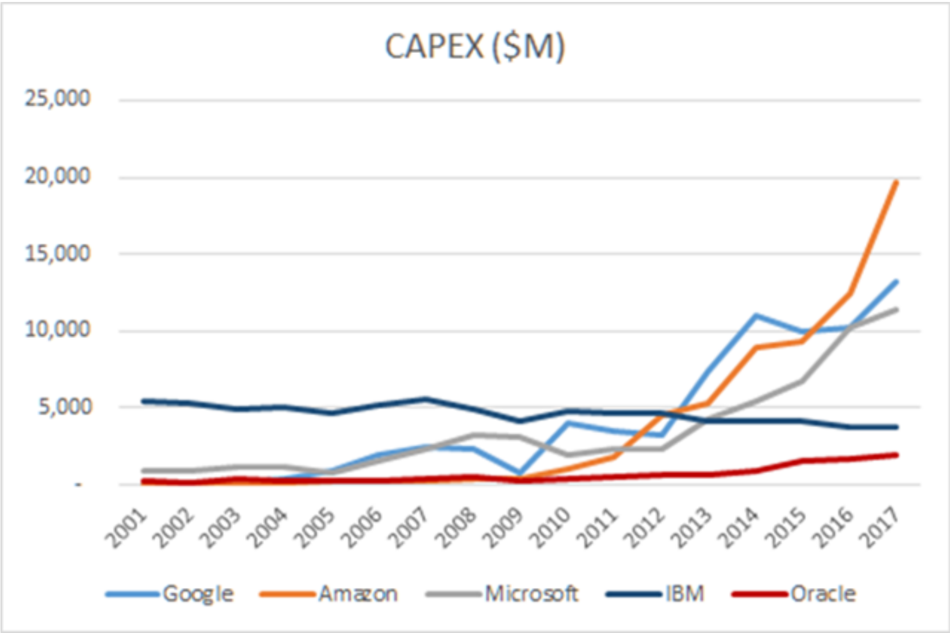
The Role of the Hosting Platform in Facilitating Computing and Network Innovation

Reigning in Complexity

Dennis R Moreau, PhD
Cybersecurity Information Architecture
VMware, Office of the CTO
dmoreau@vmware.com

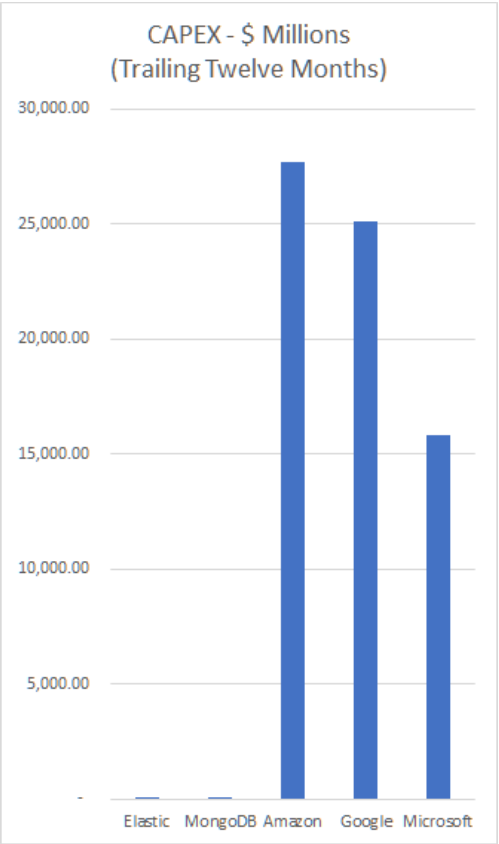
Hyper-scalers by CAPEX: Unique Scale, Distribution, Scope …



https://www.platformonomics.com/2019/03/follow-the-capex-commercial-open-source-vs-the-cloud/

AWS Launched over 1800 significant services and features in 2018

https://www.forbes.com/sites/siliconangle/2018/11/27/how-andy-jassy-ceo-of-aws-thinks-the-future-of-cloud-computing/#4efc8fd17730



DB 2019

Cloud Outages 2019

Complexity THE Cloud Management Problem

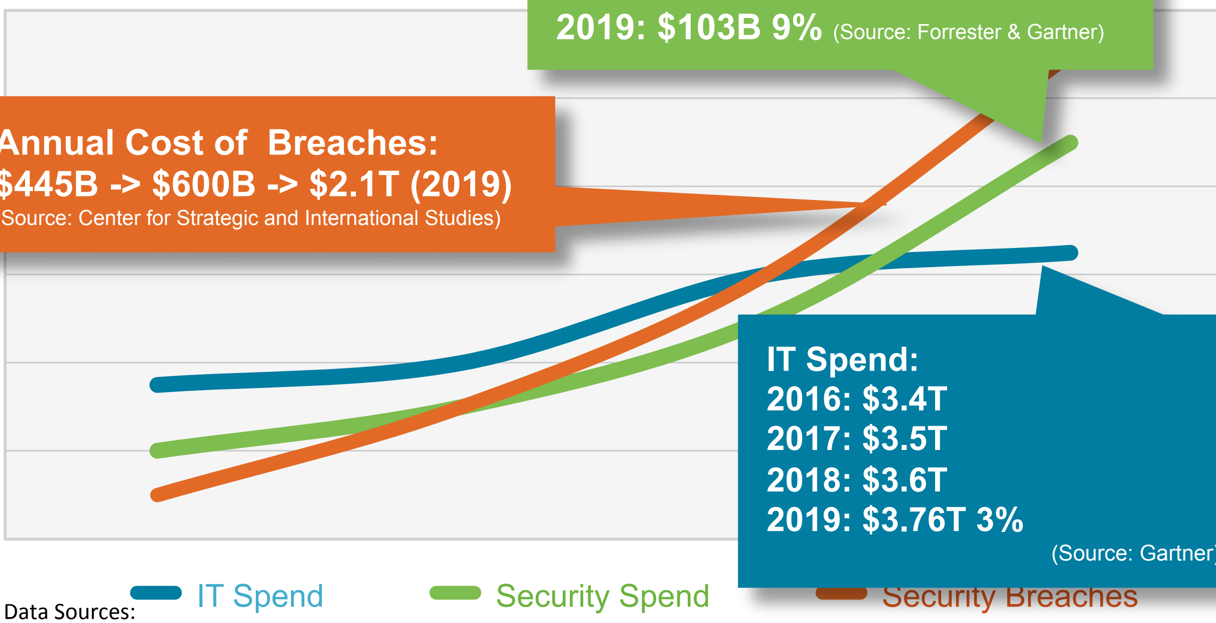
- March 13: Facebook
  - Cause: Server Configuration Change
- June 2: Google Cloud Platform
  - Cause: Routine Configuration Change (wrong servers)
- June 24: Version
  - Cause: BGP Routing Leak
- July 2: Cloudflare
  - Cause: Bad Software Deployment
- July 3-4: Facebook, Twitter, Apple
  - Cause: Routine Maintenance Operation
- July 11: Twitter
  - Cause: Inconsistent internal System Change
- August 31: AWS
  - Cause: Server Resilience/Recovery Misconfiguration
- *March 23: AWS – Capital One
  - Cause: Firewall Mis-configuration

http://techgenix.com/2019-website-outages/

Driven by urgency and complexity … we are thrashing
Example: Security

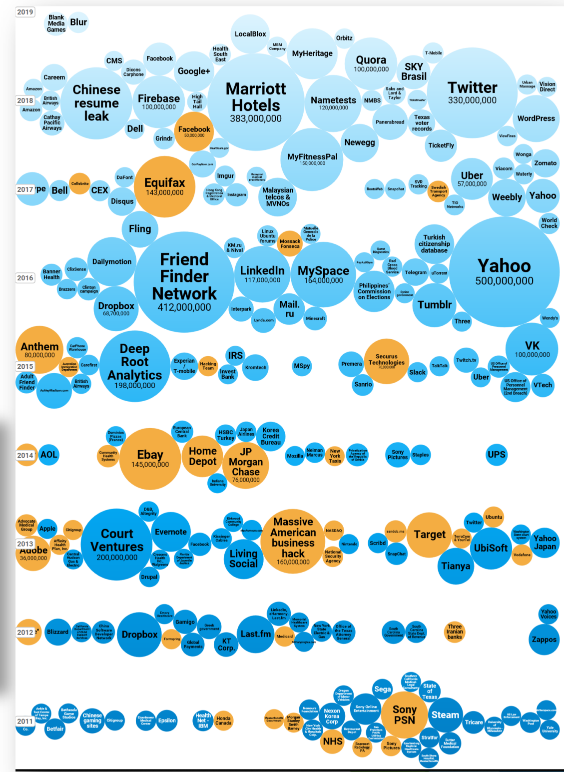
Security Spend:
2016: $83B
2017: $90B
2018: $96B
2019: $103B 9% (Source: Forrester & Gartner)

Annual Cost of Breaches:
$445B -> $600B -> $2.1T (2019)
(Source: Center for Strategic and International Studies)



IT Spend:
2016: $3.4T
2017: $3.5T
2018: $3.6T
2019: $3.76T 3%
(Source: Gartner)

Breaches 2011-2019



https://informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

Data Sources:
Identity Theft Resource Center: https://www.idtheftcenter.org
DataBreaches.Net: https://www.databreaches.net/
Visualization Source:
Information Is Beautiful: http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

Increasing complexity

Separability??.

External Complexity from the problem space… on clients and in DCs

Initial Access
- Valid Accounts
- Trusted Relationship
- Supply Chain Compromise
- Spearphishing via Service
- Spearphishing Link
- Spearphishing Attachment
- Replication Through Removable Media
- Exploit Public-Facing Application
- Hardware Additions
- Drive-by Compromise

Execution
- Scheduled Task
- Trap
- LSASS Driver
- Local Job Scheduling
- Launchctl
- XSL Script Processing
- Windows Remote Management
- User Execution
- Trusted Developer Utilities
- Third-party Software
- Space after Filename
- Source
- Signed Script Proxy Execution
- Service Execution
- Scripting
- Rundll32
- Regsvr32
- Regsvcs/Regasm
- PowerShell
- Mshta
- InstallUtil
- Graphical User Interface
- Exploitation for Client Execution
- Execution through API
- Dynamic Data Exchange
- Control Panel Items

Persistence
- Valid Accounts
- Plist Modification
- Image File Execution Options Injection
- DLL Search Order Hijacking
- Web Shell
- Startup Items
- Setuid and Setgid
- Service Registry Permissions Weakness
- Port Monitors
- Path Interception
- New Service
- Launch Daemon
- Hooking
- File System Permissions Weakness
- Dylib Hijacking
- Application Shimming
- AppInit DLLs
- AppCert DLLs
- Accessibility Features
- Winlogon Helper DLL
- Windows Management Instrumentation Event Subscription
- SIP and Trust Provider

Privilege Escalation
- Process Injection
- Extra Window Memory Injection
- Bypass User Account Control
- Access Token Manipulation
- Sudo Caching
- Sudo
- SID-History Injection
- Exploitation for Privilege Escalation

Defense Evasion
- XSL Script Processing
- Web Service
- Trusted Developer Utilities
- Timestomp
- Template Injection
- Space after Filename
- Software Packing
- SIP and Trust Provider Hijacking
- Signed Binary Proxy Execution
- Rundll32
- Rootkit
- Regsvr32
- Regsvcs/Regasm
- Redundant Access
- Process Hollowing
- Process Doppelganging
- Port Knocking
- Obfuscated Files or Information
- Indicator Removal on Host
- DLL Side-Loading
- DCShadow

CredentialAccess
- Network Sniffing
- Two-Factor Authentication Interception
- Private Keys
- Password Filter DLL
- LLMNR/NBT-NS Poisoning
- Keychain
- Kerberoasting
- Input Prompt
- Input Capture
- Hooking
- Forced Authentication
- Exploitation for Credential Access
- Credentials in Files
- Credential Dumping
- Brute Force
- Bash History
- Account Manipulation
- Securityd Memory
- Credentials in Registry

