Cyber security approaches for Cloud/Edge Environments

Alessandro Carrega

TNT Lab – DITEN
University of Genoa

Information

- Lecturer: **Dr. Alessadro Carrega**.
  - *Email*: alessandro.carrega@unige.it.
  - *Skype*: alessandro.carrega@gmail.com.
  - *Telegram / Whatsapp*: **3487485497**.
- Duration: **20 hours**.
- Language: **English**.
- Lesson in site and remote (Teams).

Information

- Dedicated Teams channel:
  - **PhD STIET Cyber security approaches for Cloud/Edge Environments**

    https://teams.microsoft.com/l/team/19%3a-Dtnw_NHUAl1AjZZV4HixIifmU8gywbskeeQwSV--uk1%40thread.tacv2/conversations?groupId=bdafff5c-0ab9-44b2-aef2-5a14e1dd6e15&tenantId=6cd36f83-1a02-442d-972f-2670cb5e9b1a
- Optional homework.
  - Available in Teams.
- Final Exam with two options:
  - **Theoretical**: *short survey with 3 papers*.
  - **Pratical**: *2 exercises*.
  - **Quiz**: *multiple choice questions*.

Outline

1

Evolving computing paradigms

- Virtualization, cloud computing, software orchestration
- Software-oriented architectures and micro-services
- Beyond the cloud: the Internet of Service and CPS

2

Cloud security aspects

- Limitations of existing approaches
- Evolving threats landscape
- Cloud security approaches

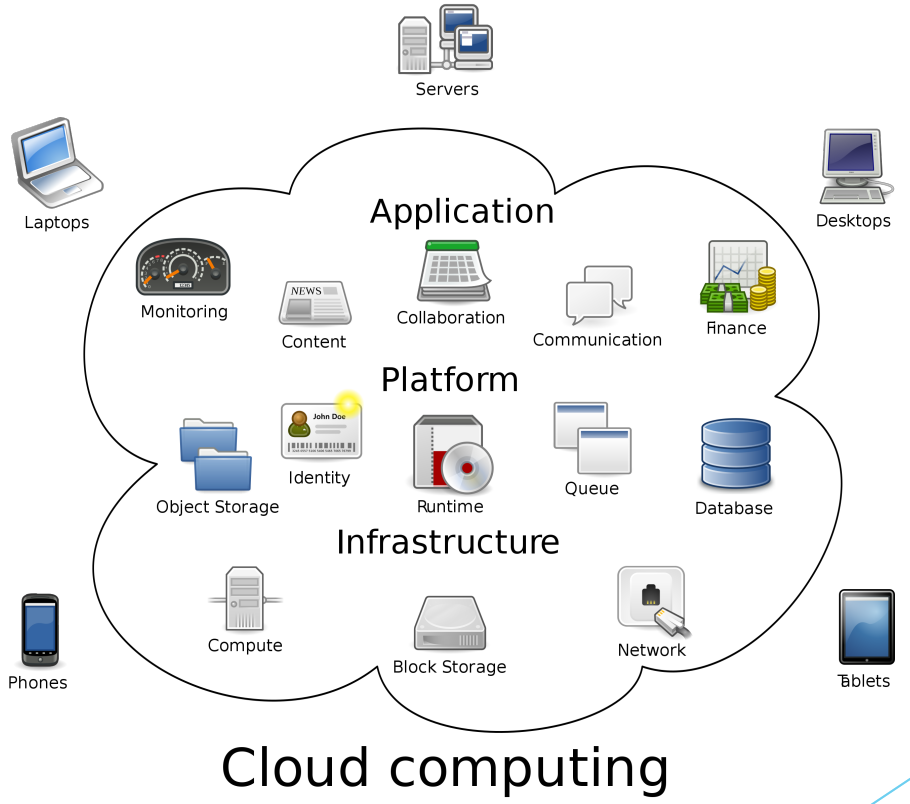
3

Protecting the workload

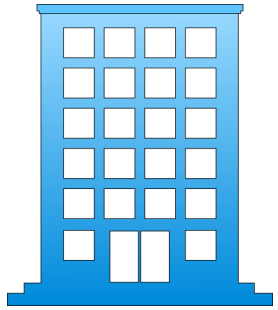
- Monitoring and visibility for cloud-based services: SIEM systems
- Enhancing Elastic Stack
- The ASTRID/GUARD architecture
- Secure development, integration, and deployment pipelines

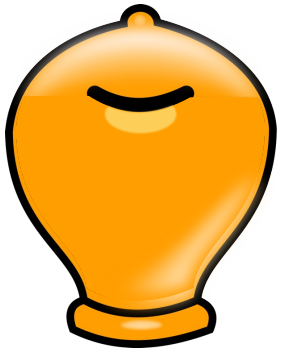
Cloud computing

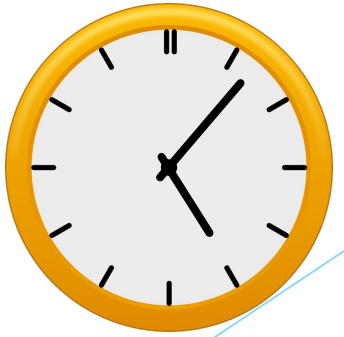
- Virtualization
- XaaS paradigms: IaaS, PaaS, SaaS, NaaS, …
- Sharing of resources (servers, networks, software, data)
  - Multitenancy
- Private vs Public systems
  - Sharing of responsibility
- Pay-as-you-go
- Self-provisioning and software-defined



