

Application security seen from an enterprise level

The role of Enterprise Security Architecture in secure application development

Stefaan Van daele

Executive Security Architect



June 2022

IBM Security

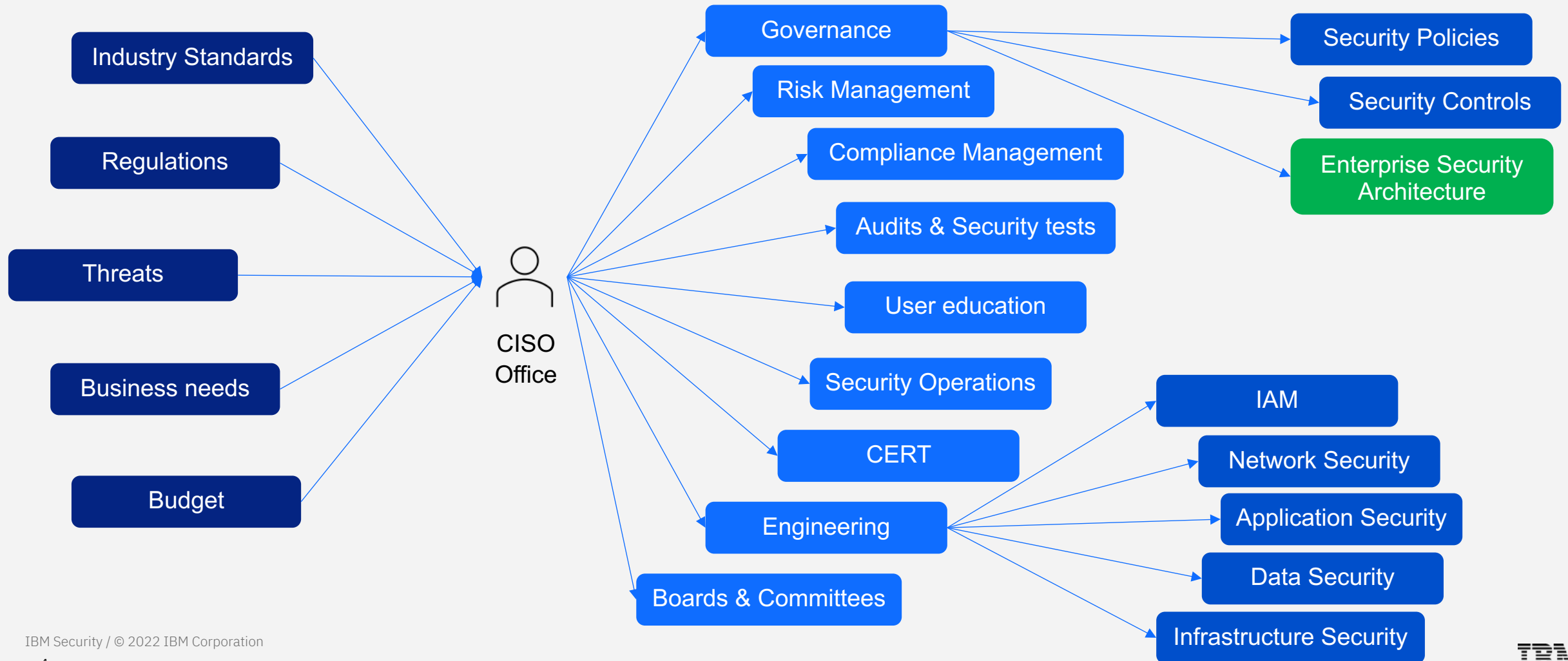


Topics

- What is an Enterprise Security Architecture?
- Zero Trust in ESA
- The type of application makes a difference in approach
- Application related security requirements from an enterprise level viewpoint

What is Enterprise
Security Architecture?

The starting context : Enterprise / Corporate Security



Enterprise Security Architecture is not Security Architecture

Enterprise Security Architecture

Enterprise Information Security architecture (EISA or ESA) is the practice of applying a comprehensive and rigorous [method for describing a current and/or future structure and behaviour for an organization's security processes](#), information security systems, personnel, and organizational sub-units so that they align with the organization's core goals and strategic direction.

ESA is the structure to realize Security Governance.

https://en.wikipedia.org/wiki/Enterprise_information_security_architecture

An ESA is mostly Risk driven

Security Architecture

Security Architecture (SA) is a subset of an IT Architecture defining all the security specific aspects of the overall solution. It typically covers topics like authentication and authorization, threat modelling, data security, the definition of security related operations and more. Security is here part of the so called non-functional requirements.

A Security Architecture is in most cases fully integrated in the overall design approach and not a stand-alone exercise.

A SA is mostly Policy driven

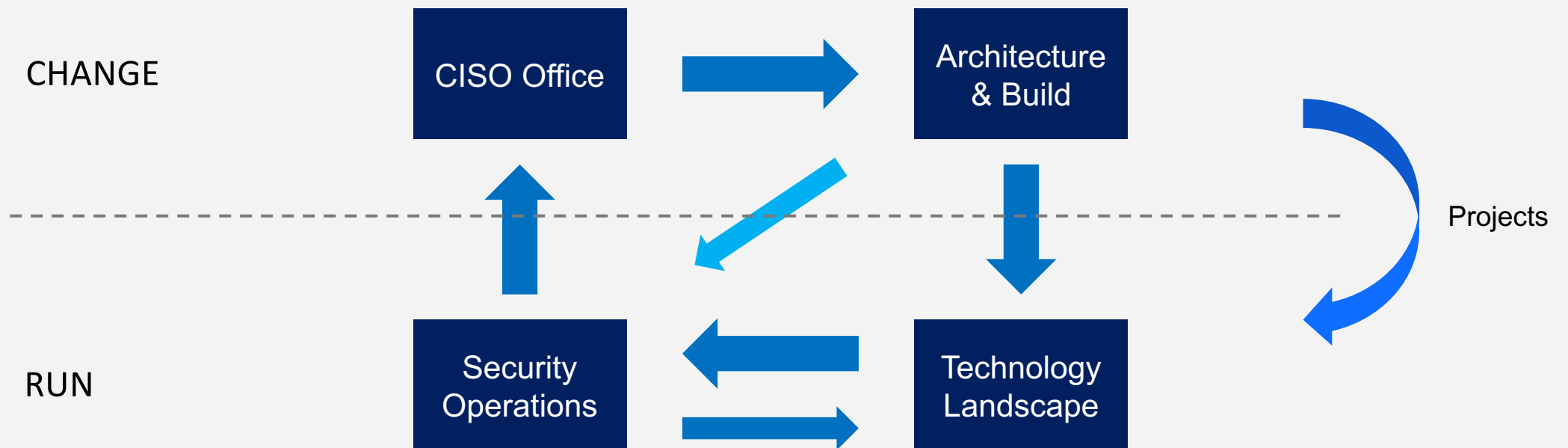
Architecture of a Security Specific Solution

The architecture of a security solution is a regular IT Architecture needed to design, build, deploy and operate a security specific solution. It contains all activities and work products like for other solutions with main difference that most of the functional requirements are now security specific.

