

RSAConference2022

San Francisco & Digital | June 6 – 9

SESSION ID: DSO-W01

Security automation for DevOps at the scale of Dell: A real life case study

Sam Sehgal

Program Leader, DevSecOps & SDL Automation
Dell Technologies

TRANSFORM



Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

©2022 RSA Conference LLC or its affiliates. The RSA Conference logo and other trademarks are proprietary. All rights reserved.



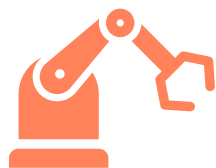
Why this talk, why now?

#RSAC



The problem

- Organizations understand DevSecOps today and plan to (or already are) implement it.
- But... face challenges on how to implement it on a large scale, measure its impact and communicate the outcomes



Dell case study

- Started as small-scale effort. Expanded incrementally on the SDL foundation
- Built with collaboration, experimentation, open mind, and technology
- Now pervasive across nearly all BUs across Dell



Takeaways for you

- Take insights from what worked (and didn't) for us. No need to reinvent the wheel
- Cherry-pick and practice architecture patterns and approach as relevant to your org
- Help you broaden the horizons beyond point solutions and think holistically



Outline of today's talk

1

Dell's scale
and context

2

Problems
driving
requirements

3

Strategy,
execution &
outcomes

4

Your action
plan

RSAConference2022

Let's start with the context

The scale of the problem



By product heterogeneity



Hardware

(personal computers, server, networking, storage...)



Ecommerce

Public facing ecommerce site
(dell.com)



Internal apps

Non-Internet facing internal
apps



APEX

As-a-service offers

By maturity journey

Long release
lifecycles



Agile. Frequent or continuous
deployment using CI/CD

Legacy monolithic
applications



Modern containerized
microservices

Risk tolerance:
High



Risk tolerance:
Low

By numbers



of products and applications to be secured

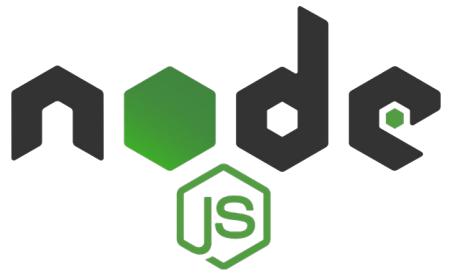
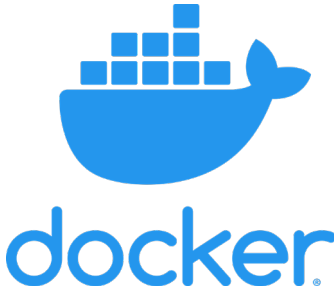


of code repos



of engineers

By tech stack and tools



RSA®Conference2022

Challenges driving requirements

How scale amplifies challenges

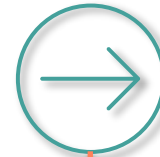


Frustrating dev and ops experience

Feedback to
DevOps on
security take
too long



Security scan
results in PDF,
email and not
consumable in
a pipeline



Handling of
false positives
further
impedes agility



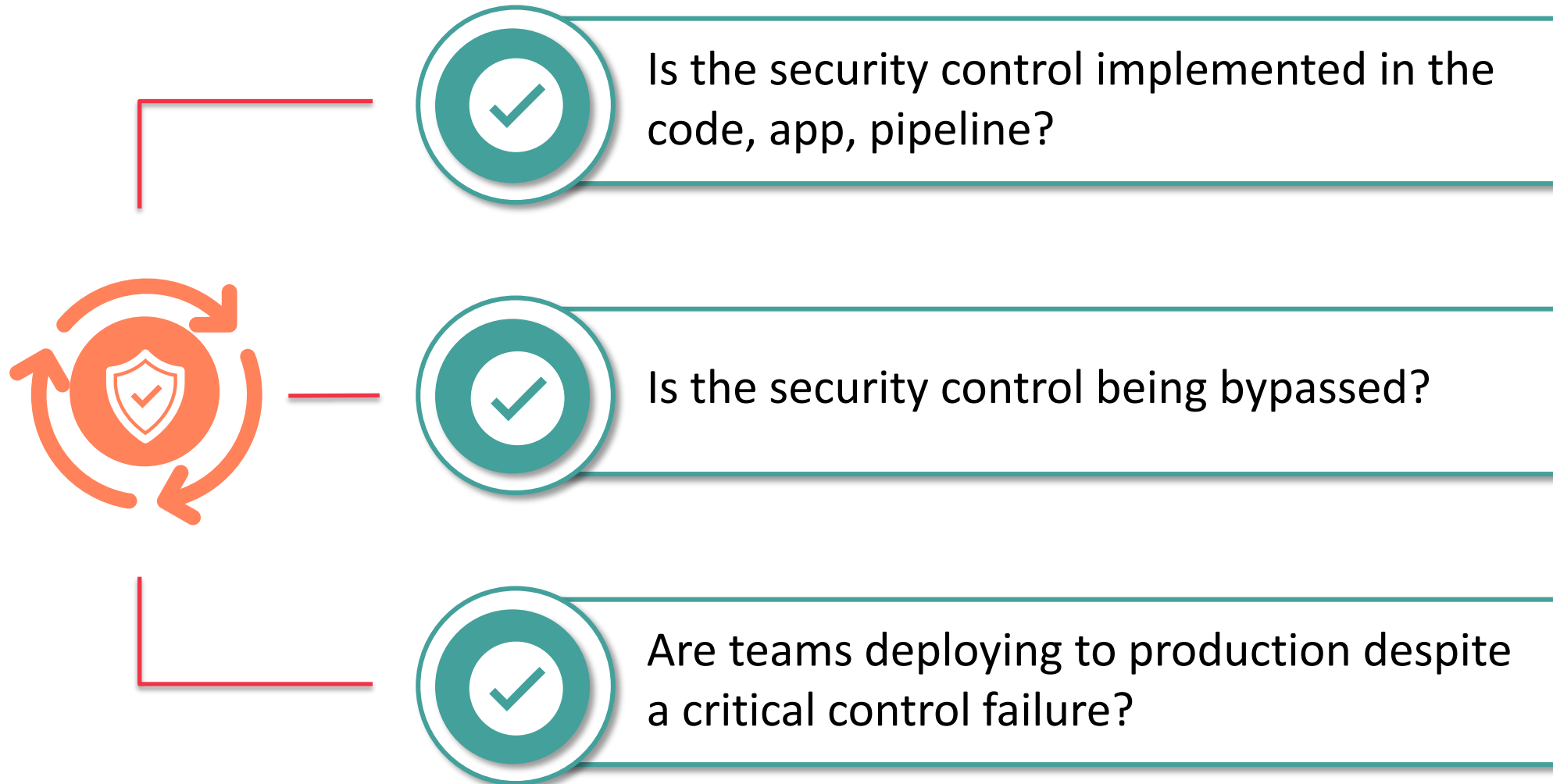
Resource intensive

Manual security assessments are...