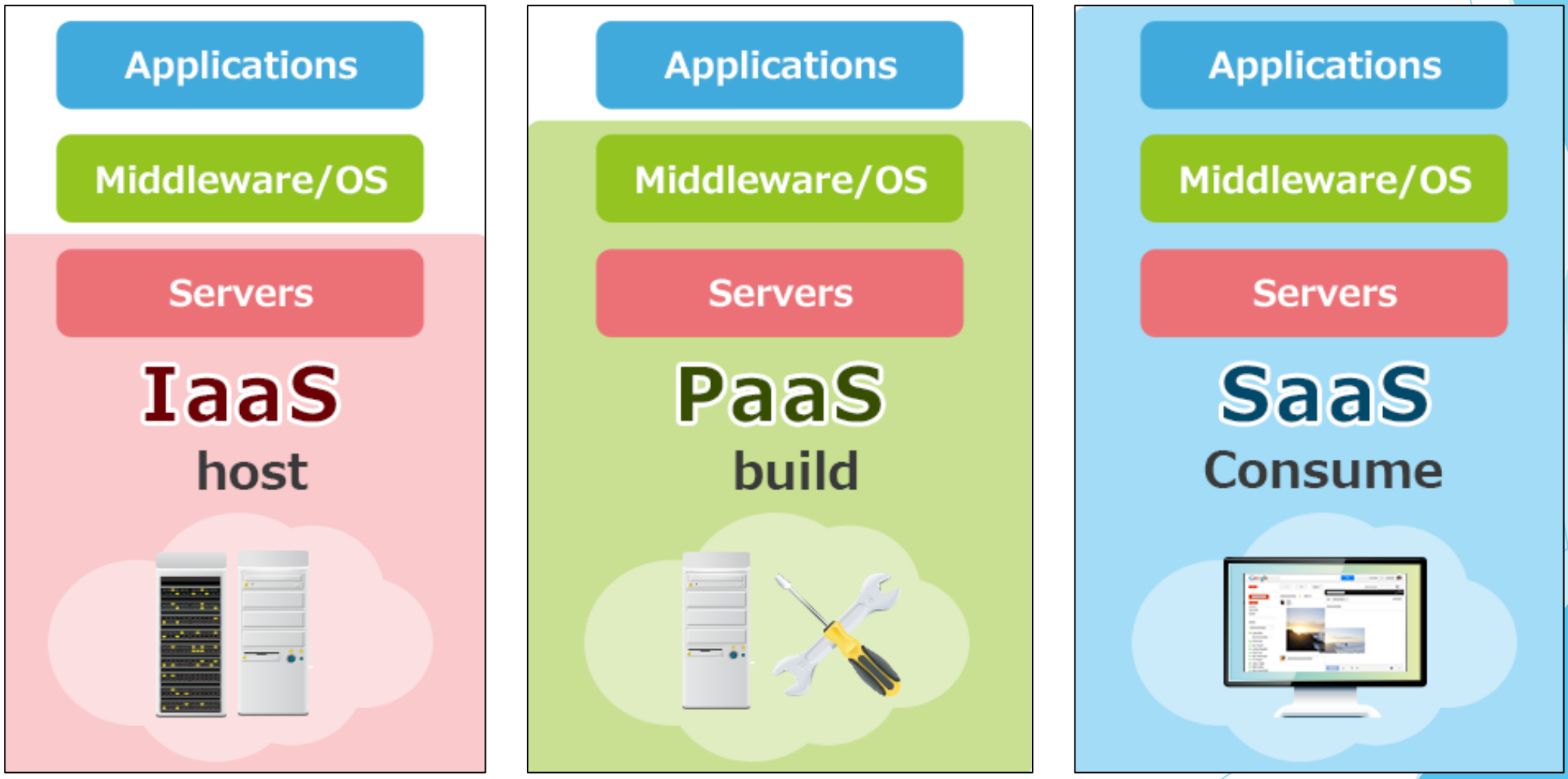
Cloud computing



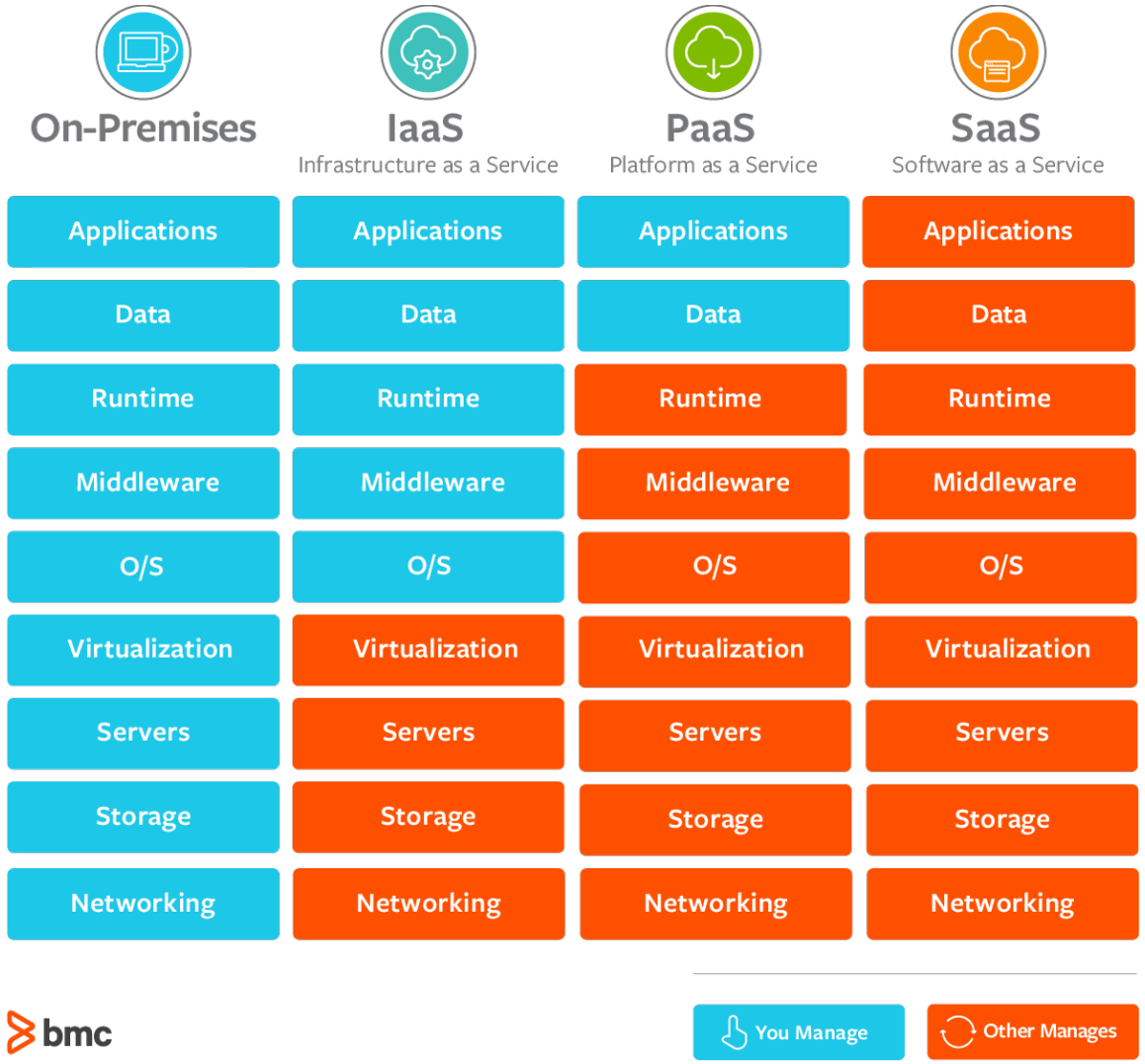




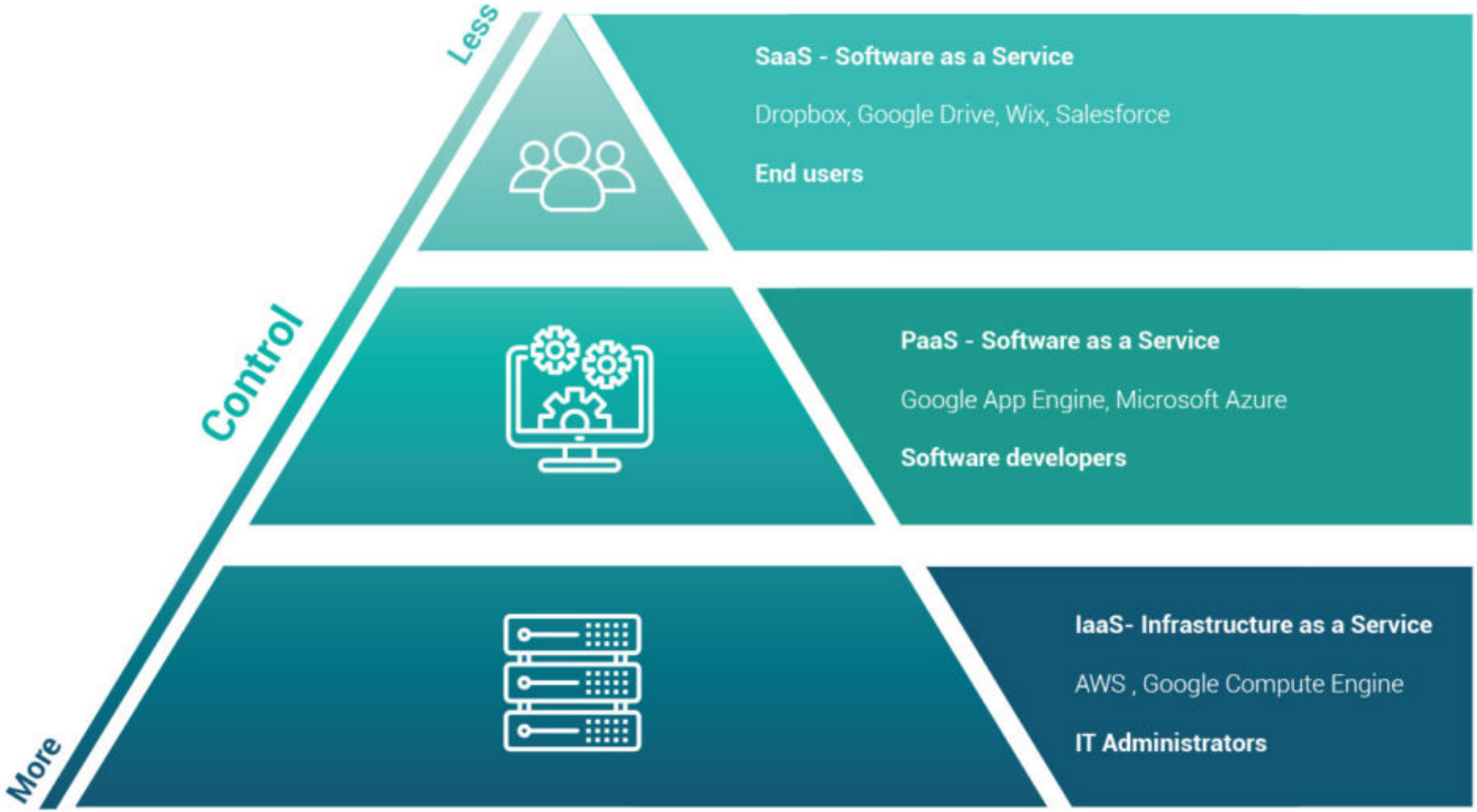
Cloud models: IaaS, PaaS, SaaS



Cloud models: management



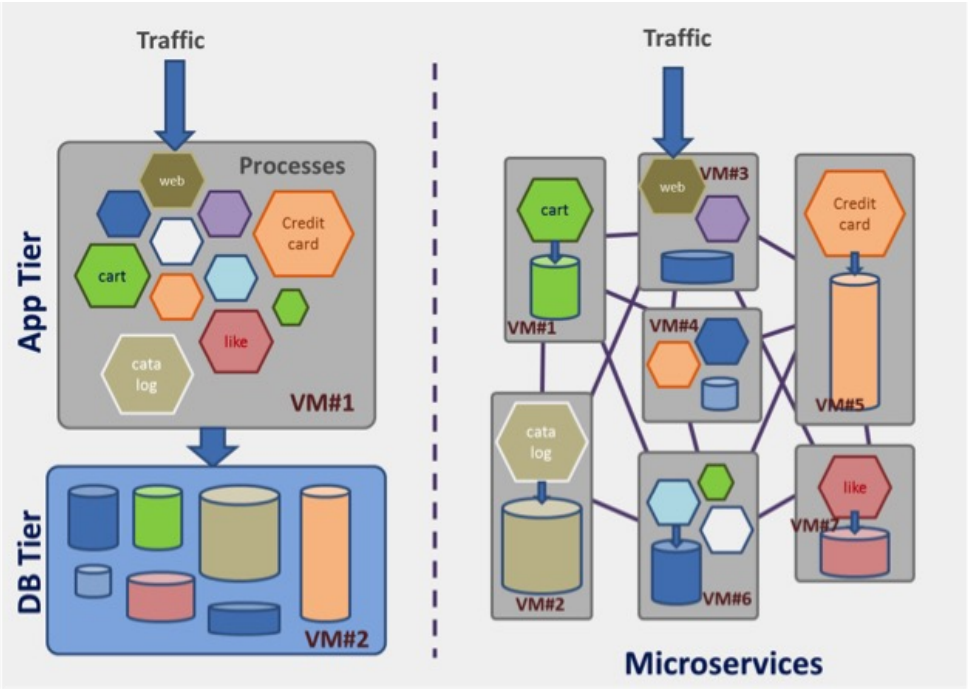
Cloud models: target users



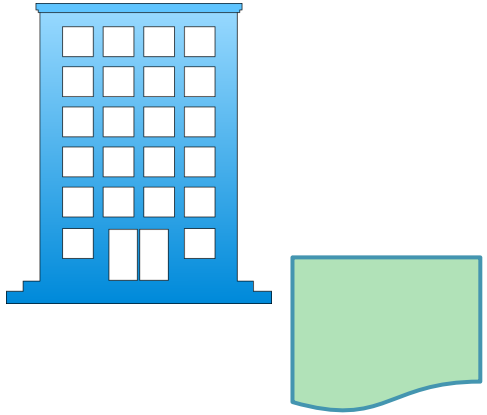
Beyond plain virtualization

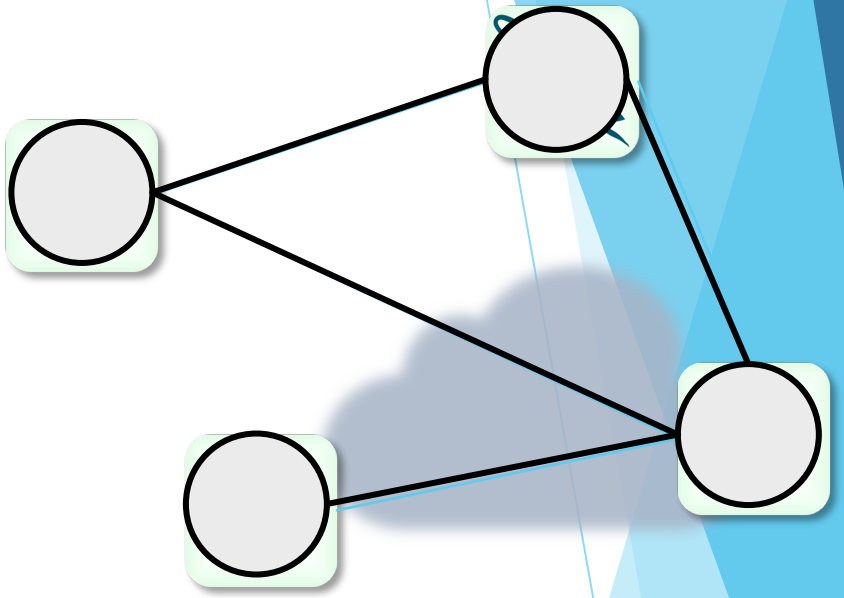
New paradigms for software development, deployment and operation:

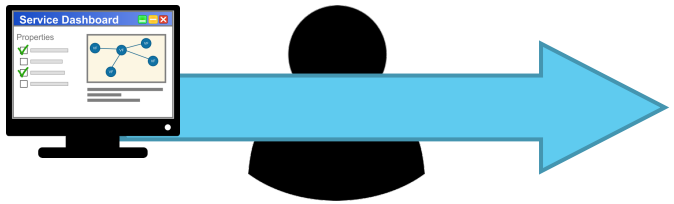
- ✓ From "code *writing*" to "*service chaining and configuration*"
- ✓ Re-usable components
- ✓ Automated lifecycle management
- ✓ Context-awareness and adaptability
- ✓ Elastic services

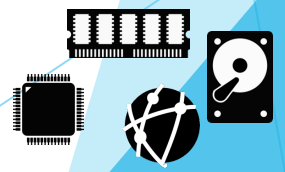


Software orchestration









Software orchestration

Software orchestration

Elasticity

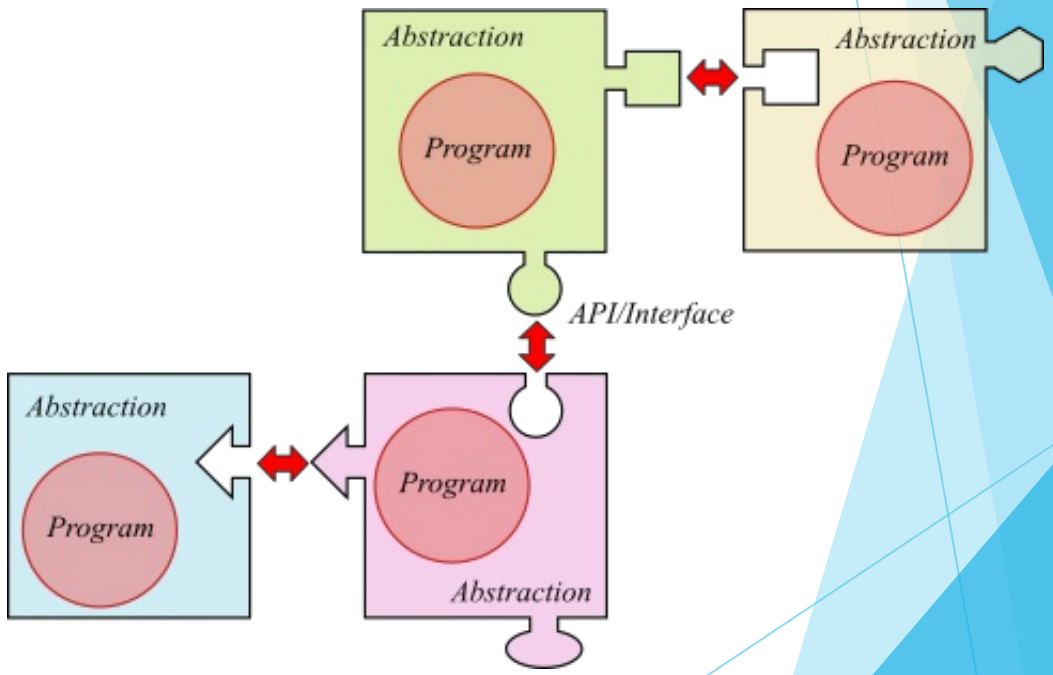
Partially unpredictable topology at design time!

Redundancy

Resilience

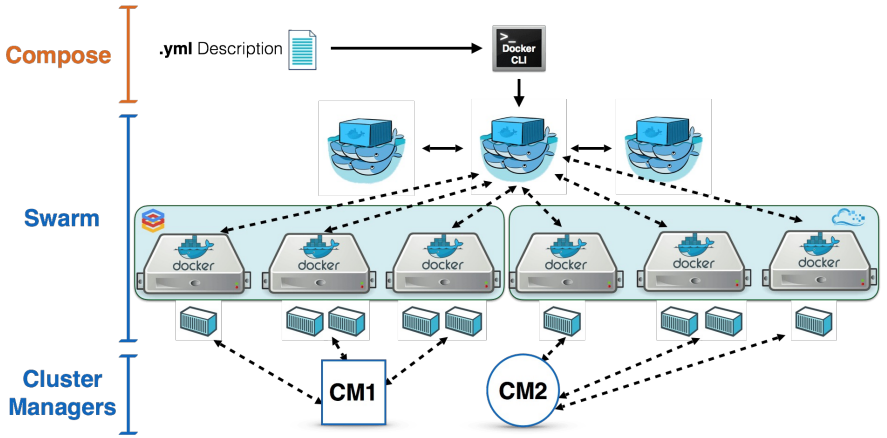
Service orchestration

- Service model
  - description
  - capabilities (what the component does)
  - properties (configuration elements),
  - requirements (virtual resources, libraries, other components), and
  - management operations (e.g., install, start, stop, scale, monitor).
- Orchestration
  - deploy the service in the underlying virtualization infrastructure
  - perform life-cycle management actions.
- Examples: TOSCA, ETSI NFV, IETF SFC, FIWARE, Kubernetes, Docker Swarm, Apache Mesos.