Discovery
- System Time Discovery
- System Service Discovery
- System Owner/User Discovery
- System Network Configuration Discovery
- Security Software Discovery
- Remote System Discovery
- Query Registry
- Process Discovery
- Permission Groups Discovery
- Peripheral Device Discovery
- Password Policy Discovery
- Network Share Discovery
- Network Service Scanning
- File and Directory Discovery
- Browser Bookmark Discovery
- Application Window Discovery
- System Network Connections Discovery
- System Information Discovery
- Account Discovery

Lateral Movement
- Windows Remote Management
- Third-party Software
- Taint Shared Content
- SSH Hijacking
- Shared Webroot
- Replication Through Removable Media
- Remote File Copy
- Remote Desktop Protocol
- Pass the Ticket
- Pass the Hash
- Logon Scripts
- Exploitation of Remote Services
- Application Deployment Software
- Windows Admin Shares
- Remote Services
- Distributed Component Object Model
- AppleScript

Collection
- Video Capture
- Screen Capture
- Man in the Browser
- Input Capture
- Email Collection
- Data Staged
- Data from Removable Media
- Data from Network Shared Drive
- Data from Information Repositories
- Automated Collection
- Audio Capture
- Data from Local System
- Clipboard Data

Exfiltration
- Scheduled Transfer
- Exfiltration Over Physical Medium
- Exfiltration Over Command and Control Channel
- Data Transfer Size Limits
- Data Encrypted
- Data Compressed
- Automated Exfiltration
- Exfiltration Over Other Network Medium
- Exfiltration Over Alternative Protocol

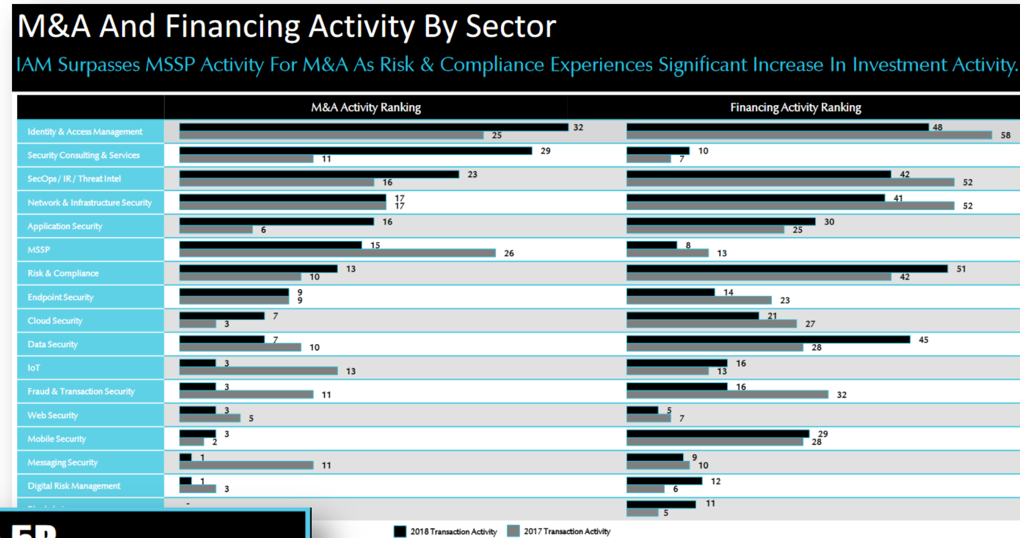
Command and Control
- Web Service
- Uncommonly Used Port
- Standard Non-Application Layer Protocol
- Standard Application Layer Protocol
- Remote Access Tools
- Port Knocking
- Multilayer Encryption
- Multiband Communication
- Multi-Stage Channels
- Multi-hop Proxy
- Fallback Channels
- Domain Fronting
- Data Obfuscation
- Data Encoding
- Custom Cryptographic Protocol
- Connection Proxy
- Communication Through Removable Media
- Standard Cryptographic Protocol
- Remote File Copy
- Custom Command and Control Protocol
- Commonly Used Port

Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Execution	Collection	Exfiltration	Command and Control
Accessibility Features	Accessibility Features	Binary Padding	Brute Force	Account Discovery	Application Deployment Software	Command-Line	Automated Collection	Automated Exfiltration	Commonly Used Port
AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Application Window Discovery	Exploitation of Vulnerability	Execution through API	Clipboard Data	Data Compressed	Communication Through Removable Media
Basic Input/Output System	Bypass User Account Control	Code Signing	Credential Manipulation	File and Directory Discovery	Logon Scripts	Graphical User Interface	Data Staged	Data Encrypted	Custom Command and Control Protocol
Bootkit	DLL Injection	Component Firmware	Credentials in Files	Local Network Configuration Discovery	Pass the Hash	PowerShell	Data from Local System	Data Transfer Size Limits	Custom Cryptographic Protocol
Change Default File Handlers	DLL Search Order Hijacking	DLL Injection	Exploitation of Vulnerability	Local Network Connections Discovery	Pass the Ticket	Process Hollowing	Data from Network Shared Drive	Exfiltration Over Alternative Protocol	Data Obfuscation
Component Firmware	Exploitation of Vulnerability	DLL Search Order Hijacking	Input Capture	Network Service Scanning	Remote Desktop Protocol	Rundll32	Data from Removable Media	Exfiltration Over Command and Control Channel	Fallback Channels
DLL Search Order Hijacking	Legitimate Credentials	DLL Side-Loading	Network Sniffing	Peripheral Device Discovery	Remote File Copy	Scheduled Task	Email Collection	Exfiltration Over Other Network Medium	Multi-Stage Channels
Hypervisor	Local Port Monitor	Disabling Security Tools	Two-Factor Authentication Interception	Permission Groups Discovery	Remote Services	Service Execution	Input Capture	Exfiltration Over Physical Medium	Multiband Communication
Legitimate Credentials	New Service	Exploitation of Vulnerability		Process Discovery	Replication Through Removable Media	Third-party Software	Screen Capture	Scheduled Transfer	Multilayer Encryption
						Windows Management			

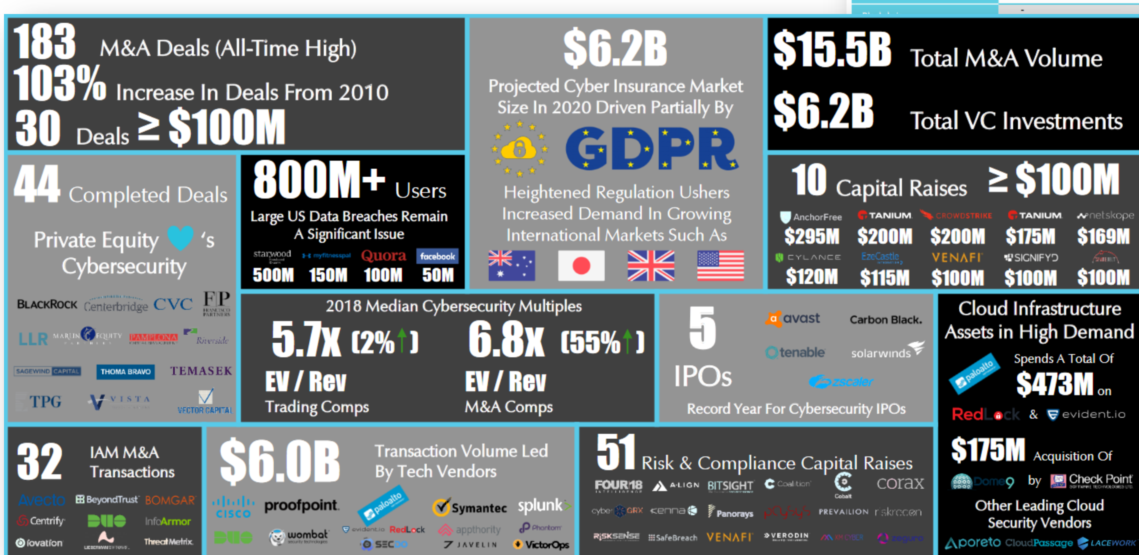
MITRE

https://attack.mitre.org/docs/MITRE_ATTACK_Enterprise_11x17.pdf

Innovation:
Security is hot, motivated by the complexity and intensity of the problem.



https://www.cbronline.com/news/cybersecurity-ma

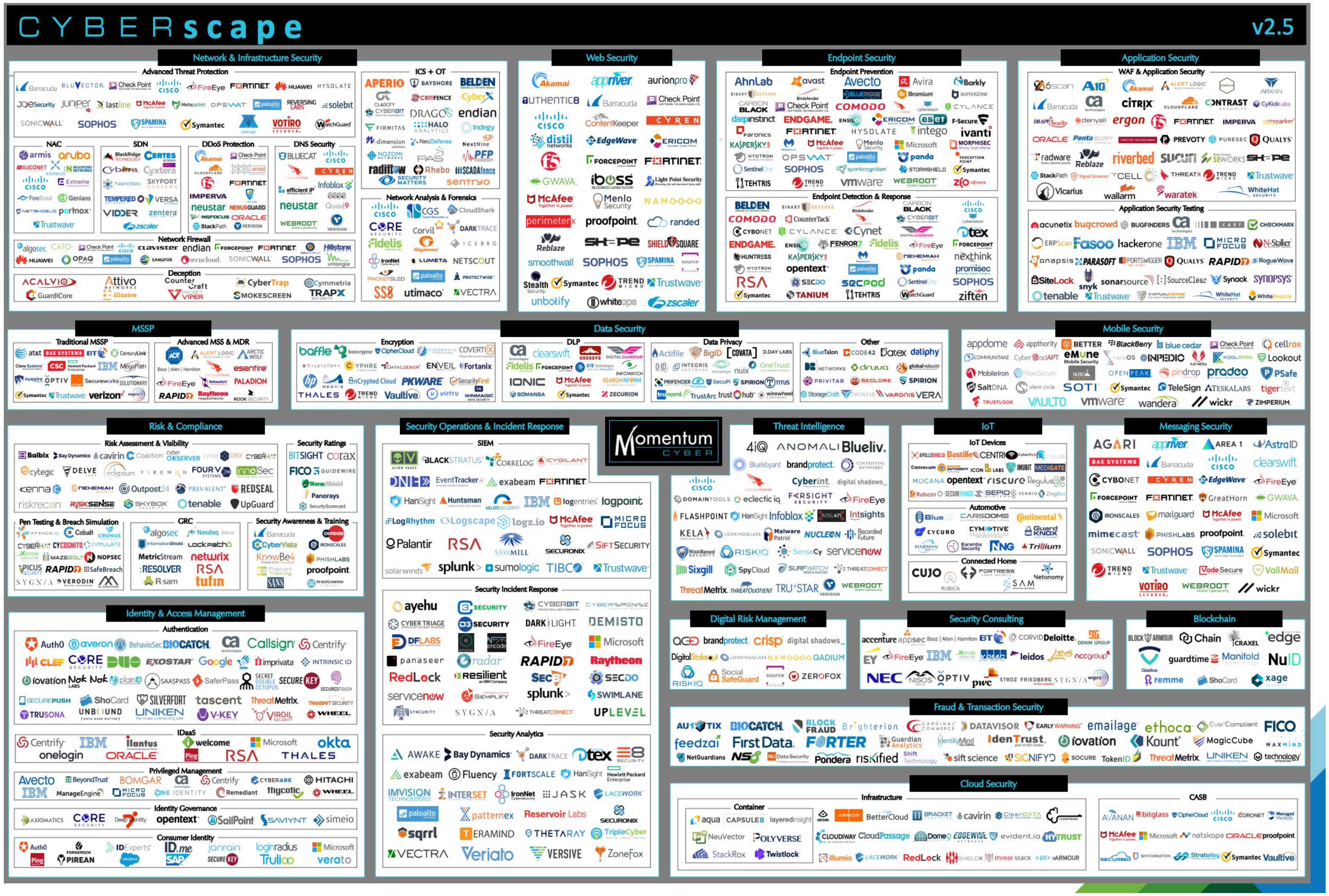


… but this innovation mechanism has led to 1500 security vendors and hundreds of startups looking for funding and exists …

