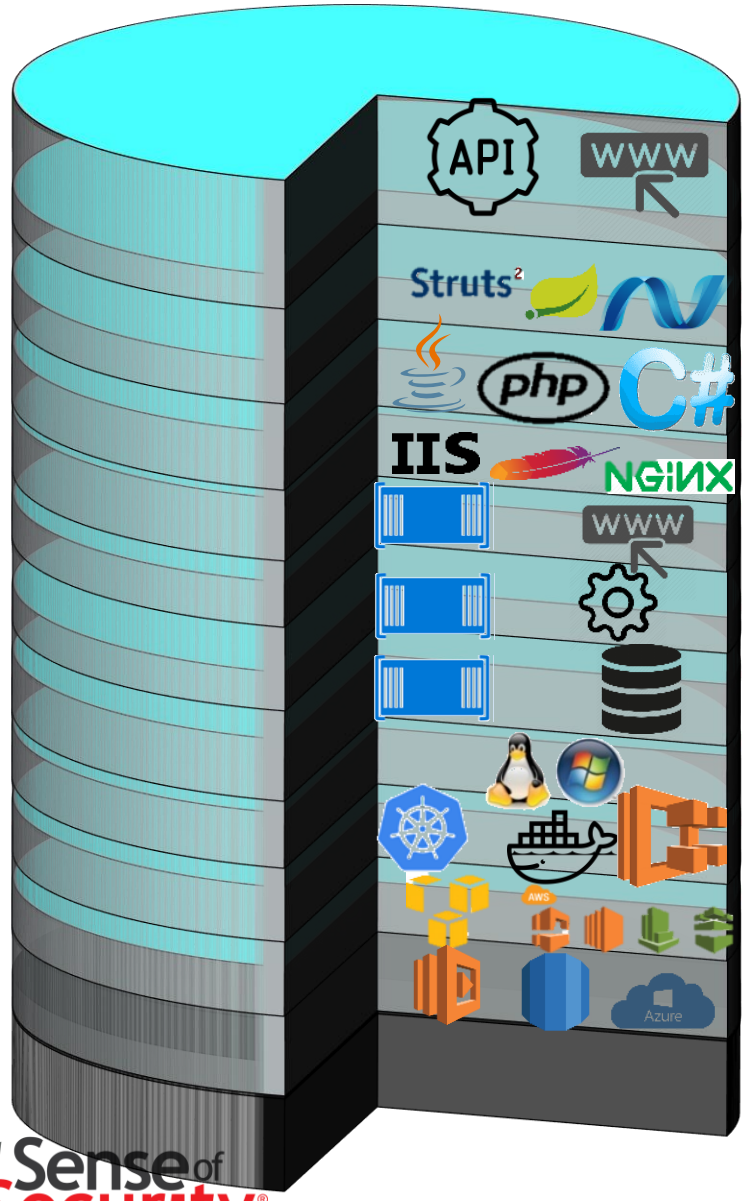
Security Testing Needs to Go Down The Stack

User Interface (WebApps, forms, logons, API's)
Framework (Struts, Spring, .NET)
Language (Java, PHP, .NET)
AppServer (IIS, Apache, Nginx)
Process UI (Container, presentation layer)
Process App (Container, application processing)
Process BackEnd (Container, database)
Operating System (Linux, Windows)
Clustering/Orchestration (CaaS, Swarm, Kubernetes)