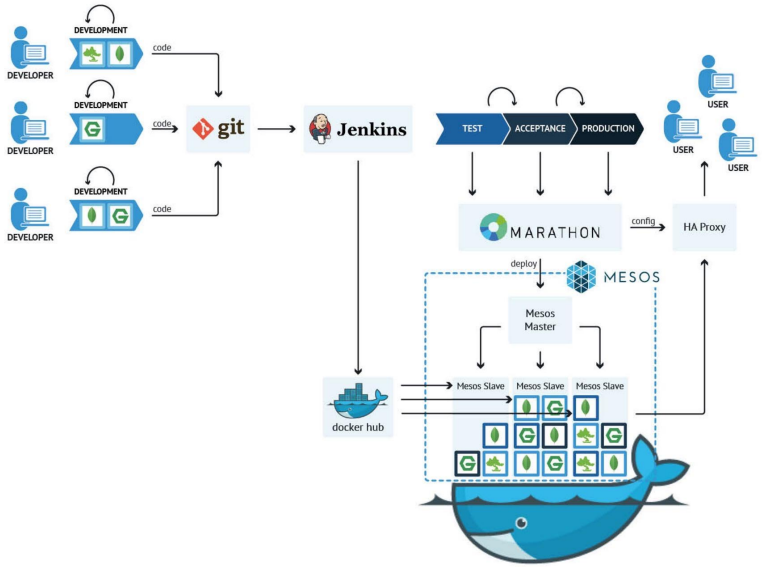


Orchestration - Examples

Docker Swarm

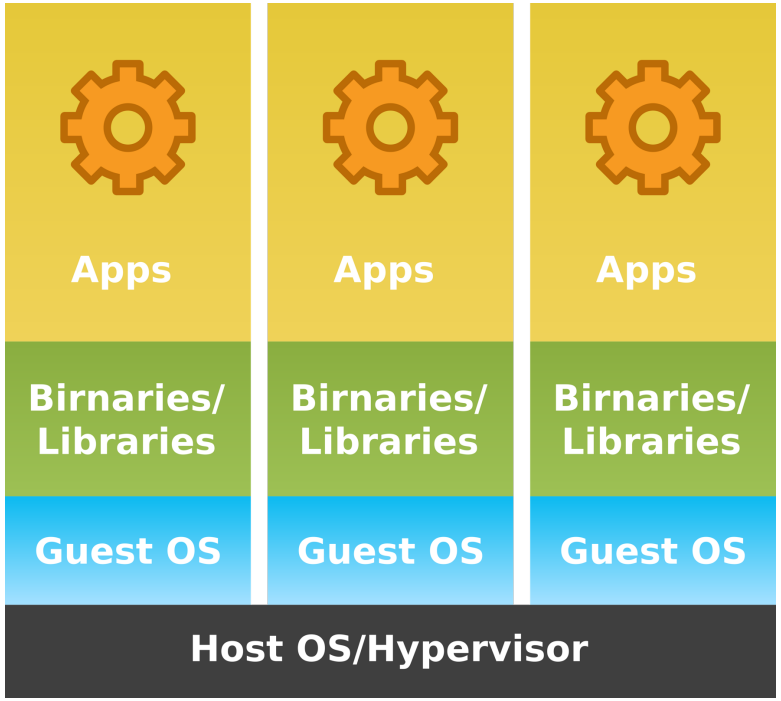


Apache Mesos

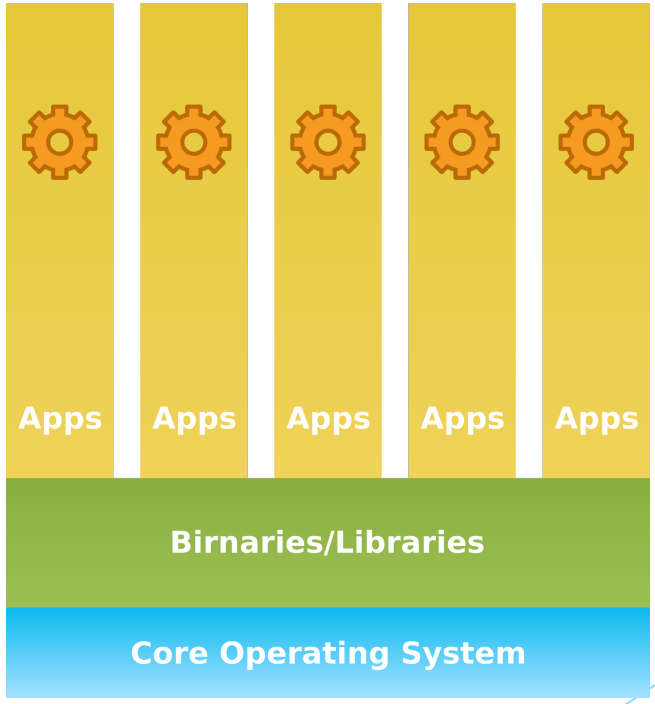


Trends in virtualization…

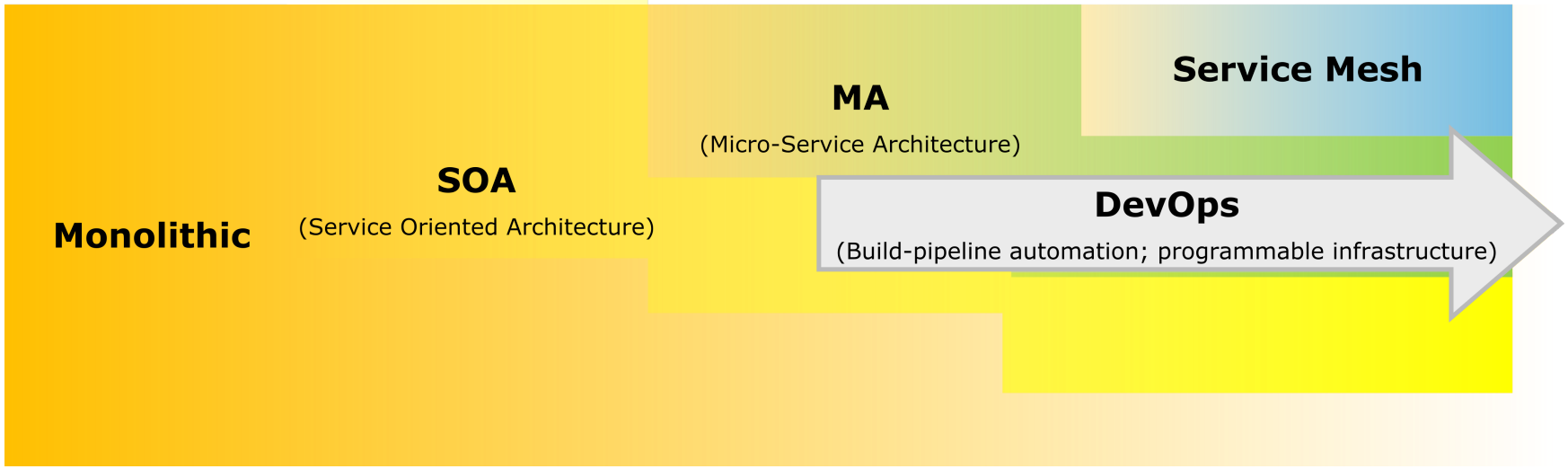
From VMs…

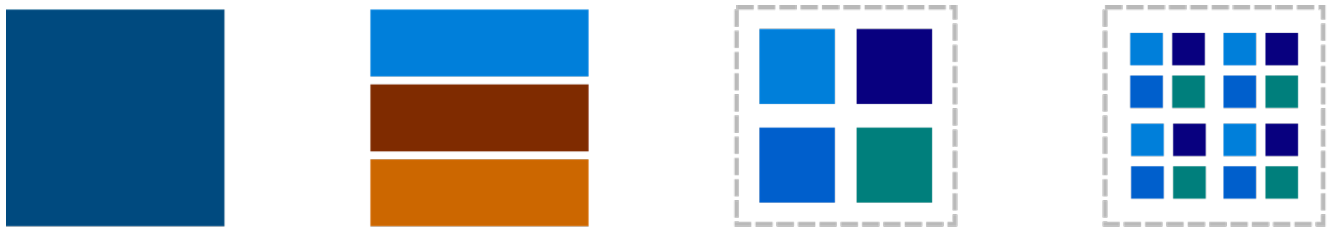


to lightweight containers



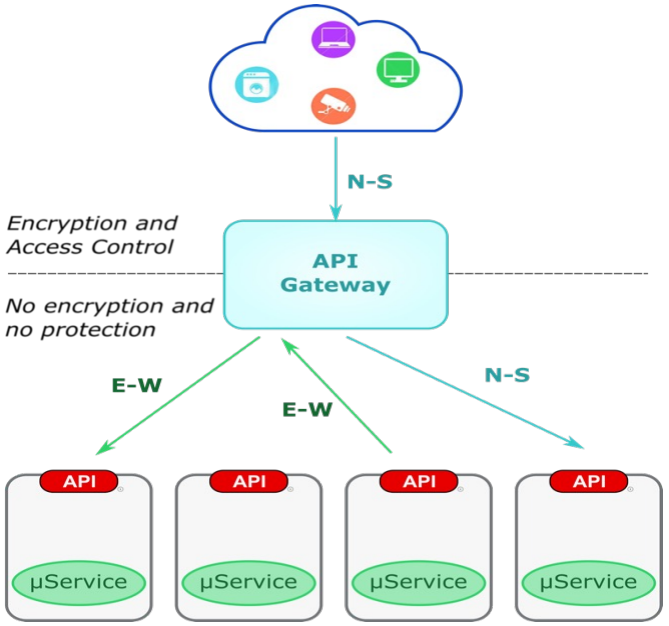
… and software architectures



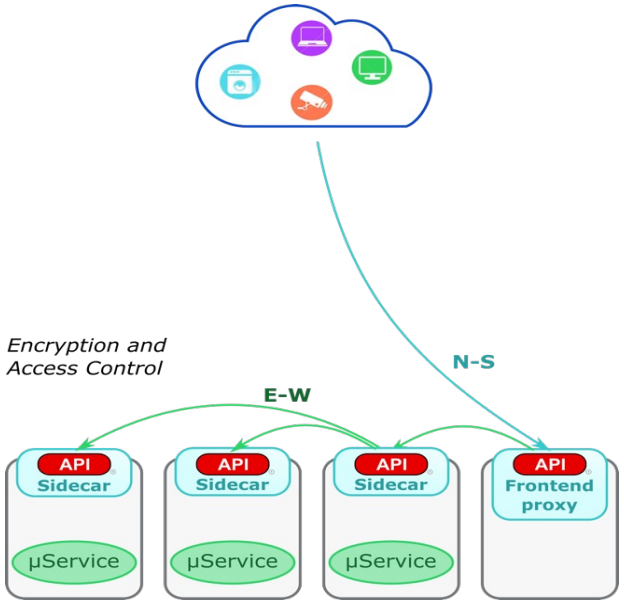


(Micro-)service-oriented architectures

Micro-services



Service meshes



Beyond the cloud: the Internet of Services

Internet of Things

+

Communication networks

+

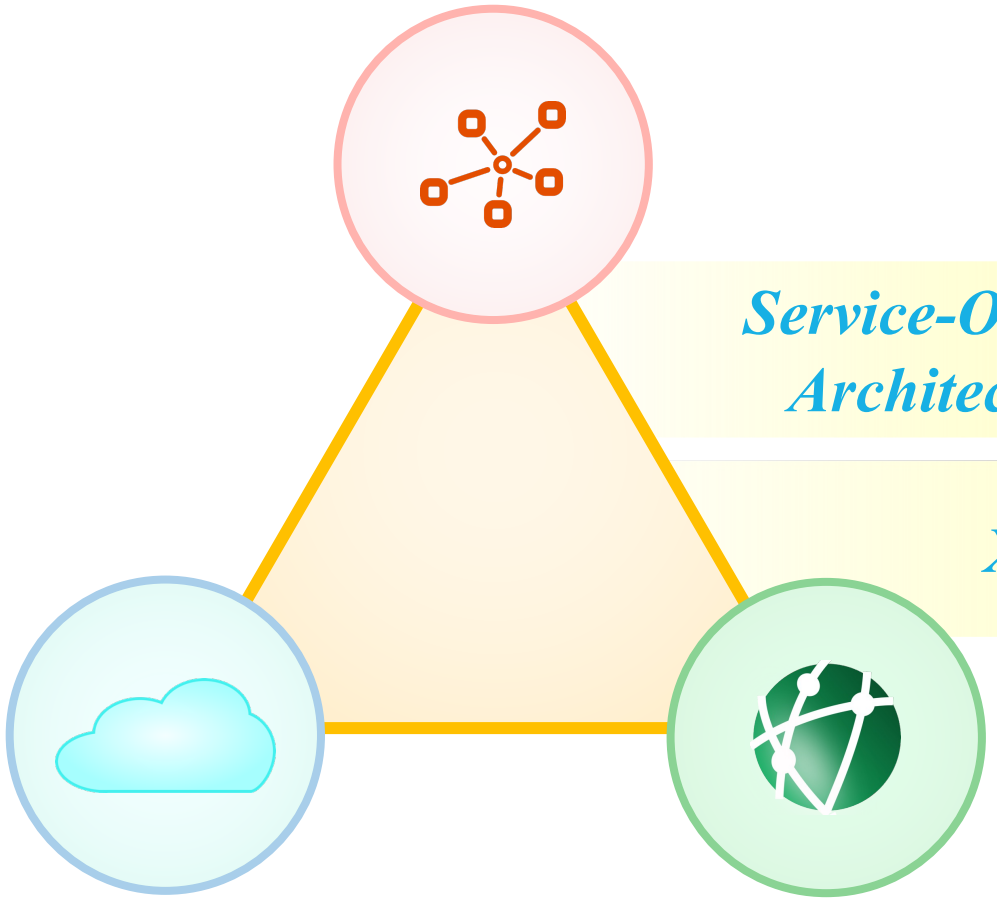
Internet of services

Physical surrounding

=

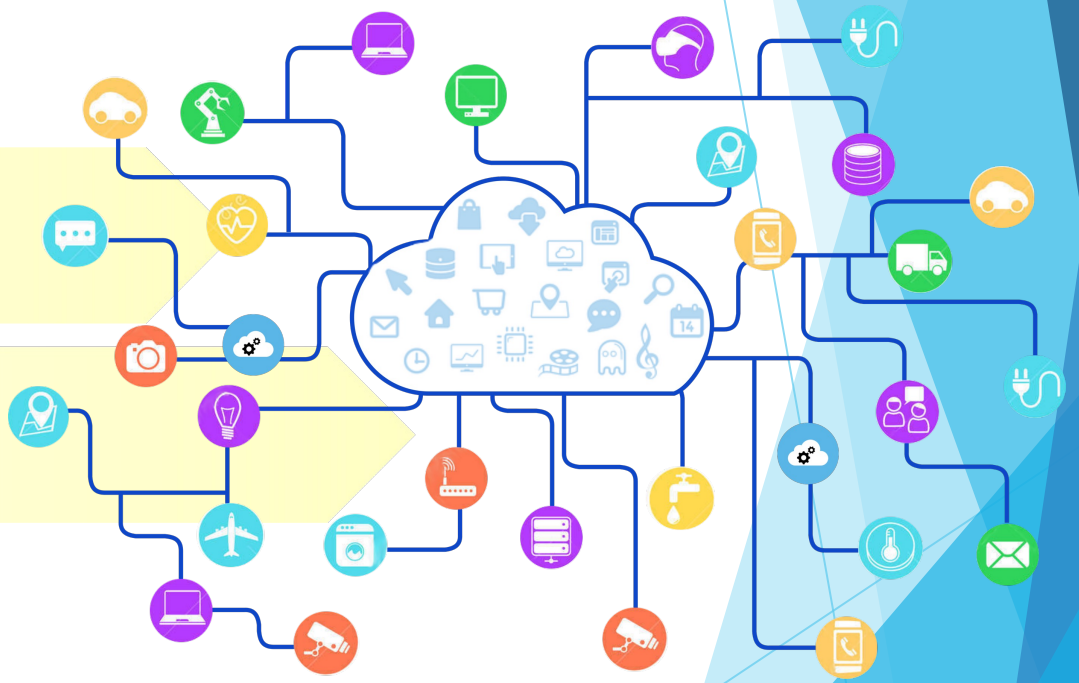
Smart services

The Internet of Services

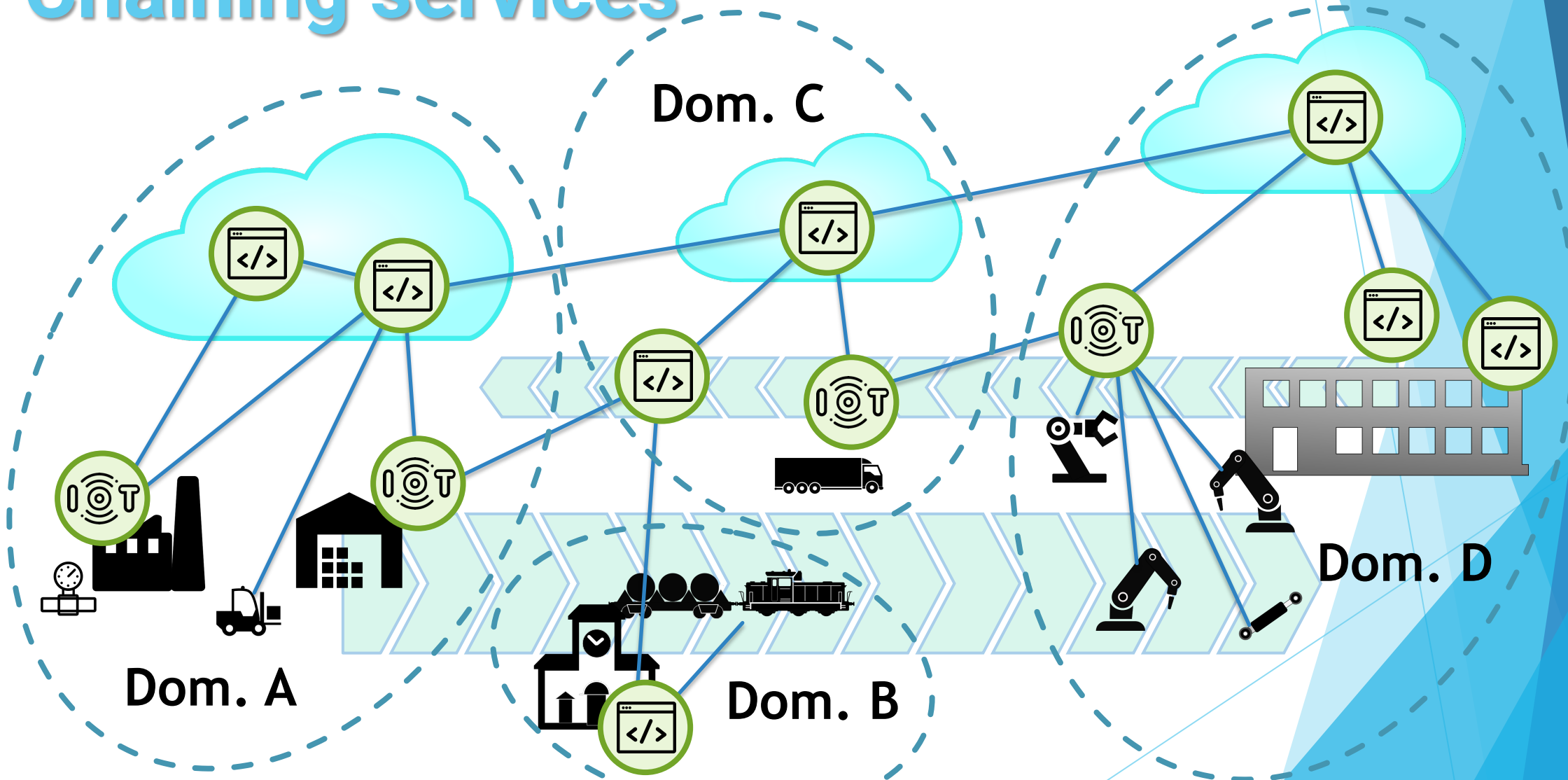


Service-Oriented Architectures

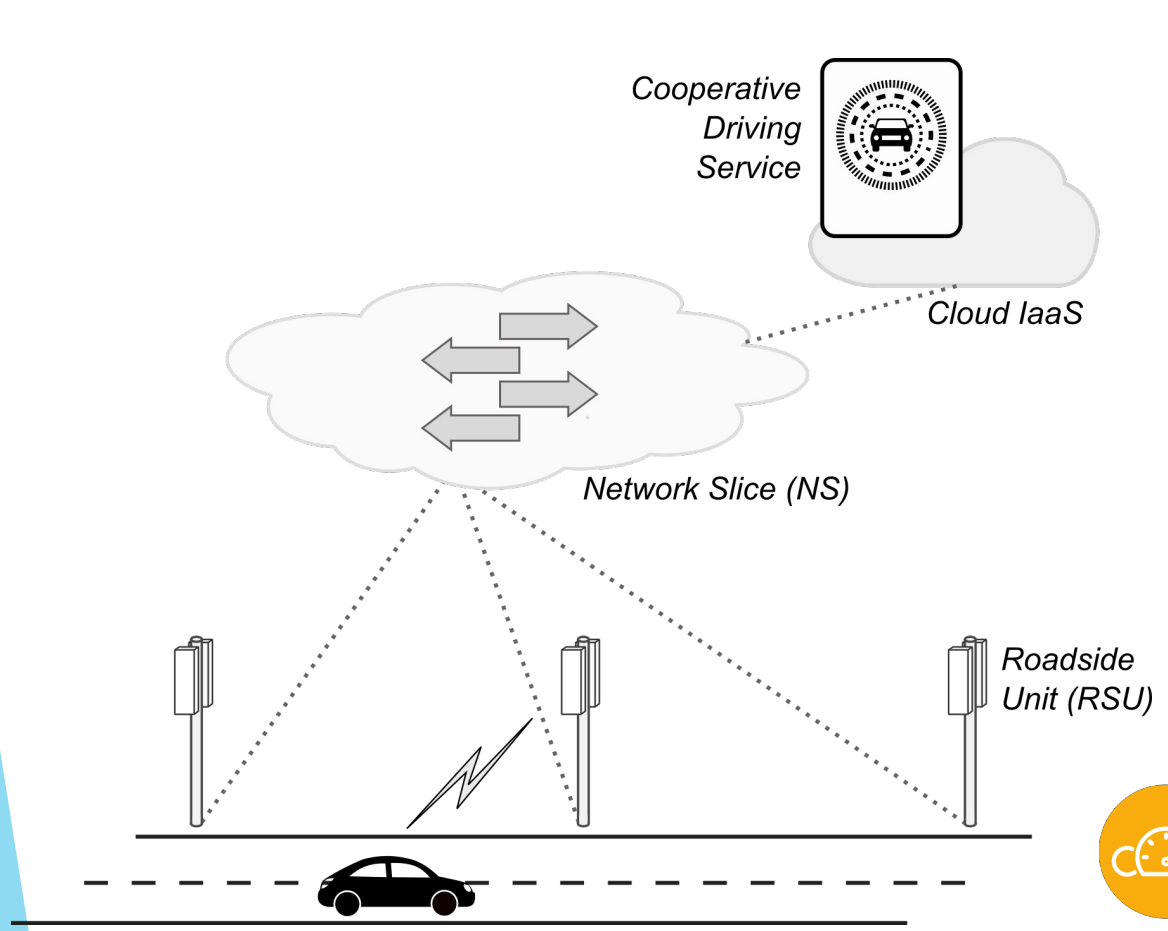
XaaS

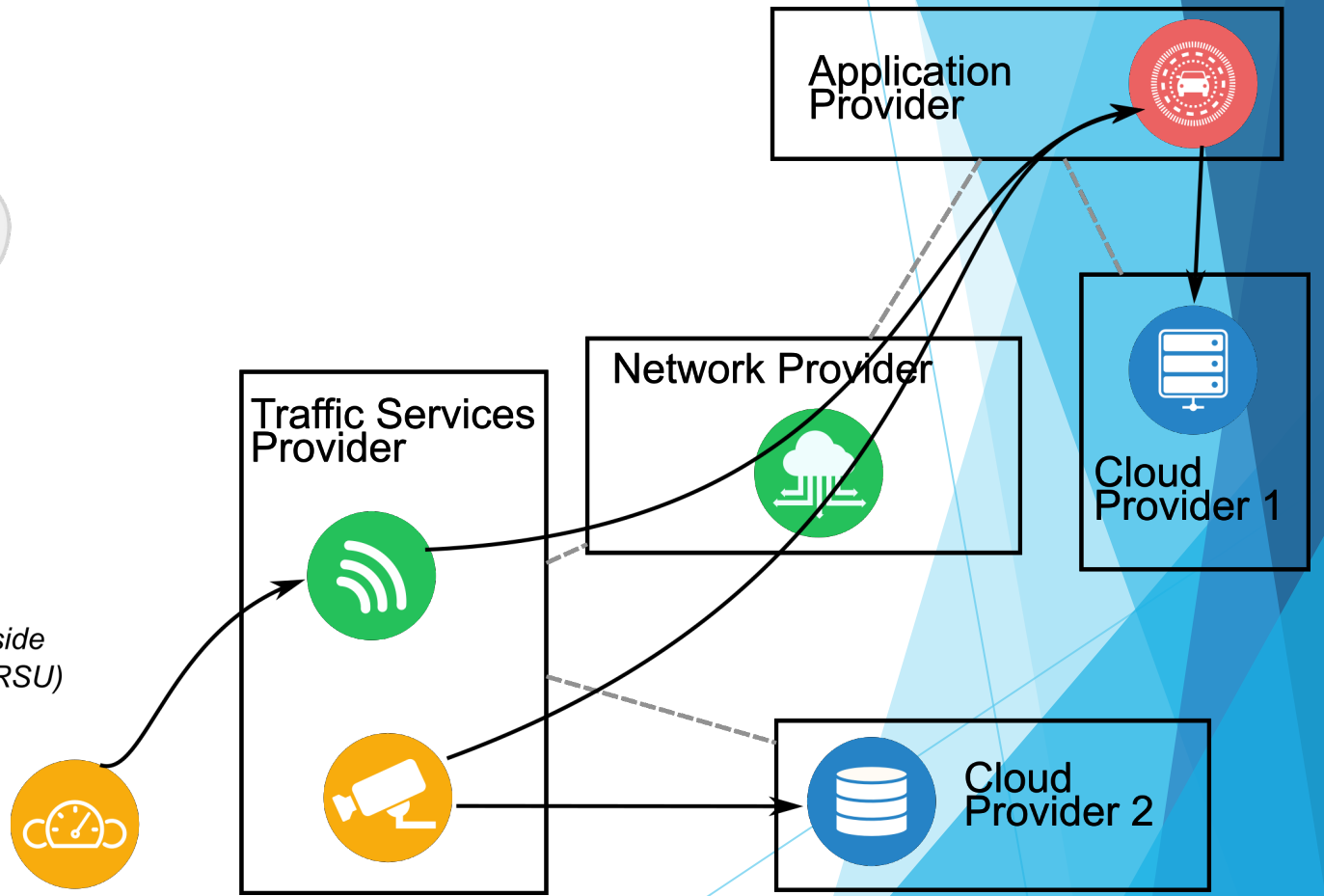


Chaining services



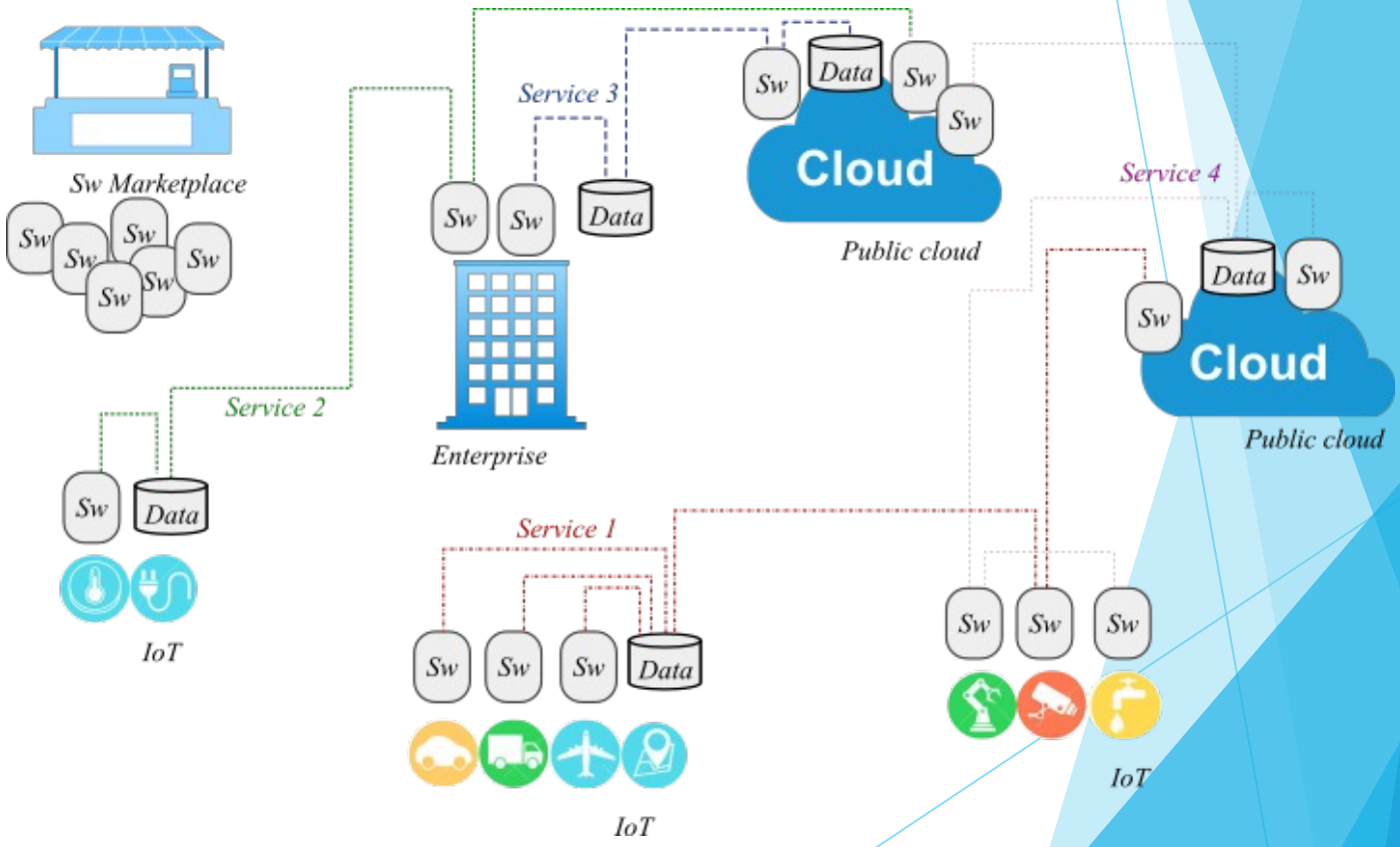
Chaining services: example





Digital business chains

- Dynamic composition of
  - software
  - services
  - data
  - smart devices
- Elastic topologies
  - model-driven or data-driven
- Programmable infrastructure
  - cloud, SDN, IoT



… and security?

