



DevOpsSec: Building CI/CD with Security Teams

Shawn Wells
Chief Security Strategist
Red Hat Public Sector
shawn@redhat.com || 443-534-0130



Late 1990's

YOU CAN CLICK BUT YOU CAN'T HIDE

I

ILLEGAL DOWNLOADING

Inappropriate for All Ages





Early 2000's

Don't be an iPirate.
You'll Lose Yer Loot.





**FBI ANTI-PIRACY
WARNING**

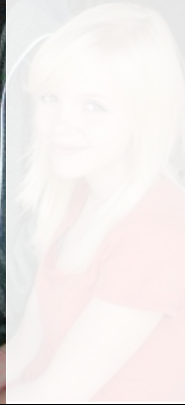
Find the “software pirate”



Y2K



15
years old







© 2007 CNET Networks, Inc.









**Selling access to
music,
“Movies”,
software**



Monetizing “Sensor data”

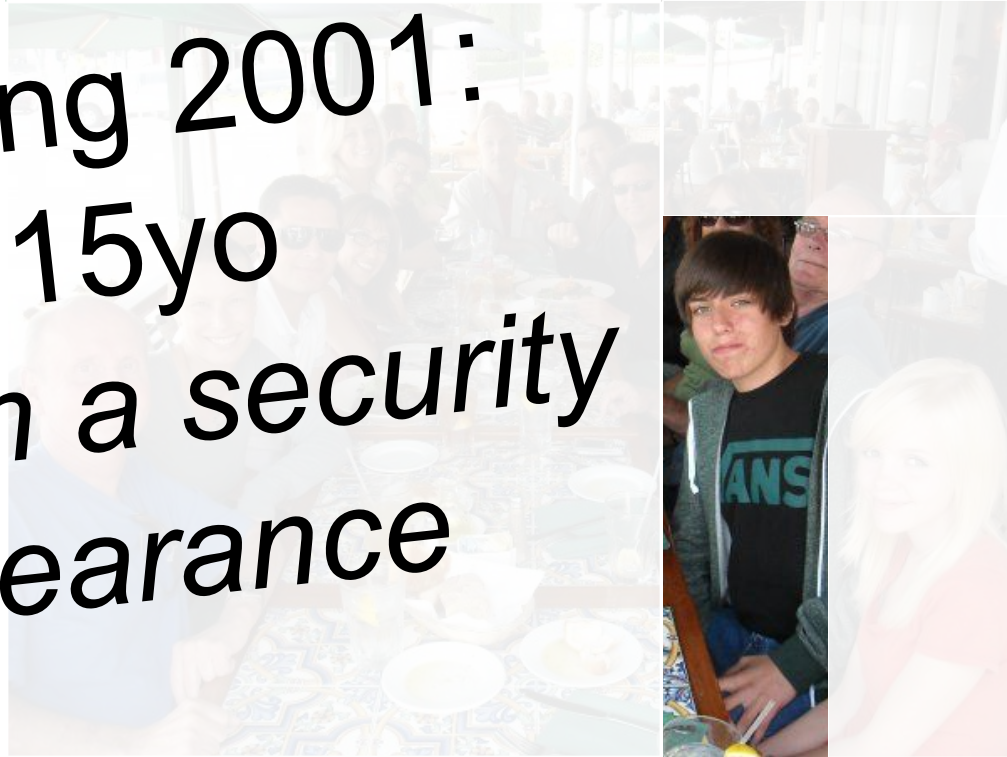




Busted
Fall 2000
:(

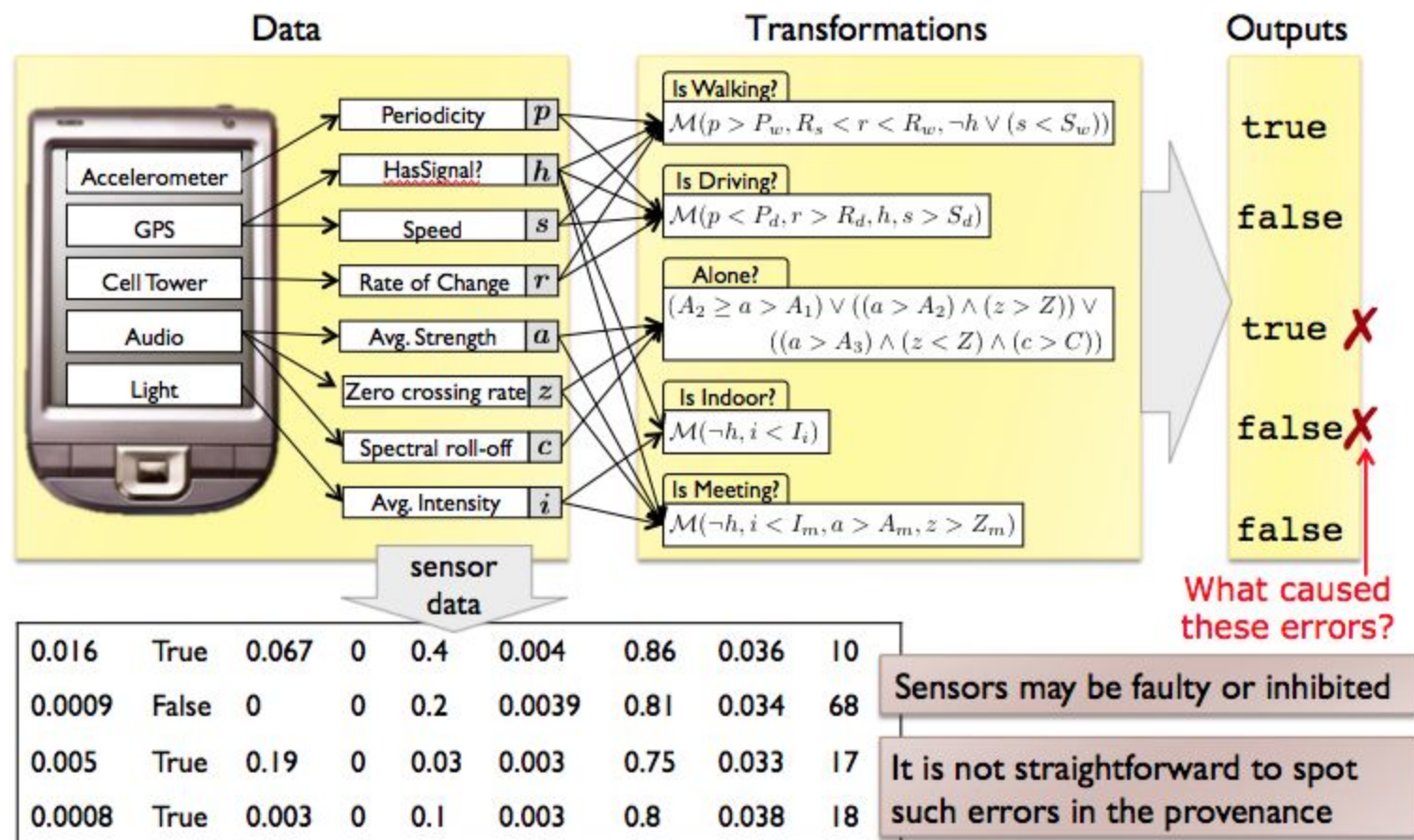


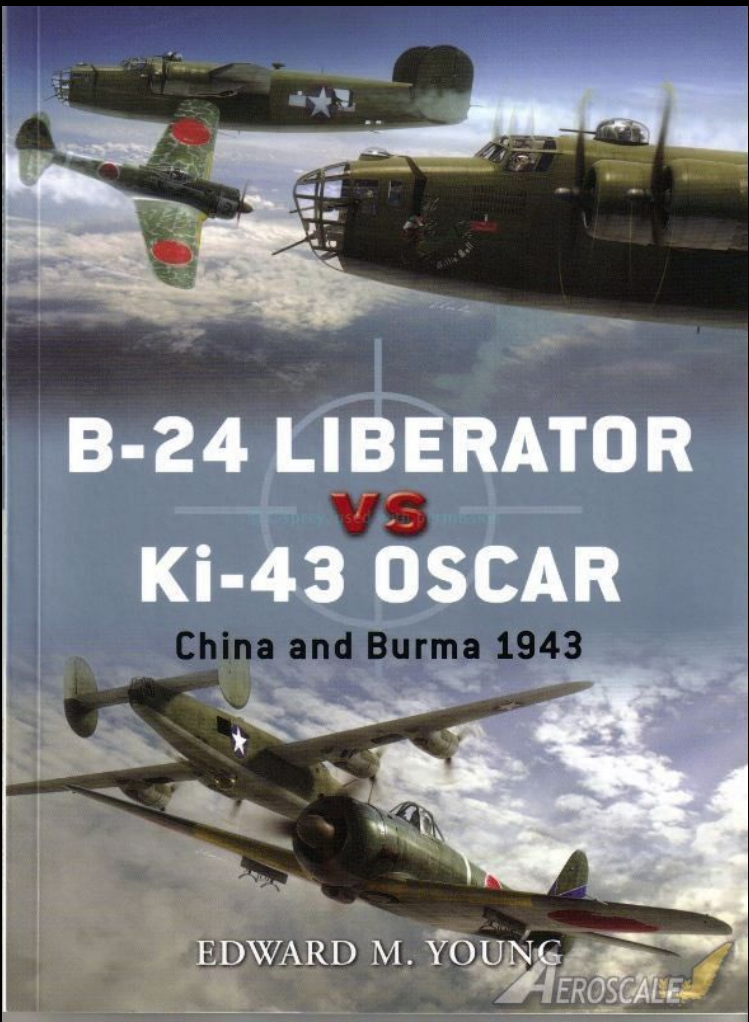
Spring 2001:
15yo
...with a security
clearance



Exploiting
~~Monetizing~~
“Sensor
data”







B-24 LIBERATOR

VS

Ki-43 OSCAR

China and Burma 1943

EDWARD M. YOUNG

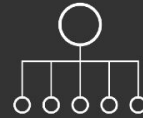
AEROSCALE







SORENSEN'S LESSONS



**STANDARD
PARTS**

**STANDARD
PROCESS**

**STANDARD
INFRASTRUCTURE**

**BUILD
FOR CHANGE**



STANDARD PARTS

- Standardized
- Interoperable
- Multi-vendor

STANDARD PROCESS

STANDARD INFRASTRUCTURE

BUILD FOR CHANGE



STANDARD
PARTS

**STANDARD
PROCESS**

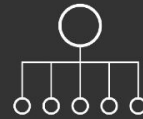
STANDARD
INFRASTRUCTURE

BUILD
FOR CHANGE

Eliminate redundancy

Encourage flexibility

Drive modularity



STANDARD
PARTS

STANDARD
PROCESS

**STANDARD
INFRASTRUCTURE**

BUILD
FOR CHANGE

Process drives tools.
Not the other way around.



STANDARD
PARTS

STANDARD
PROCESS

STANDARD
INFRASTRUCTURE

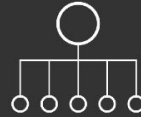
**BUILD
FOR CHANGE**



Turn craftwork
into commodities.

Design for improvement,
not function.