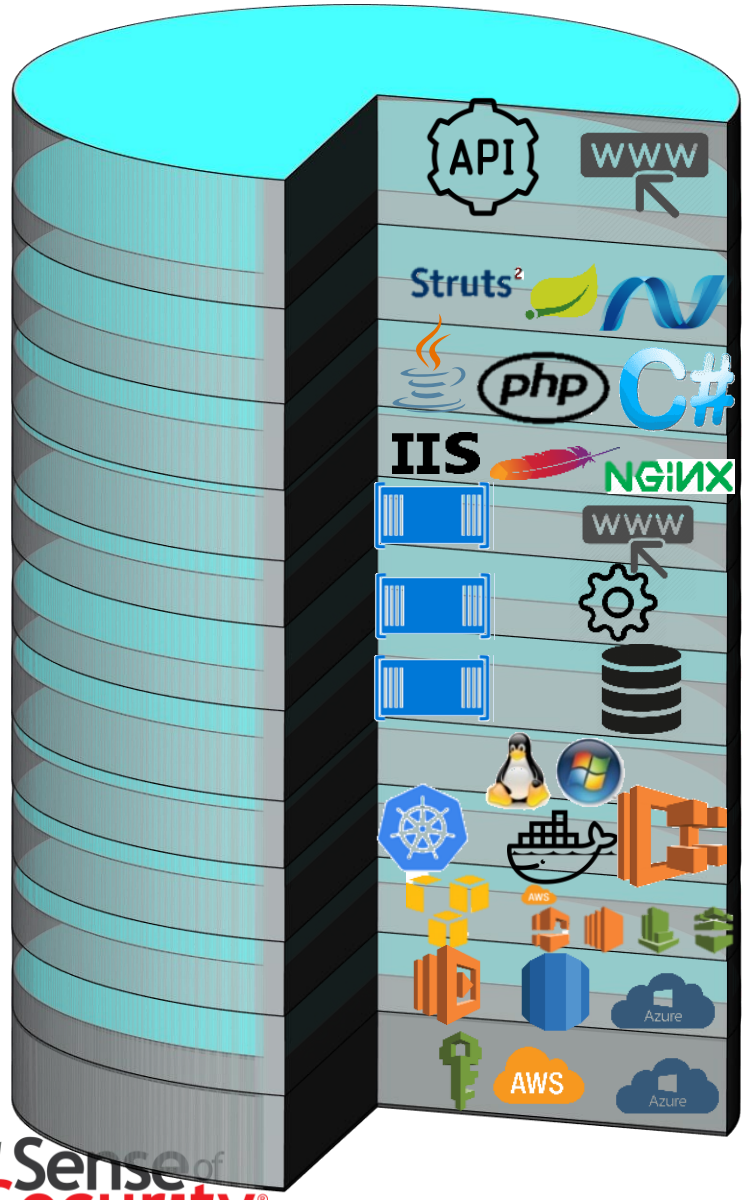
Security Testing Needs to Go Down The Stack

User Interface (WebApps, forms, logons, API's)
Framework (Struts, Spring, .NET)
Language (Java, PHP, .NET)
AppServer (IIS, Apache, Nginx)
Process UI (Container, presentation layer)
Process App (Container, application processing)
Process BackEnd (Container, database)
Operating System (Linux, Windows)
Clustering/Orchestration (CaaS, Swarm, Kubernetes)
Networking (SDN, SecGroups)



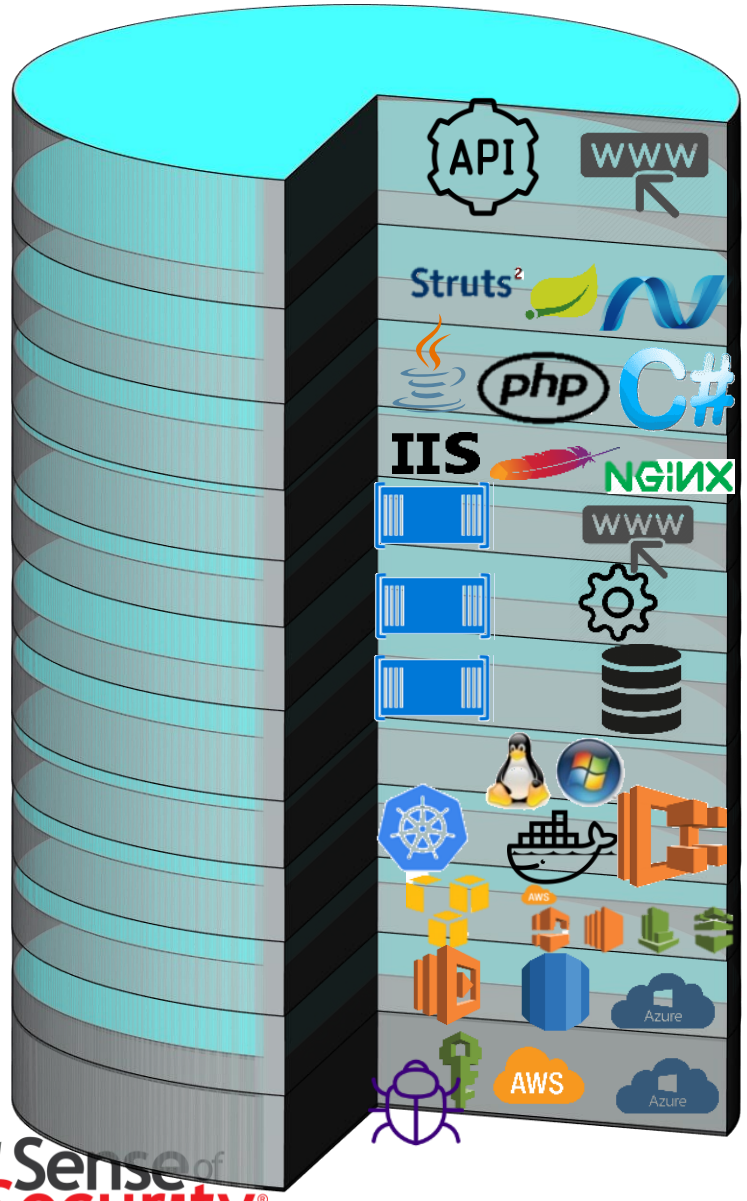
Security Testing Needs to Go Down The Stack