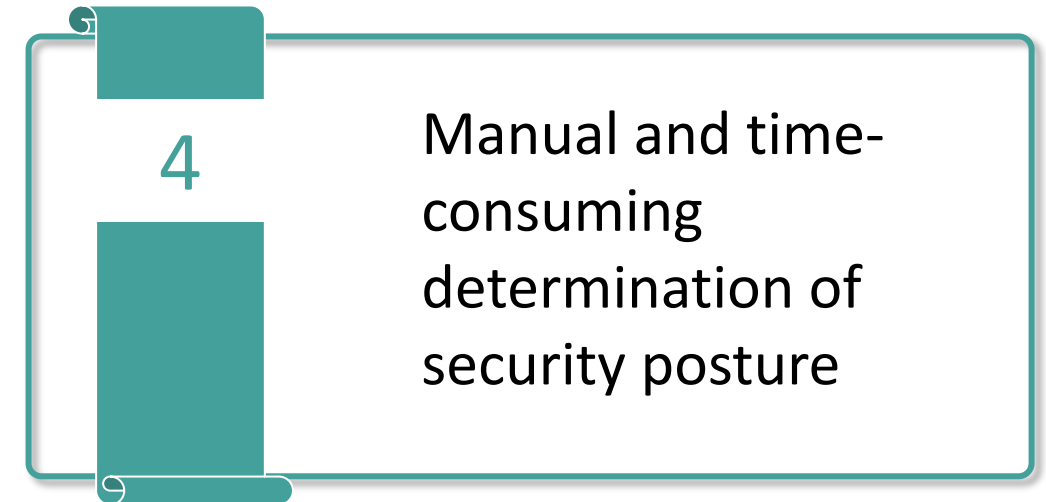
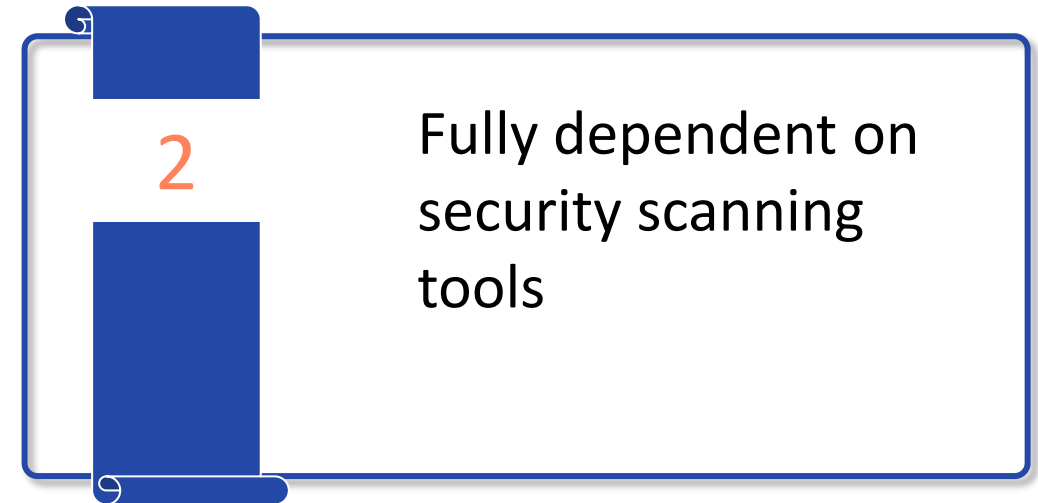
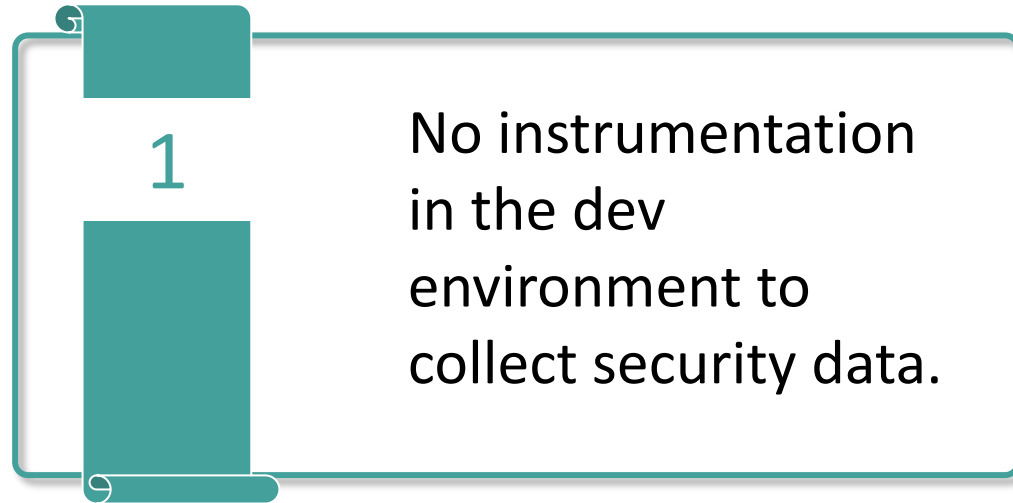


... so need a multi-option approach

Did you do what the security control says?



Limited observability



And...

the challenge of solving all the above at scale

RSA[®]Conference2022

Strategy, implementation & outcomes

How we are solving it and delivering outcomes



7-point implementation strategy

- 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

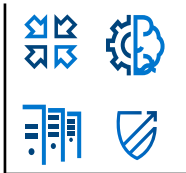
**Lowest common denominator



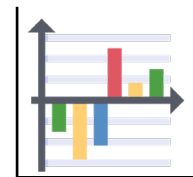
Build on existing SDL foundation

SDL
for Scale

Embrace & Adopt.