SORENSEN'S LESSONS



**STANDARD
PARTS**

**STANDARD
PROCESS**

**STANDARD
INFRASTRUCTURE**

**BUILD
FOR CHANGE**

RELEASES PER YEAR

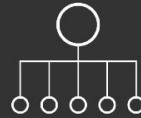
1/day



1/hour

DevOps

~~SORENSEN'S LESSONS~~



**STANDARD
PARTS**

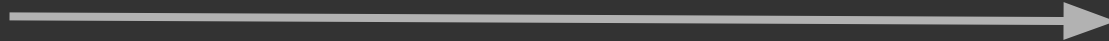
**STANDARD
PROCESS**

**STANDARD
INFRASTRUCTURE**

**BUILD
FOR CHANGE**

RELEASES PER YEAR

1/day



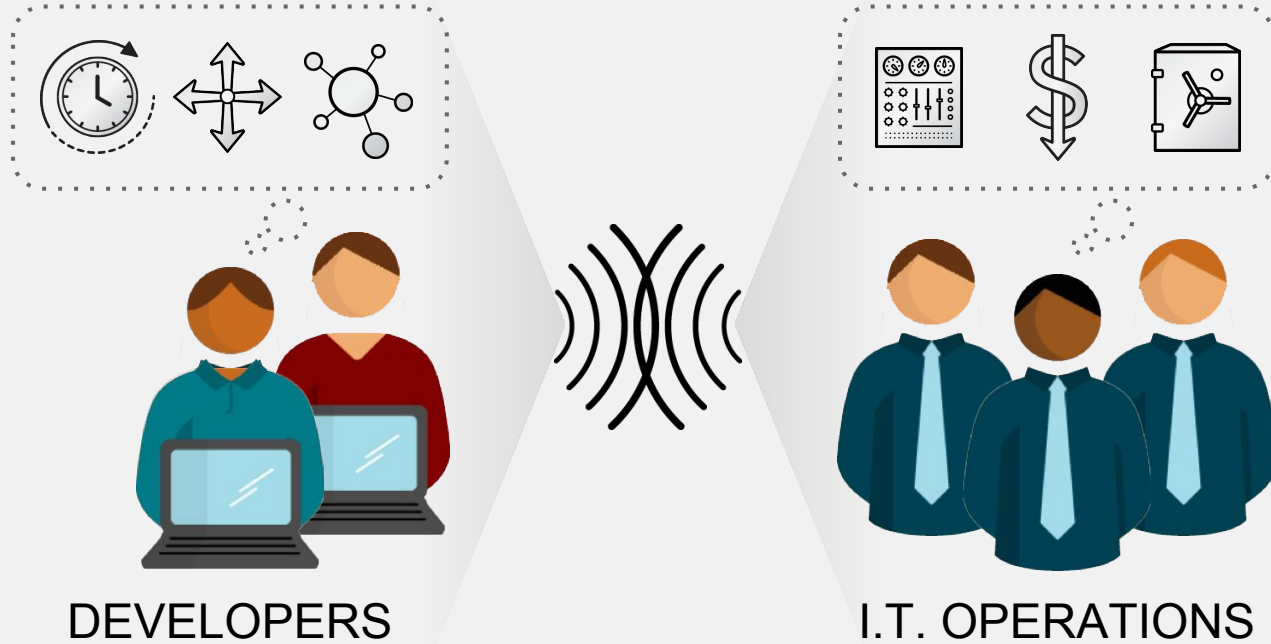
1/hour

The Problem

Applications require complicated installation and integration every time they are deployed



THE PROBLEM



DEVOPS

Everything as code

Automate everything

Continuous Integration/Delivery

Application is always “releaseable”

Application monitoring

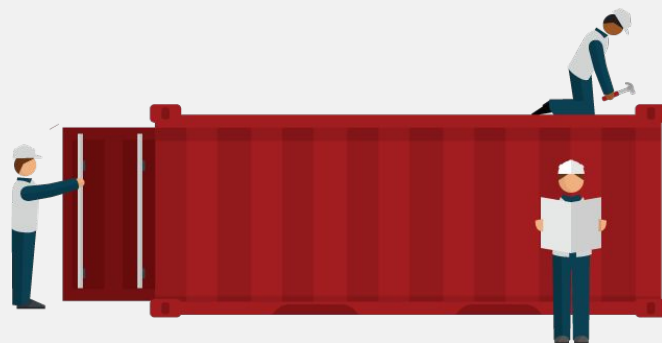
Rapid feedback

Rebuild vs. Repair

Delivery pipeline

A Solution

Adopting a container strategy will allow applications to be easily shared and deployed.



WHAT ARE CONTAINERS?