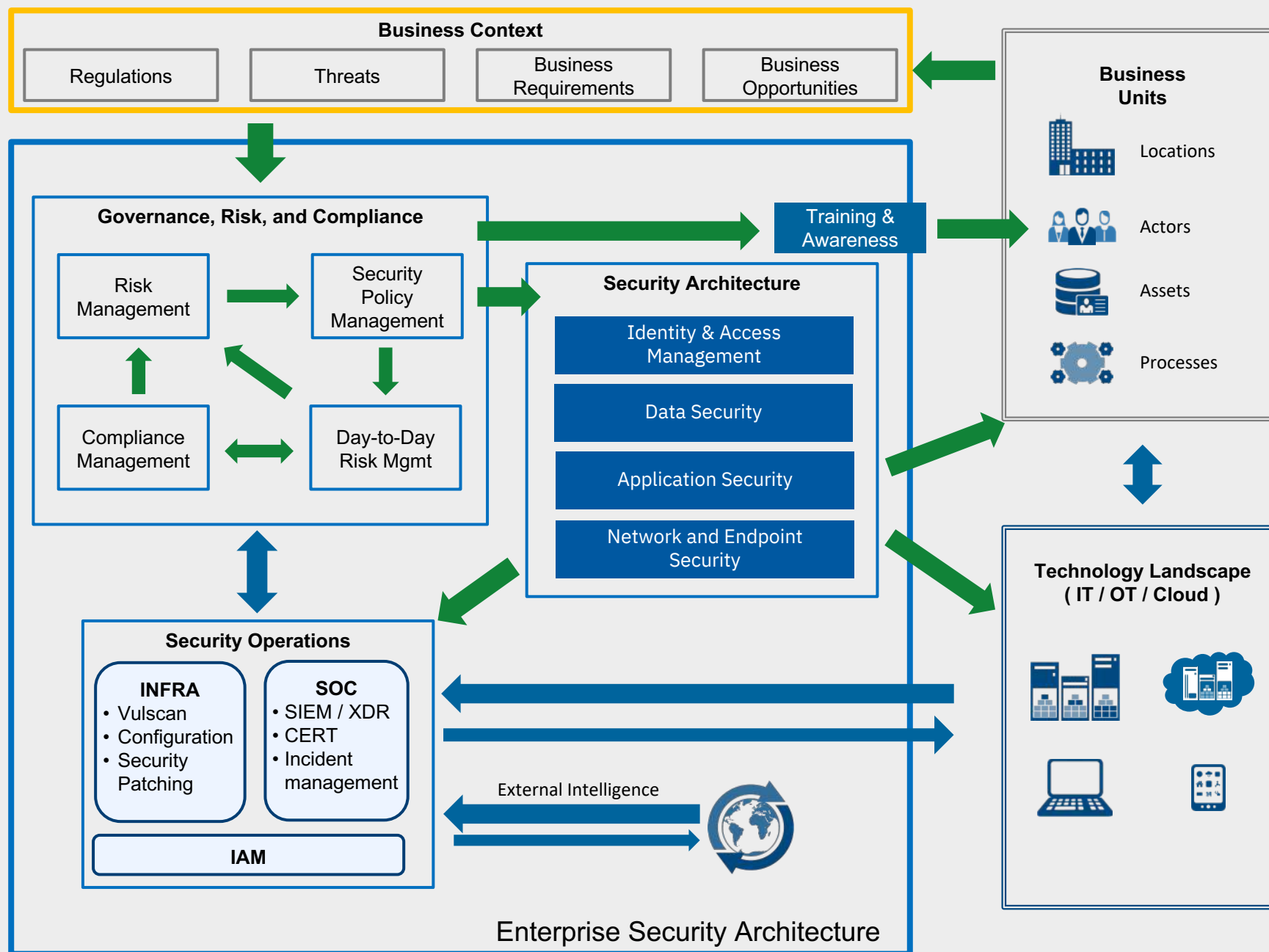
Our company is using Agile and DevOps, so we don't need an ESA

- That doesn't change the goal of an ESA nor it removes the need for security architectures
- If a Product Owner would like to deploy an application in a production environment, then all the security requirements (user stories) have to be implemented or the missing ones have to be formally accepted as risks by the Risk Committee...
- We will discuss this later