Measure and progress Apps. against
four Maturity Practice Levels



Effectively reduce risk prior to
release (GA)



Invest and train security
champions



Champion



Security Engineer

Threat Modeling

DevOps & SRE

Fostering Continuous & Frictionless Security



Static Code Analysis

Open Source Component Mgmt.

Container Scanning

Web Security Testing

Network Vulnerability Scanning



Champion



Security Engineer

Independent Security Testing

SDL Security Assessment

Design

Development

Testing

Release

62+ Controls

One Standard



Compliant



Cutting Edge



Standard



Leading

Four Maturity Practice Levels

L1

Awareness

L3

Knowledge/Skill change

L2

Knowledge

L4

Demonstrate Skill

L5

Demonstrate Skill

Five Levels of Security Training

Offer multiple consumption options

1 SDL Engineer Led



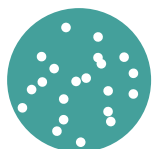
Led by Security (SDL) Engineer



Manual



For high value applications



Most comprehensive

2 SDL as Self Service



Led by Security Champion



Manual

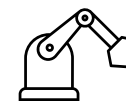


For all applications



Comprehensive

3 SDL as API



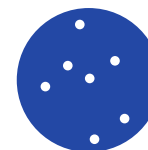
Performed pragmatically via Git workflows and CI/CD events