User Interface (WebApps, forms, logons, API's)
Framework (Struts, Spring, .NET)
Language (Java, PHP, .NET)
AppServer (IIS, Apache, Nginx)
Process UI (Container, presentation layer)
Process App (Container, application processing)
Process BackEnd (Container, database)
Operating System (Linux, Windows)
Clustering/Orchestration (CaaS, Swarm, Kubernetes)
Networking (SDN, SecGroups)
Cloud Platform



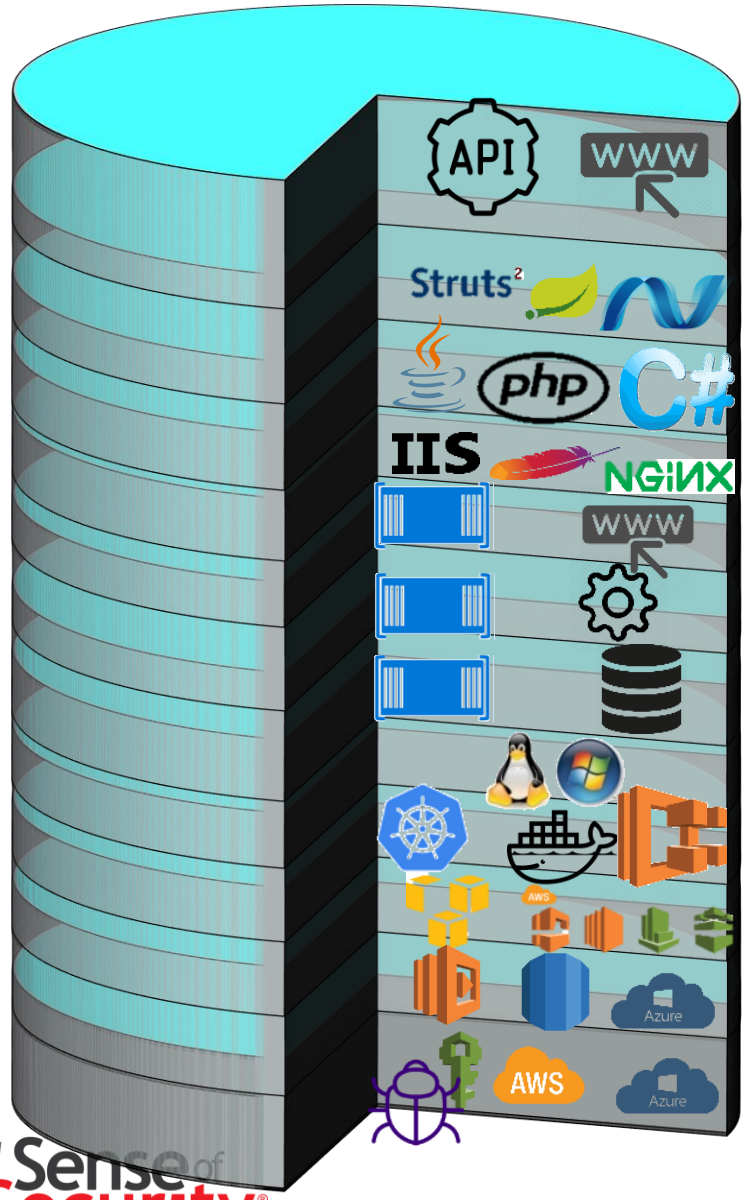
Security Testing Needs to Go Down The Stack