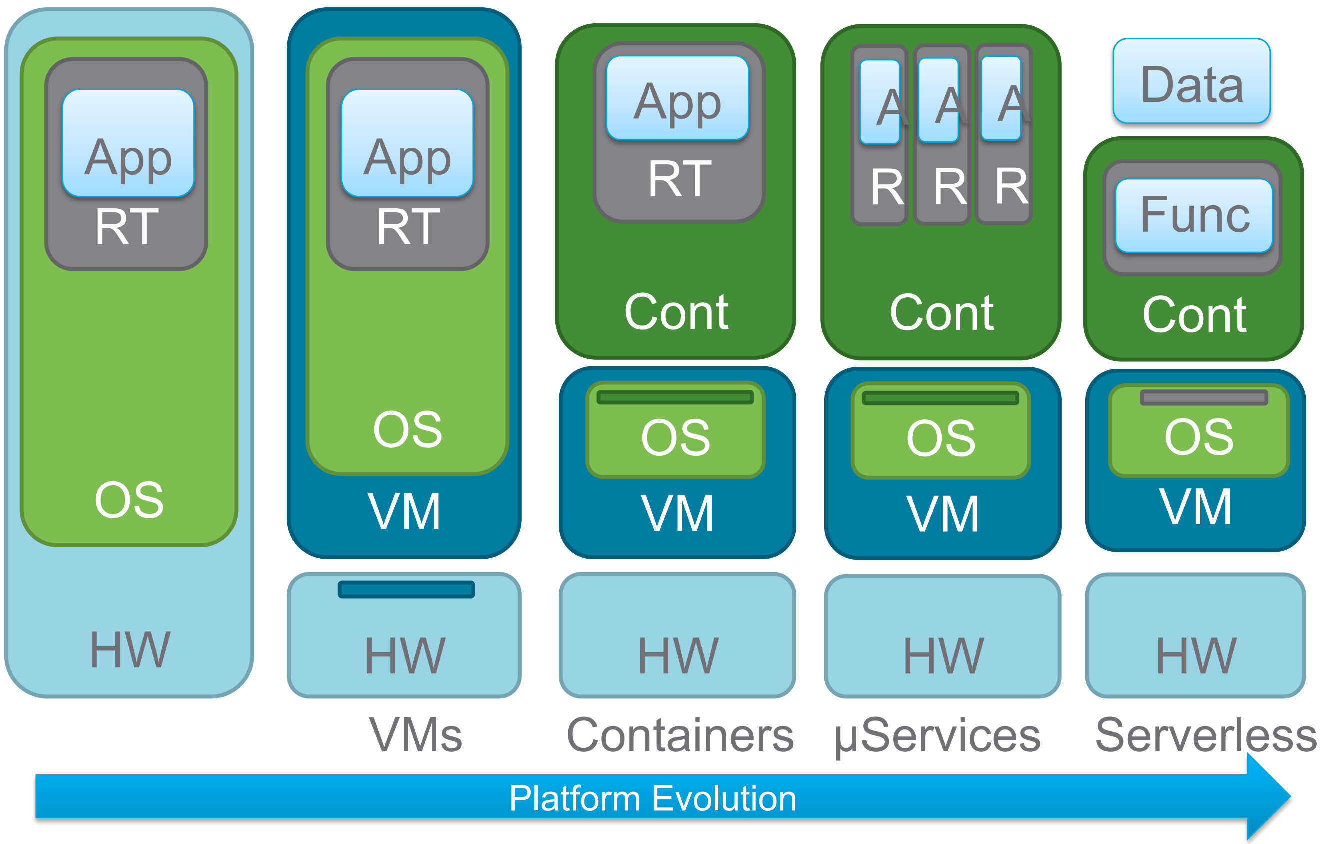
The future of security cannot look like this

Complexity triggered misconfiguration opens the door to >80% of breach

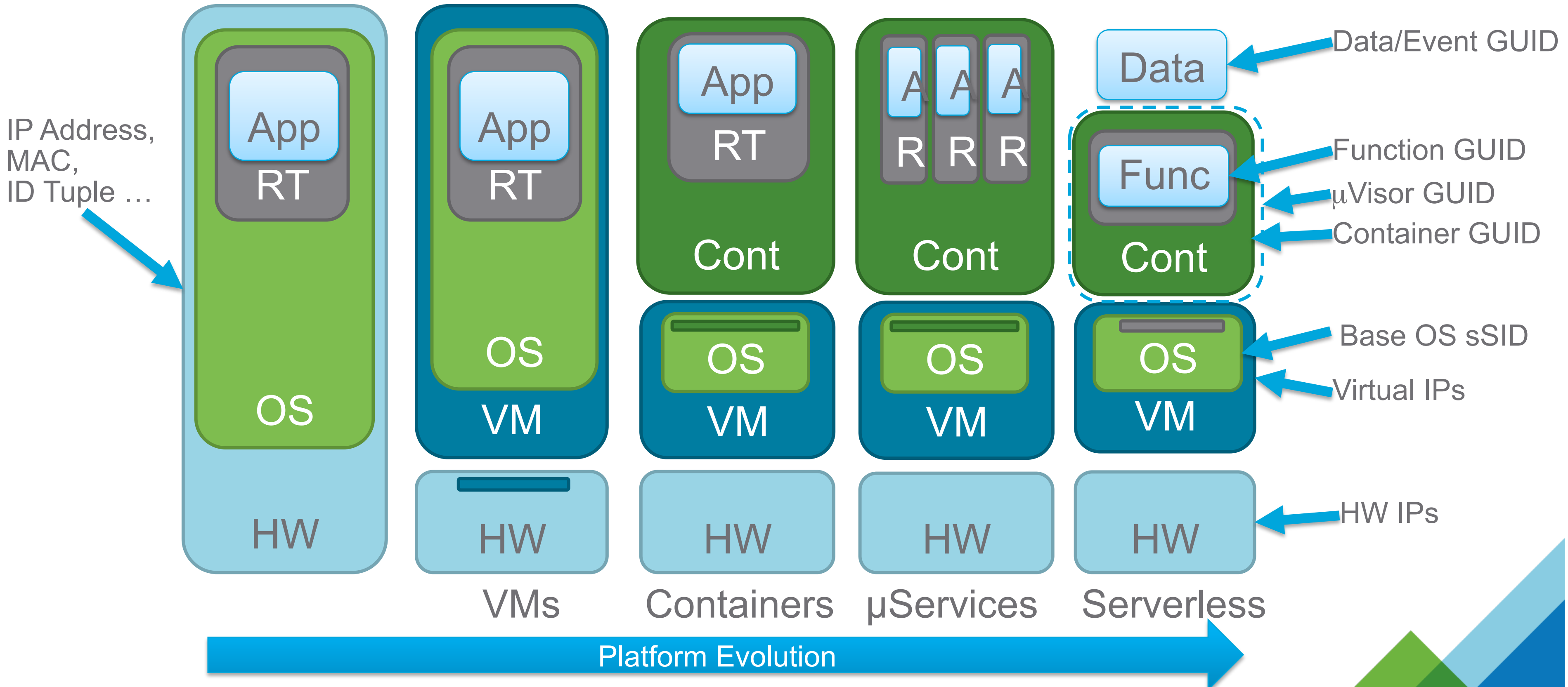
External Complexity from the solution space.



... Applications stacks are more decoupled, distributed and dynamic



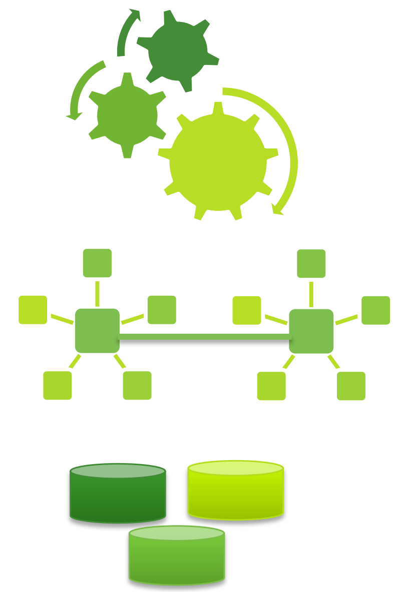
…consequently identifiers, structures and behaviors more complex.



Stack identifiers misaligned due to decoupling, distribution and dynamics.

Internal Platform complexity has grown too …

- Compute
  - Server Isolation, Processor Virtualization, Container Isolation – Process & Namespace, SM
  - White-Listing, Anti-Virus, Endpoint Detection & Response
  - TPMs (Titan, Intel, …), FPGAs, GPUs, Enclaves, ASLR, Control Flow Integrity, Smart NICs
- Network
  - VLANs, VPNs, Micro-segments
  - Firewalls, IPSs, WAFs, Sandboxes
  - Application Gateways, API Microgateways (JSON/APIs/gRPC), Layers (Functions)
- Storage
  - Volumes
  - ACLs
  - Encryption
- Composite Abstractions
  - PODs, STNs, VPCs, ASEs,…
- Future – More Dynamics (Moving Target), Encryption, Distribution (MPC, CryptoLedgers, …)