Virtualization *Cloud* *IoT* *Fog/edge computing* *Service meshes*

ility **Multi-tenancy** **Dynamicity** **Heterogeneity** **Autonomicity** **Multi-domain**

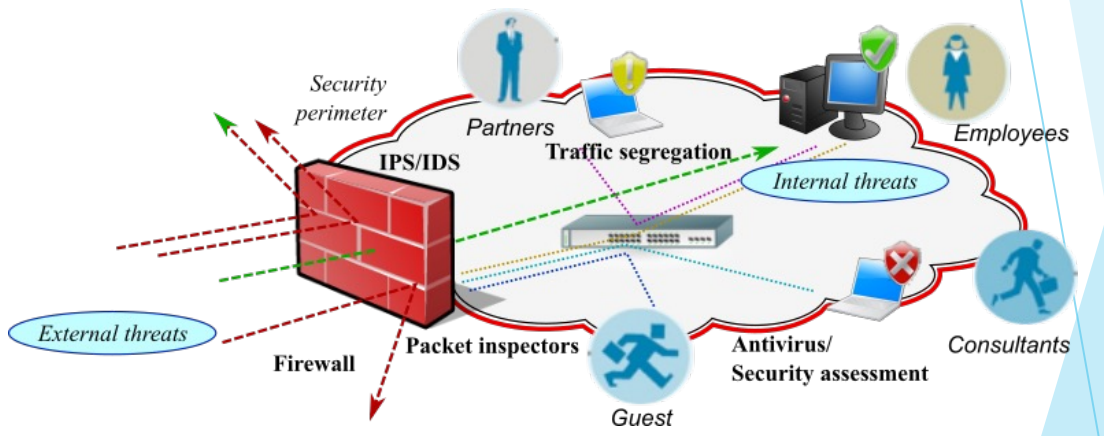
urity perimeter

lity **Inefficiency**

Security concerns

The “security perimeter” model

- Shortcomings:
  - rigidity
    - network partitioning
    - HW/SW security appliances
    - routing/switching policies
  - market segmentation
    - network packets bounced across security appliances
      - for analysis, inspection, mitigation, and processing
    - redundant inspection/analysis
    - limited scope security service

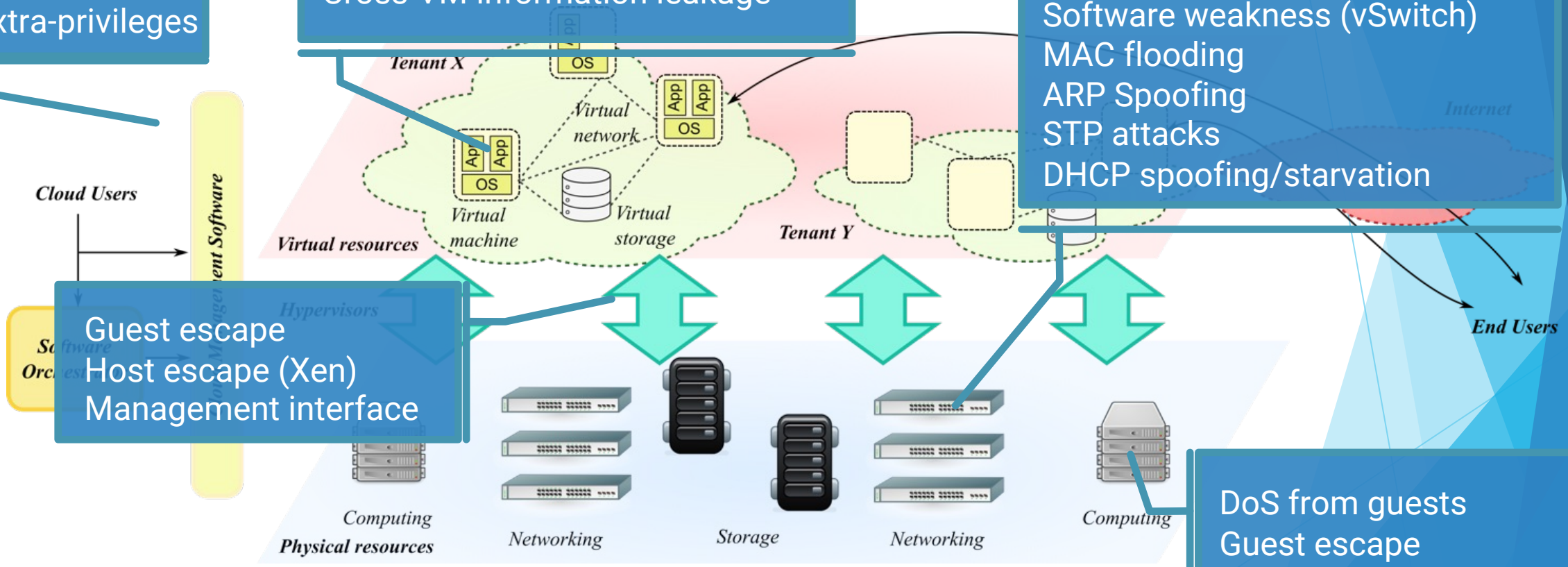


Security threats in cloud infrastructures

Extra-privileges

Internet to guest (OS and applications)
Privilege escalation
Cross-VM information leakage

Lack of encryption in management networks (storage, migration)
VLAN hopping
Software weakness (vSwitch)
MAC flooding
ARP Spoofing
STP attacks
DHCP spoofing/starvation



Guest escape
Host escape (Xen)
Management interface

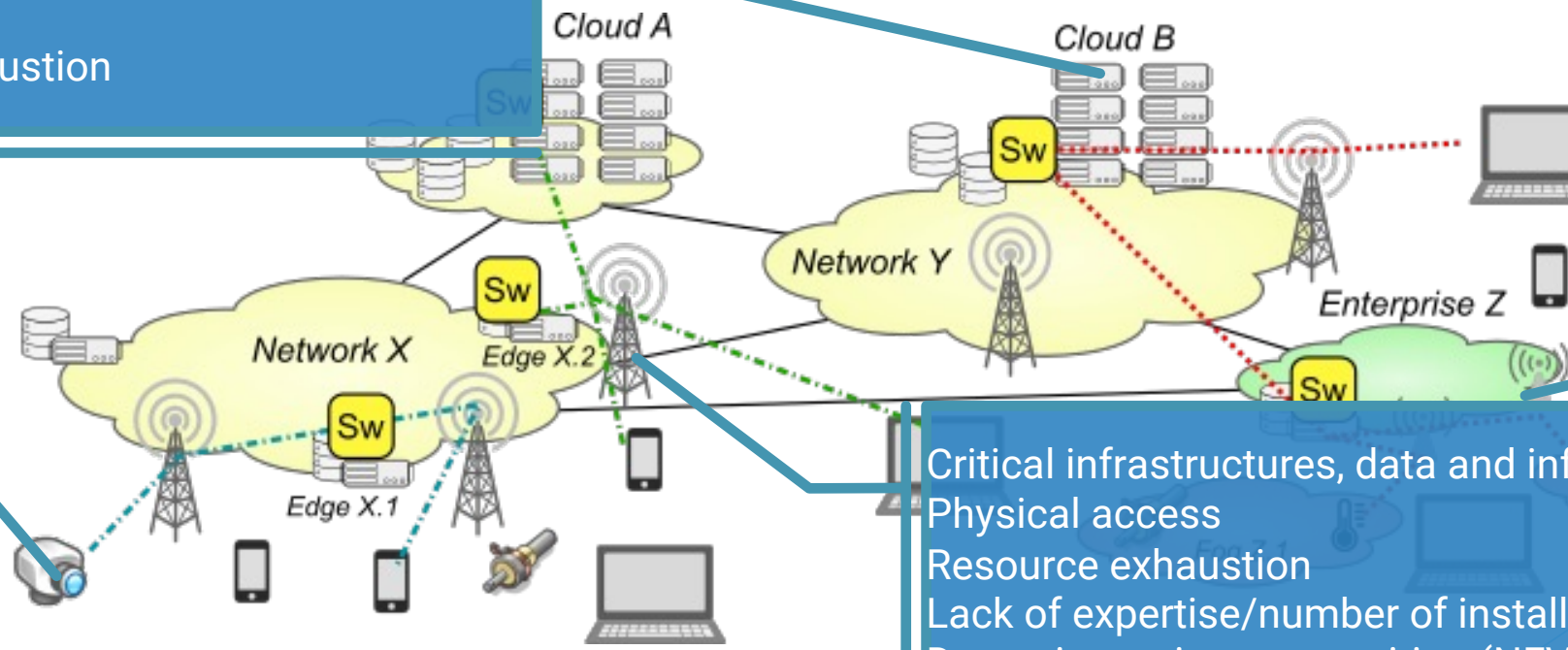
DoS from guests
Guest escape
Arbitrary code execution
Arbitrary file write
Internet to host

Security threats: multi-infrastructures

Broader range of attack models
Increased attack surface
Multi-tenancy
Lack of hardware acceleration
Lack of trusted platform modules

Public Internet connection (LTE, WiFi)
Weak security mechanisms by design
Poor configuration
Tampering
Resource exhaustion

Personal and portable devices, multihoming
Road warriors
Removable storage
Outsourcing



Critical infrastructures, data and information leakage
Physical access
Resource exhaustion
Lack of expertise/number of installations
Dynamic service composition (NFV, MEC)
Intrusion and privilege escalation

Security challenges

- Slow and ineffective **detection** of attacks
  - low visibility, limited scope of algorithms
- Difficulty in identifying **new threats** and vulnerabilities
  - ineffective correlation of data from multiple sources, lack of big data techniques
- Outdated and inefficient **architectures**
  - sharing the security context, balance granularity with overhead
- Technology and business **lock-in**
  - no common interfaces/architectures

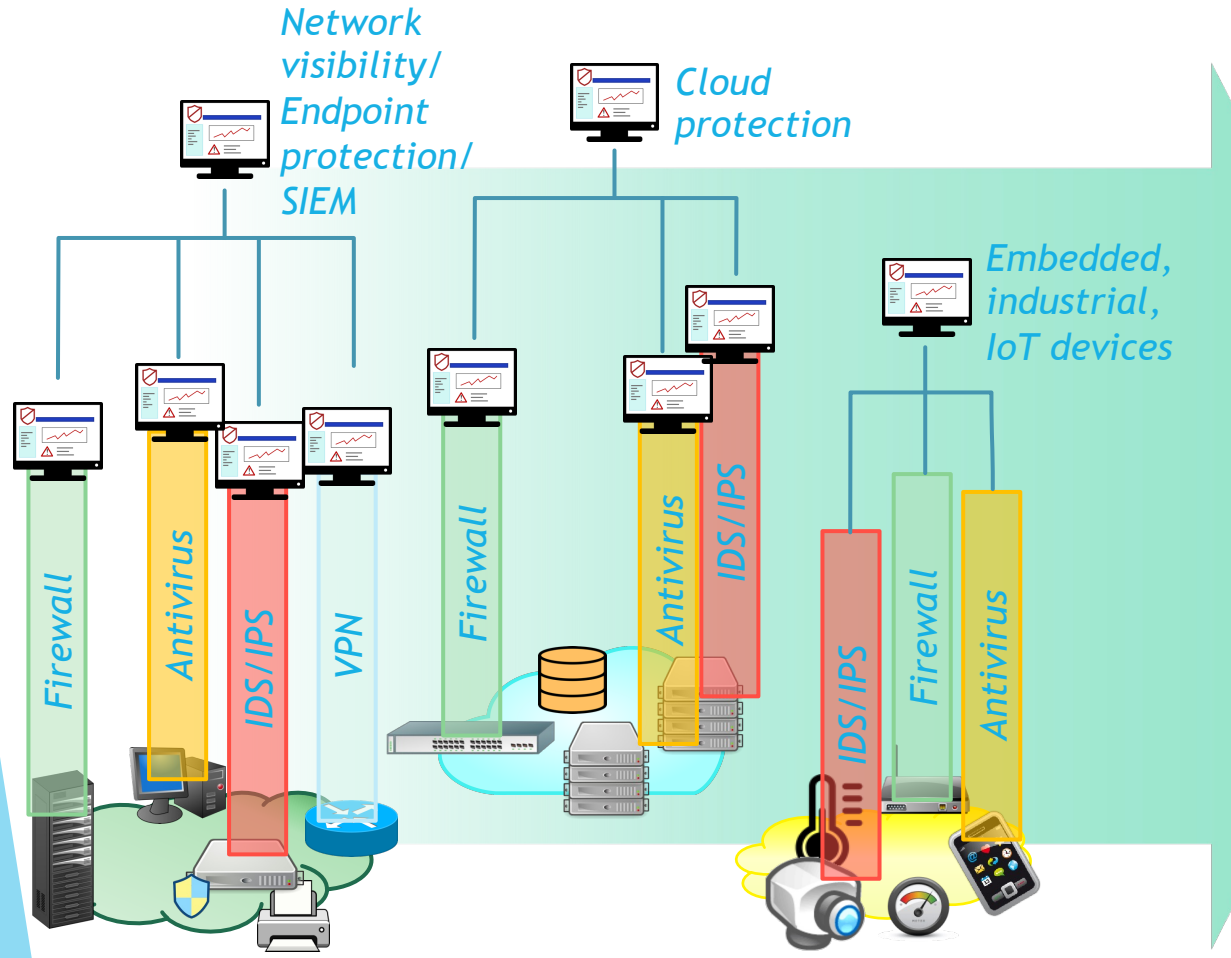
Security challenges

- **Rigidity**
  - network partitioning, security processes, routing and switching policies
- Slowness to **share** the knowledge
  - mostly based on paperwork and manual processes
- Ineffective **awareness** to users
  - dashboards mainly designed for technical operators (no end users, management, legal staff)

Security challenges

- Growing **complexity**
  - human errors in design, implementation, configuration, and management
- **Trustworthiness** and reliability of the overall end-to-end service
  - integrity and dependability of the whole chain beyond identity management and access control
  - tracking the propagation of private data and sensitive information
  - weak links represent a privileged attack vector
- **Dynamic composition**
  - the chain topology and composition are usually unknown
  - difficulty to assess and certify trustworthiness for multi-domain and multi-tenancy services

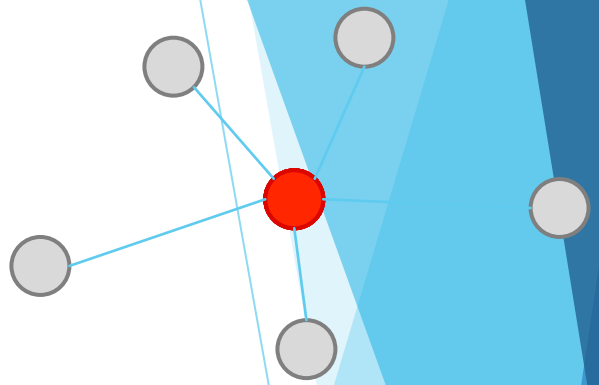
Emerging trends



- *From centralized to distributed architectures*
- *Leveraging the programmability of the infrastructure*
- *Efficiency vs performance*
- *Reliability and data protection*
- *Orchestration and automation*
- *Data correlation*
- *Visualization, representation, and sharing*

Emerging trends

- From centralized to more distributed architectures
  - Micro-firewalls, distributed firewalls;
  - Log collectors, integrated security solutions, SIEM.
- Leverage network programmability
  - From SNMP, NetFlow, sFlow, IPFIX to OpenFlow and NetConf;
  - From stateless to stateful inspection and monitoring.