User Interface (WebApps, forms, logons, API's)
Framework (Struts, Spring, .NET)
Language (Java, PHP, .NET)
AppServer (IIS, Apache, Nginx)
Process UI (Container, presentation layer)
Process App (Container, application processing)
Process BackEnd (Container, database)
Operating System (Linux, Windows)
Clustering/Orchestration (CaaS, Swarm, Kubernetes)
Networking (SDN, SecGroups)
Cloud Platform
Core Infrastructure



Security Testing Needs to Go Down The Stack

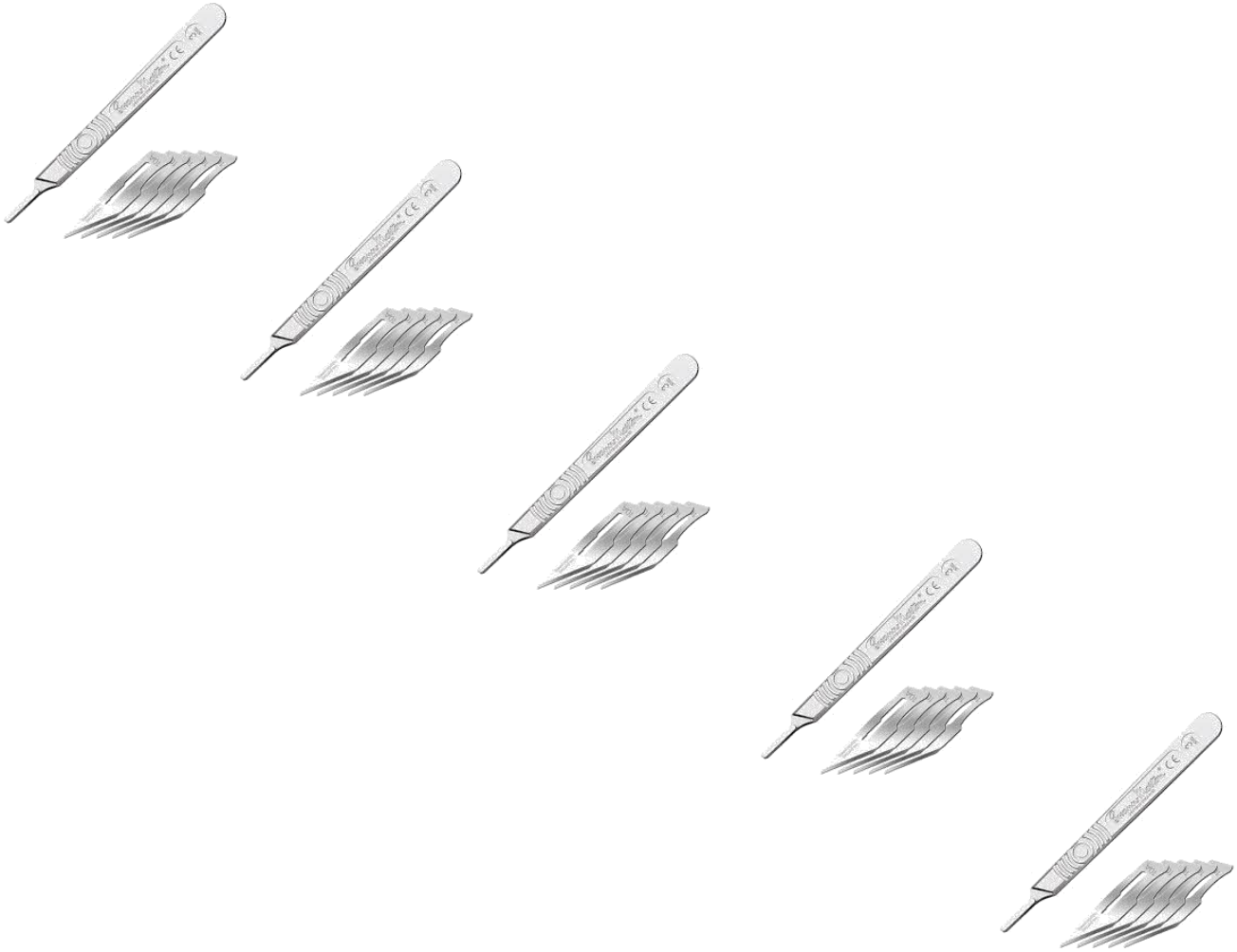
User Interface (WebApps, forms, logons, API's)
Framework (Struts, Spring, .NET)
Language (Java, PHP, .NET)
AppServer (IIS, Apache, Nginx)
Process UI (Container, presentation layer)
Process App (Container, application processing)
Process BackEnd (Container, database)
Operating System (Linux, Windows)
Clustering/Orchestration (CaaS, Swarm, Kubernetes)
Networking (SDN, SecGroups)
Cloud Platform
Core Infrastructure