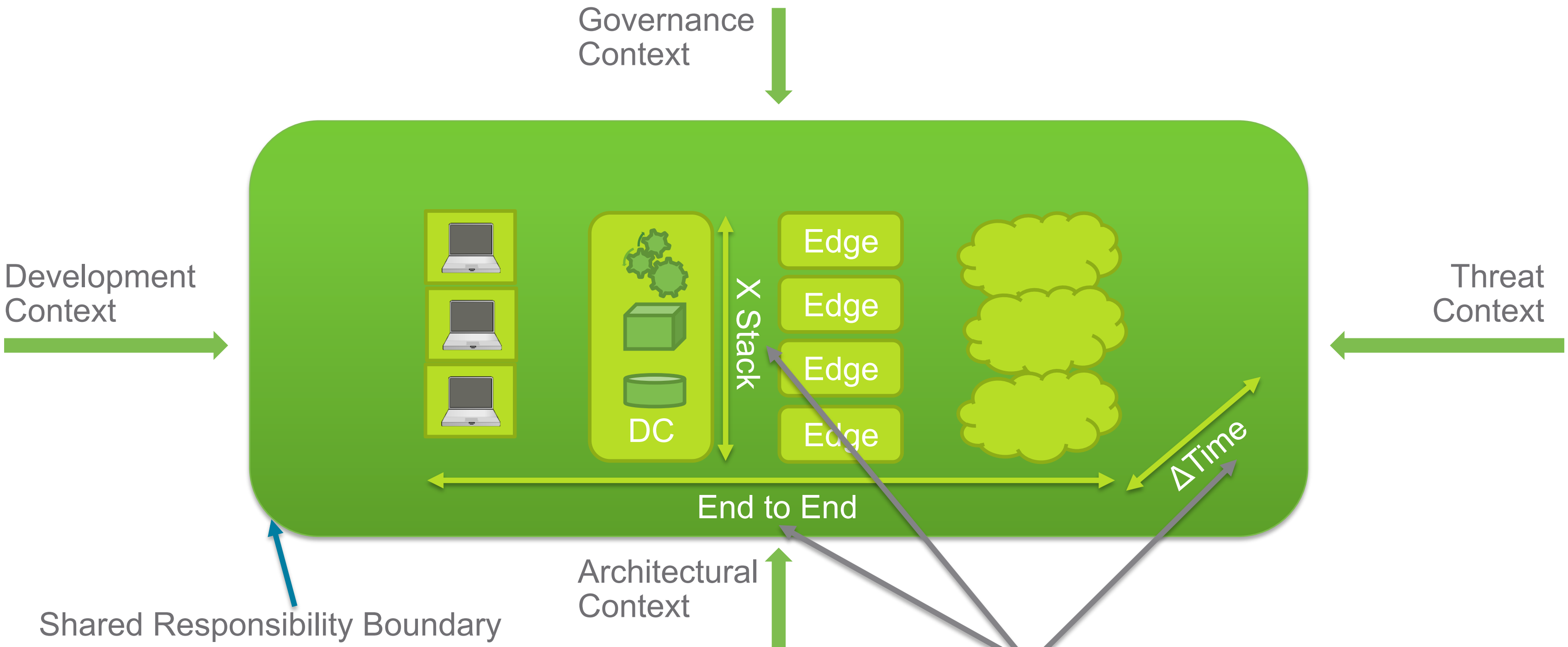


But none of these is ever perfect in implementation, or perfectly managed …

Reference: Engineering Trustworthy Systems, O. Sami Saydjari, 2018

Context (Internal and External) enlighten a way forward

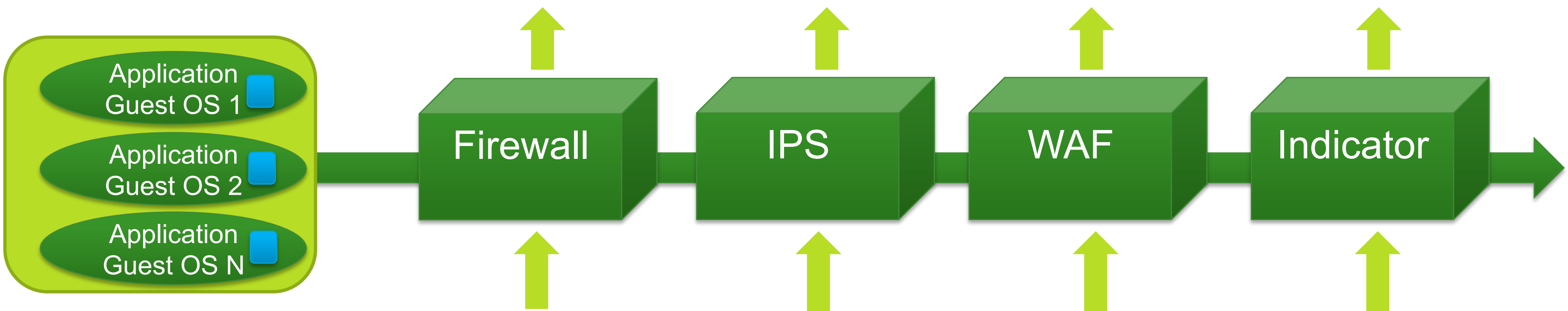
External Intentional Context is largely invariant over Platforms



… and key locus of correlation across contextual perspectives (internal and external)

Internal Context is inherently Platform-specific

Example: Security policy complexity



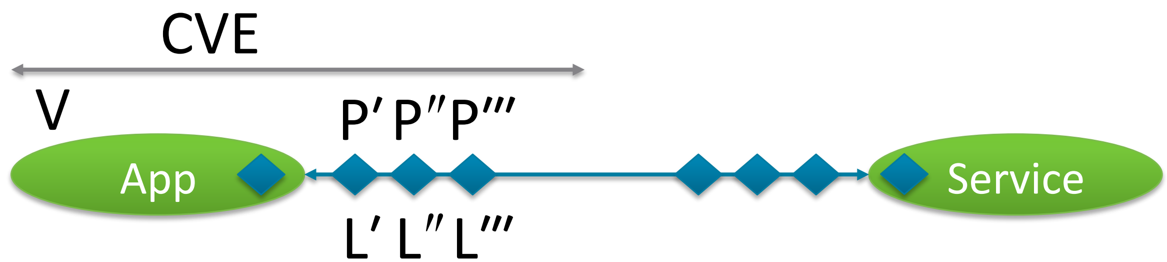
FW:
rule 1002 at 4 inout protocol udp from any to any port 67 accept;
rule 1002 at 5 inout protocol udp from any to any port 68 accept;
rule 1001 at 6 inout protocol any from any to any accept with log;

SNORT:
alert ip any any - > any any (msg:“ ET POLICY SSN Detected in Clear Text (dashed)”;
pcre:”/ ([0-6]\ d\ d | 7[0-2 56]\ d | 73[0-3] | 77[0-2])-\ d{ 2}-\ d{ 4} /”; reference:url,
doc.emergingthreats.net/ 2001328; classtype:policy-viol ation; sid: 2001328; rev: 13;)

OWASP MSRL: SecRule REQUEST_LINE "@contains /wp-content/plugins/firestats/php/window-add-excluded-ip.php“
"chain,phase:2,block,t:none,t:urlDecodeUni,t:htmlEntityDecode,t:normalisePathWin,capture,logdata:'%{TX.0}',severit
:'2',id:2011256,rev:2,msg:'SLR: ET WEB_SPECIFIC_APPS FireStats window-add-excluded-ip.php CSS Attempt',tag:'web-
application-attack'"

YARA:
$mz = { 4d 5a}   …
Condition: ($mz at 0) and ((1 of ($s*)) or (2 of ($x*) and all of ($z*))) and filesize < 4000

Example: End to End External Contextual Alignment Network Policy informed by endpoint context (e.g. Vuln(App(OS(SSID?)))

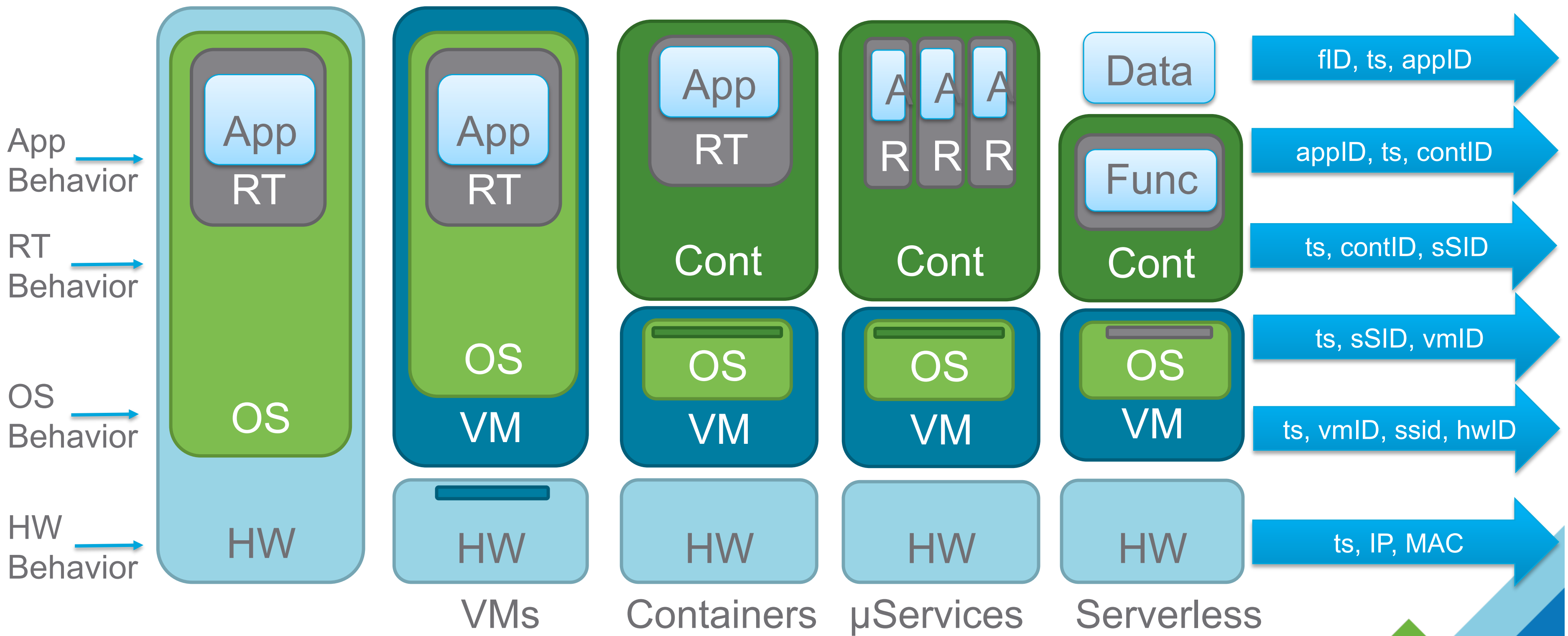


Context for Aligning Policy:

Intended Behavior (app)

- DevOps & Testing – RAML, Open API, API Blueprints
- IoT – IETF MUD (YANG descriptions)
- Integration – HL7 FIHR Integration Policy, API Lifecycle Management Facilities
- API Micro-gateways – Google Apigee (Swagger), Salesforce Mulesoft (RAML), AWS API Gateways, Tyk, …)
- Structure/Dependency OAM (Microsoft), TOSCA, Blueprints, Accelerators, Zachmann (audit), Helm, … VMware VRA Blueprints
- Reference Architectures – AWS Quick Starts, Azure Blueprints, VMware VVD

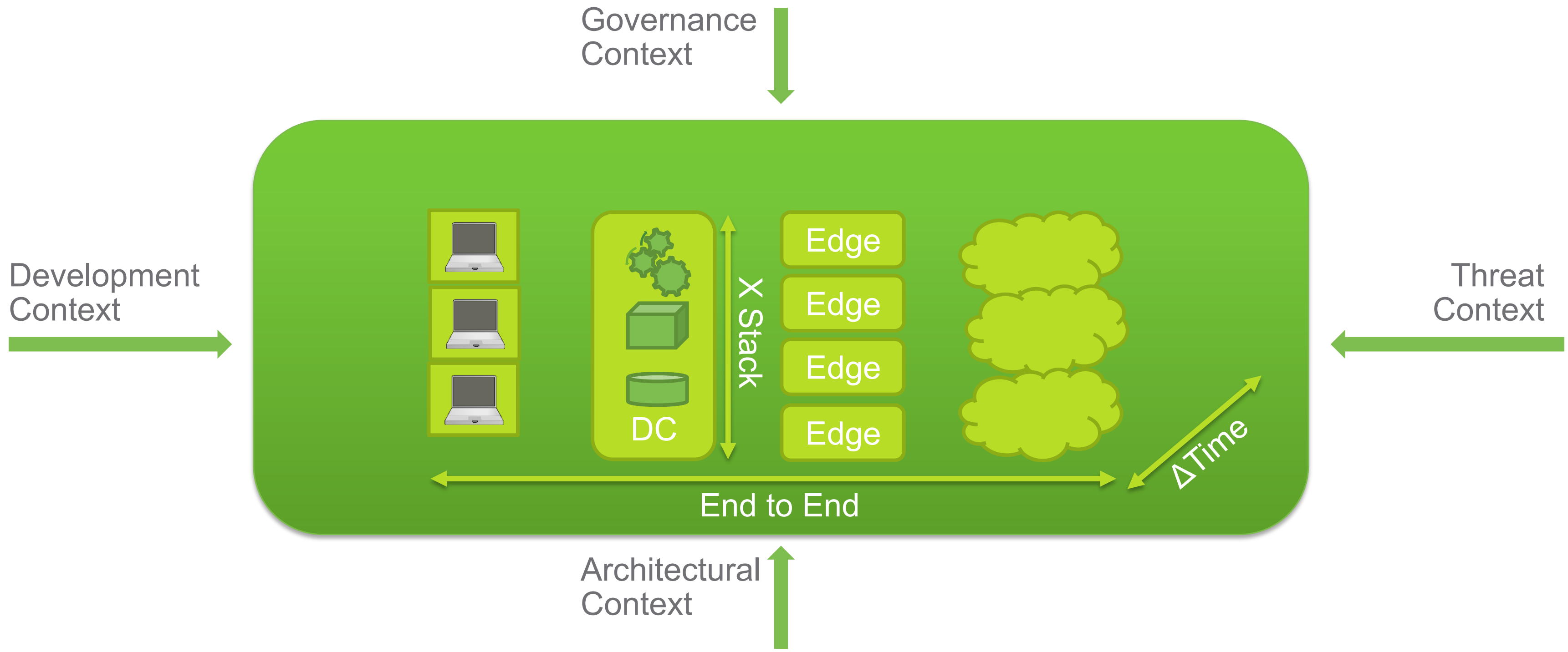
Example: Cross Stack Internal Contextual Alignment, …



Note: The hosting platform has the context to align decoupled logs and policy.