An ESA addresses Corporate Security as a closed loop

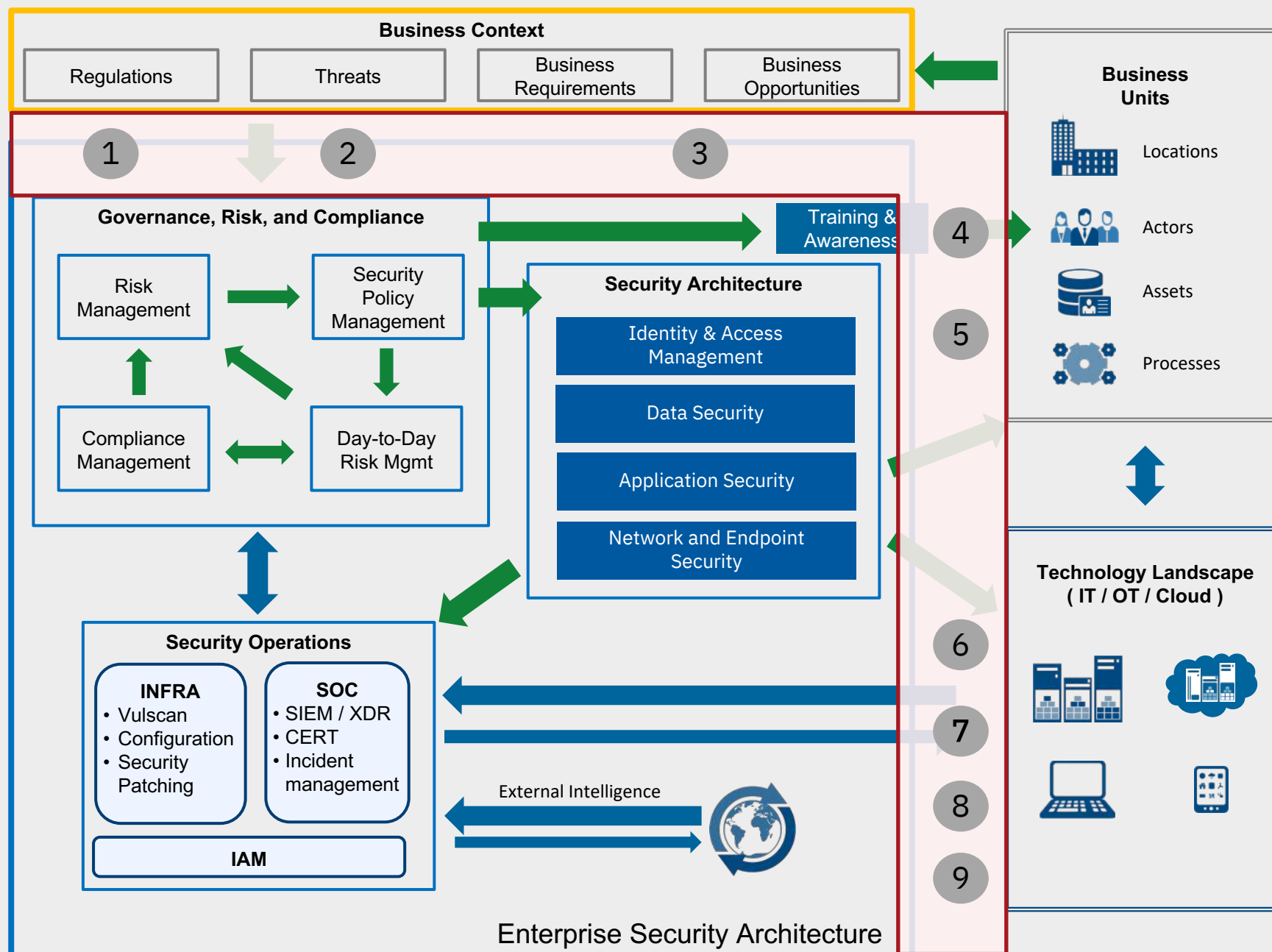


A more detailed version of the closed loop

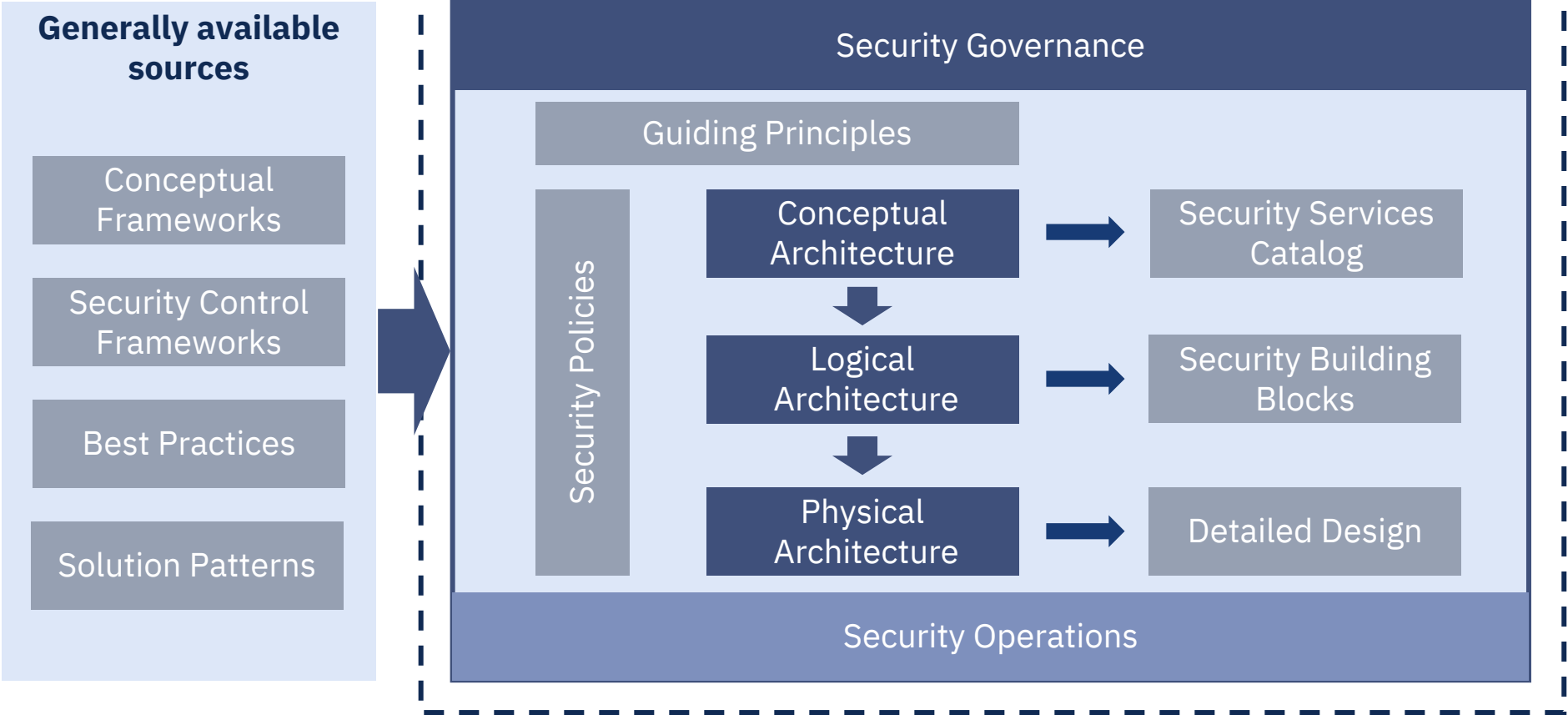


CISO Office Service Catalog

1. Audit Support
2. Risk Analysis
3. Demand management
4. Training
5. BU specific services
6. Security Tooling
7. Standards & Guidelines
8. Security operations
9. Incident Response

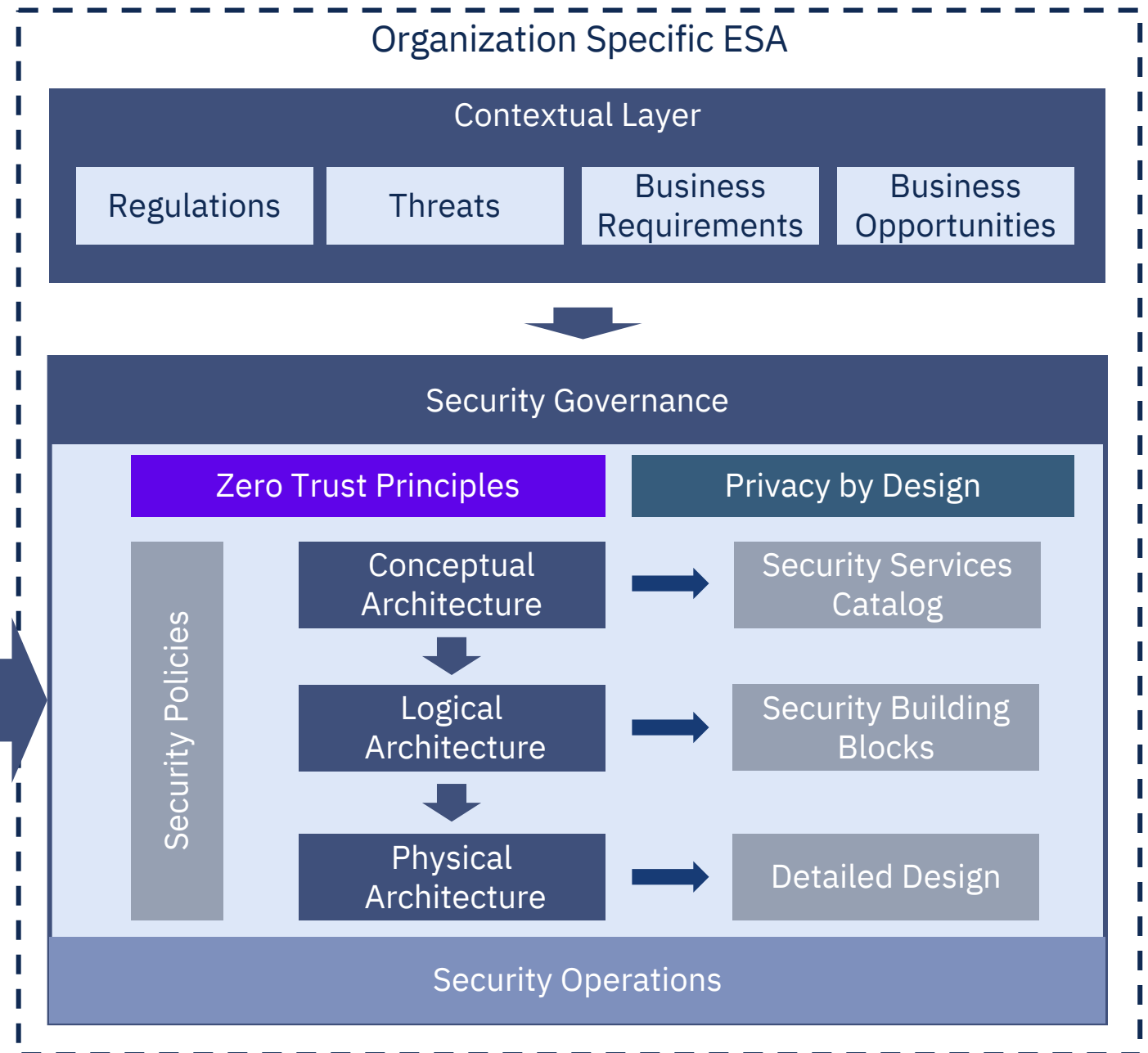


Quid Zero Trust?

