Implementing SDL at Scale

Secure Development Lifecycle - DevSecOps

Sam Sehgal
Program Leader, SDL automation and DevSecOps

Sep 19, 2022

DELL Technologies

Dell Context

Scale and complexity

By product heterogeneity

By maturity journeys

By numbers

By tech stack & tools

Terraform node Java CI/CD

Pain points

Manual, slow security feedback

Security findings not actionable in CI/CD

Time consuming security assessments

Security control bypassed?

7-point implementation strategy for SDL at scale

1 Solid SDL* foundation

2 Multiple consumption options

3 Customer agnostic architecture

4 Act one team everyday

5 Integrate at LCD**

6 Optimize for DevOps experience

7 Instrument for measurement

*Secure Development Lifecycle
**Lowest common denominator

SDL Overview

OBJECTIVES

SDL
for Scale

Embrace & Adopt.

Measure and progress Apps. against four **Maturity Practice Levels**

Effectively **reduce risk** prior to release (GA)

L1 L3
L2 L4
L5

Invest and train **security champions**

SERVICES

DevOps & SRE — **Fostering Continuous & Frictionless Security**

Design	Development	Testing	Release
Champion Security Engineer Threat Modeling	Static Code Analysis Open Source Component Mgmt. Container Scanning	Web Security Testing Network Vulnerability Scanning	Champion Security Engineer Independent Security Testing SDL Security Assessment

FOUNDATIONS

62+ Controls

One Standard

Compliant

Cutting Edge

Standard

Leading

Four Maturity Practice Levels

L1 Awareness

L2 Knowledge

L3 Knowledge/Skill change

L4 Demonstrate Skill

L5 Demonstrate Skill

Five Levels of Security Training

DELL Technologies

Offer multiple consumption options

2

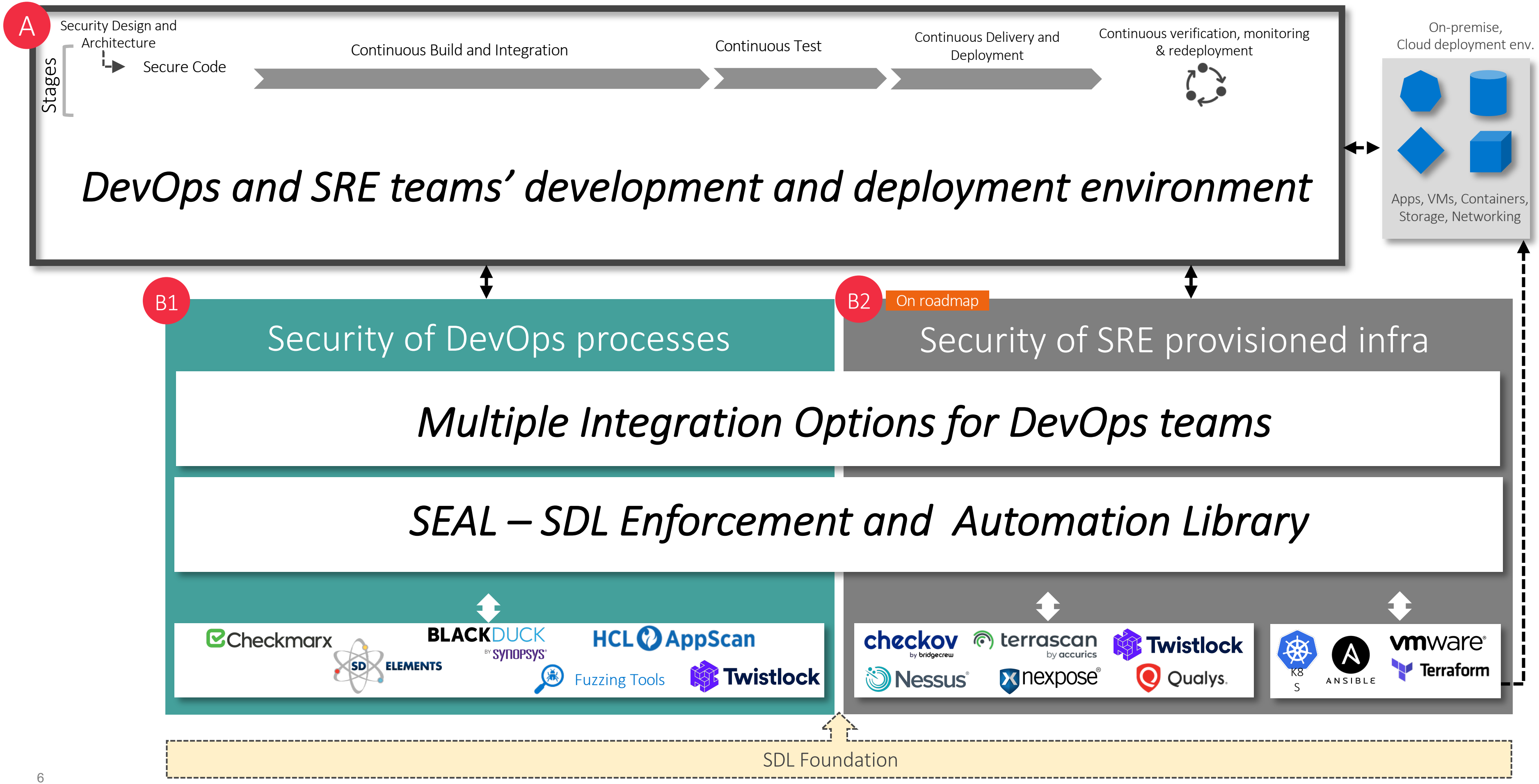
1 SDL Engineer Led	2 SDL as Self Service	3 SDL as API
Led by Security (SDL) Engineer	Led by Security Champion	Performed pragmatically via Git workflows and CI/CD events
Manual	Manual	Automated
For high value applications	For all applications	For apps with frequent & automated deployments
Most comprehensive	Comprehensive	Balanced