Context (Internal and External)
Where does it come from?

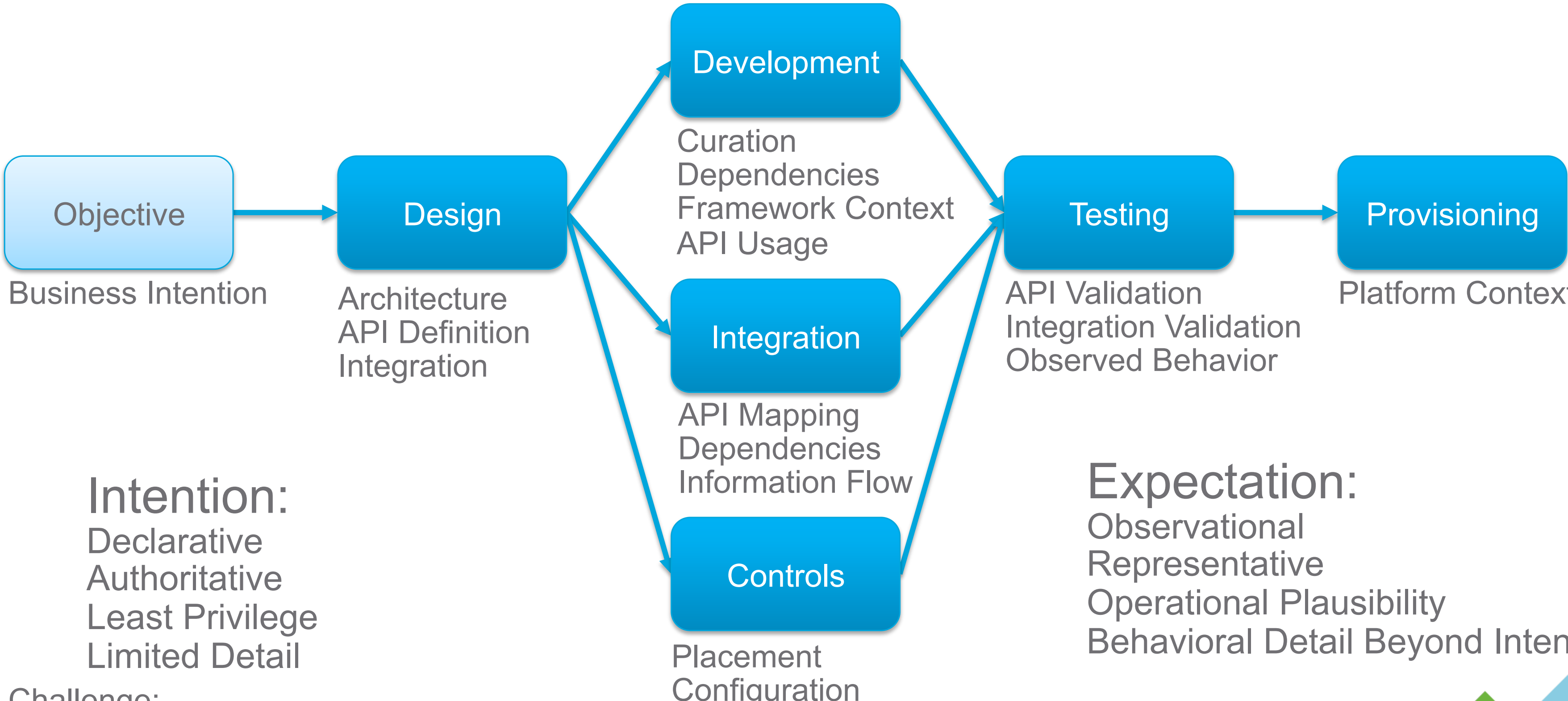
Platform-enabled Context: Aligning Security



Policy at this boundary has different authors, with different objectives, different change rhythms …
the Platform is where these intersect

Development Intentional Context (from the Left):

Leveraging API-First methodology (e.g. Pivotal –Home Depot, JPMC, ...)



Intention:

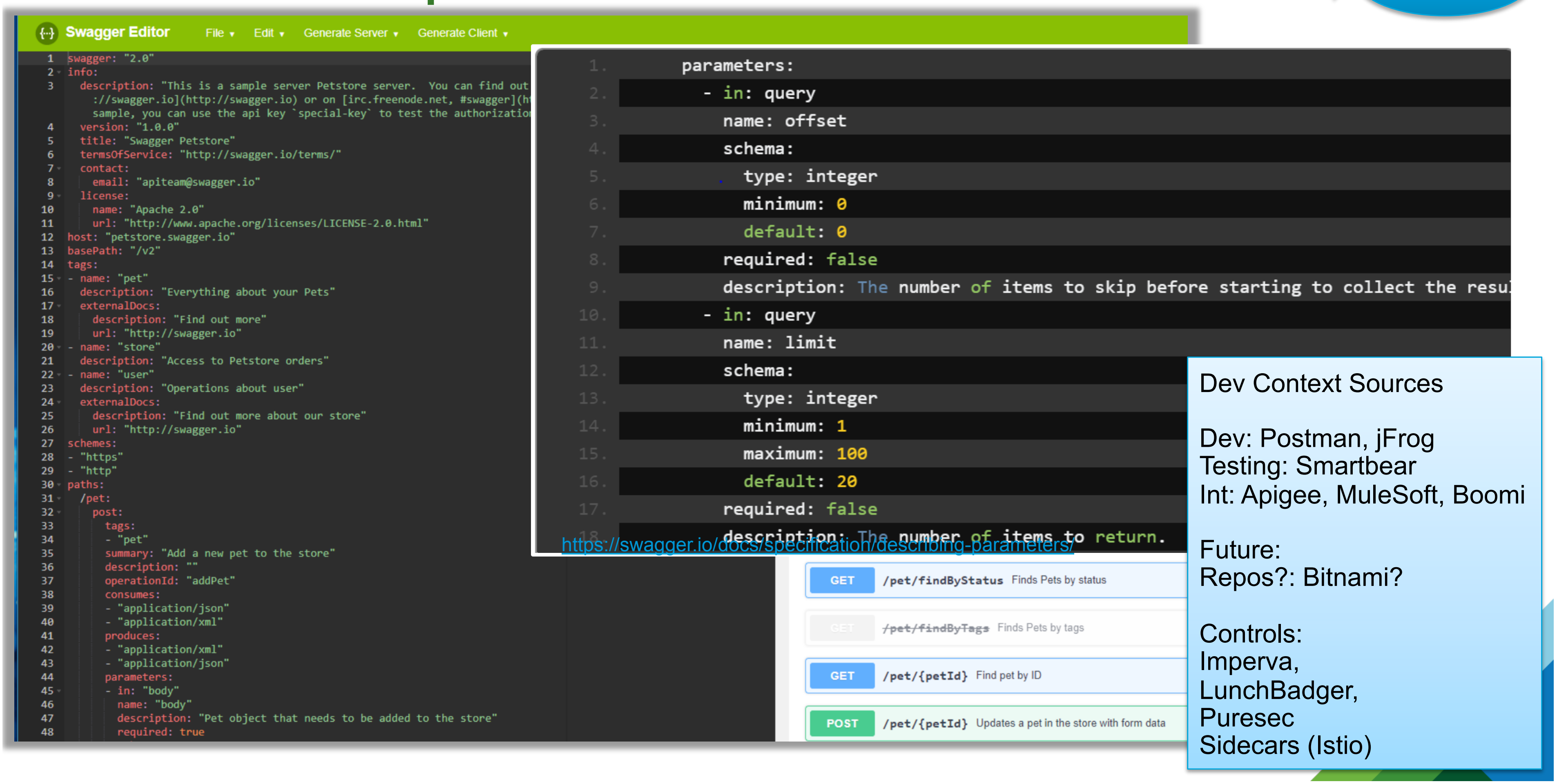
Declarative
Authoritative
Least Privilege
Limited Detail

Expectation:

Observational
Representative
Operational Plausibility
Behavioral Detail Beyond Intention

Challenge:
How can app repositories be leveraged to a) amortize the cost and amplify the value of derived context, b) improve SNR and context trustability, and to b) leverage a consumer population for feedback and trust?

Context: Development Context: Intended Behavior



https://swagger.io/docs/specification/describing-parameters/

Dev Context Sources

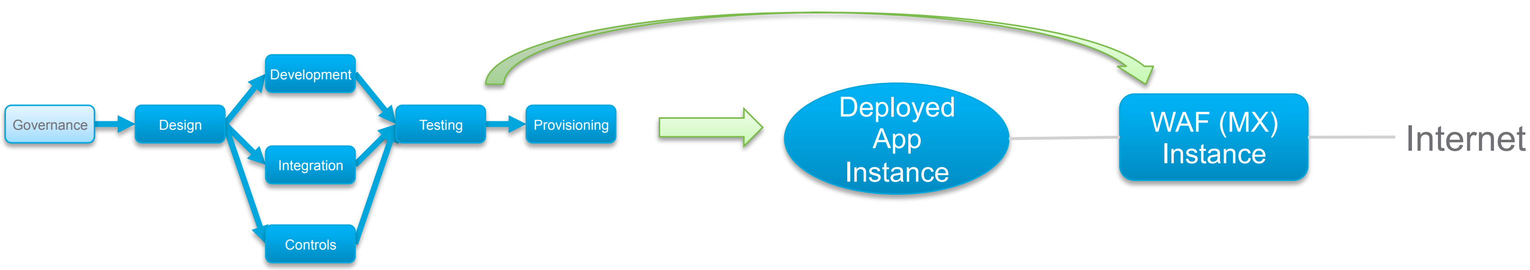
Dev: Postman, jFrog
Testing: Smartbear
Int: Apigee, MuleSoft, Boomi

Future:
Repos?: Bitnami?