It Depends Who You Ask

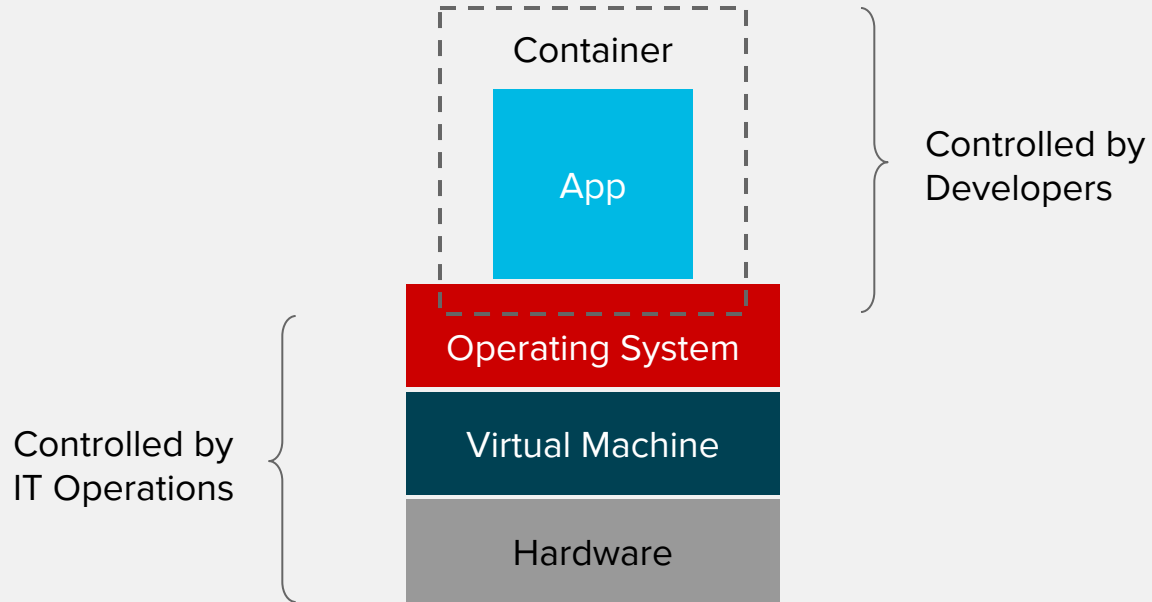
INFRASTRUCTURE

- Sandboxed application processes on a shared Linux OS kernel
- Simpler, lighter, and denser than virtual machines
- Portable across different environments

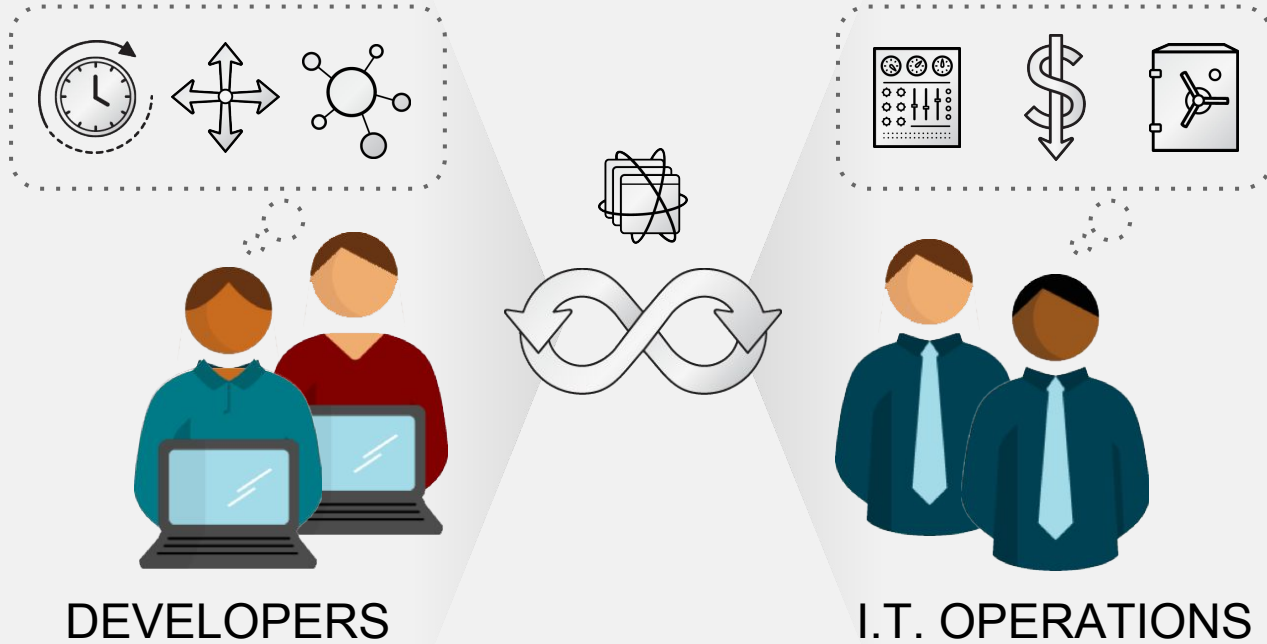
APPLICATIONS

- Package my application and all of its dependencies
- Deploy to any environment in seconds and enable CI/CD
- Easily access and share containerized components