Customer agnostic automation architecture

3

A Maintained by Dell DevOps & SRE teams

B Provided by Product and Application Security



A

Stages

Security Design and Architecture

Secure Code

Continuous Build and Integration

Continuous Test

Continuous Delivery and Deployment

Continuous verification, monitoring & redeployment

DevOps and SRE teams’ development and deployment environment

On-premise, Cloud deployment env.

Apps, VMs, Containers, Storage, Networking

B1

Security of DevOps processes

B2

On roadmap

Security of SRE provisioned infra

Multiple Integration Options for DevOps teams

SEAL – SDL Enforcement and Automation Library

Checkmarx, SD ELEMENTS, BLACKDUCK by SYNOPSYS, HCL AppScan, Fuzzing Tools, Twistlock

checkov by bridgecrew, terrascan by accurics, Twistlock, Nessus, nexpose, Qualys

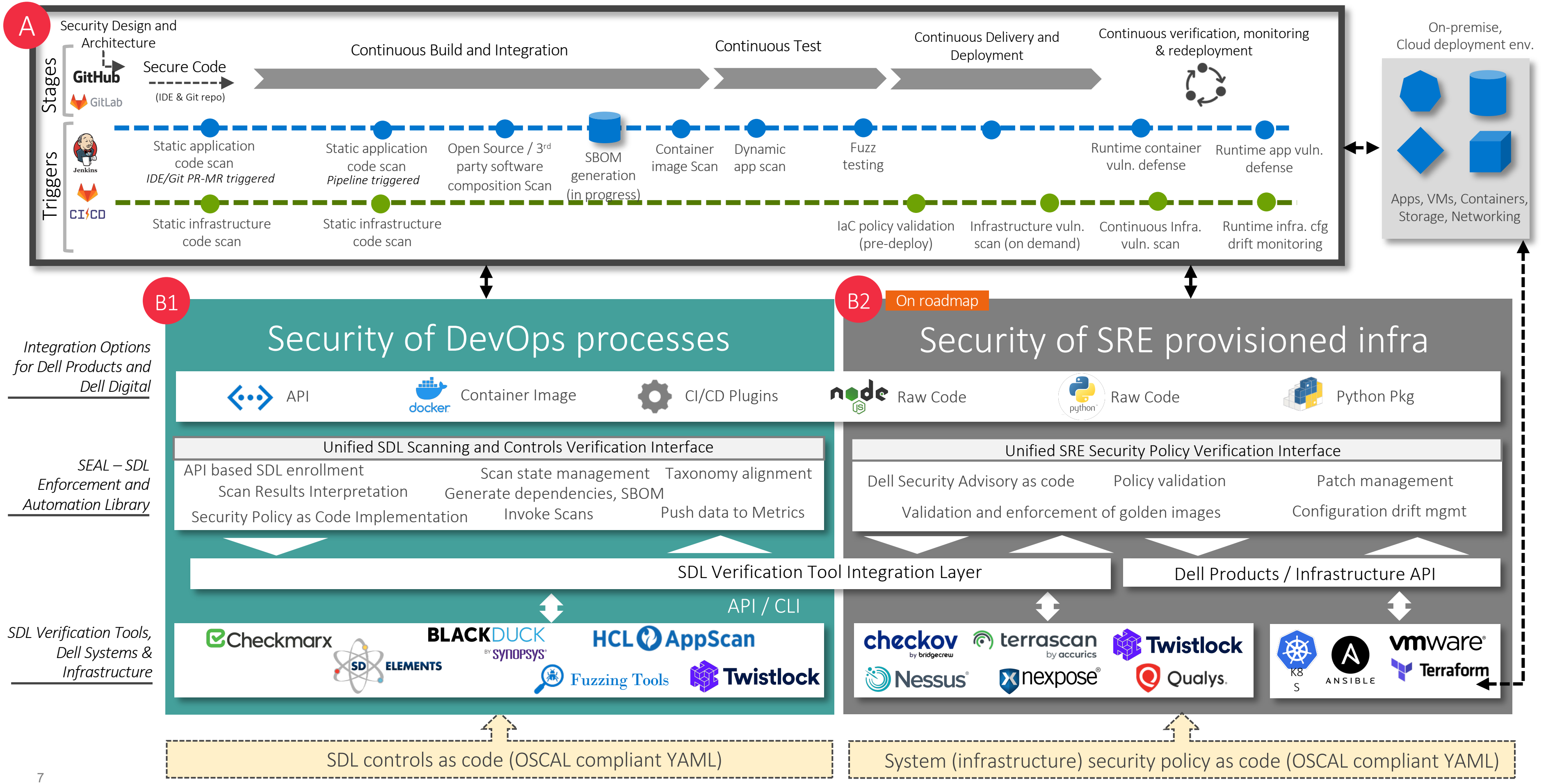
K8S, ANSIBLE, vmware, Terraform

SDL Foundation

Customer agnostic automation architecture 3

A Maintained by Dell DevOps & SRE teams

B Provided by Product and Application Security



A

Stages

Security Design and Architecture

GitHub

GitLab

Secure Code

(IDE & Git repo)

Continuous Build and Integration

Continuous Test

Continuous Delivery and Deployment

Continuous verification, monitoring & redeployment

Triggers

Jenkins

CI/CD

Static application code scan *IDE/Git PR-MR triggered*

Static application code scan *Pipeline triggered*

Open Source / 3rd party software composition Scan