Controls:
Imperva,
LunchBadger,
Puresec
Sidecars (Istio)

WAF Example
Imperva Ingestion of Swagger for SecureSphere

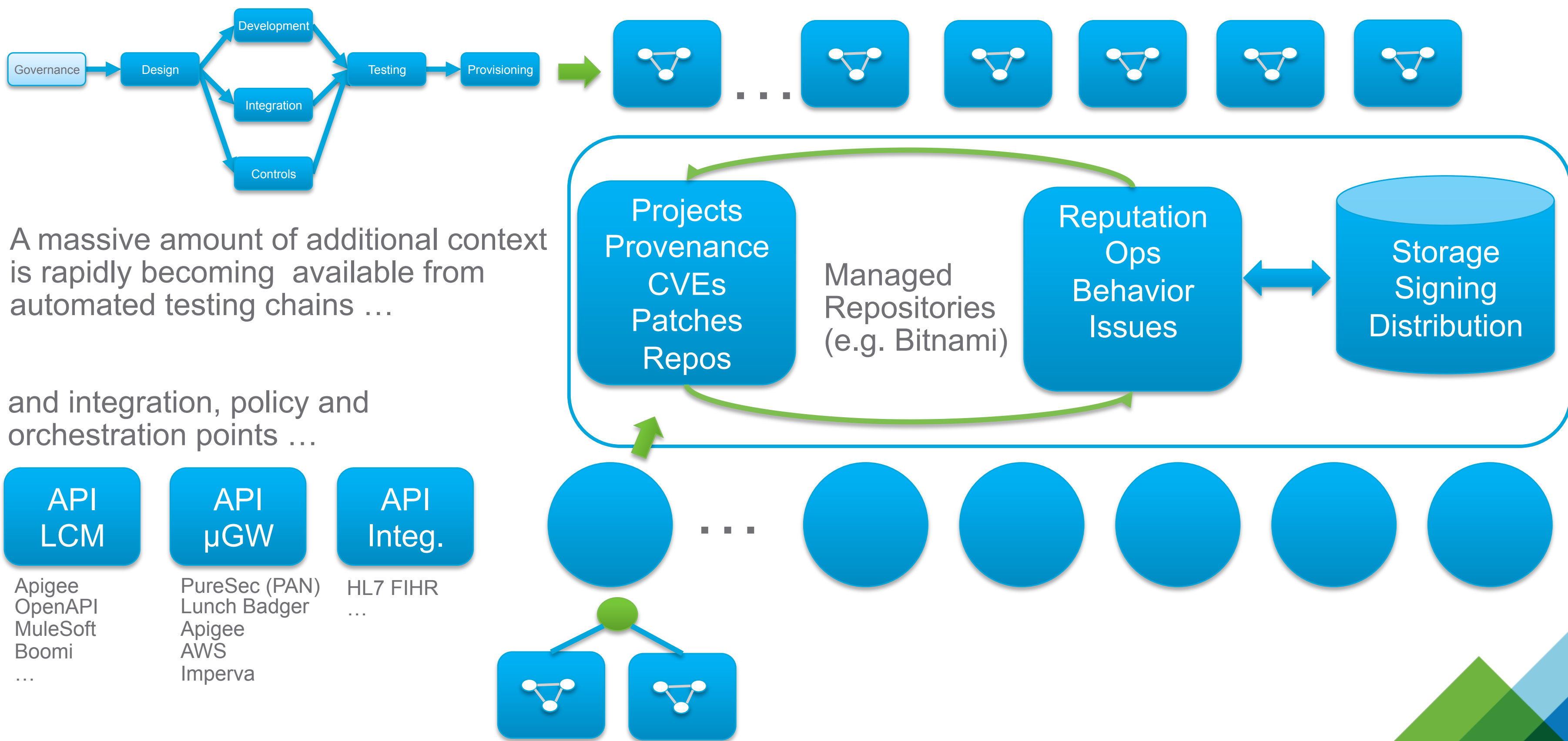
```
import imperva_sdk
from imperva_sdk.SwaggerJsonFile import SwaggerJsonFile
import json

# Connect to MX
mx = imperva_sdk.MxConnection("10.0.0.1", Password="password")

# Load swagger file as JSON
swagger_json = SwaggerJsonFile('swagger_file.json')

# Select Web Application
app = mx.get_web_application(Name="app", Site="site", ServerGroup="sg", WebService="ws")

# Apply swagger as profile
app.update_profile(SwaggerJson=swagger_json)

# Log out
mx.logout()
```



https://imperva.github.io/imperva-sdk-python/examples.html

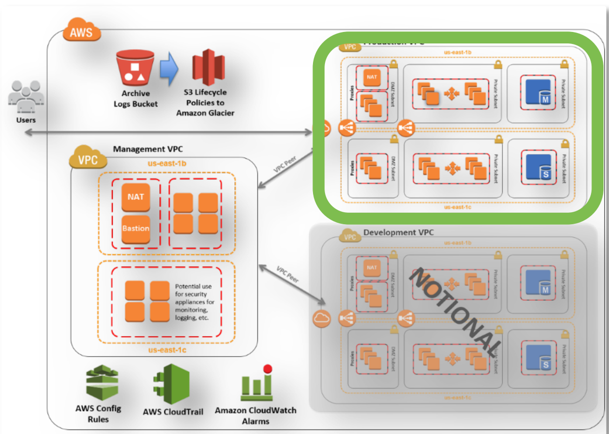
The Context Economy of Repositories: OSS & 3rd Party

Costs are amortized & Benefits are multiplied across repo users.



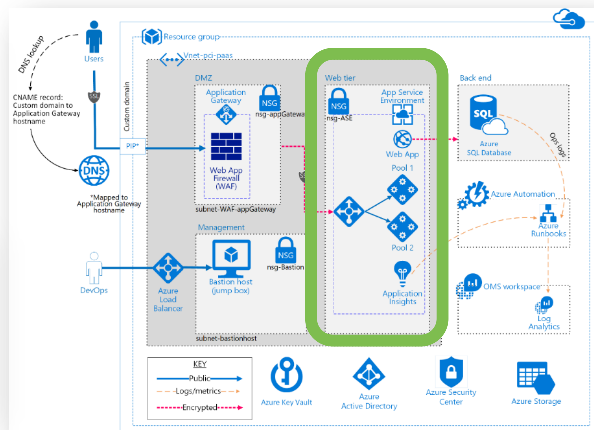
Architectural Context: Compliance Reference Models

AWS (34 Pages)



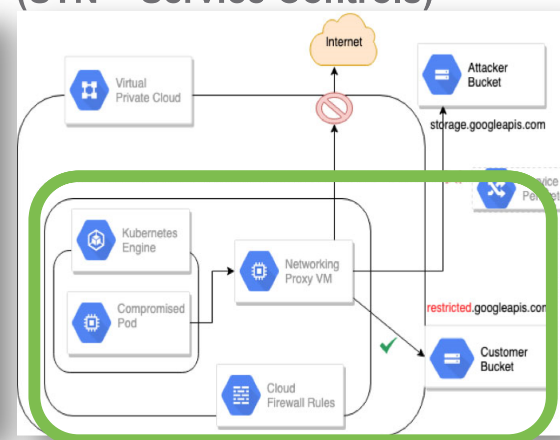
https://aws.amazon.com/quickstart/architecture/accelerator-pci/

Azure (19 Pages)



https://docs.microsoft.com/en-us/azure/security/blueprints/payment-processing-blueprint

Google (STN + Service Controls)



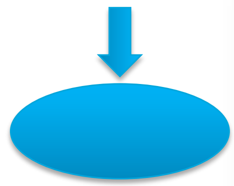
https://p16.praetorian.com/blog/cloud-data-exfiltration-via-gcp-storage-buckets-and-how-to-prevent-it

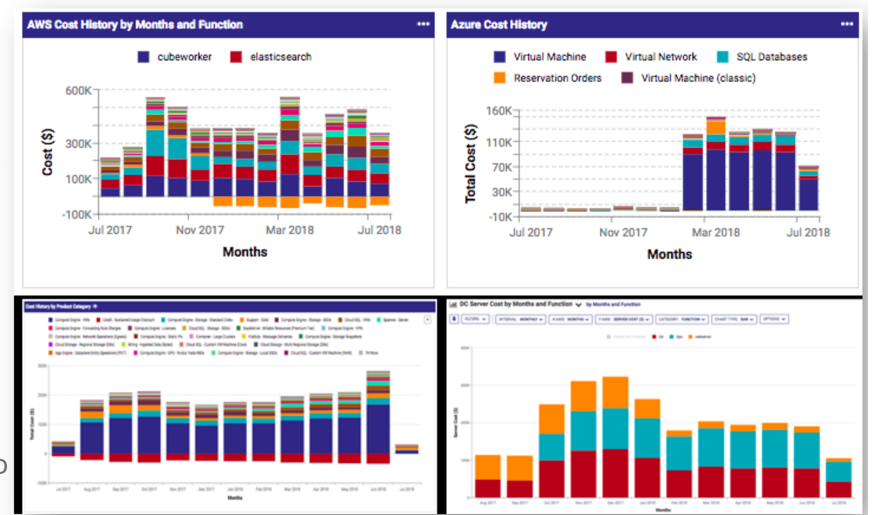
VMware VVD+ – Reference Arch …

- Audit once, comply many – costs amortized, benefits multiplied …
- Creates a community of highly normalized argumentation – learn/detect here … protect everywhere else
- Control topology provides actionability, guardrails, and semantics to telemetry

Challenge: How to bring new controls (dynamics, distribution, AR …) to this SD but conventional model.
System security argumentation beyond the compliance reference architecture -> system posture.
Accommodate “tense” of tagging – PCI classification vs PCI qualification vs PCI validation

Governance Context
Exploit Risk to Impact Risk

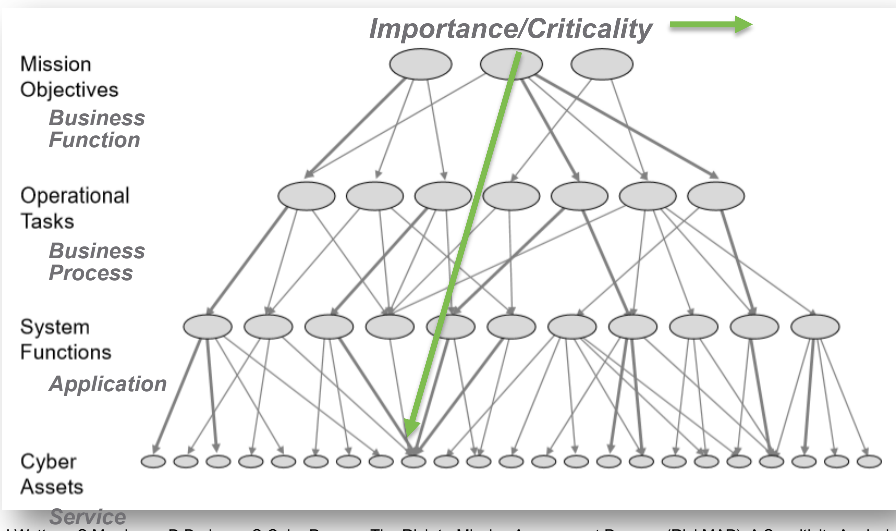


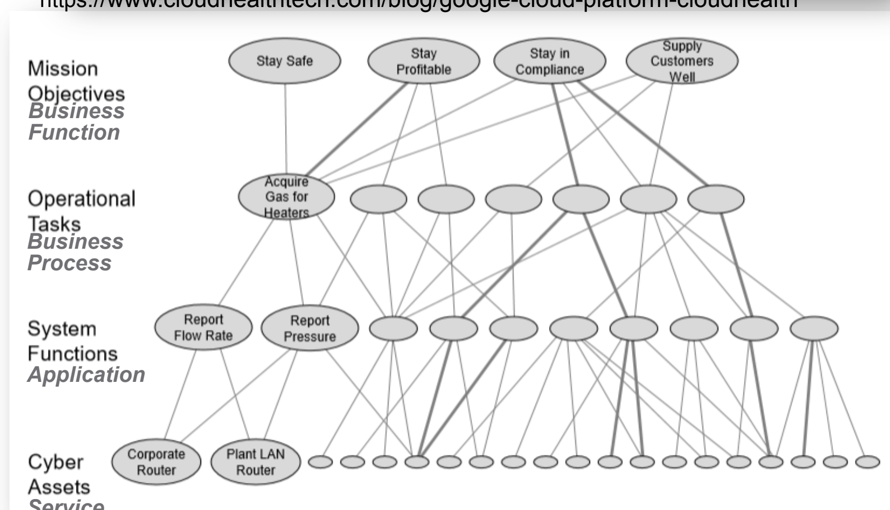


https://www.cloudhealthtech.com/blog/google-cloud-platform-cloudhealth

Risk = Impact (Criticality) * Exploitability (CVEb) * Probability (CVEt)

Challenge: There can is no algebra for risk, due to coupling over intimately & implicitly shared resources. How then can we connect labeled service criticality to underlying component logs/alerts/forensics severity?