Security Testing Needs to Go Down The Stack

User Interface (WebApps, forms, logons, API's)
Framework (Struts, Spring, .NET)
Language (Java, PHP, .NET)
AppServer (IIS, Apache, Nginx)
Process UI (Container, presentation layer)
Process App (Container, application processing)
Process BackEnd (Container, database)
Operating System (Linux, Windows)
Clustering/Orchestration (CaaS, Swarm, Kubernetes)
Networking (SDN, SecGroups)
Cloud Platform
Core Infrastructure

Finesse

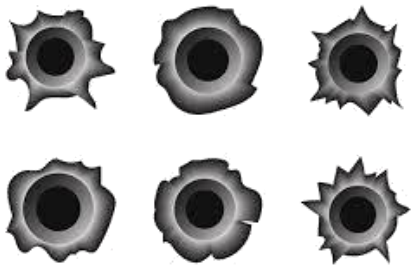



Right Decision
Wrong Decision

Lower Cost
Predictable
Even if a Web App/Service Pen Test not suitable for current technologies
Doesn't really assess the threats
More North-South than East-West
Check Box

More considered
Requires expert capability, R&D
Requires understanding of the full stack incl implications of -aaS
Requires persistence in an ephemeral setting
Yes – it will cost more
Assurance, Validation & Compliance

There are Pen Tests & There are Pen Tests!



Lower Cost	More considered
Predictable	Requires expert capability, R&D
Even if a Web App/Service Pen Test not suitable for current technologies	Requires understanding of the full stack incl implications of -aaS
Doesn't really assess the threats	Requires persistence in an ephemeral setting
More North-South than East-West	Yes – it will cost more
Check Box	Assurance, Validation & Compliance

Blue Team: Key Steps to App Container Security

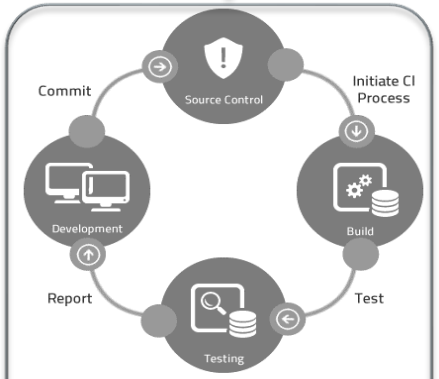
1	**End-to-End Vulnerability Management**
2	Container Attack Surface Reduction
3	User Access Control
4	Hardening the Host OS & the Container
5	SDLC Automation (DevOps)

Solutioning

1	End-to-End Vulnerability Management

Automated Vuln Mgt

SHIFT LEFT

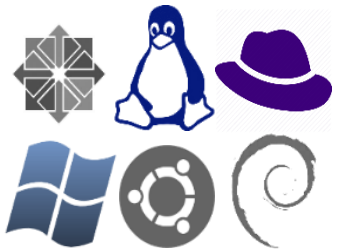


Build

- API's & Plug-ins
- Third Party Components
- Vuln Mgt Automation

Registry

- Automated Scan of Pub/Priv Registry



Host

- Compliance Scanning
  - OS
  - CaaS

Runtime

- Audit logging
- Event logging

Image adapted from Qualys materials

Hosts

Filter hosts by status: All

Filter hosts by compliance: All

Filter hosts by CVE: All

Search by: Name

Name

k8

Vulnerability	Severity	Resource Type	Resource
CVE-2019-7308	Medium	DEB Package	linux-azure
CVE-2019-7308	Medium	DEB Package	linux-azure
CVE-2019-6133	Medium	DEB Package	linux-azure
CVE-2019-6133	Medium	DEB Package	policykit-1
CVE-2019-6133	Medium	DEB Package	policykit-1
CVE-2019-6133	Medium	DEB Package	linux-azure
CVE-2019-5736	High	DEB Package	runc

Resource: runc

Description: [runc container breakout]

Fix Version: 1.0.0~rc2+docker1.13.1-0ubuntu1~16.04.1

Solution: Upgrade package runc to version 1.0.0~rc2+docker1.13.1-0ubuntu1~16.04.1 or above.