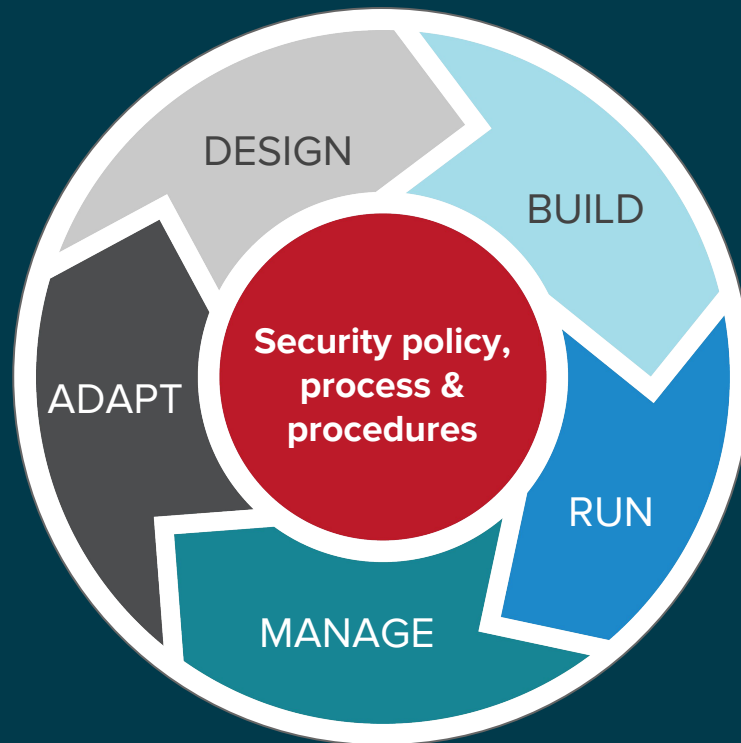
A SOLUTION



A SOLUTION

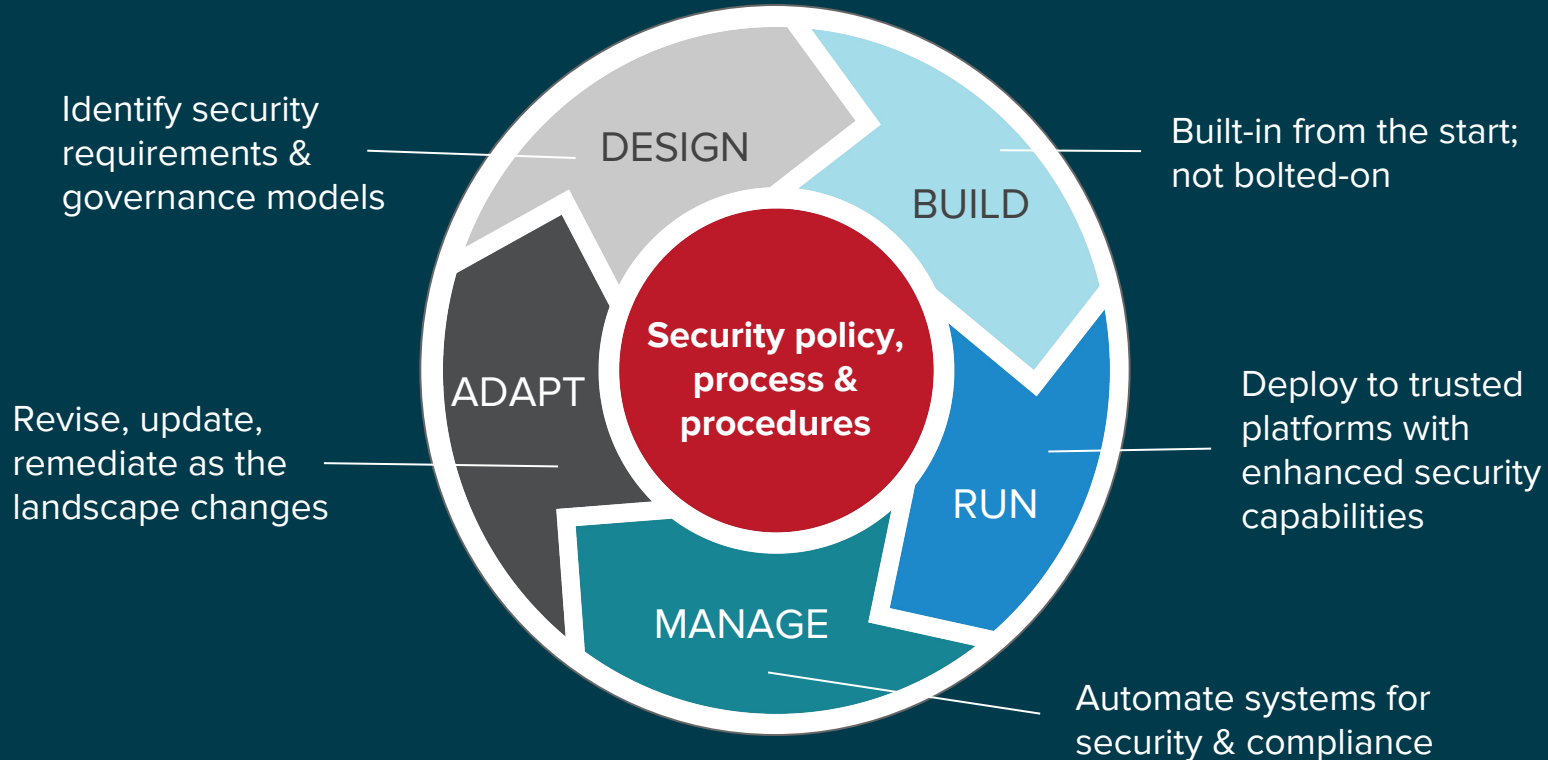


SECURITY MUST EVOLVE