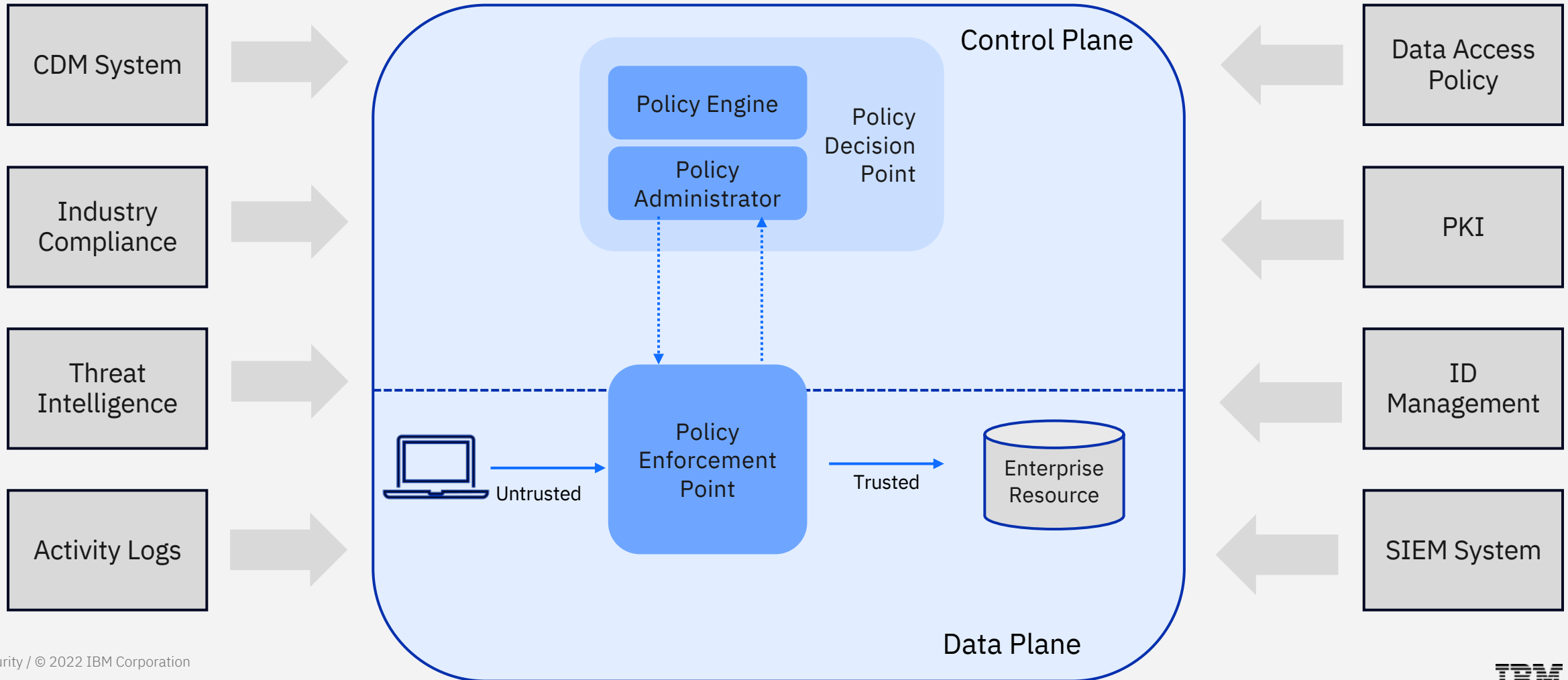


Zero Trust is a set of principles driving the outcome of the design

- Always verify, never trust
- Assume breach
- Least privilege



NIST 800-207 Conceptual Zero Trust Architecture



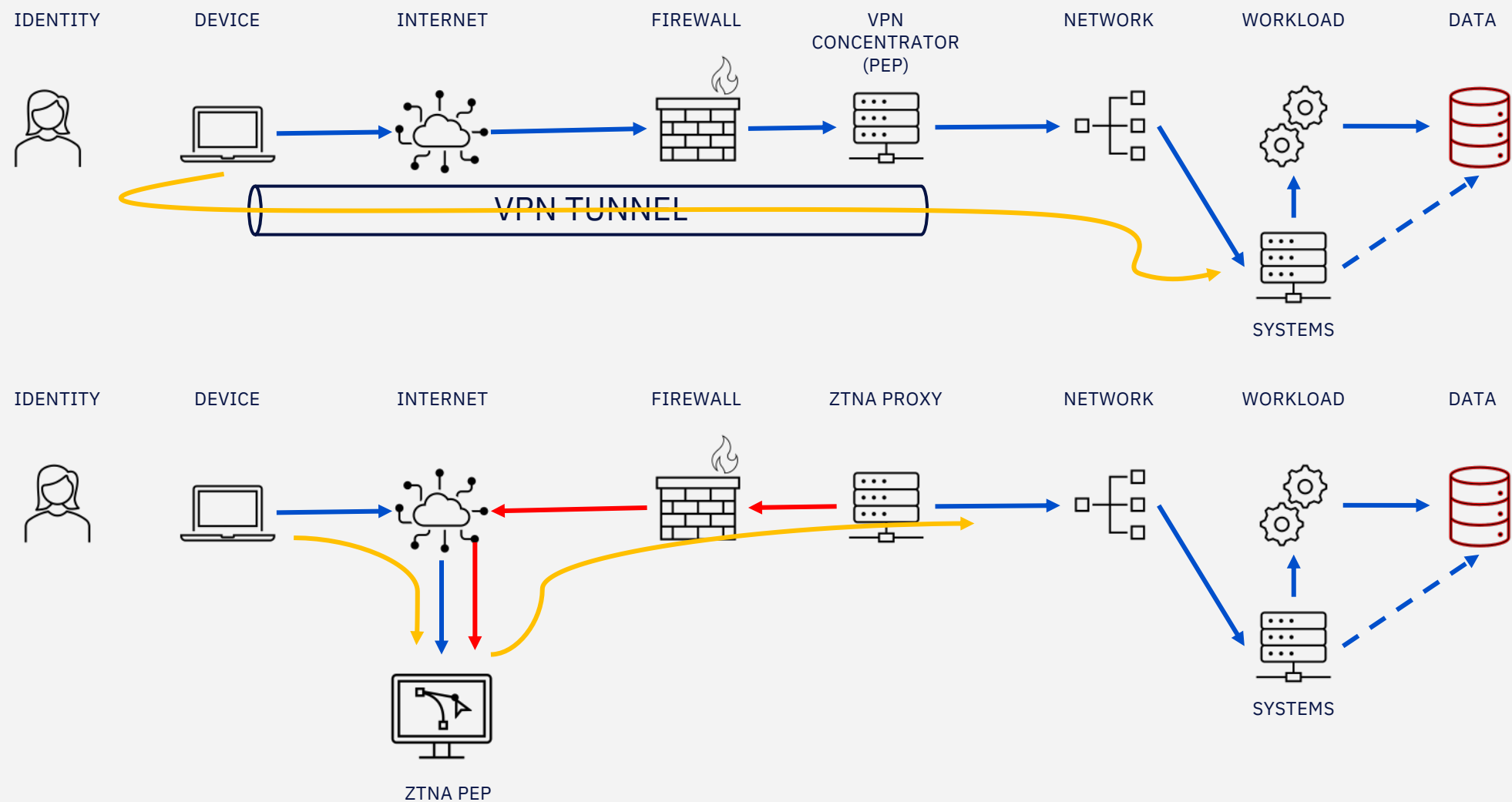
Zero Trust is not so much about new building blocks from an ESA perspective, rather new patterns and guidelines

Example of possible building blocks

	IAM	Data	Application	Endpoint	Network
Least Privilege	<ul style="list-style-type: none">• JIT• PAM• RBAC	<ul style="list-style-type: none">• Encryption• DRM	<ul style="list-style-type: none">• Adaptive Access Control	<ul style="list-style-type: none">• Hardening• PAM	<ul style="list-style-type: none">• Micro-segmentation• ZTNA
Always Verify	<ul style="list-style-type: none">• Continuous Authentication• Adaptive Access Control	<ul style="list-style-type: none">• Adaptive Access Control	<ul style="list-style-type: none">• Adaptive Access Control• Process Allow List• TCB	<ul style="list-style-type: none">• TCB	<ul style="list-style-type: none">• Adaptive Access Control• NAC
Assume Breach	<ul style="list-style-type: none">• UEBA	<ul style="list-style-type: none">• DLP	<ul style="list-style-type: none">• EDR	<ul style="list-style-type: none">• EDR	<ul style="list-style-type: none">• NDR