Automated

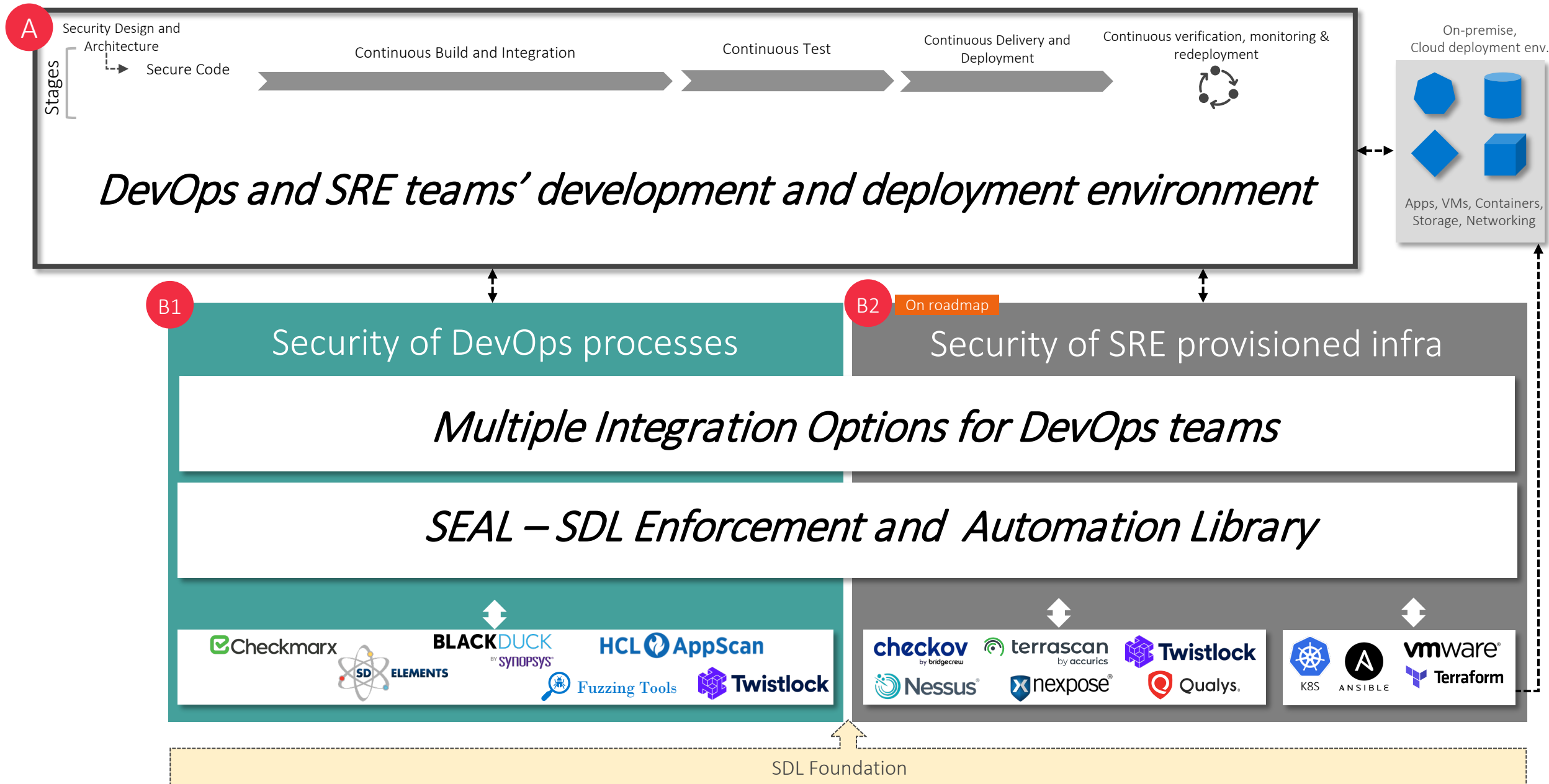


For apps with frequent & automated deployments



Balanced

Customer agnostic automation architecture



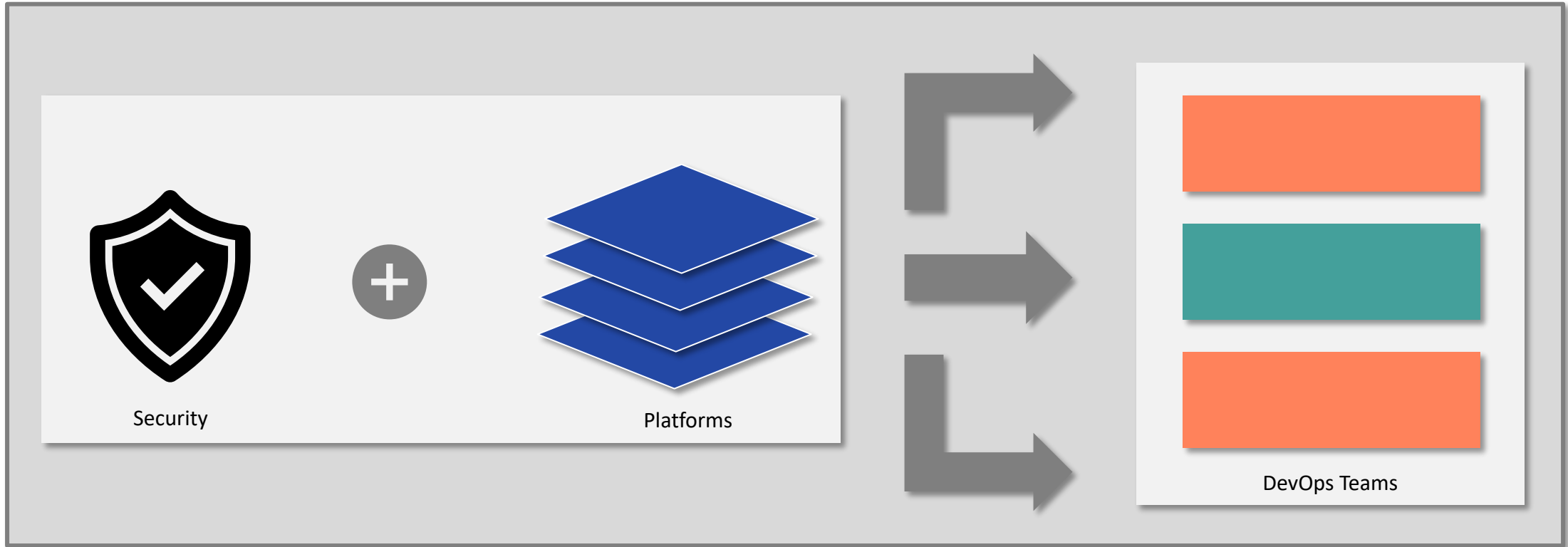


Act as one team everyday

- 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



Integrate at the *Lowest Common Denominator*

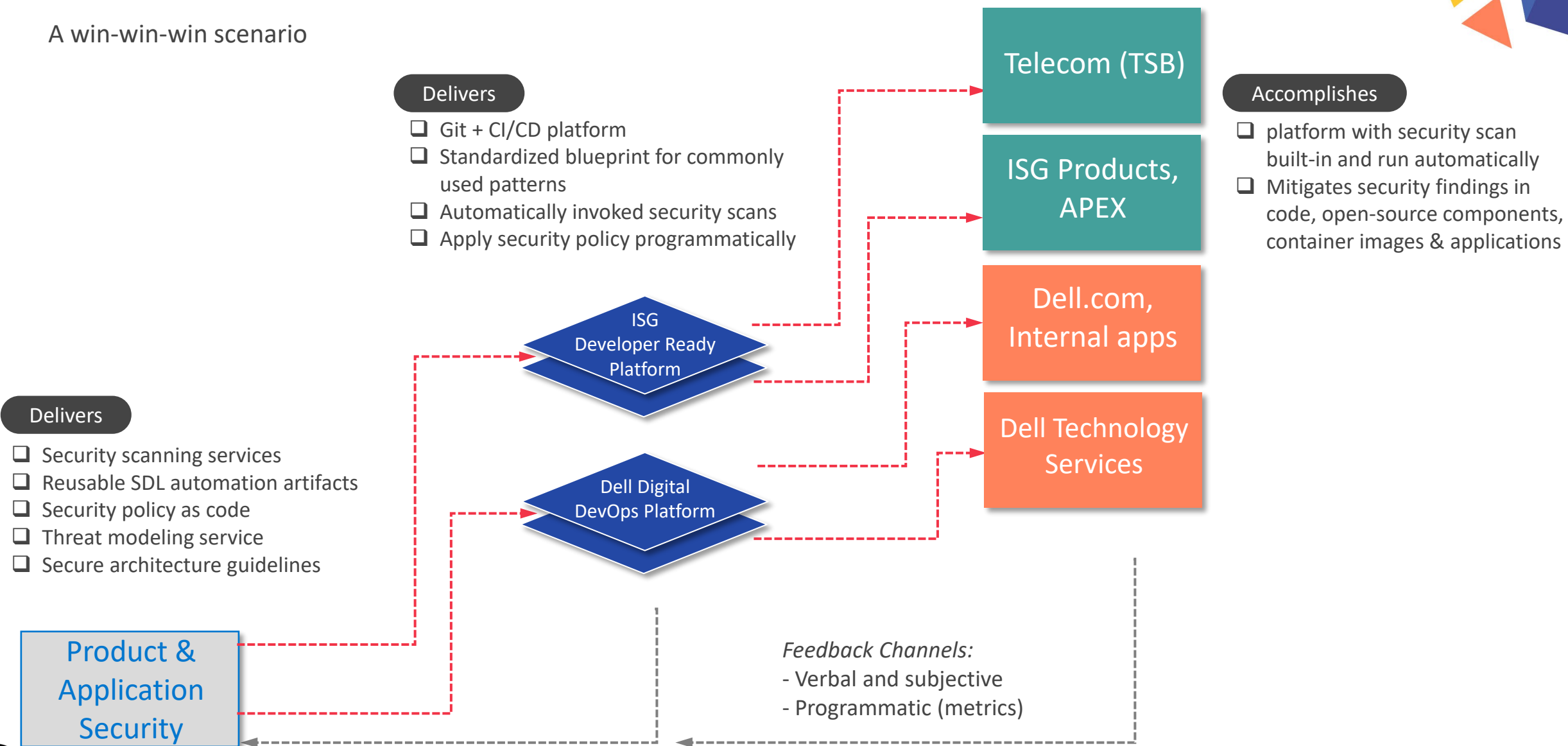


Identify opportunities for platform integration...