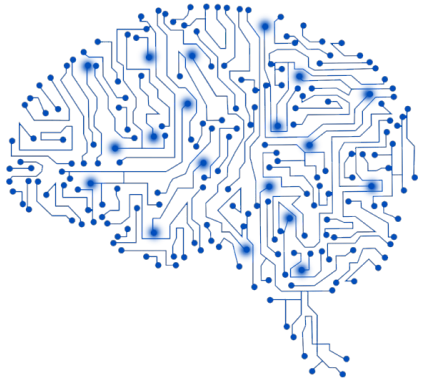


Emerging trends

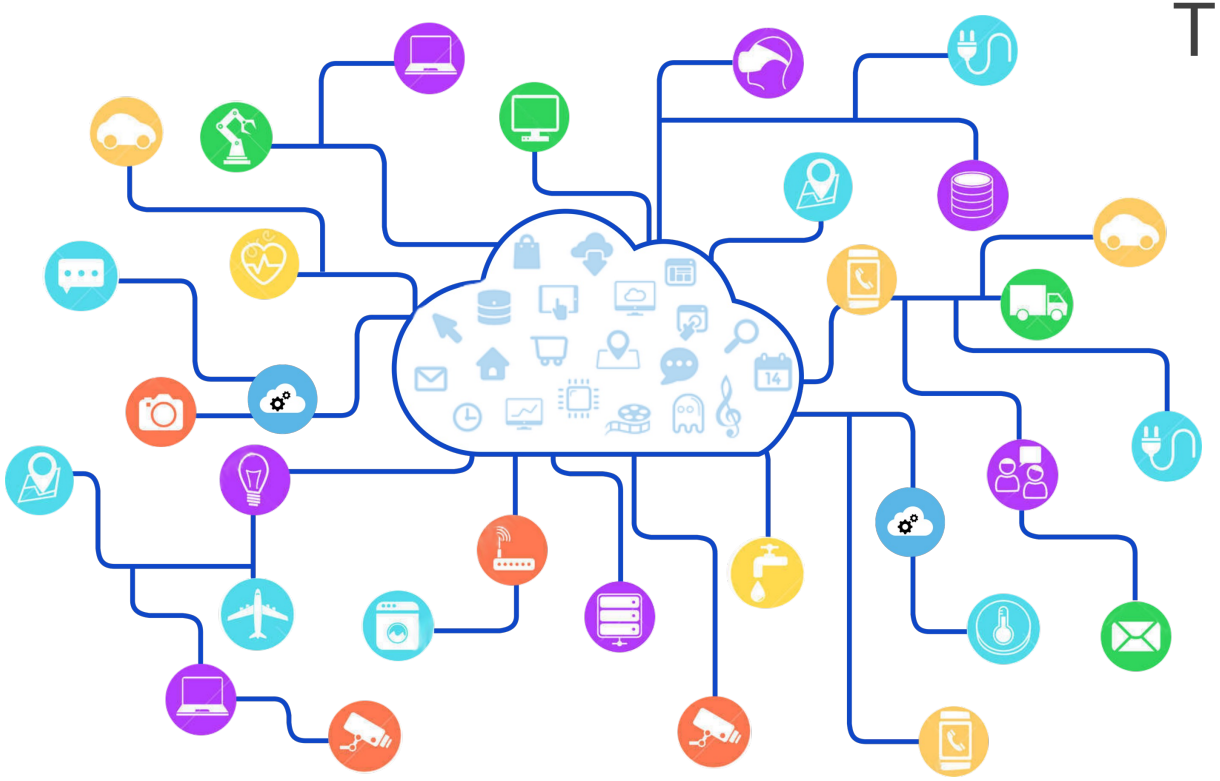
- Boost efficiency and performance
  - from simple memory-less string matching to stateful rules;
  - applications are the main targets for attacks;
  - growing number and complexity of protocols;
  - multi-vector attacks.
- Trustworthiness and data protection
  - certification of origin and integrity
  - data location and propagation

Emerging trends

- Software orchestration
  - smart devices and applications with programmable reporting capabilities;
  - policy-driven automation;
  - formal verification methods.
- Data correlation
  - machine learning and AI;
  - information sharing.
- Situational awareness
  - visualization, representation, sharing.



The cloudification wave



The “cloudification” wave

- massive virtualization
- multi-tenancy
- externalization
- DevOps
- software orchestration
- elastic services
- cloud-native applications

Security aspects for cloud-based services

- Service integrity
  - timely detect any attack or threat that may compromise the integrity, confidentiality, or availability of data and processes.
- Service trustworthiness
  - trust software developers, things, vendors, service providers, infrastructures, data sources, …
- Data sovereignty
  - know who, how, and where will process private data and sensitive information.

Challenges

- Infrastructure vs Virtual Resources
  - many providers and different cloud models
  - different responsibilities
- heterogeneous interfaces
- ever-evolving attack patterns
- broad threat landscape
- dynamic topologies
- visibility
  - software, configuration, communications

Azure

aws

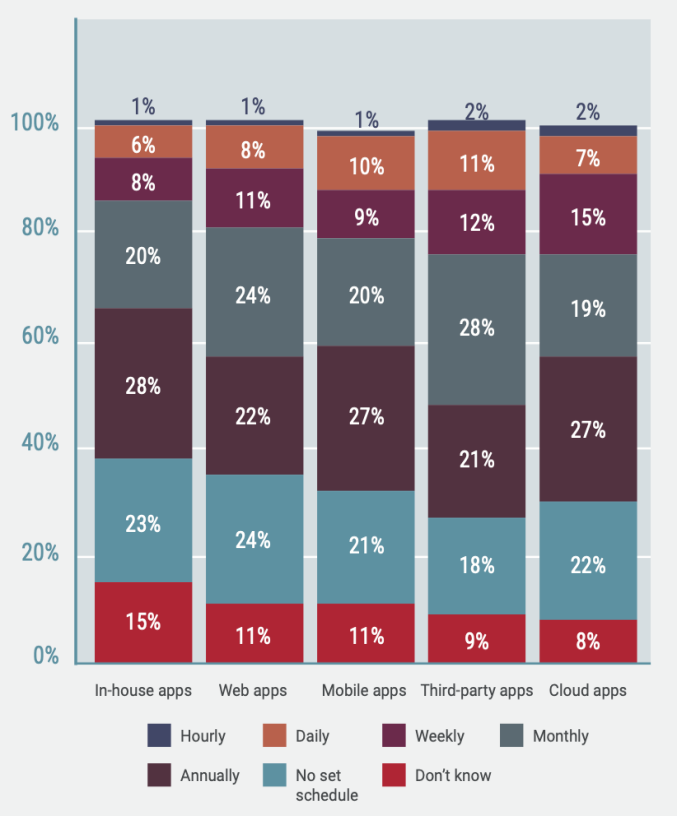
ORACLE Cloud

IBM Cloud

Technical issues: faster software releases

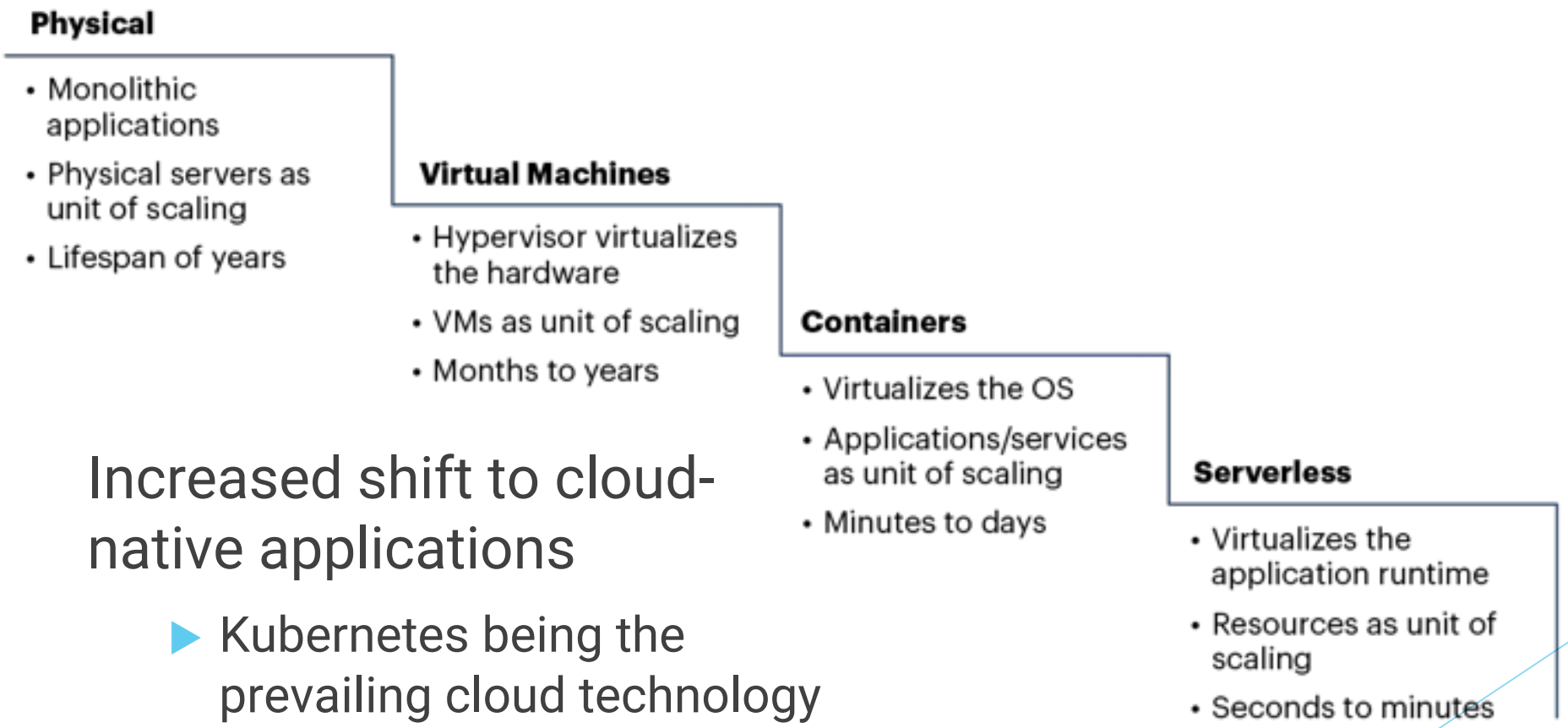
Faster release cycles of applications

- new computing paradigms improve the speed at which new software is delivered;
- this complicates security analysis, which has not improved at the same pace.



Frequency of application changes

Technical trends: evolution of workloads



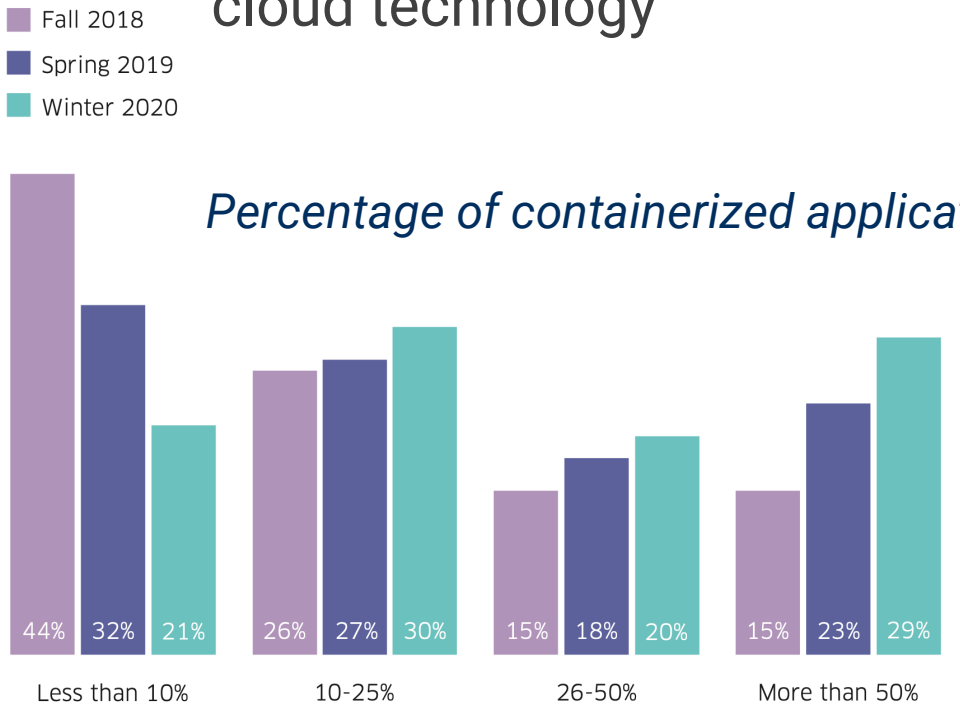
Increased shift to cloud-native applications

- Kubernetes being the prevailing cloud technology

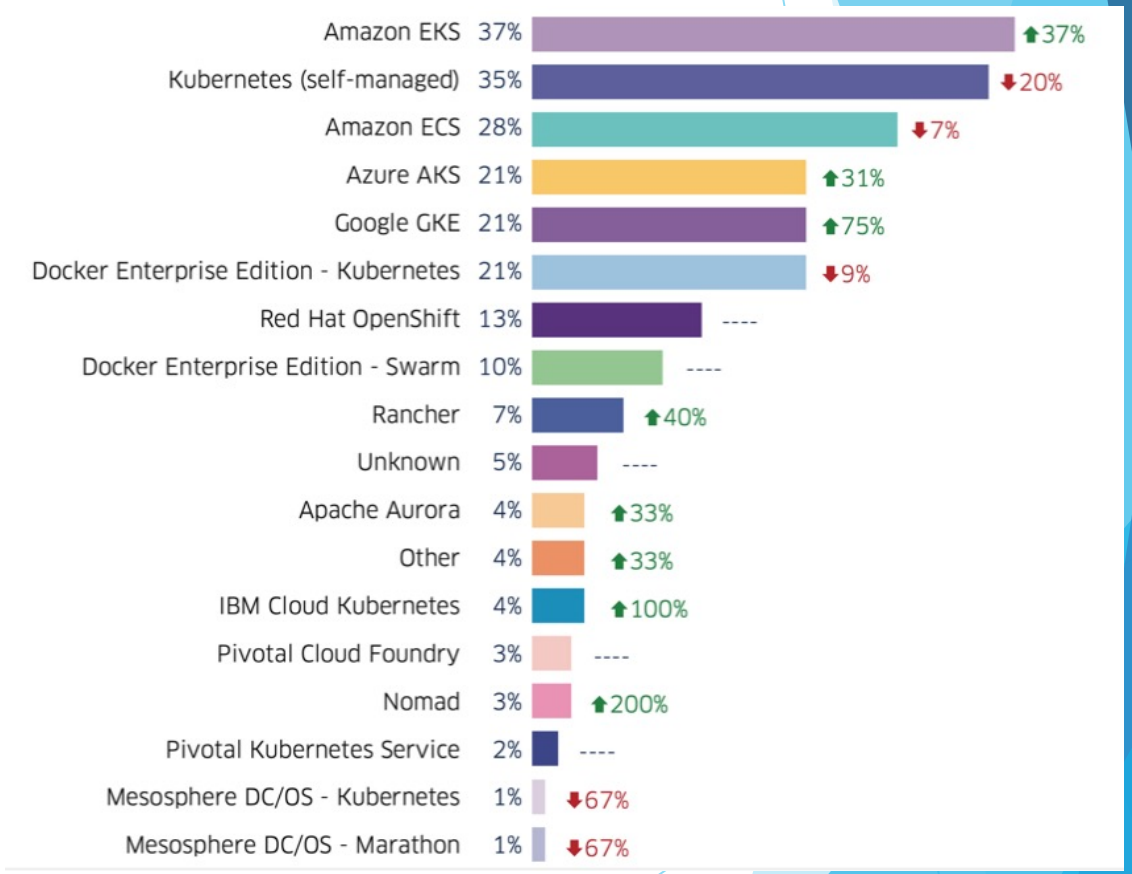
Technical trends: containers and Kubernetes