SECURITY MUST BE CONTINUOUS

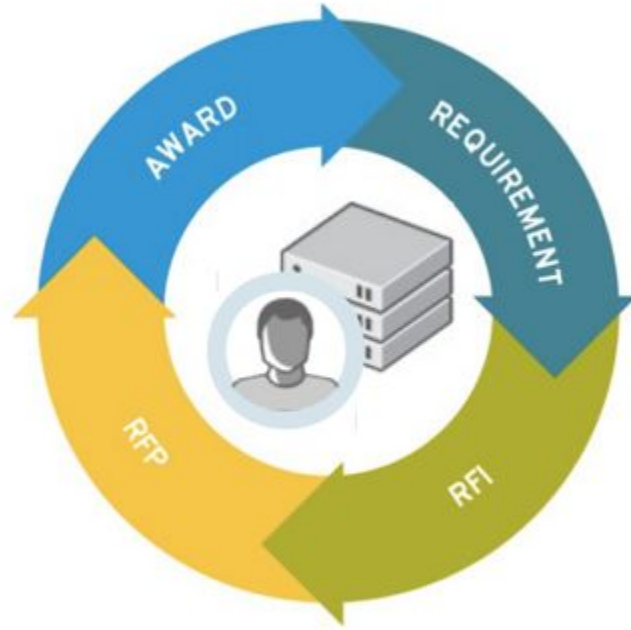
And integrated throughout the IT lifecycle