SBOM generation (in progress)

Container image Scan

Dynamic app scan

Fuzz testing

Runtime container vuln. defense

Runtime app vuln. defense

Static infrastructure code scan

Static infrastructure code scan

IaC policy validation (pre-deploy)

Infrastructure vuln. scan (on demand)

Continuous Infra. vuln. scan

Runtime infra. cfg drift monitoring

On-premise, Cloud deployment env.

Apps, VMs, Containers, Storage, Networking

Integration Options for Dell Products and Dell Digital

SEAL – SDL Enforcement and Automation Library

SDL Verification Tools, Dell Systems & Infrastructure

B1

Security of DevOps processes

API

docker Container Image

CI/CD Plugins

Raw Code

Unified SDL Scanning and Controls Verification Interface

API based SDL enrollment

Scan state management

Taxonomy alignment

Scan Results Interpretation

Generate dependencies, SBOM

Security Policy as Code Implementation

Invoke Scans

Push data to Metrics

SDL Verification Tool Integration Layer

API / CLI

Checkmarx

SD ELEMENTS

BLACKDUCK by SYNOPSYS

HCL AppScan

Fuzzing Tools

Twistlock

B2 On roadmap

Security of SRE provisioned infra

python Raw Code

Python Pkg

Unified SRE Security Policy Verification Interface

Dell Security Advisory as code

Policy validation

Patch management

Validation and enforcement of golden images

Configuration drift mgmt

Dell Products / Infrastructure API

checkov by bridgecrew

terrascan by accurics

Twistlock

Nessus

nexpose

Qualys.