Increased shift to cloud-native applications

- Kubernetes being the prevailing cloud technology



Percentage of containerized applications



Usage of container orchestrators

DITEN Università di Genova

Technical trends: multi-cloud deployments

No vendor lock-in

Increased availability and resilience

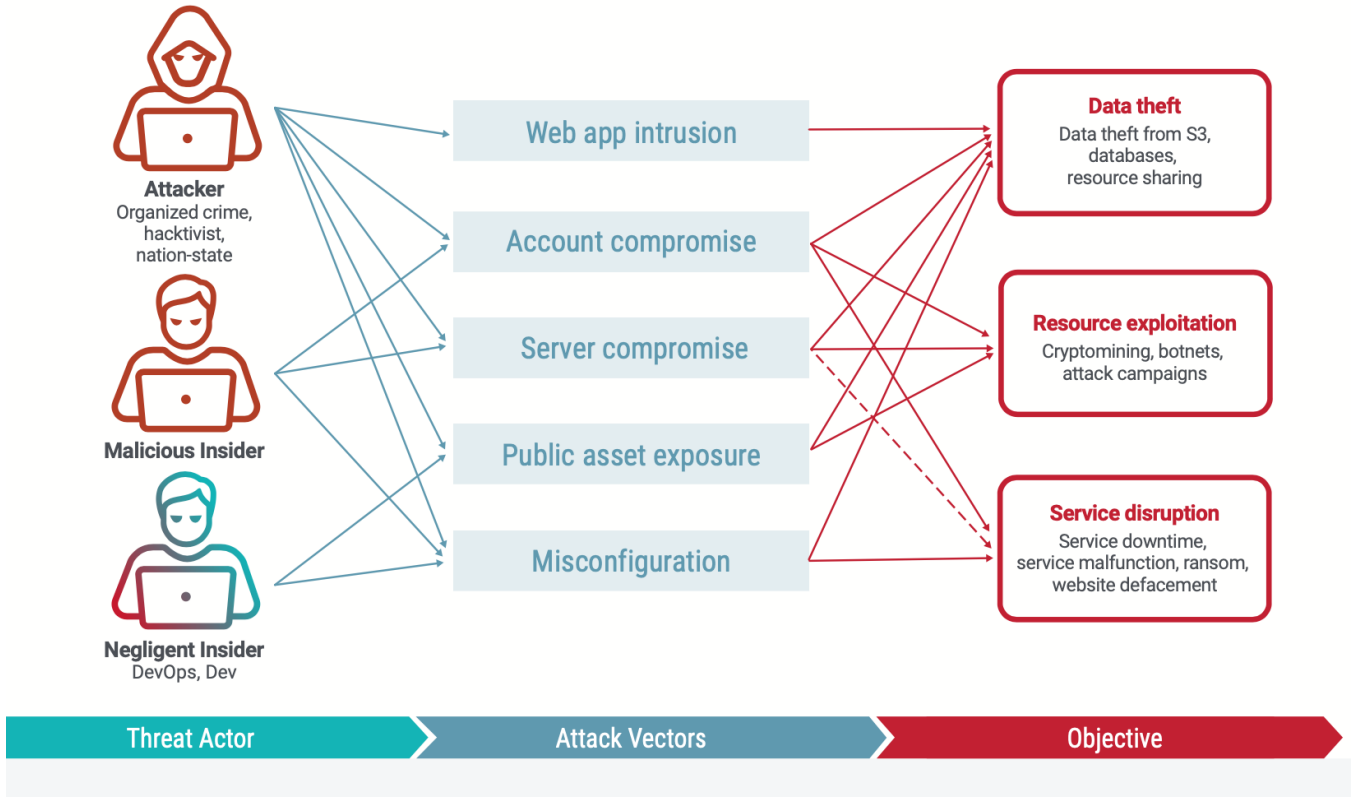
Geographical coverage

- performance, regulations

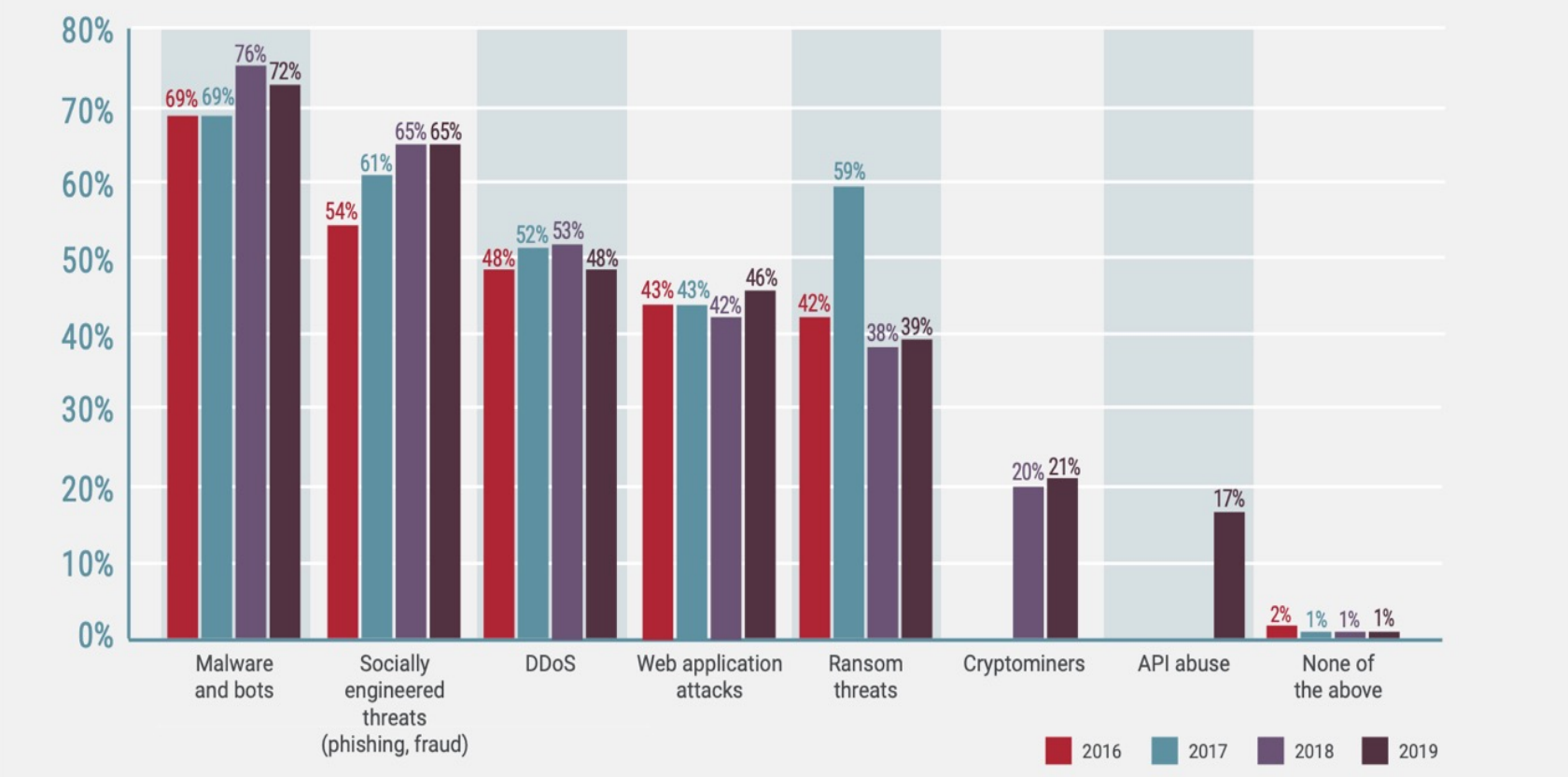
⛔ Heterogeneous interfaces and capabilities

- deployment, management, security

The cloud threat model



The cloud threat landscape



Top cloud threats

- Many security tools and features available for cloud applications
  - incorrect usage is a common issue.
- Typical causes:
  - low understanding;
  - limited visibility;
  - fast software release cycles
  - multi-cloud deployments and heterogeneous security capabilities.

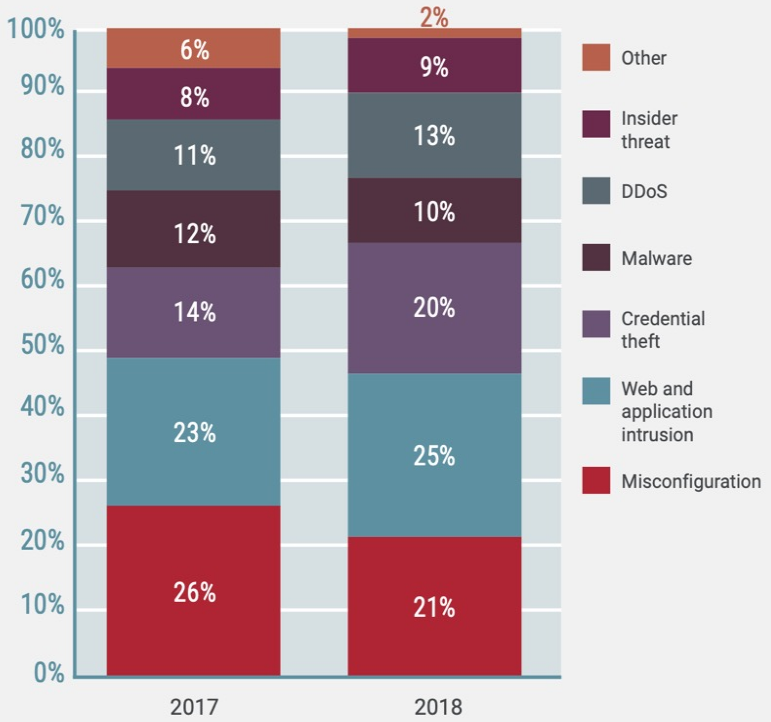
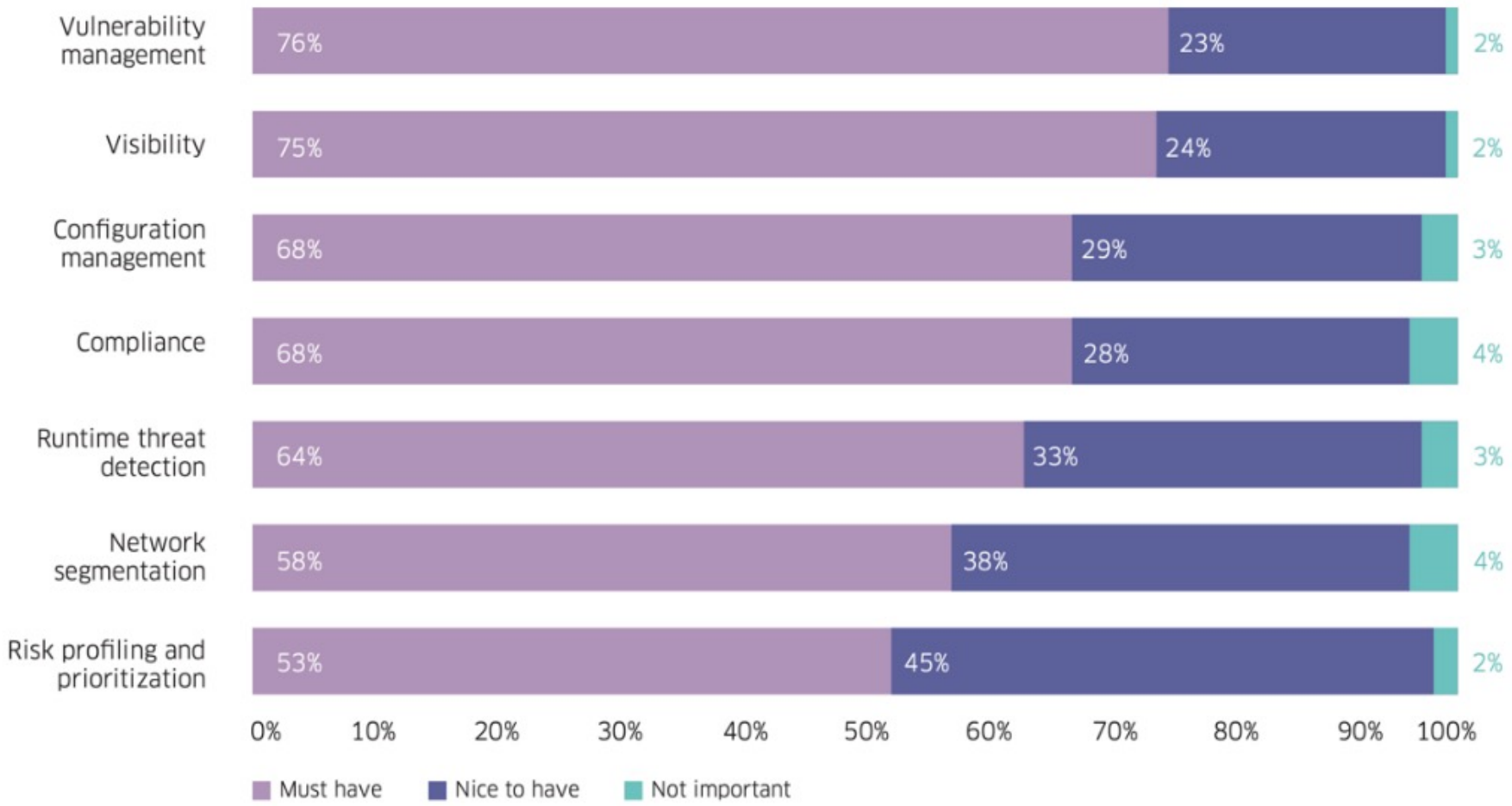


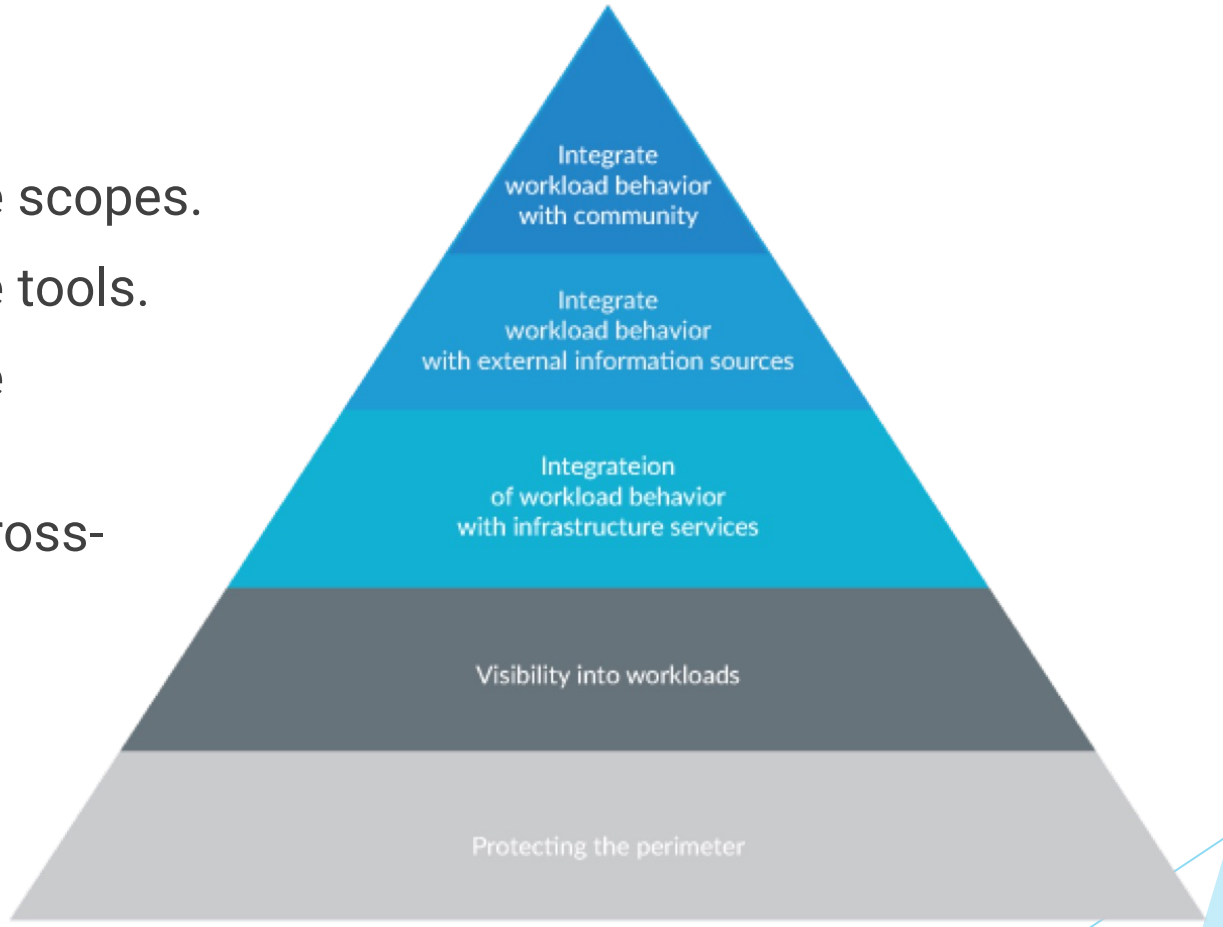
Figure 52. Future security threats to the cloud environment.

Top security features

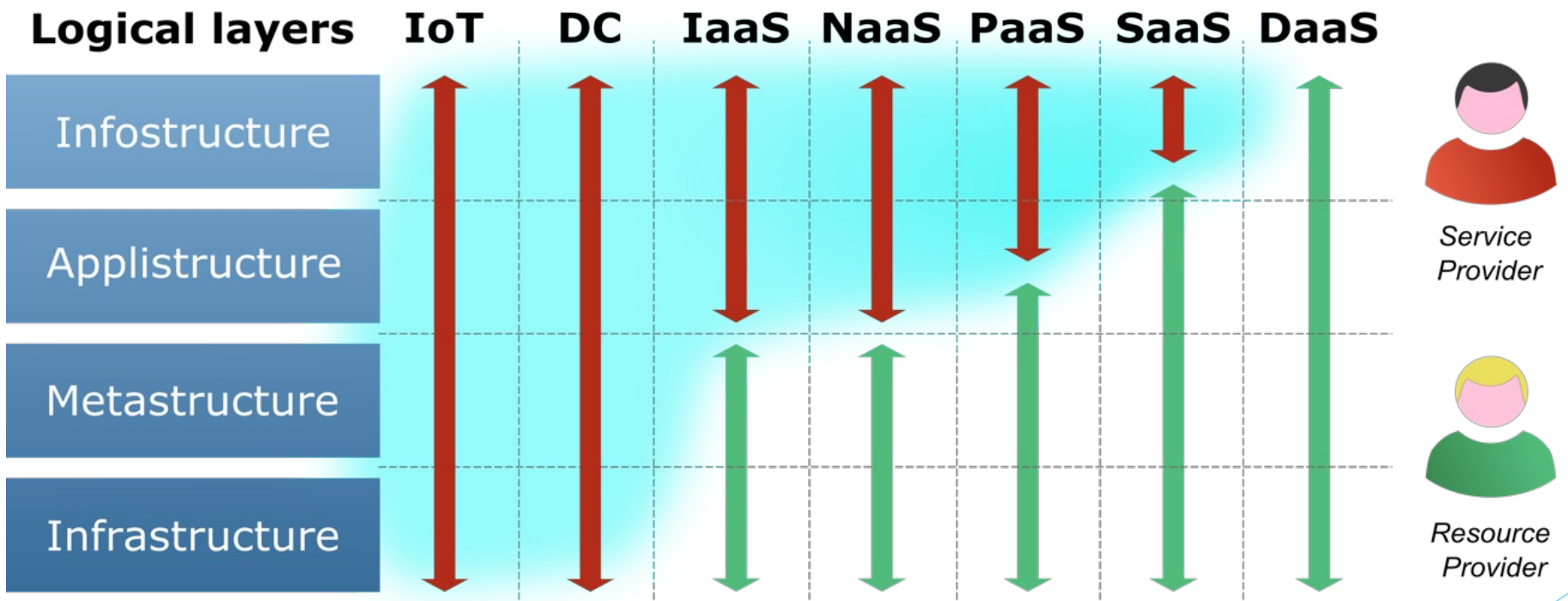


Cloud protection

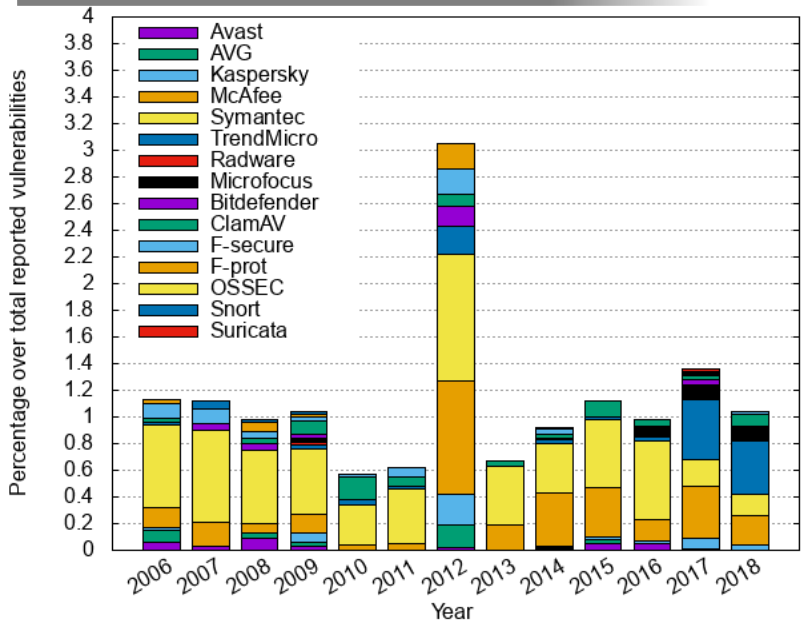
- Multiple layers, multiple scopes.
- Multiple layers, multiple tools.
- Multiple layers, multiple responsibilities.
- Multiple layers, many cross-relationships.