Aqua Score: 7.2 High

Solutioning

2 Container Attack Surface Reduction

Solutioning

3 User Access Control

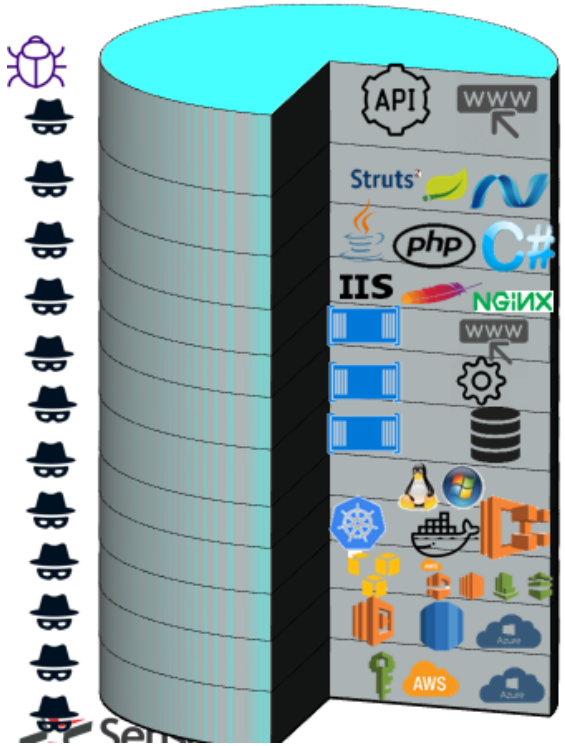
Solutioning

4 Hardening the Host OS & the Container

See NIST SP 800-190 and various others incl https://www.cisecurity.org/benchmark/docker/

Solutioning

5 SDLC Automation (DevOps)



Security Testing Needs to Go Down The Stack

User Interface (WebApps, forms, logons, API's)
Framework (Struts, Spring, .NET)
Language (Java, PHP, .NET)
AppServer (IIS, Apache, Nginx)
Process UI (Container, presentation layer)
Process App (Container, application processing)
Process BackEnd (Container, database)
Operating System (Linux, Windows)
Clustering/Orchestration (CaaS, Swarm, Kubernetes)
Networking (SDN, SecGroups)
Cloud Platform
Core Infrastructure

Recap

1	**Serverless, Microservices and Container Security**	4	**CI/CD Integration for Automated Security**
2	Key Implications for Penetration Testing Programs		End to End Vulnerability Management
3	Key Security features for Container Deployments		Continuous Monitoring, Governance & Compliance Reporting

Apply What You Have Learned Today – Exec/Procurement

- Next week you should:
  - Reset your review criteria for Penetration Testing
  - Explicitly incorporate testing of Cloud Technologies into your Vuln Mgt Program

- In the first three months following this presentation you should:
  - Review suppliers' capability to test Cloud Technologies
  - Develop the Blue Team side of the equation
  - Have A functional Shift Left feature in your Vuln Mgt Program for Cloud

- Within six months you should
  - Have performed an effective Penetration Test on your Cloud investment
  - Fine tune your blue team response to cloud technology attacks

Apply What You Have Learned Today – Pen Testers

- Next week you should:
  - Shortlist all the relevant cloud technologies in use by your clients
  - Re-calibrate your approach to test PaaS and Container
- In the first three months following this presentation you should:
  - Demonstrate the ability to breakout of containers
  - Demonstrate the ability to live off the land
- Within six months you should
  - Perfect methods for persistence in highly dynamic environments
  - Determine how to integrate Pen Test with client Blue Team (Purple Team)

RSAConference2019

Murray Goldschmidt
Chief Operating Officer
Sense of Security

murrayg@senseofsecurity.com.au

Office: +61 2 9290 4444

Mobile: +61 422 978 311