...without losing sight of the "downstream" DevOps teams

Our lowest common denominators

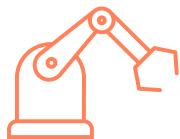
A win-win-win scenario



Optimize for developer experience



Embedding a security activity in not enough



Integration must be optimized to reduce friction



N factor model for optimization

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

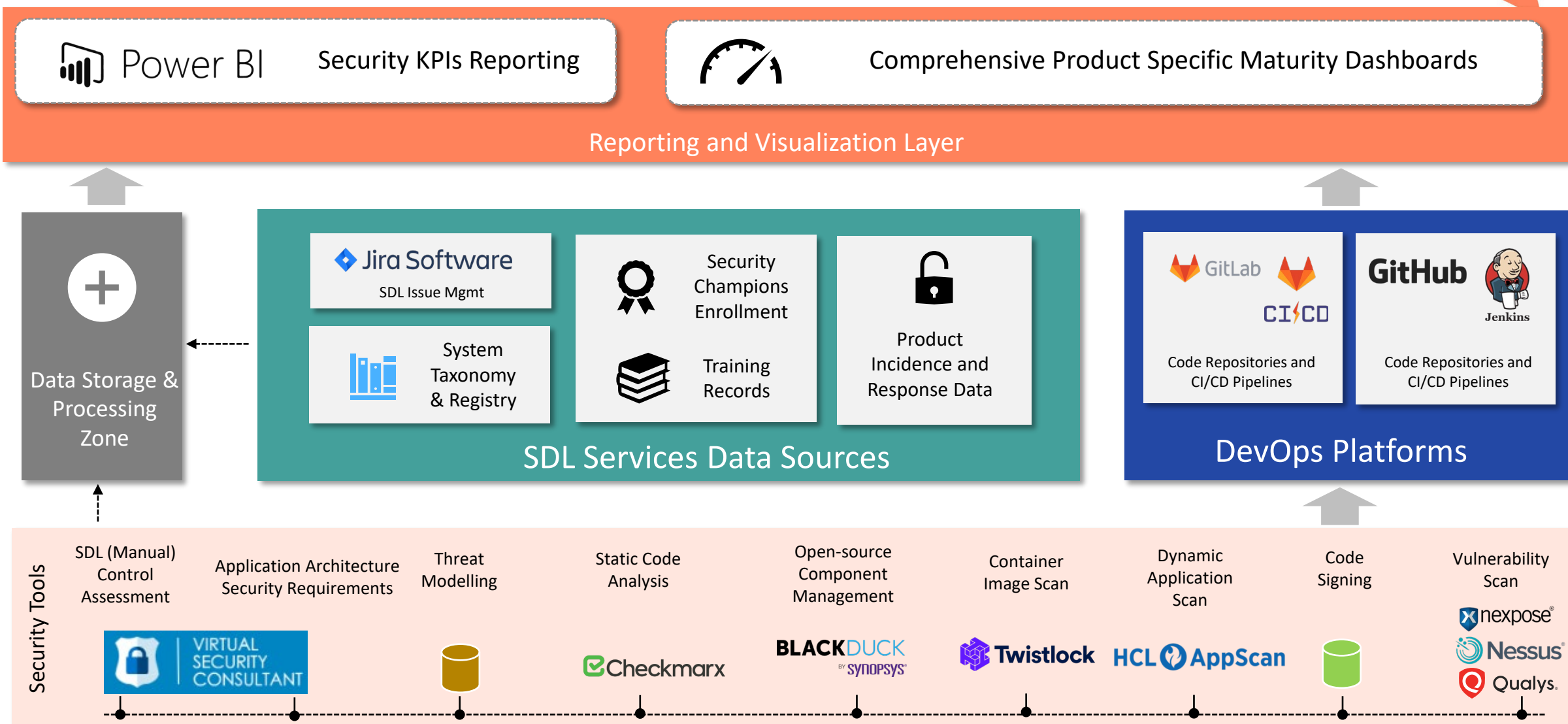
Enforcement

Audit mode vs. Strict mode (block the merge/build/pipeline)