Adaptive Access Control = combination of features from Role Based Access Control, Attribute Based Access Control, Risk Based Access Control and Multi-Factor Authentication

A practical example: Zero Trust Network Access



Application types

Type of applications and ownerships results in a specific approach

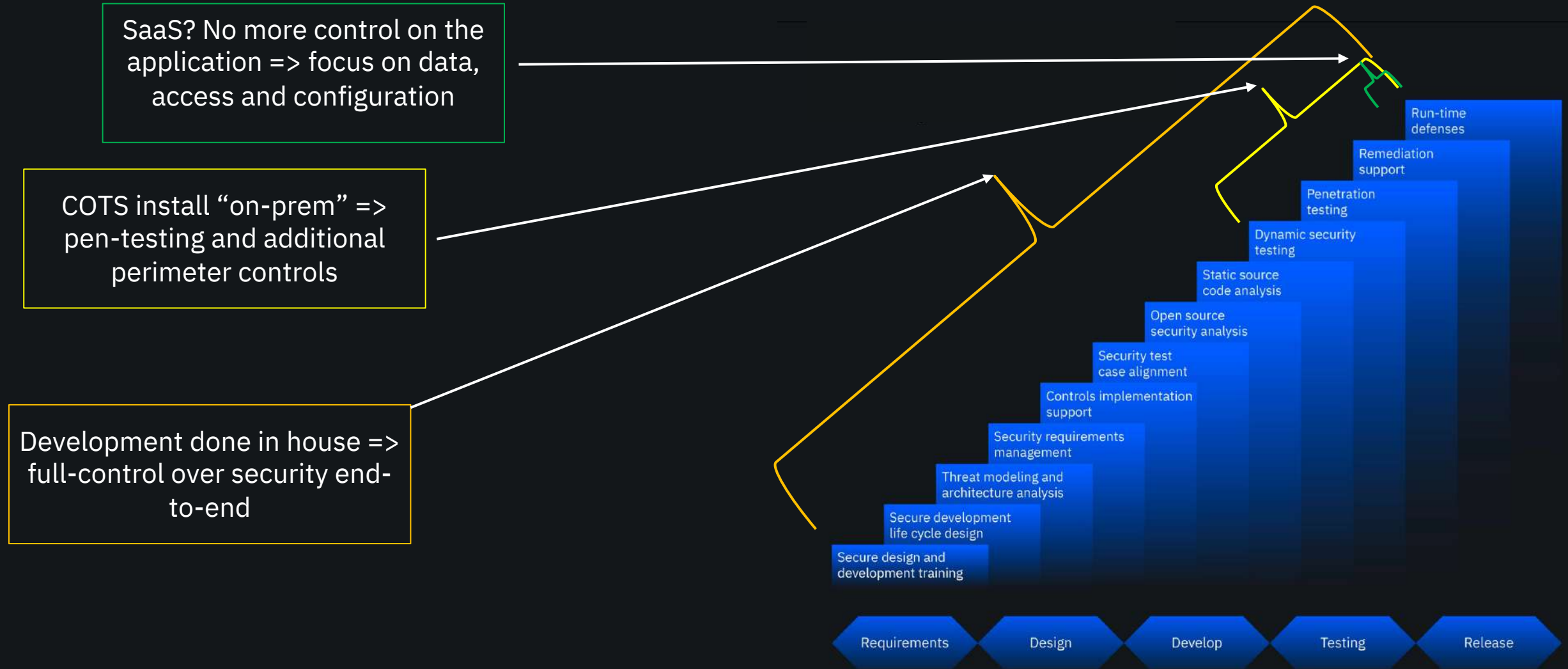
Type of application / software

- Consumer App on mobile phone
- Workforce App on mobile phone
- Public facing Web Application running in Cloud
- Public facing Web Application running in Data Centre
- Private Web Application
- Client-Server application
- OT software (PLC, ...)
- Firmware
- Operating system
- ...

Ownership

- Developed in house (all code)
- Developed in house making use of libraries
- Developed by a partner (turnkey solution)
- **Commercial of the shelf software (COTS) to be installed**
- **Open Source Software (OSS) to be installed**
- **SaaS Application**

Software Development Life Cycle and ownership



Security measures will depend on the type and ownership

Example 1 : COTS to be installed on Virtual Machines used by the company employees (private web app)

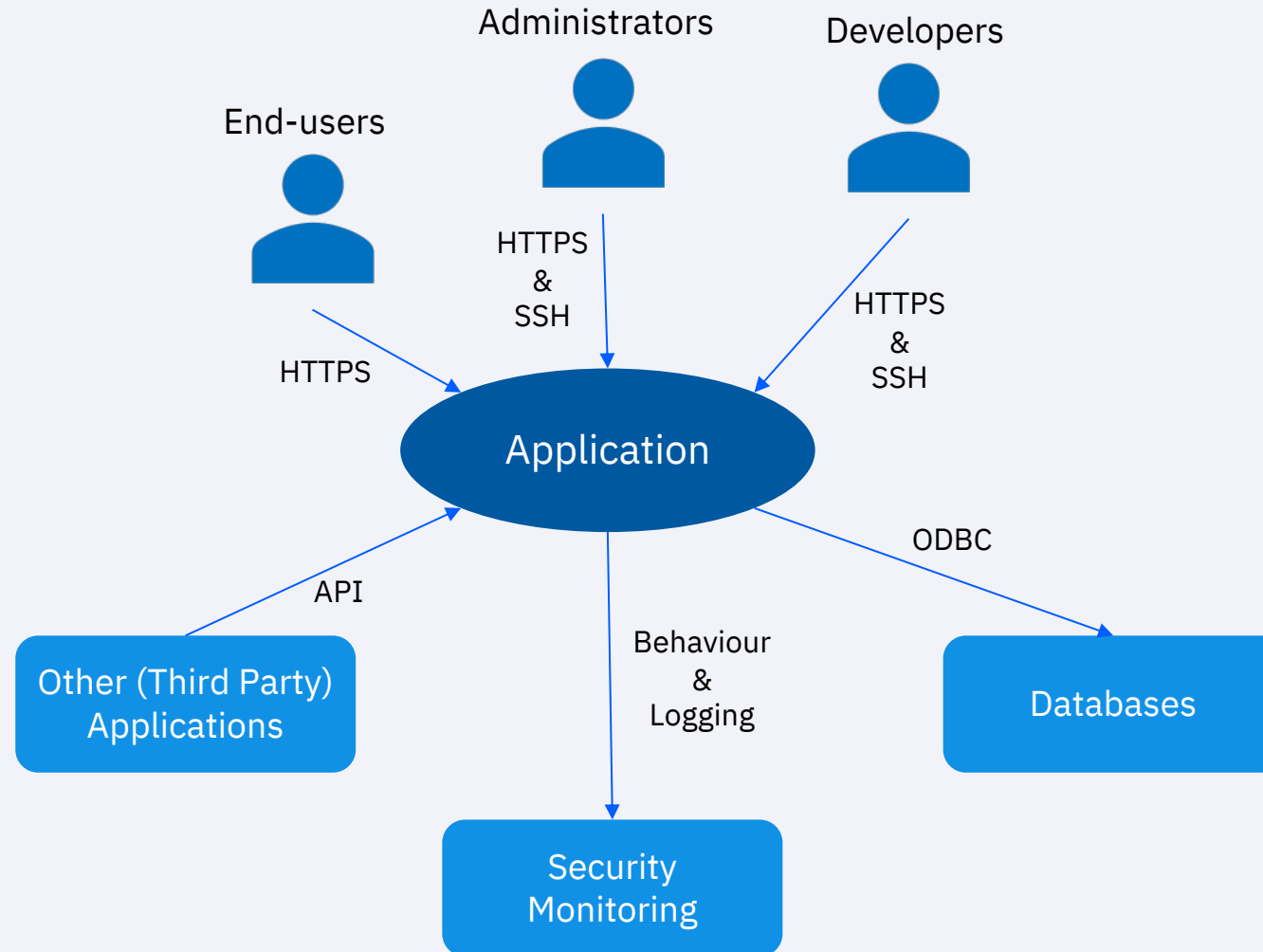
- No control on software development or patches
- Pen-testing to find
 - Software vulnerabilities
 - Configuration errors
 - Design flaws
- Web Application Firewall (WAF) as extra protection
- Zero Trust Network Access to secure “remote” network access
- Micro-segmentation to secure network access to other systems and applications
- Manage privileged access