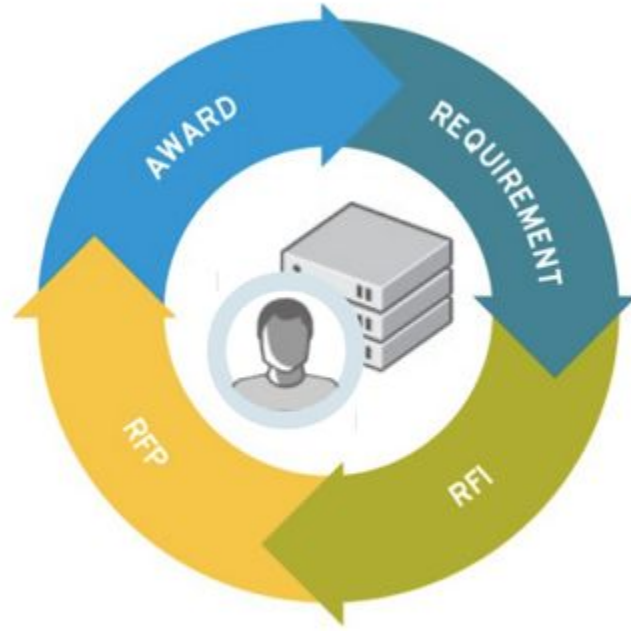


U.S. ARMY

CYBER PROTECTION BRIGADE



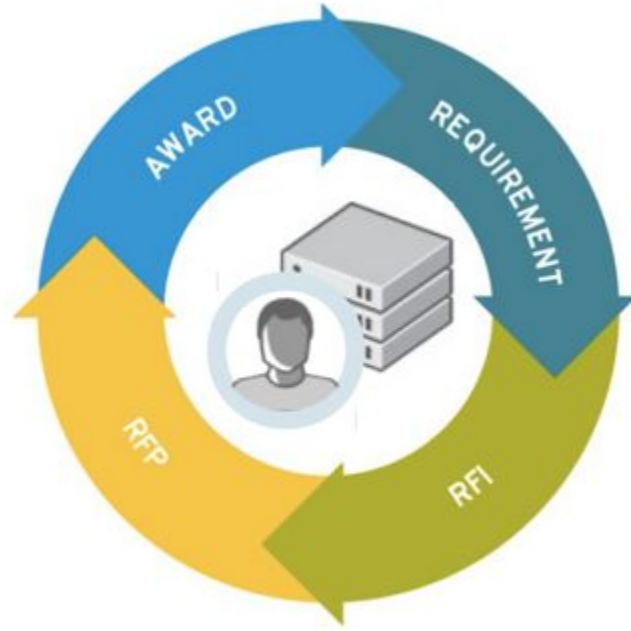
We have to share to iterate quicker.



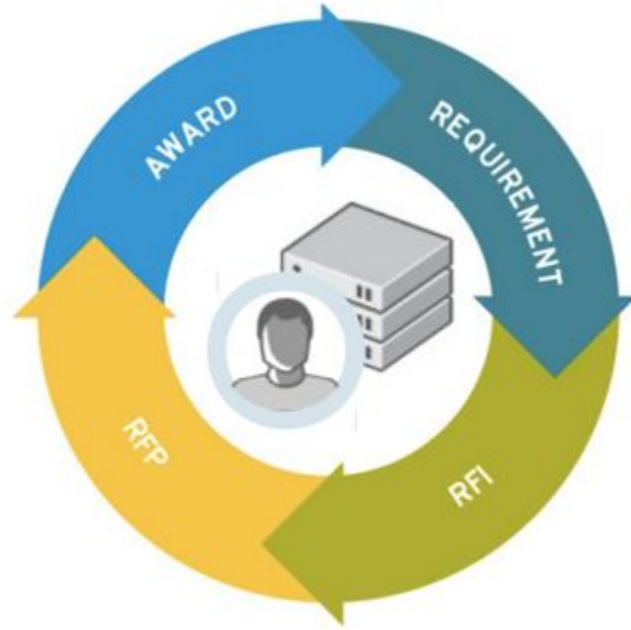
A cloud is sharing at run-time.



Open source is sharing at dev-time.



Agility is the capability.



You are the catalyst.

YOU ARE NOT AN
IT CRAFTSMAN.

YOU ARE AN
IT MANUFACTURER.

**WE CAN DO MORE
WHEN WE WORK
TOGETHER**



Contact Info

LinkedIn: <https://www.linkedin.com/in/shawndwells/>

E-Mail: shawn@redhat.com

Cell: 443-534-0130 (US EST)

Blog: <https://shawnwells.io>