Packaging

As package / As container image / As code

Instrument for measurement



RSA[®]Conference2022

Outcomes and path forward

We are not done yet



Outcomes

Frictionless Experience

of places to get security feedback

1

Speed of getting security feedback

5 days →
5 min

Adoption at Scale

of BUs onboarded

> 90%

of products enabled

> 1000

of projects/repos enabled

> 10,000

Security Effectiveness

of products/apps with security issues mitigated before deployment

> 600

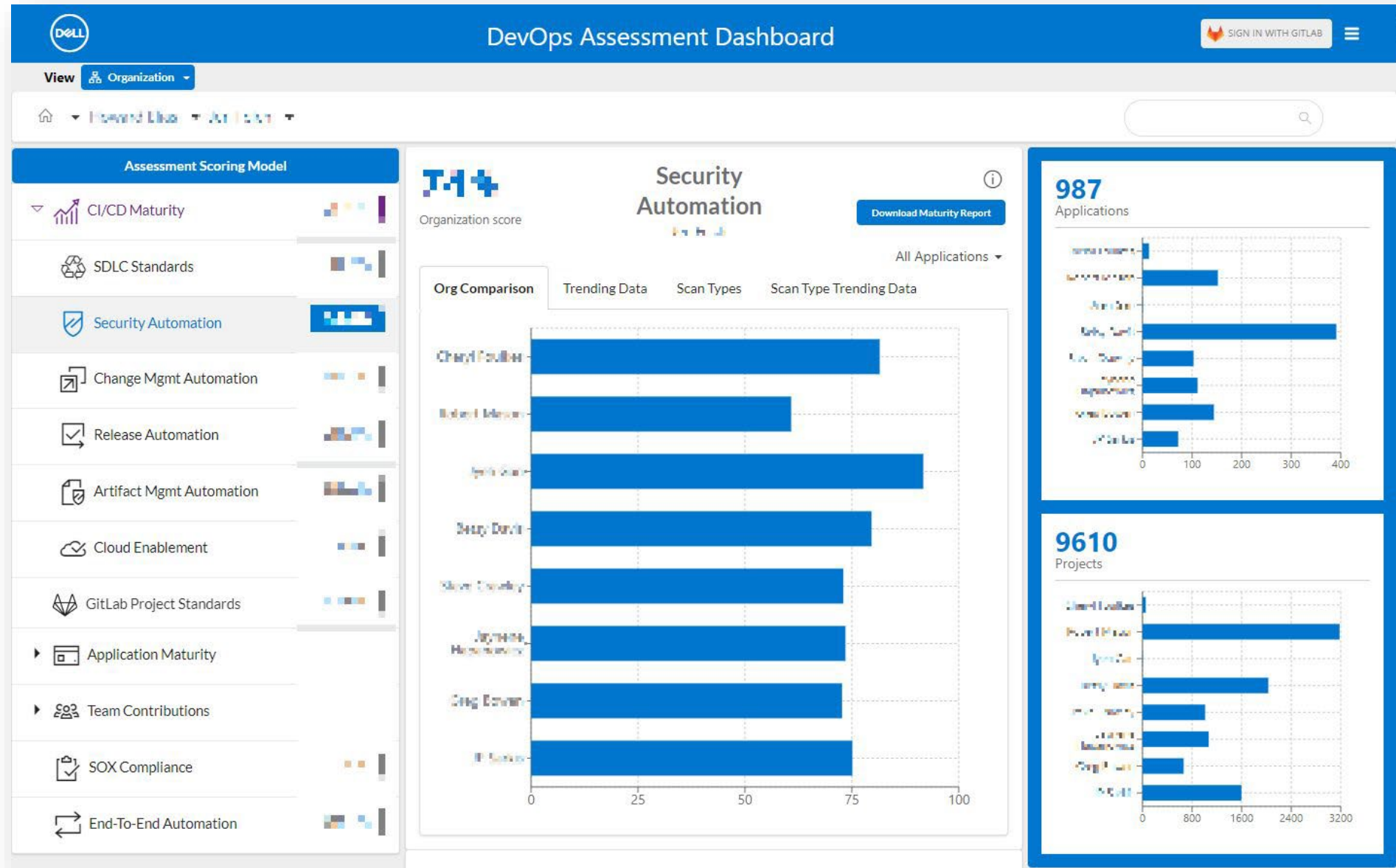
of critical/high security issues discovered and mitigated

> 400

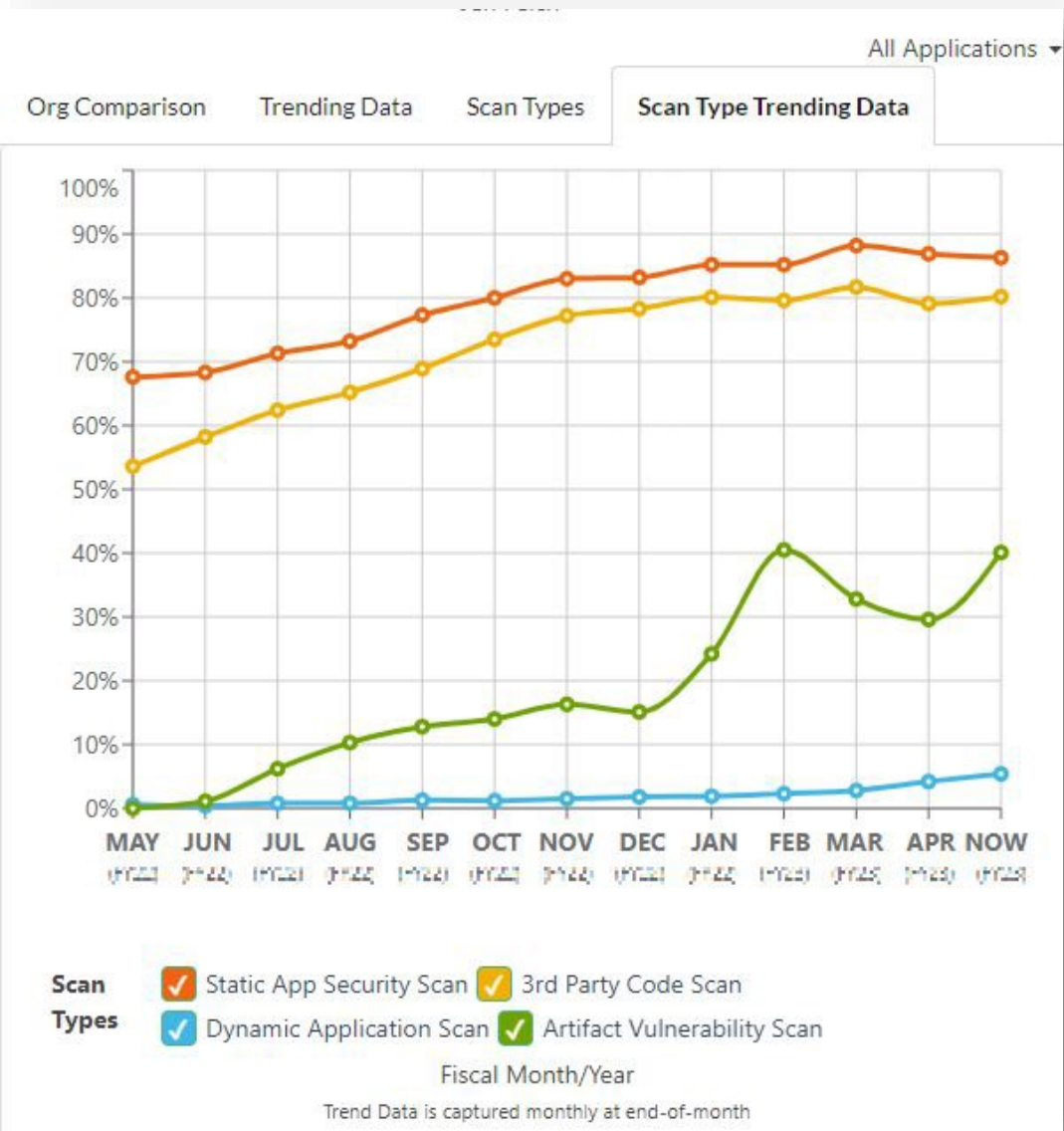


Outcomes

#RSAC



Outcomes



Build & Verify Non-Prod Deploy & Validation Scan History

TIP: Click on each Dimension Feature Item in the list below to see the validation's purpose, criteria, validation method, and support details!