Example 2: Public web facing in house developed application

- Provide requirements and guidelines before application design starts (if applicable, Security as a code)
- Thread modelling on the design
- Source code scanning
- Dynamic and Integrated application testing
- Regular Pen-testing
- DDoS protection
- WAF
- Micro-segmentation to secure network access to other systems and applications

Security Requirements for applications

As security architect I look at an application as a black box



Applications developed “in house”

Where to start? Threats / Risks

Survey Results Rank	Survey Average Score	Issue Name
1	7.729927	Insufficient ID, Credential, Access and Key Mgt, Privileged Accounts
2	7.592701	Insecure Interfaces and APIs
3	7.424818	Misconfiguration and Inadequate Change Control
4	7.408759	Lack of Cloud Security Architecture and Strategy
5	7.275912	Insecure Software Development

List of Mitigating Controls

AIS

Application and Interface Security

AIS-02: Application Security Baseline Requirements

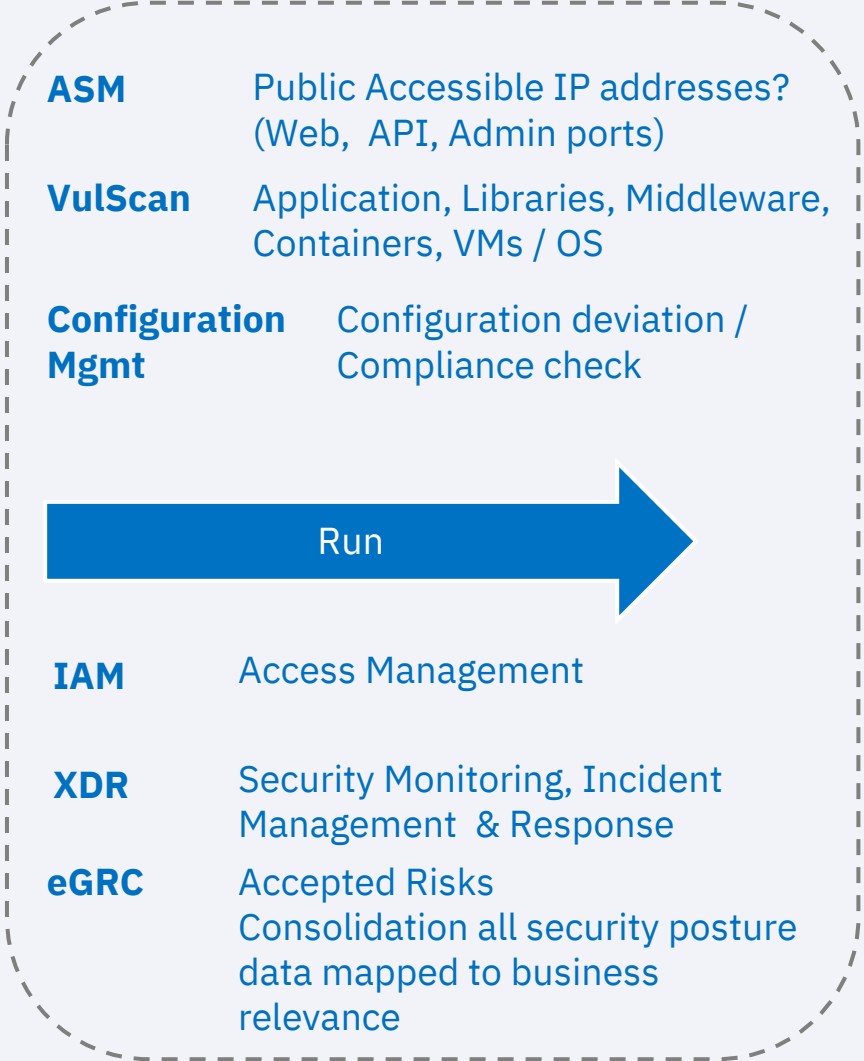
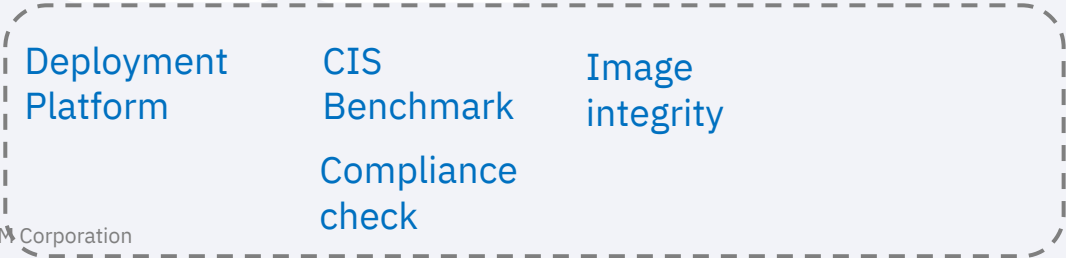
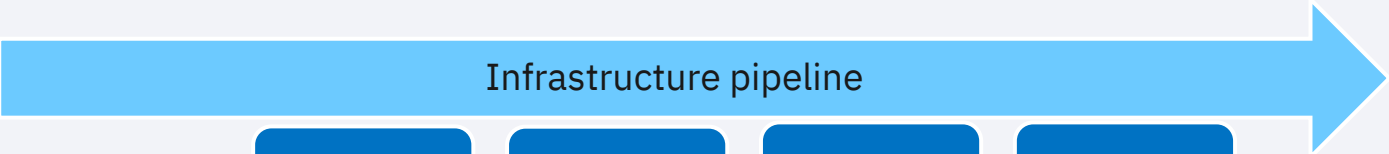
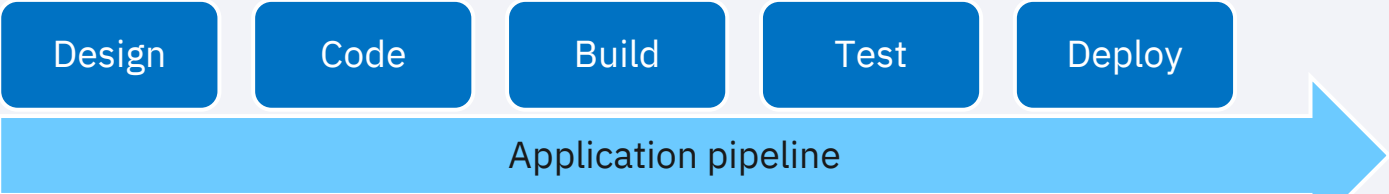
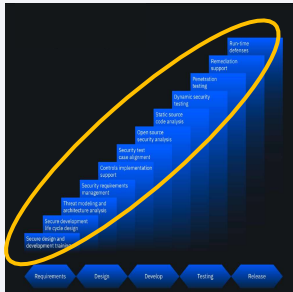
Control Specification and mapping

Control Domain	Control Title	Control ID	Updated Control Specification
Application & Interface Security	Application Security Baseline Requirements	AIS-02	Establish, document and maintain baseline requirements for securing different applications.

Translated into guidelines and procedures with assignment of ownership (and automate where possible)

Sources: *Top Threats* to Cloud Computing - Pandemic Eleven (2022) – Cloud Security Alliance
Cloud Controls Matrix Version 4.0 (2021) – Cloud Security Alliance

What could be the baseline security requirements?



How could a dev(ops) team facilitate / anticipate these requirements?