K8S

ANSIBLE

vmware

Terraform

SDL controls as code (OSCAL compliant YAML)

System (infrastructure) security policy as code (OSCAL compliant YAML)

Act as one team

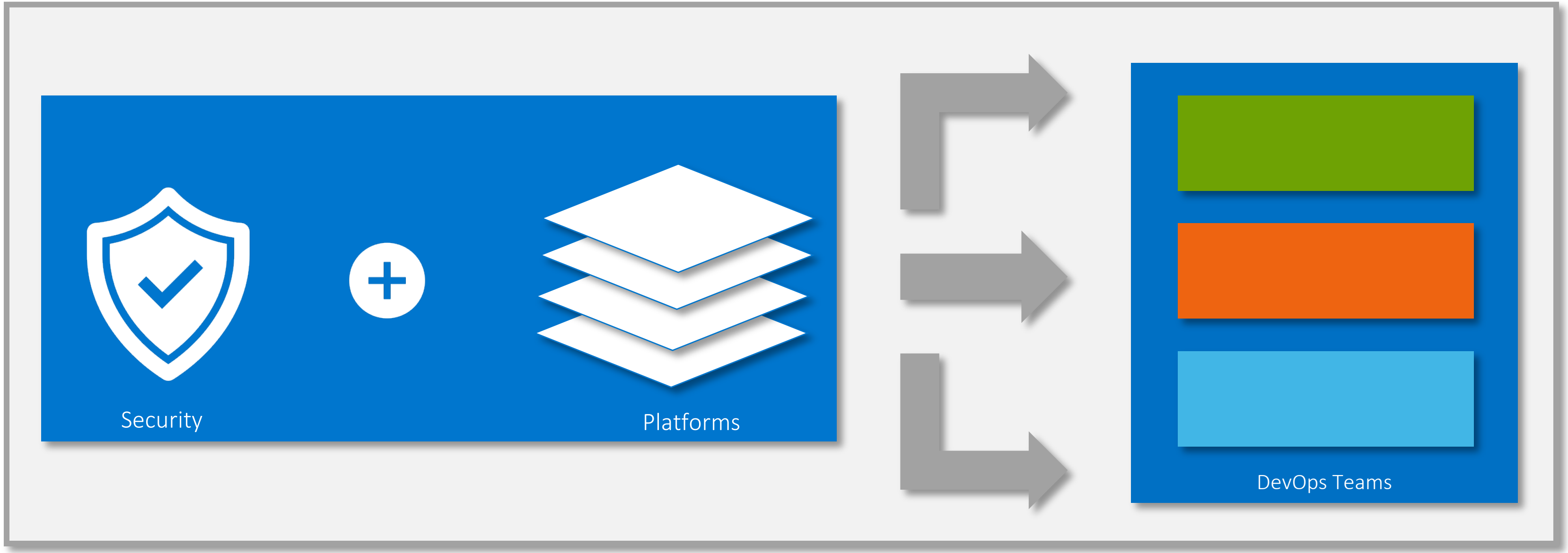
4

- Do not “confront” – Build partnerships instead
- Establish joint scrum teams
- Get into a common backlog
- Resource challenged?
  - Bring security champions to the challenge
  - Reinforce through security awareness and training