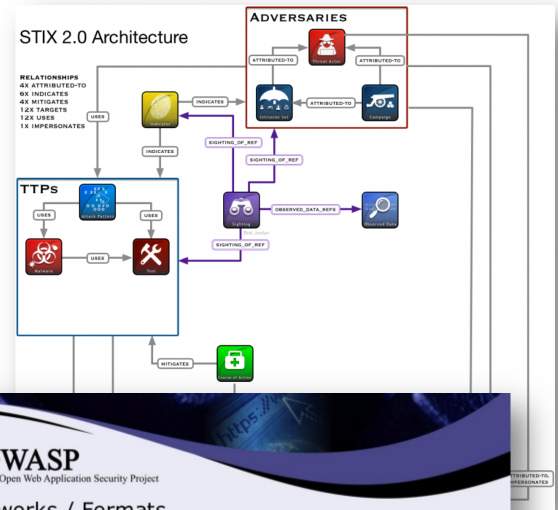


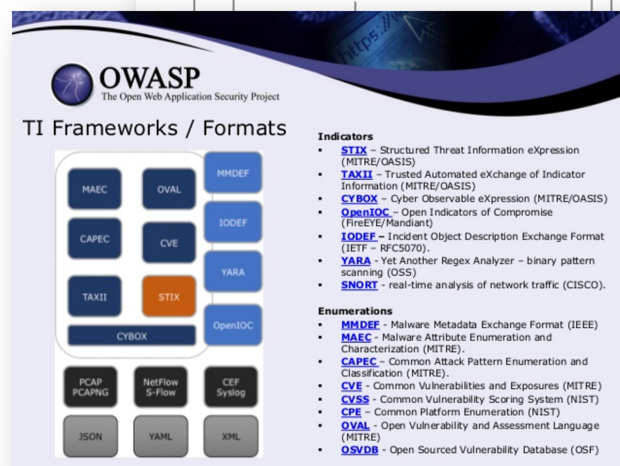
J Watters, S Morrissey, D Bodeau,, S Cohn Powers, The Risk-to-Mission Assessment Process (RiskMAP): A Sensitivity Analysis and an Extension to Treat Confidentiality Issues, The Mitre- DHS Research Report, July 2009

Threat Essential Context When hygiene fails.
Connecting the dots from indications to exploit

Description

- **Summary**
- **Attack_Execution_Flow**
  - Attack_Phase$^{1..3}$ (Name(Explore, Experiment, Exploit))
    - **Attack_Step$^{1..*}$**
      - **Attack_Step_Title**
      - **Attack_Step_Description**
      - **Attack_Step_Technique $^{0..*}$**
        - **Attack_Step_Technique_Description**
        - **Leveraged_Attack_Patterns**
        - **Relevant_Attack_Surface_Elements**
        - **Observables$^{0..*}$**
        - **Environments**
      - **Indicator$^{0..*}$ (ID, Type(Positive, Failure, Inconclusive))**
        - **Indicator_Description**
        - **Relevant_Attack_Surface_Elements**
        - **Environments**
      - **Outcome$^{0..*}$ (ID, Type(Success, Failure, Inconclusive))**
        - **Outcome_Description**
        - **Relevant_Attack_Surface_Elements**
        - **Observables$^{0..*}$**
        - **Environments**
      - **Security Control$^{0..*}$ (ID, Type(Detective, Corrective, Preventative))**
        - **Security_Control_Description**
        - **Relevant_Attack_Surface_Elements**
        - **Observables$^{0..*}$**
        - **Environments**
      - **Observables$^{0..*}$**

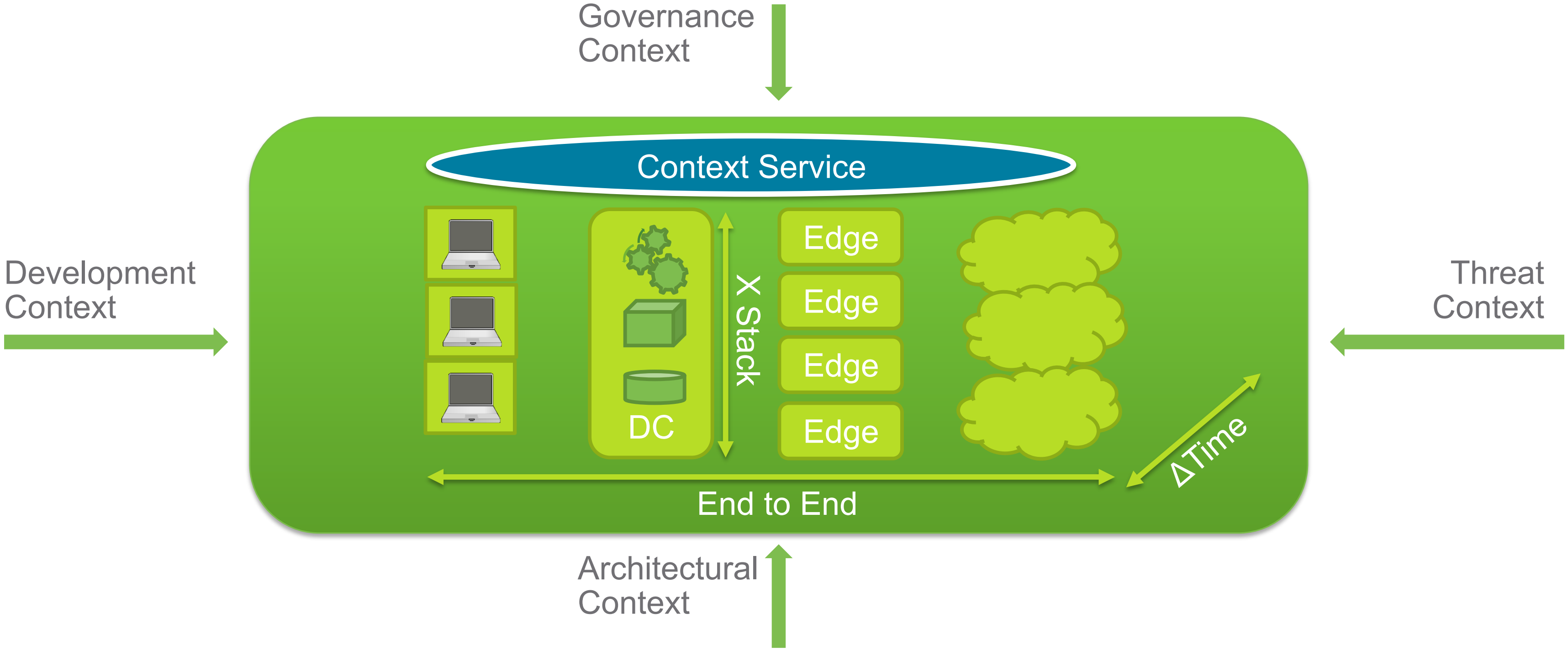




https://image.slidesharecdn.com/attackiseasyletstalkdefencev3-151026104559-lva1-app6891/95/bucharest-attack-is-easy-lets-talk-defence-20-638.jpg?cb=1445856555

Challenge: Behaviors on SDI are less representable by normal indicators due to decoupling & dynamics. Need behavior abstractions.

Platform-enabled Context (Internal and External): Single source of truth … end to end, X stack and over time.



Shifted Context (and Policy) has different authors, with different objectives, different change rhythms …
the Platform is where these intersect

Takeaways:

- At the context at the platform boundary between the External consumption of the and the Internal presentation is very useful
- This boundary is a disciplinary bridge across concerns.
- The modern hosting platform can automate internal context… conventionally difficult to construct
  - End to End
  - Cross Stack
  - Across Time
- The modern hosting platform can collect, protect and distribute external context (to entire management/ security portfolios
- Platform providers host vibrant innovation ecosystems, partnerships, residencies, internships and collaboration opportunities.

Emerging Research Challenges at the Platform Boundary

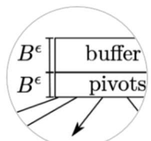
- Platform enlightened AI – toward more interpretable, explainable, actionable and therefore trustable intelligence and automation.
  - Purely statistical, regression based and ML techniques don't leverage the intentional structures and behavior, constraints, … resulting in adversarial, supply chain, explain-ability, actionability and trust-ability challenges.
  - Ex. XAI and 3rd Wave AI momentum: Causality models, embedded ML, intentional guardrails, …
- With richer sets of context over development lifecycles, we need models that can capture and support reasoning over intentional, expected and observed behaviors.
  - Existing tagging/labeling models are hobbled by ambiguity and semantic mismatches across disciplinary and lifecycle boundaries.
  - Ex. OASIS OCA (Security Portfolio), Mitre System Argumentation efforts
- More expressive policy logics/languages: As we shift testing and security "left", we increasingly cultivate more and different policy authors, who have different objectives and act in different rhythms.
  - First order policy languages require completeness and consistency that don't exist across diverse sources of dynamic policy . We need more expressive and embedded logic schemes that can provide useful inference in the face of incompleteness, inconsistency and evolutionary change.
  - Ex. AWS AR, Defeasible Logics, …
- With the increasing use of GPUs, TPUs, FPGAs, … as processor extensions and in shared resource pools, we extended trusted execution and attestation approaches that are less brittle (than hash extension) and leverage the isolation and dynamics of modern platforms.
  - Example: Trusted Blue line/Green line code models.
- As we face the emergence of Quantum computation, and the intrinsic uncertainties over post-quantum crypto techniques, a level of crypto agility and/or resilience will be needed. What are new abstractions might make migration of crypto technologies, less disruptive to application and services.
  - Example: Microsoft Post-Quantum Crypto VPNs

VRG Active Research Areas
(Frequently 1-2 Researchers + Research Interns)

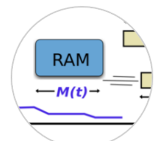
https://research.vmware.com/projects

Active Research Areas

Anomaly Detection
Anomaly detection algorithms that intuitive, rigorous and scalable.



BetrFS
A right-optimized write-optimized file system

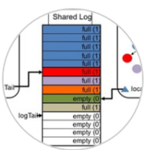


Cache-Adaptive Algorithms
Tools for analyzing algorithm performance in the real world

CloudCast
CloudCast is a world-wide and expandable measurements and analysis system, co...

Data center Network Topology Design
Designing performant, practical data center networks for cost and operational...

Networking for the 99%
This projects studies "non-hyperscalar" networks, their features and pain poi...



NR
A method to implement any concurrent data structure.

P4
P4: Programmable data-planes

RADE
Resource-efficient supervised anomaly detection framework that reduces memory...

Scalable and Precise Stream Processing
Algorithms and data structures for real-time processing of streams that are t...

Towards Predictable Low Latency Networks
Data center network stack that can provide predictable low latency

Opportunities for Academic Research with VMware

Faculty Research Collaborations

https://www.vmware.com/company/research/faculty-programs.html#research

Faculty Research Collaborations

VMware is committed to sponsoring academic research in areas of importance to the future of computing. Our support for faculty enables graduate student researchers and post-docs, and helps to cover the expenses involved in developing new technology in a university setting. Some recent faculty research collaborations include:

- Arizona State University
- Bar Ilan University
- Brown University
- Carnegie Mellon University
- Cornell University
- École Polytechnique Fédérale de Lausanne (EPFL)
- ETH Zürich
- Georgia Institute of Technology
- Imperial College London
- Indian Institute of Technology, Delhi
- Massachusetts Institute of Technology
- Politecnico di Torino
- Princeton University
- Stanford University
- Technion
- Tel Aviv University
- Texas A&M University
- University College London
- University of California, Berkeley
- University of California, Santa Cruz
- University of Cambridge
- University of Colorado at Boulder
- University of North Carolina at Chapel Hill
- University of Texas at Austin
- University of Texas at Dallas
- University of Utah
- University of Washington
- University of Wisconsin at Madison

Systems Research Awards

https://www.vmware.com/company/research/faculty-programs.html#system

Tiark Rompf is an Assistant Professor of Computer Science at Purdue University.

Professor Rompf received the 2018 VMware Systems Research Award. He is recognized for radically new approaches to performance- and safety-critical systems, in particular through rethinking the role and relationship between high-level and low-level languages. His systems-oriented approach is illustrated well by his far-ranging explorations of lightweight modular staging (LMS), a platform and methodology for enabling run-time code generation.