IAM

- Run code at lowest privilege possible
- Higher privilege needed? => just in time & temporary (check-out / check-in via APIs or Short lifetime tokens)
- Anticipate for delegation of authentication
- Anticipate for delegation of authorization (where possible)
- End-to-end traceability and accountability (who is updating the database)
- Do I have to mention this? No hard coded credentials / secrets /

DATA

- Sensitive data? Then the DBA shouldn't be allowed to see it (nor the devops team by the way)
- Long living sensitive data? Anticipate that the encryption algorithm for data-at-rest might have to change during the data retention time.
- Take care of metadata (e.g. data classification)

APPLICATION

- Test, test, test => Automated as much as possible: Threat Modelling, SAST, DAST, IAST, Compliance
- Meaningful security logging (authentication, change of authorizations, unexpected action)
- List of libraries included in the build are clearly documented (SBOM)

ENDPOINT

- Run code on hardened systems e.g. CIS Benchmark, yes even in test environments
- Serverless based solution ? Still some security work to do (next slide)

NETWORK

- Don't assume that network access to other systems, internet, ... is a given => document clear requirements and a baseline for normal network connection behaviour

Even in a serverless environment there are still security requirements

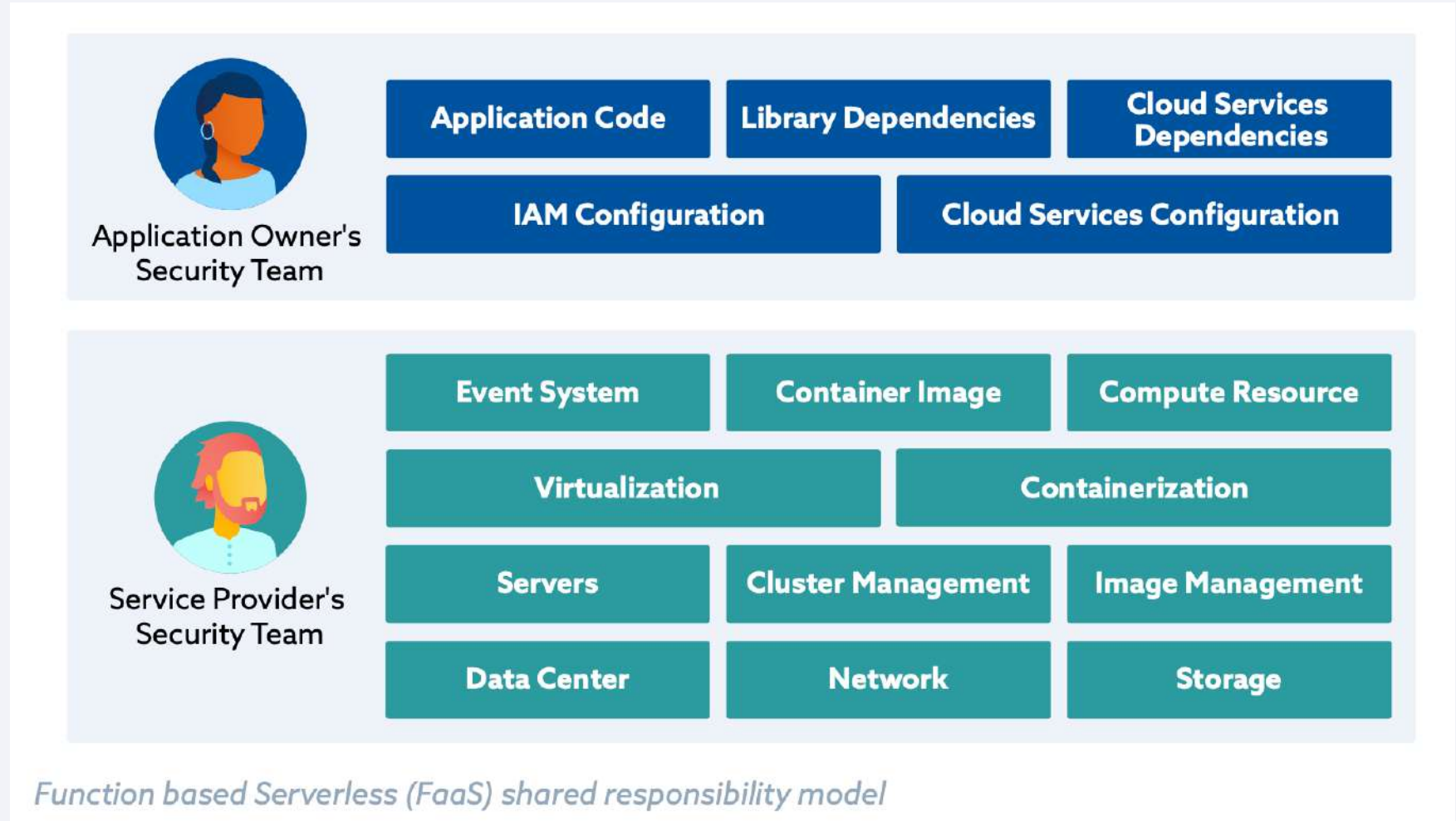


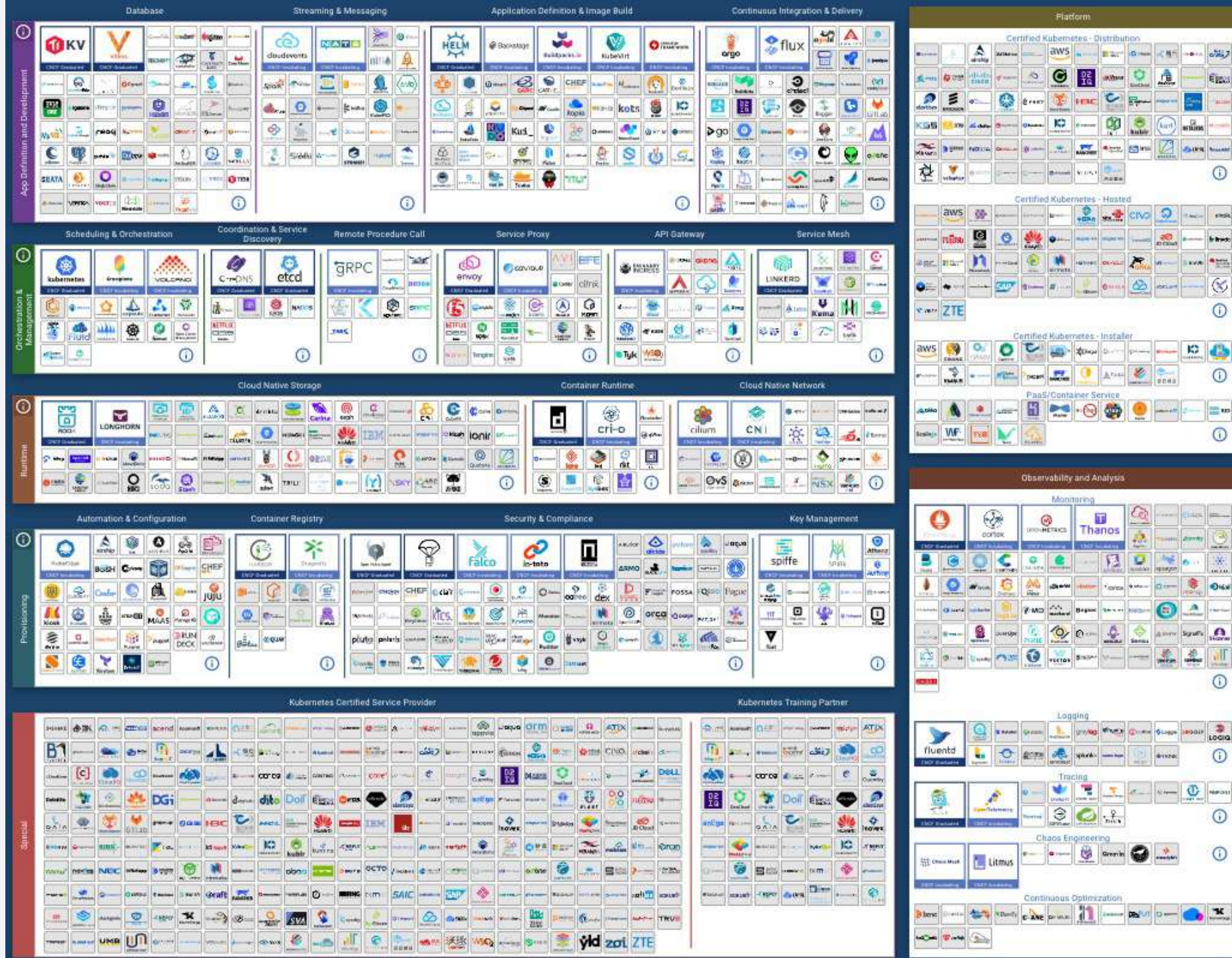
Image taken from “How to Design a Secure Serverless Architecture” Cloud Security Alliance

So you have a developer toolkit?