DELL Technologies

Integrate at the *Lowest Common Denominator*

5



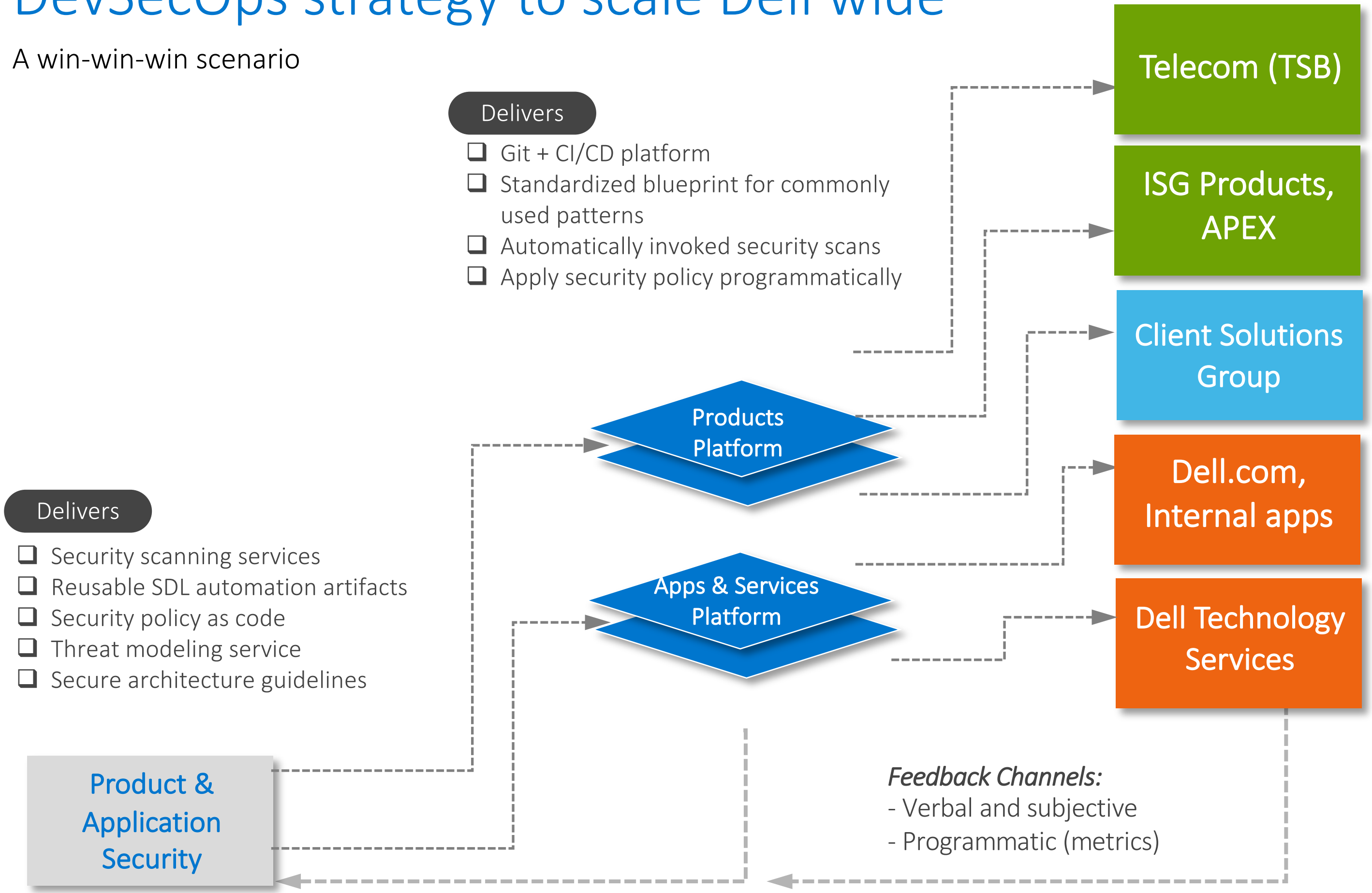
Identify opportunities for platform integration…

…without losing sight of the "downstream" DevOps teams

DevSecOps strategy to scale Dell wide

5

A win-win-win scenario



Delivers

- ❑ Git + CI/CD platform
- ❑ Standardized blueprint for commonly used patterns
- ❑ Automatically invoked security scans
- ❑ Apply security policy programmatically

Delivers

- ❑ Security scanning services
- ❑ Reusable SDL automation artifacts
- ❑ Security policy as code
- ❑ Threat modeling service
- ❑ Secure architecture guidelines

Accomplishes

- ❑ platform with security scan built-in and run automatically
- ❑ Mitigates security findings in code, open-source components, container images & applications