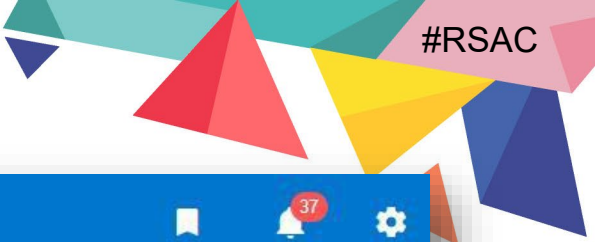
Dimension Feature Item Name	Validation	Pipeline Id
Static Application Security Testing	✓	9797571
3rd Party Code Scan	✓	9797571
Artifact Vulnerability Scan	✓	9797571

Build & Verify Non-Prod Deploy & Validation Scan History

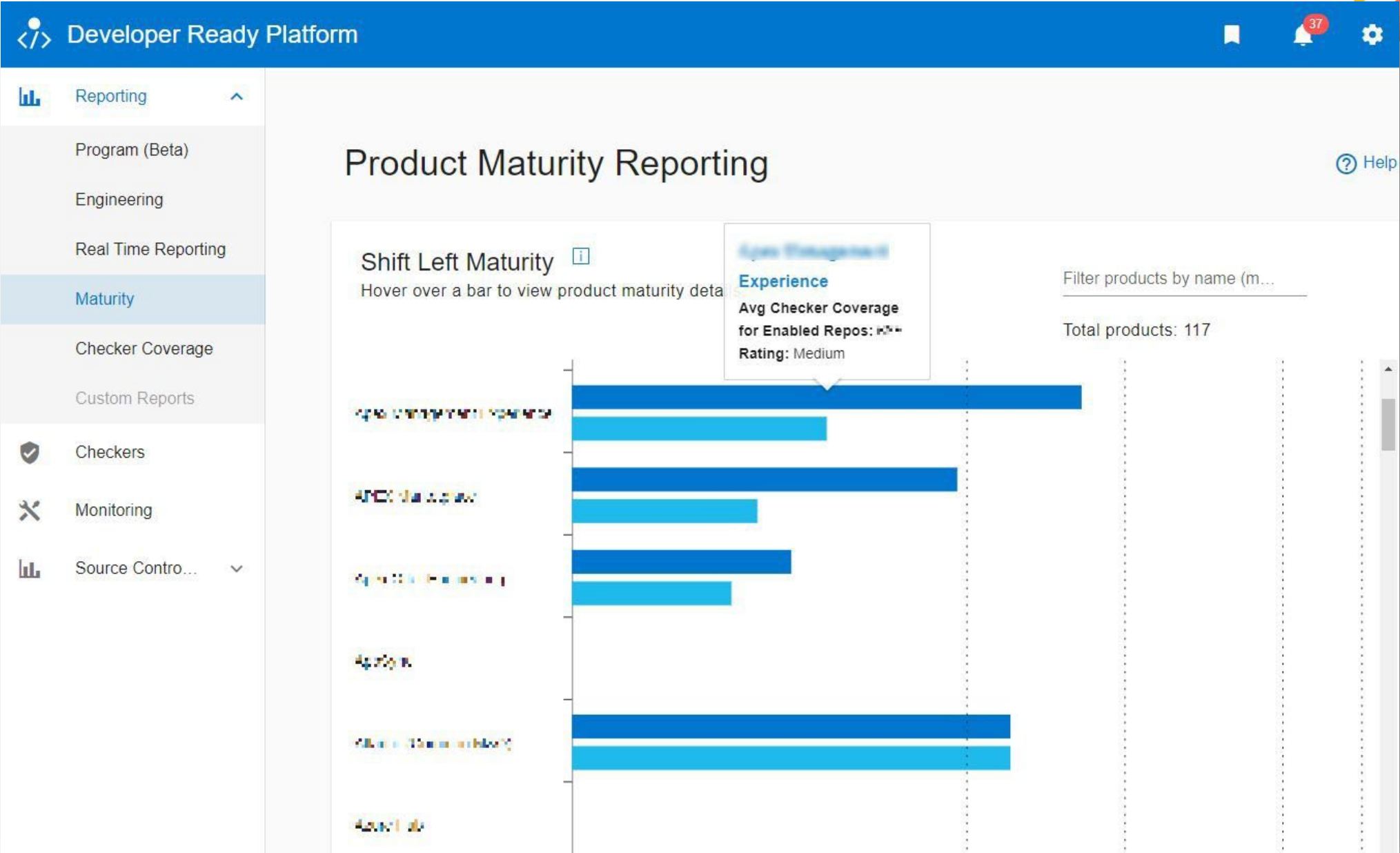
Security Scan Type	Scan run in last		
	30 Days	180 Days	12 Months
Static Application Security Testing	✓	✓	✓
3rd Party Code Scan	✓	✓	✓
Artifact Vulnerability Scan	✓	✓	✓
Dynamic Application Scan	✓	✓	✓