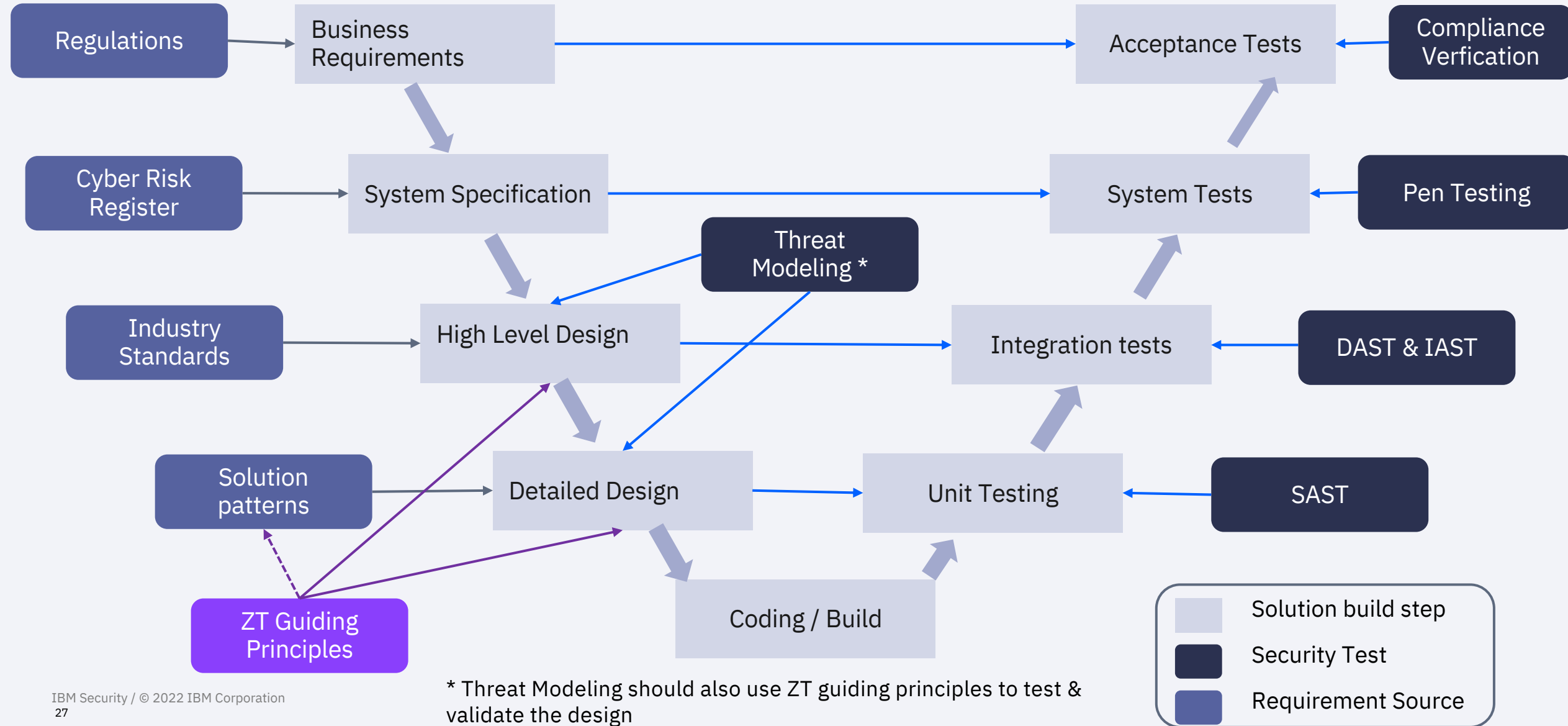
Before writing the first line of code let's think about securing the IDE:

- Selecting & configuring tools
- Securing the SDLC processes
- Protecting the source code (integrity and confidentiality) => e.g. Who has access to the repo? A developer who has left the company since 3 months

Source: <https://landscape.cncf.io/>



A side step – Zero Trust in a Security Test Strategy



Summary

- Security is a continuous process and in large(r) organisations an Enterprise Security Architecture helps to create a structure and an approach to realise security objectives
- A CISO office has many challenges, secure application development being one of them
- There is more secure application development than writing code without vulnerabilities
- Zero Trust is a set of guiding principles improving maturity, reducing implicit trust and it should be part of every IT initiative as part of the overall strategy
- Approach depends on the type of application
- Risks => Mitigating Controls => Security Control Framework => Guidelines, procedures and Standards => Implementation (in an automated way)
- Most challenges in a company are about COTS

References

<https://devops.jaxlondon.com/blog/devops-conference/most-important-devops-metric/>
<http://www.redbooks.ibm.com/abstracts/sg248100.html?Open>
<https://sabsa.org/>
<https://www.opensecurityarchitecture.org/cms/index.php>
<https://www.ibm.com/cloud/garage/architectures>
<https://go.forrester.com/zero-trust/zero-trust-model/>
<https://www.ibm.com/security/zero-trust>
<https://exchange.xforce.ibmcloud.com/botnet>
<https://exchange.xforce.ibmcloud.com/activity/map>
<https://www.ibm.com/security/resources/xforce/xfisi/>
<https://publications.opengroup.org/g112>
<https://www.maastrichtuniversity.nl/um-cyber-attack-symposium-%E2%80%93-lessons-learned>
<http://www.redbooks.ibm.com/abstracts/sg246014.html>
<https://securityintelligence.com/the-five-most-critical-tasks-in-the-ciso-job-description/>
<https://securityintelligence.com/posts/how-to-transform-from-devops-to-devsecops/>
<https://securityintelligence.com/posts/dev-ops-error-cloud-security-attackers/>
<https://www.ibm.com/security/services/application-security-services>
<https://www.ibm.com/thought-leadership/institute-business-value/en-us/report/ai-cybersecurity>
<https://www.ibm.com/garage/method/practices/code/threat-modeling>
<https://www.ibm.com/cloud/learn/devsecops>
<https://www.ibm.com/cloud/architecture/architectures/secure-devops-arch/overview>
<https://www.shadowserver.org/news/over-380-000-open-kubernetes-api-servers/>
<https://nvlpubs.nist.gov/nistpubs/ir/2012/NIST.IR.7864.pdf>
<https://cloudsecurityalliance.org/artifacts/cloud-controls-matrix-v4/>
<https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-pandemic-eleven/>
<https://www.cvedetails.com/index.php>
<https://www.techworm.net/2020/01/german-government-windows-7-updates.html>

Thank you

Follow us on:

ibm.com/security

securityintelligence.com

ibm.com/security/community

xforce.ibmcloud.com

[@ibmsecurity](https://twitter.com/ibmsecurity)

youtube/user/ibmsecuritysolutions

© Copyright IBM Corporation 2019. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

Statement of Good Security Practices: IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a lawful, comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM does not warrant that any systems, products or services are immune from, or will make your enterprise immune from, the malicious or illegal conduct of any party.

IBM Security