Tim Kraska is an Associate Professor of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology (MIT).

Professor Kraska received the 2017 VMware Systems Research Award. He has been widely recognized for his early work on hybrid human-machine data management. On the systems side, his work includes a pioneering reference architecture (CrowdDB) for hybrid crowdsourced queries. He has continued to role-model a style of holistic systems treatment in his early research by formulating and tackling research problems that together represent a powerful new vision for the future of database systems.

Matei Zaharia is an Assistant Professor of Computer Science at Stanford University.

Professor Zaharia received the 2016 VMware Systems Research Award. His accomplishments as a young researcher include developing the well-known and widely used open source projects Apache Spark, Apache Mesos, and Tachyon (now Alluxio). Zaharia's academic work has received thousands of citations, and his software is being used by thousands of developers worldwide.

Early Career Faculty Grants and Scholar-in-Residence Program

https://www.vmware.com/company/research/faculty-programs.html#scholar

Early Career Faculty Grants

The Early Career Faculty Grant program is intended to recognize the next generation of exceptional faculty members. A gift to the researcher's university is made in support of his/her research and to promote excellence in teaching. Early career faculty must be within five years of their first tenure-track appointment. Recent grants include:

- Ding Yuan, University of Toronto
- Bharath Raghavan, University of Southern California
- Aurojit Panda, New York University
- Aruna Balasubramanian, Stony Brook University
- Taesoo Kim, Georgia Institute of Technology

Scholar-in-Residence

The Scholar-in-Residence (SiR) program brings together exceptional university faculty with VMware researchers for deeper collaboration over a specific time period. SiRs often takes place during a faculty member's summer break or sabbatical year. Collaboration focuses on research objectives mutually defined in advance. Recent scholars include leading faculty from:

- Carnegie Mellon University
- Bar Ilan University
- Technion
- Tel Aviv University
- University of North Carolina at Chapel Hill

For a listing of current open positions, visit our careers page.

Thank You.

Darleen Fisher will email a copy of the slides.

Questions?

Dennis R Moreau
Cybersecurity Information Architecture
dmoreau@vmware.com

Abstract

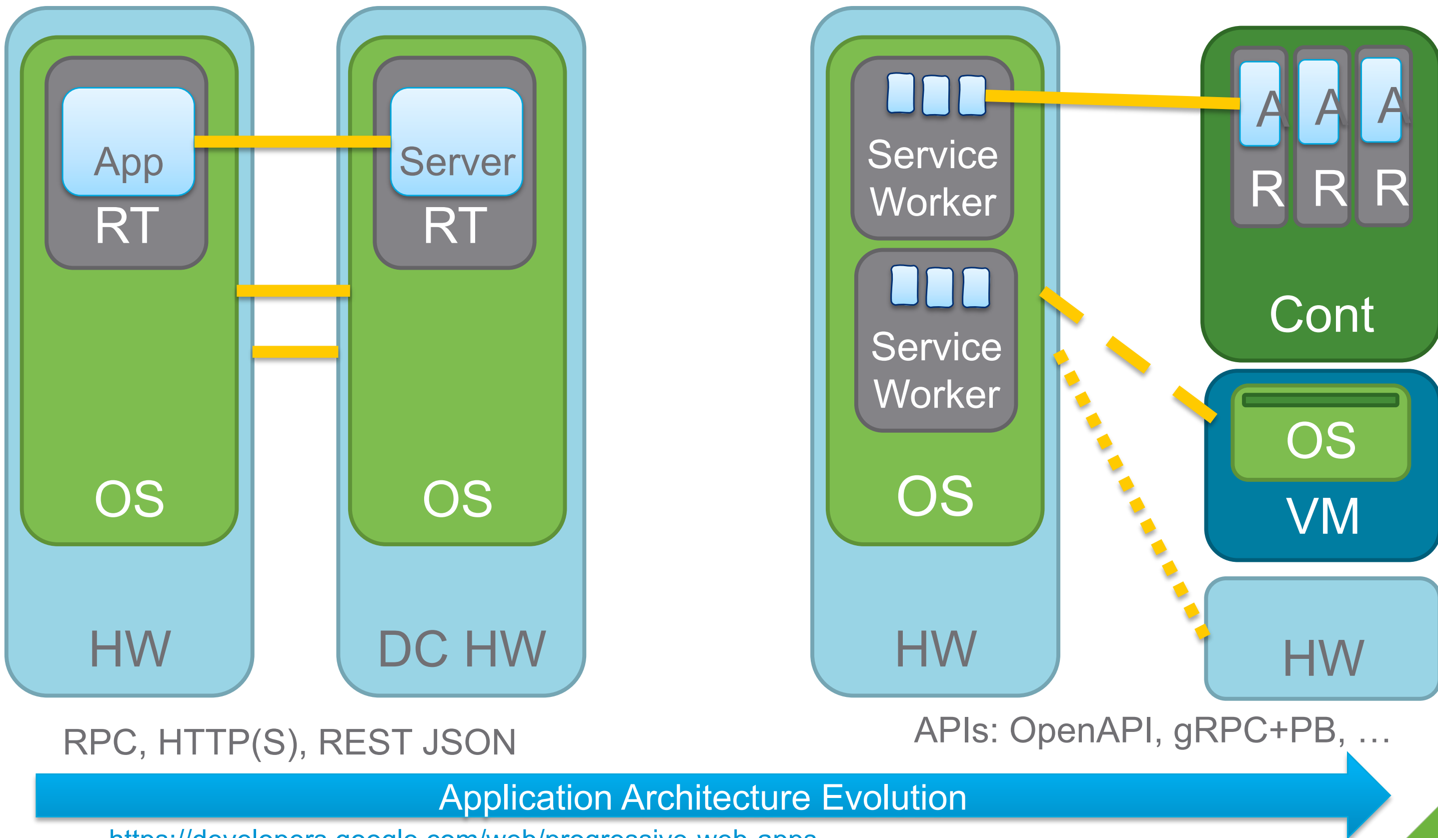
The rapid growth in the adoption of modern application development and hosting technologies has brought with it, unprecedented levels of complexity, in terms of stack decoupling, instance dynamics, and system distribution. The underlying hosting platforms readily span multiple on-premise, co-hosted and cloud-hosted sites, easily extending across geographic and regulatory boundaries. Within individual platforms there is an accompanying convergence of computation, networking and storage capabilities, realized over common resources and shared fabrics. The result is that services, applications, platforms, infrastructure and even bare metal can all be consumed on demand at incredible scale.

Unfortunately, complexity driven misconfiguration, recurrent outages and massive breaches are stimulating the growing realization that we must cultivate innovations that deliver much simpler, more efficient, more effective and more trustable information systems. The current turbulent tension between agility and manageability is challenging the conventional technological underpinnings of the management, operation and security of information systems hosted on modern platforms.

However, the very characteristics that precipitate these challenges also light the way forward in addressing them, making the modern hosting platforms ideal environments for supporting computing and networking research programs, across the innovation lifecycle including discovery, analysis, experimentation, prototyping, validation, and commercialization, extending to delivery and consumption of innovation at scale.

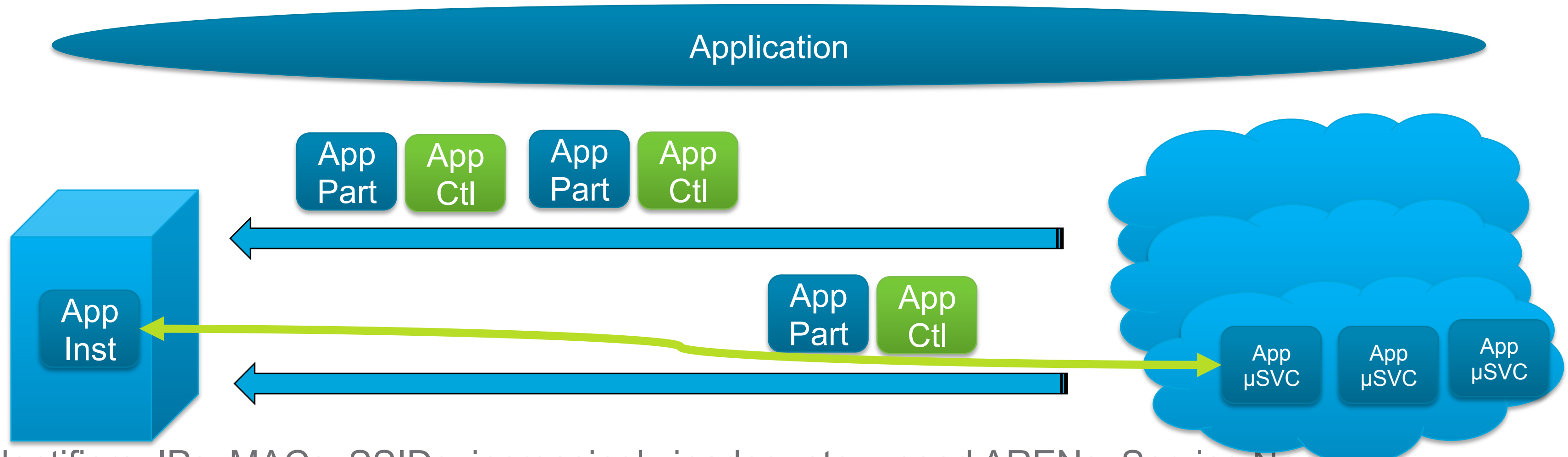
In this session we will consider emerging challenges and opportunities for modern information systems on hosting platforms, that are addressable by individual and collaborating researchers, and their teams. We will also consider the role of those platforms in facilitating innovation aimed at addressing these challenges, and how researcher engagement with platform providers and their user communities has evolved.

Client applications too, are more decoupled, dynamic, distributed



https://developers.google.com/web/progressive-web-apps

Looking Forward: Application footprint is dynamically expressed across client and backing services. Progressive Apps



- Identifiers: IPs, MACs, SSIDs increasingly inadequate – need ARENs, Service Names,
- Behavior: State, Behavior increasingly dynamic wrt Client and backing Services
- Analytics: Due to Identifier and Behavior challenges, even correlation, ML are more challenged
- Actionability: What, Where and Why are all tougher to resolve into an actionable context

The Hosting Platform Role:

For Specific Persona and roles like Root Cause Analysis, Security Response, Behavioral Analytics.... provide authenticated access to:

- 1) Vertical dependencies across abstractions layers and dynamics
  - From Applications/Services, to Containers, to Pods, to VMs, to Servers, …
- 2) Horizontal interactions/connections, end to end.
  - From clients to backing services
- 3) Context by Identifier.
  - Provenance, Templates, Instances, Tests, Attestations, Hosts, Policy Sets, Accounts, …
  - Intention, Expectation, Observation

The decoupling, distribution and dynamics that cause this complexity, are also enablers of the solution…

- "… Cloud-native architectures should extend this idea (granular Defense in Depth) beyond authentication to include things like rate limiting and script injection. Each component in a design should seek to protect itself from the other components. This not only makes the architecture very resilient, it also makes the resulting services easier to deploy in a cloud environment, where there may not be a trusted network between the service and its users…"
  - Google: https://cloud.google.com/blog/products/application-development/5-principles-for-cloud-native-architecture-what-it-is-and-how-to-master-it
- Where coupling increases (Netflix's appropriate coupling) context enhancement reigns in complexity.
  - Netflix: https://www.infoq.com/news/2019/01/netflix-evolution-architecture/
- DevSecOps & Context
  - DoD DevSecOps Ref Design: https://dodcio.defense.gov/Portals/0/Documents/DoD%20Enterprise%20DevSecOps%20Reference%20Design%20v1.0_Public%20Release.pdf?ver=2019-09-26-115824-583
  - Mitre Security & DevSecOps: https://www.mitre.org/sites/default/files/publications/pr-19-0769-devsecops_security_test_automation-briefing.pdf