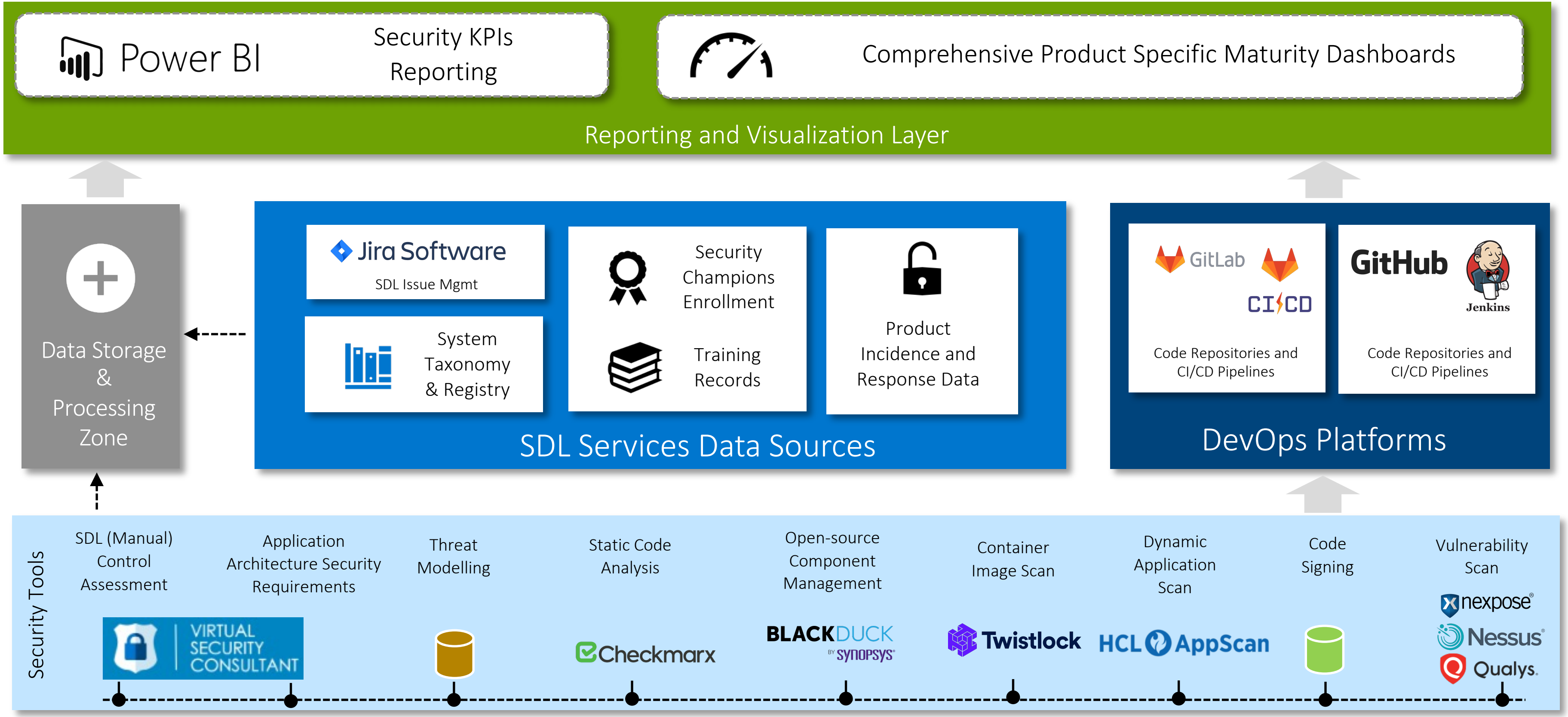
Feedback Channels:
- Verbal and subjective
- Programmatic (metrics)

N factor model for optimization

6

Scope	Be precise about “what” needs to security verified
Branch	Not all branches need same treatment for security verification
Stage	Allocate security activities appropriately among CI/CD/CD
Trigger	Security activity triggered with: Time / Git event / Pipeline event
Frequency	The sweet spot: how often to perform a security activity
Enforcement	Audit mode vs. Strict mode (block the merge/build/pipeline)
Packaging	As package / As container image / As code

Instrument for measurement



Power BI

Security KPIs Reporting

Comprehensive Product Specific Maturity Dashboards

Reporting and Visualization Layer

Data Storage & Processing Zone

Jira Software

SDL Issue Mgmt

System Taxonomy & Registry

Security Champions Enrollment

Training Records

Product Incidence and Response Data

SDL Services Data Sources

GitLab CI/CD

Code Repositories and CI/CD Pipelines

GitHub Jenkins

Code Repositories and CI/CD Pipelines

DevOps Platforms

Security Tools

SDL (Manual) Control Assessment

VIRTUAL SECURITY CONSULTANT

Application Architecture Security Requirements

Threat Modelling

Static Code Analysis

Checkmarx

Open-source Component Management

BLACKDUCK by SYNOPSYS

Container Image Scan

Twistlock

Dynamic Application Scan

HCL AppScan

Code Signing

Vulnerability Scan

nexpose

Nessus

Qualys

DELL Technologies

Thank you

DELL Technologies