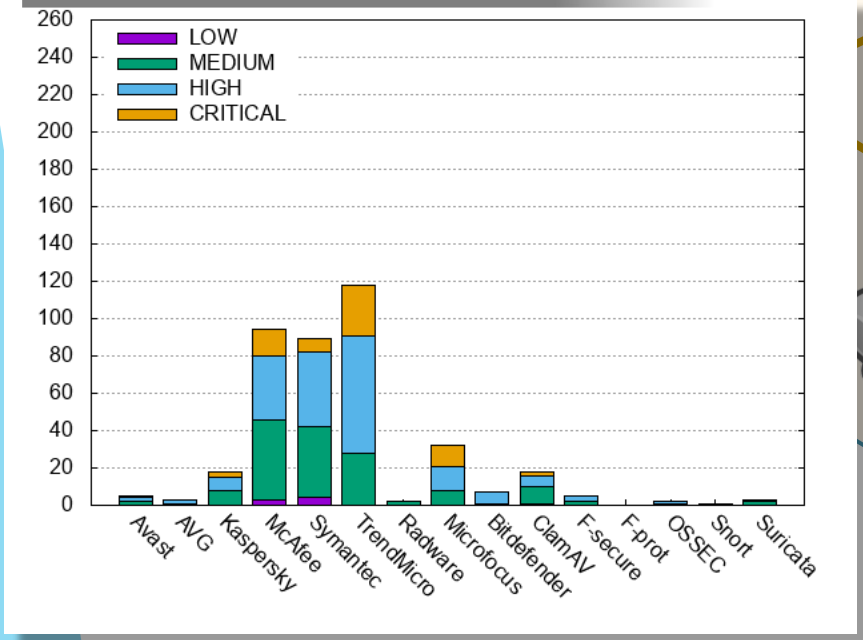
Understanding the sharing of responsibility



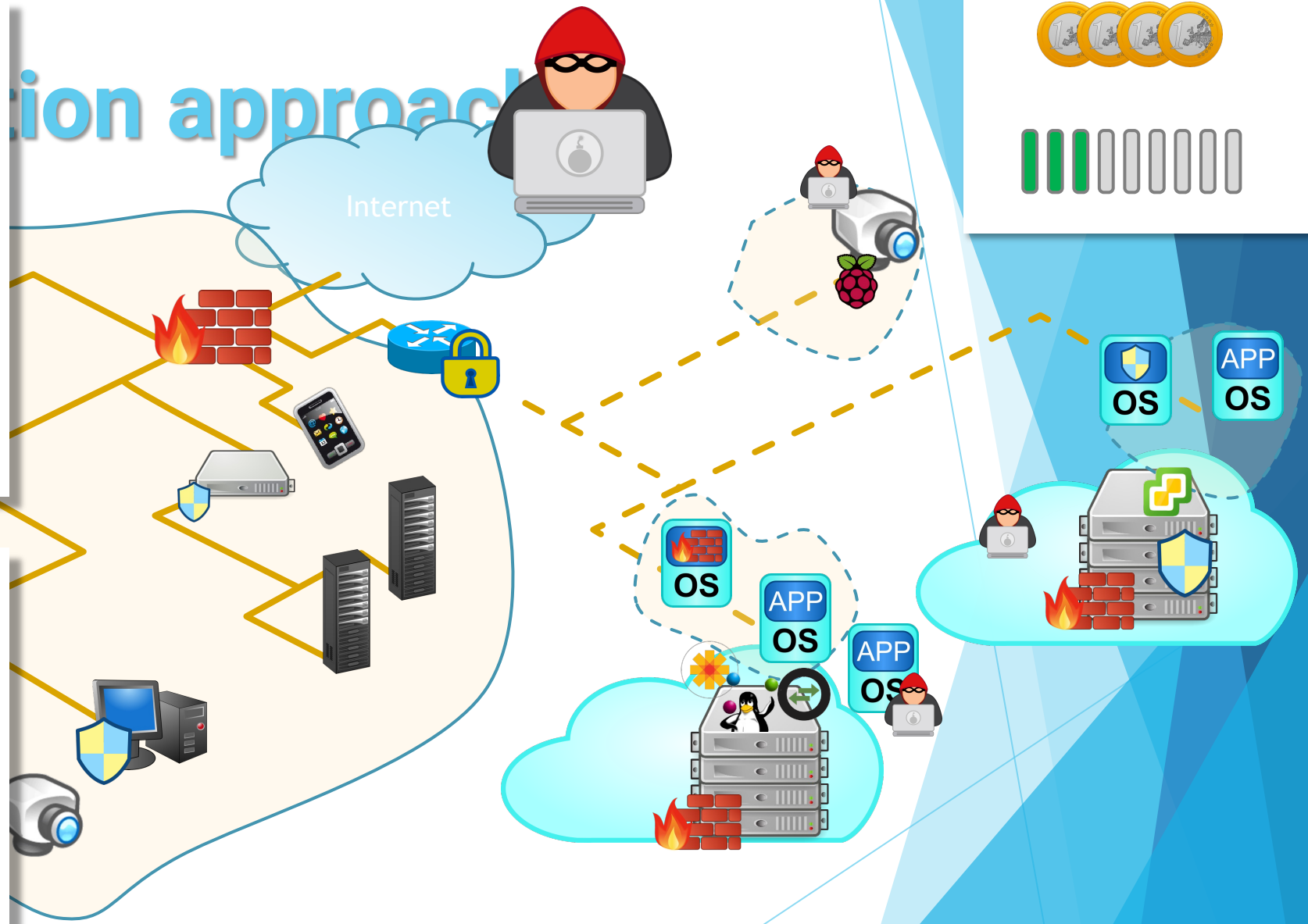
Reported vulnerabilities by year



Criticality



tion approach

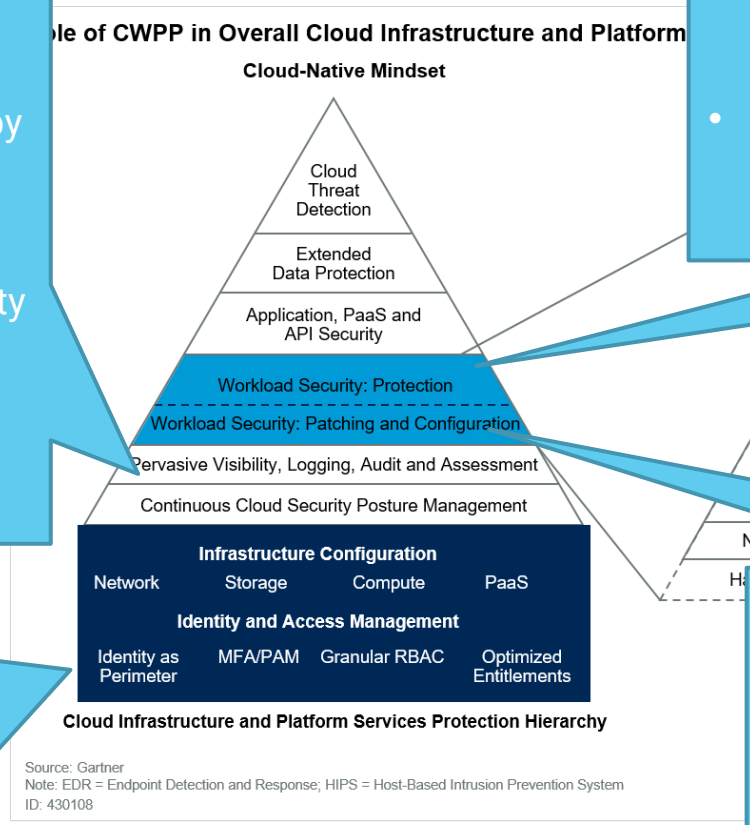


DITEN Università di Genova

Cloud protection hierarchy

Cloud Security Posture Management (CSPM)
- integrates security capabilities offered by cloud providers with the customer's security processes
- Cloud Access Security Brokers (CASB) - translate internal security policies to specific interfaces

Cloud Workload Protection Platforms (CWPP)
- provides "consistent visibility and control for physical machines, virtual machines, containers, and serverless workloads, regardless of location"
- brings end-point protection concepts to the cloud, taking into account specific characteristics: multi-tenancy, elasticity, resource limitations

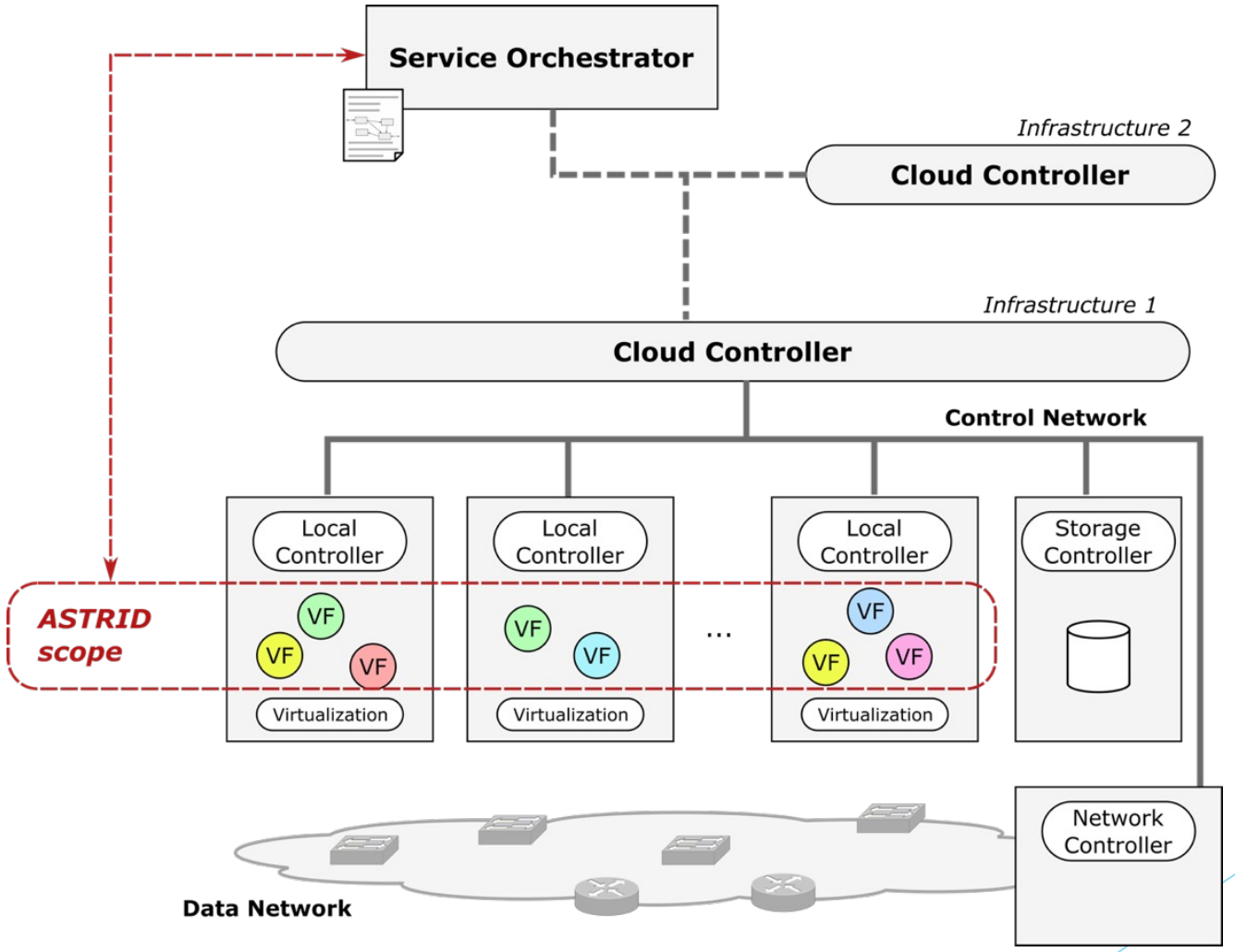


Cloud infrastructure:
- cloud providers
- IDS, antivirus, firewalls and similar tools
- as good as or better than most private DCs

Cloud Native Application Protection Platform (CNAPP)
- leverages the synergy in combining CWPP and CSPM capabilities
- scan workloads and configurations in development and protect workloads and configurations at runtime

Università di Genova

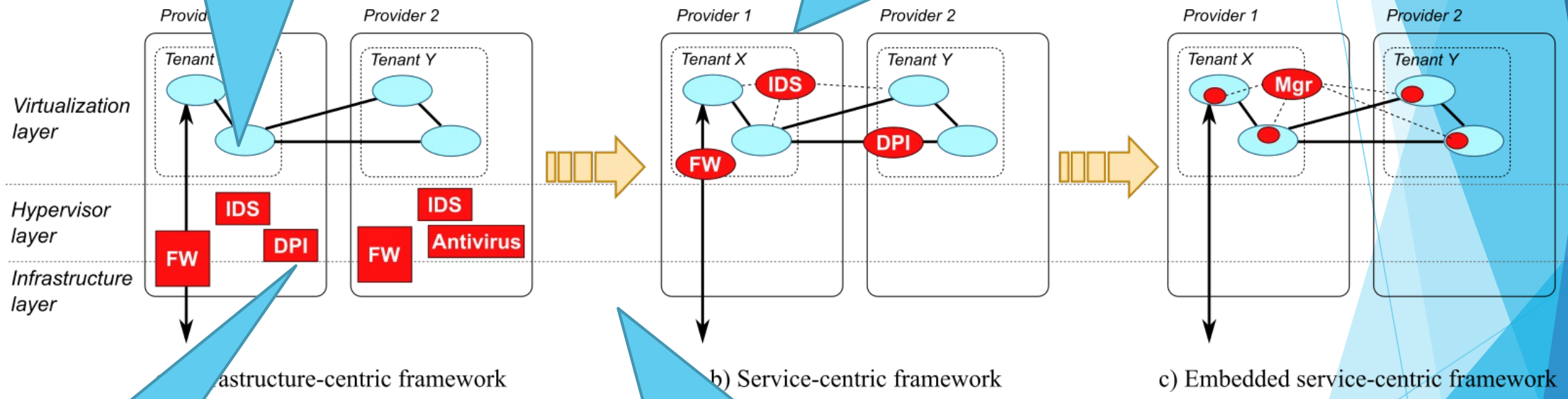
Protecting the workload: the scope



The need for visibility

Cloud Security Posture Management (CSPM)

Cloud Workload Protection Platforms (CWPP)



astructure-centric framework

b) Service-centric framework

c) Embedded service-centric framework

Cloud infrastructure

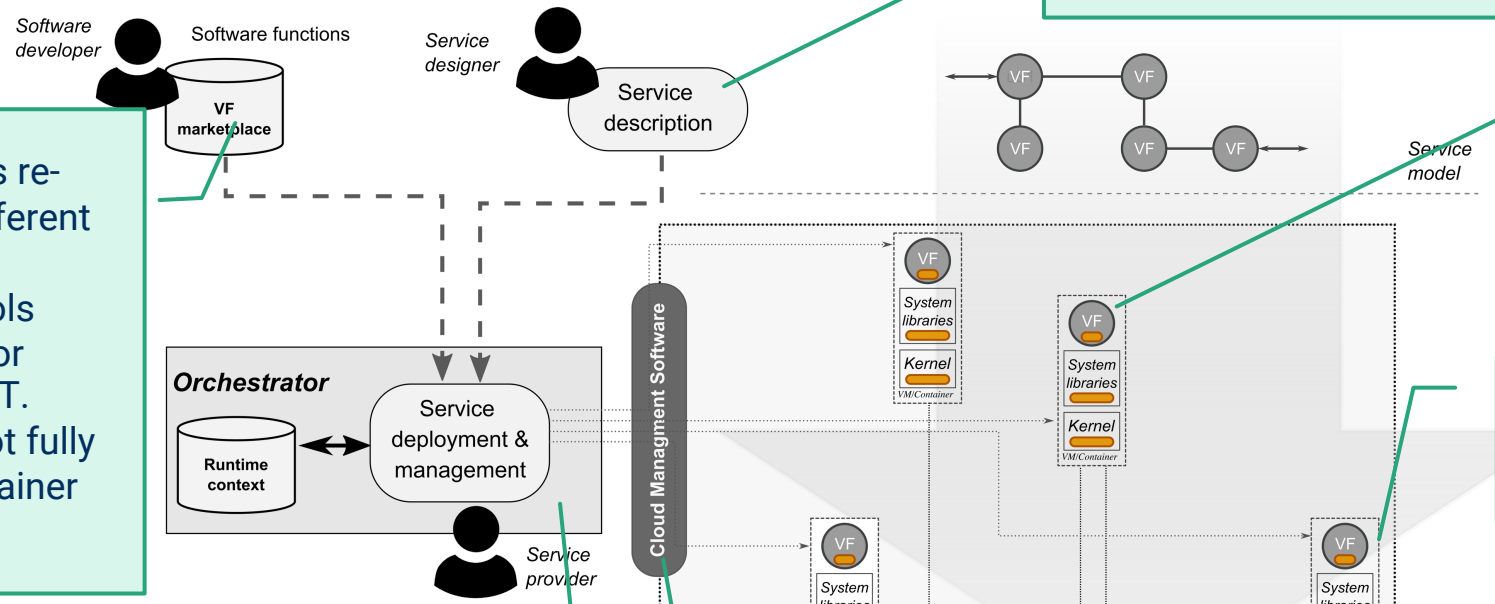
Cloud Native Application Protection Platform (CNAPP) (not shown)

DITEN Università di Genova

Protecting the workload

Analysis of all possible operating conditions is largely unpractical (software is often not known).

Software is re-used in different projects
Several tools available for SAST/DAST.
They do not fully cover container images.



Effective run-time missing, especially for serverless computing.