Outcomes

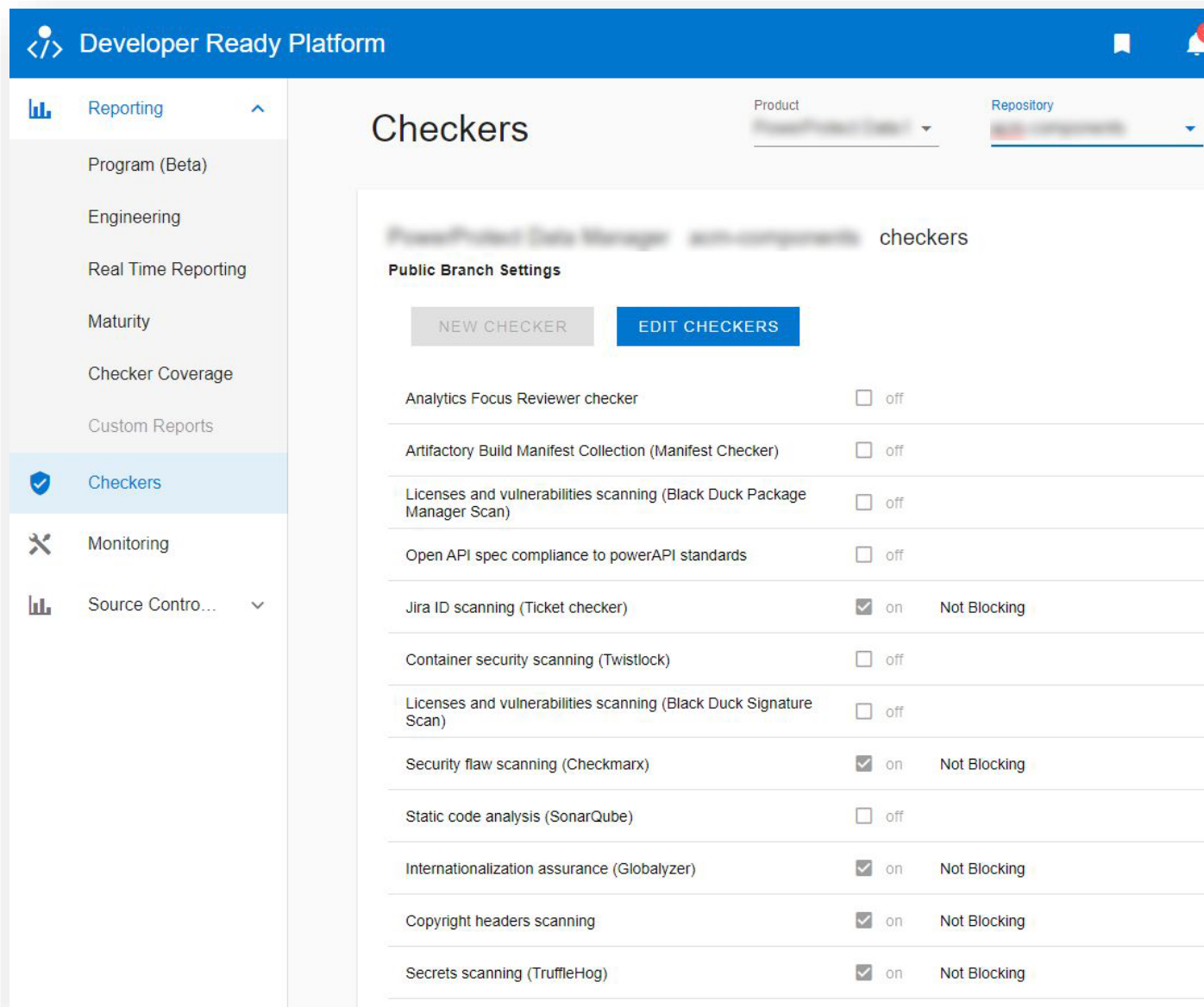


#RSAC



Outcomes

#RSAC



The screenshot shows the 'Developer Ready Platform' interface. The left sidebar contains navigation links: Reporting, Program (Beta), Engineering, Real Time Reporting, Maturity, Checker Coverage, Custom Reports, Checkers (selected), Monitoring, and Source Contro... The main content area is titled 'Checkers' and displays a table of various security and compliance checks. The table has columns for the check name, its status (on/off), and a 'Not Blocking' indicator. The checks listed include Analytics Focus Reviewer checker, Artifactory Build Manifest Collection (Manifest Checker), Licenses and vulnerabilities scanning (Black Duck Package Manager Scan), Open API spec compliance to powerAPI standards, Jira ID scanning (Ticket checker), Container security scanning (Twistlock), Licenses and vulnerabilities scanning (Black Duck Signature Scan), Security flaw scanning (Checkmarx), Static code analysis (SonarQube), Internationalization assurance (Globalyzer), Copyright headers scanning, and Secrets scanning (TruffleHog).

Check Name	Status	Not Blocking
Analytics Focus Reviewer checker	off	
Artifactory Build Manifest Collection (Manifest Checker)	off	
Licenses and vulnerabilities scanning (Black Duck Package Manager Scan)	off	
Open API spec compliance to powerAPI standards	off	
Jira ID scanning (Ticket checker)	on	Not Blocking
Container security scanning (Twistlock)	off	
Licenses and vulnerabilities scanning (Black Duck Signature Scan)	off	
Security flaw scanning (Checkmarx)	on	Not Blocking
Static code analysis (SonarQube)	off	
Internationalization assurance (Globalyzer)	on	Not Blocking
Copyright headers scanning	on	Not Blocking
Secrets scanning (TruffleHog)	on	Not Blocking



Reimagining the future

Rapid Response Support Model

Conversational support via a chatbot. Instantaneous response to questions/issues

Security Control Verification

Expand beyond security testing automation
Enable DevOps to verify all (qualified) SDL controls with minimal manual effort

SBOM & EO enablement

Automated generation of software bill of material and recorded in source of truth

Secure Development Lifecycle

Frictionless enrollment to SDL tools

Slash enrollment time from days to seconds →
Superior customer experience
Near-zero human intervention → Focus instead on high impact tasks

Seamless app security testing

Security feedback time reduced from days to minutes
In context feedback to developers

Seamless IaC and NW security testing

Enable SRE and Ops team to secure IaC code
Automated network and systems scanning

Legend

- ★ Available
- ★ Ideation
- ★ Active new initiatives

RSA[®]Conference2022

Your action plan

For implementing or scaling DevSecOps



Assess and measure

#RSAC

30 days



- Time taken by DevOps/SRE teams to get security feedback: minutes/days/weeks
- Time and approach taken by Security org to get security posture
 - Security issues discovered, reported, fixed
 - Days of open before remediation
- % of security bugs slipped through SDL and ended up in production
 - Worst case: caught by customers or external entities
- SDLC stages where are security and DevOps interactions happen: (early / mid /later)

3 to 6 months



- If offering only manual SDL assessment: pilot with self-service option
- If already offering self-service SDL assessment
 - Start proof of concept with a DevOps team. Ideally with a platform team
- Embed a security engineer or a security champion in the development team's scrum process

Retrospective and scale

#RSAC

1 year



- Remeasure and compare with metrics from “Assess” stage
- Conduct joint retrospective sessions with development and security teams
 - What worked or didn’t work
- Plan for scaling across larger number of teams across organization

RSA[®]Conference2022

Questions