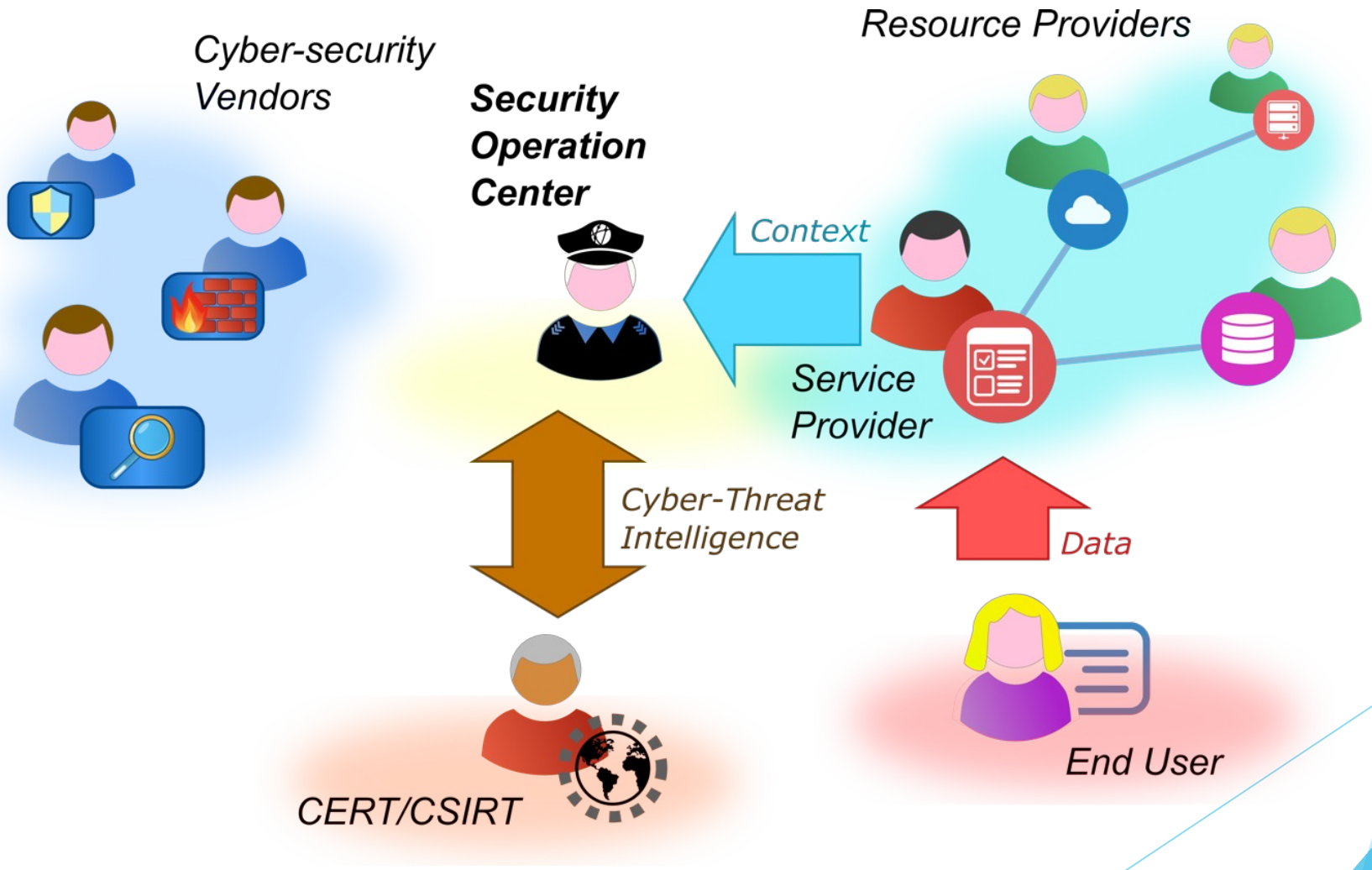
API remains the most vulnerable point.

Kubernetes offers a rich set of controls to effectively secure clusters and their applications.
Defaults are usually the least secure settings.

Network instrumentation, logs, and event monitoring.

Static configuration.
Consolidated tools available.

Protecting the workload



Security Information and Event Management

- **Collection** of events and logs from many sources
  - preferably in raw formats to fulfil legal requirements for forensics.
- **Decisions** based on rules
  - look for known patterns in the reported data.
- Incident **generation**
  - logs, reports, or alerts.
- **Notification** to humans
  - including remediation, if possible.

SIEM – Known limitations

PROBLEM	
Cannot use required data	
Difficult to maintain and operat	
High false negative and positive	
Unstable	
Inflexible data	→ annot adapt to critical cases
Static workflows	→ Limited and restrictive
Cannot detect modern threats	→ Business risk

Make the system *programmable*:

programmatic visibility

+

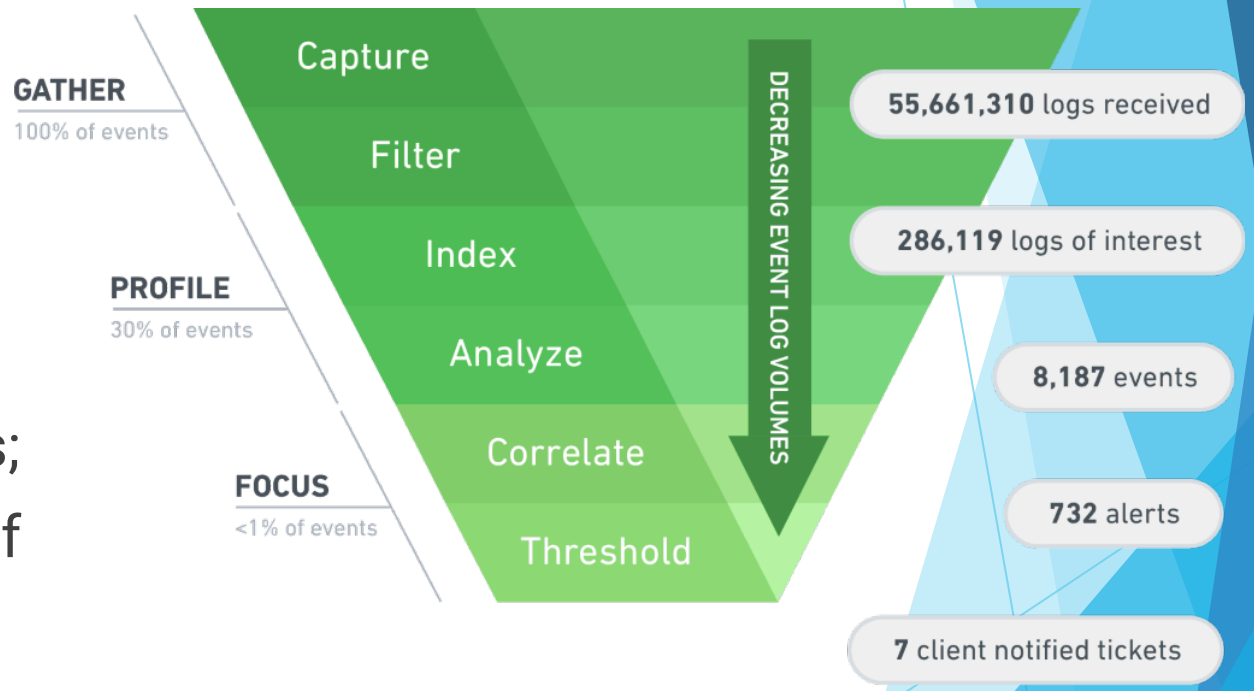
dynamic security pipelines

SIEM – Evolution

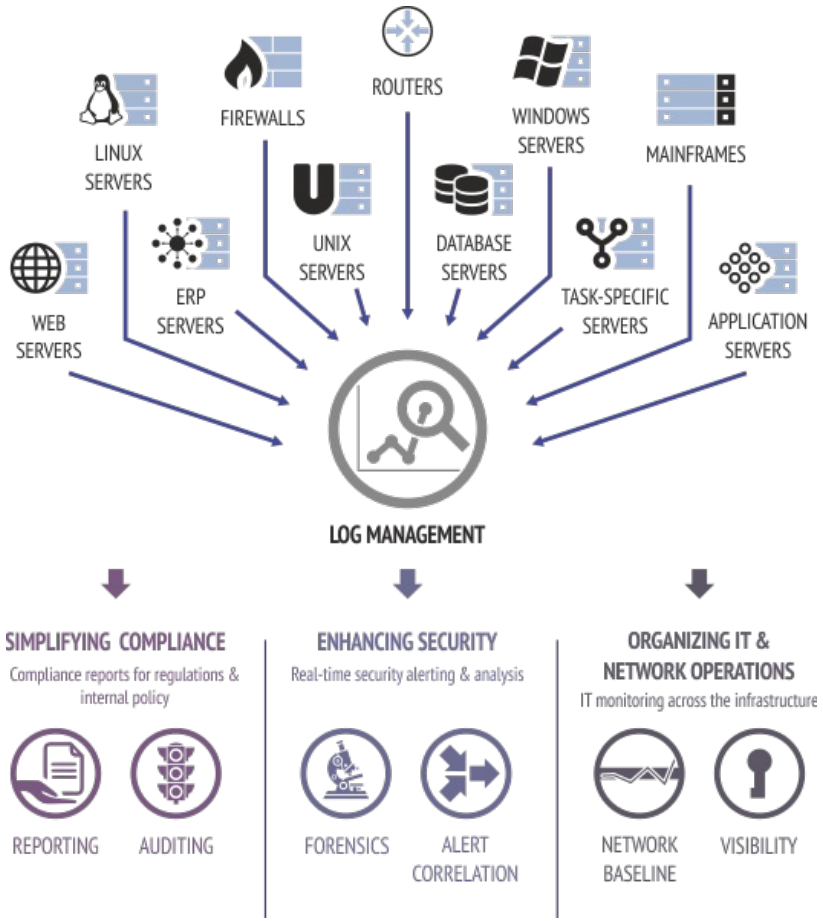
- Analytics-driven SIEM
  - beyond plain correlation rules for data analysis;
  - sophisticated quantitative methods to gain insight into and prioritize efforts;
  - ML and AI for anomaly detection.
- Flexibility
  - on premises, in the cloud or hybrid deployment.
- Beyond logs and security events
  - broader insights from data sources, including cloud workloads.
- Improved decision-support
  - rapid response, incident investigation and coordination of CSIRT.

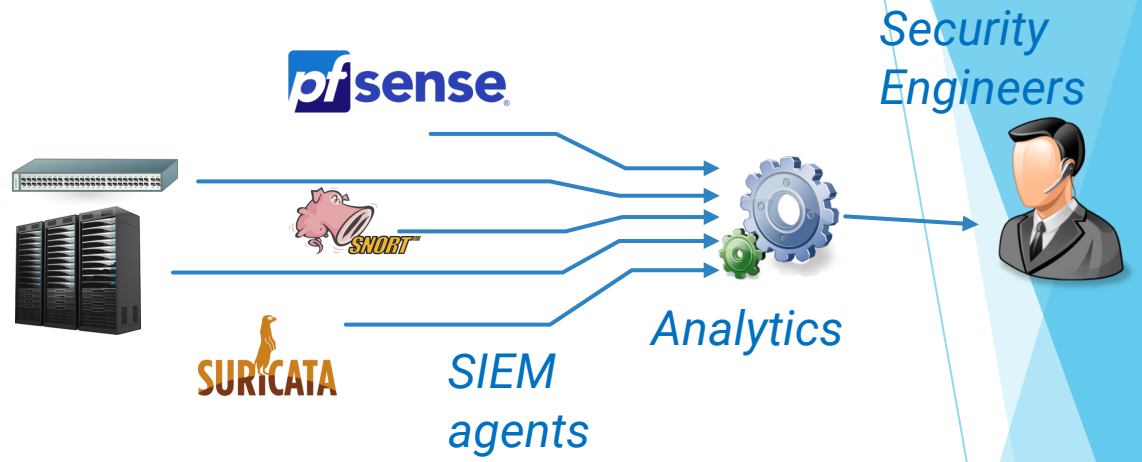
SIEM – Evolution

- Scalability
  - to follow the organization grow;
  - filter out useless information and events.
- Elasticity
  - to detect new cloud workloads;
  - to process variable amounts of data;
  - to balance resource consumption and performance.



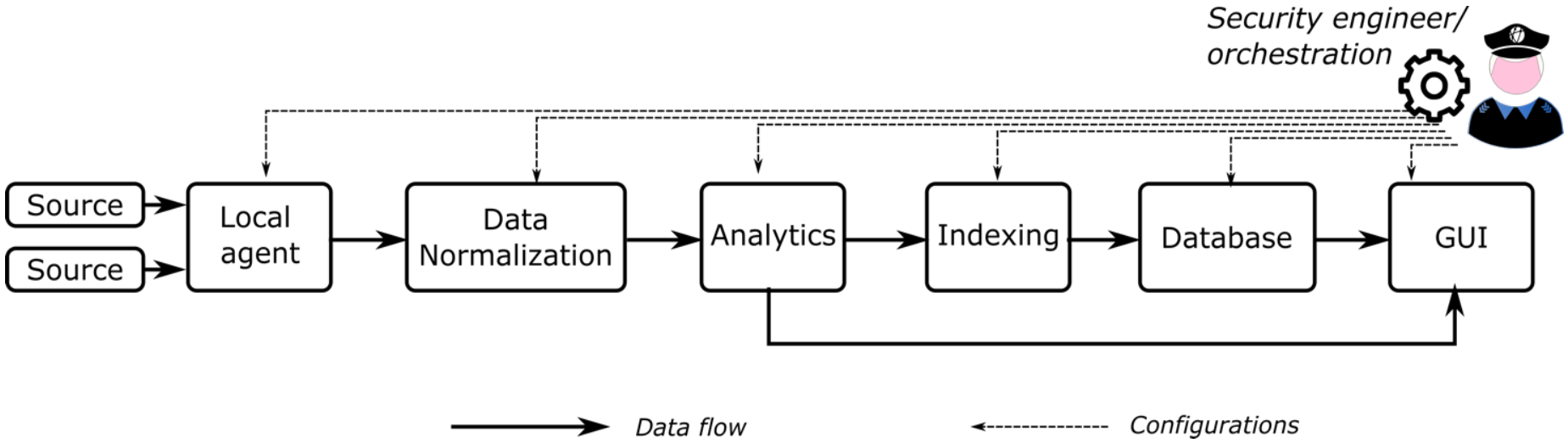
SIEM - Challenges





- Beyond bare metal servers and static service architectures:
  - virtualization, multi-tenancy, cloud-native applications, software repositories, elastic and dynamic topologies
- Beyond rigid appliances and services for cybersecurity
  - programmable agents, flexible streaming and pipelining

SIEM common architecture



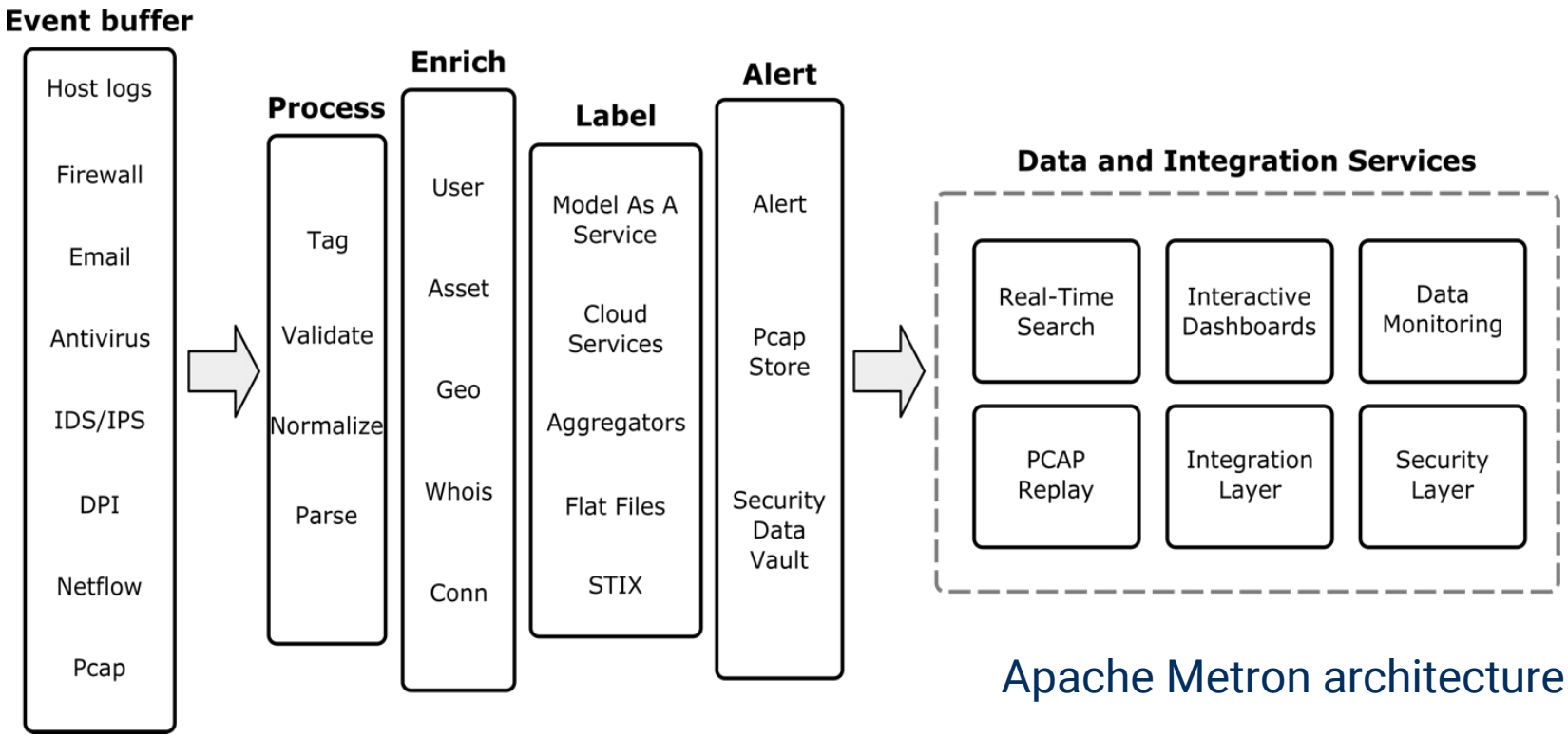
Log and data collection
File integrity monitoring
Rootkit and malware detection
Security policy monitoring
Configuration assessments
Software inventory

Security analytics
Intrusion detection
Log data analysis
File integrity monitoring

Vulnerability detection
Configuration assessment
Incident response
Regulatory compliance

Wazuh architecture

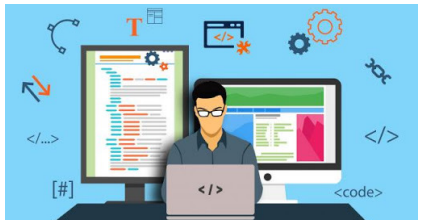
SIEM layers



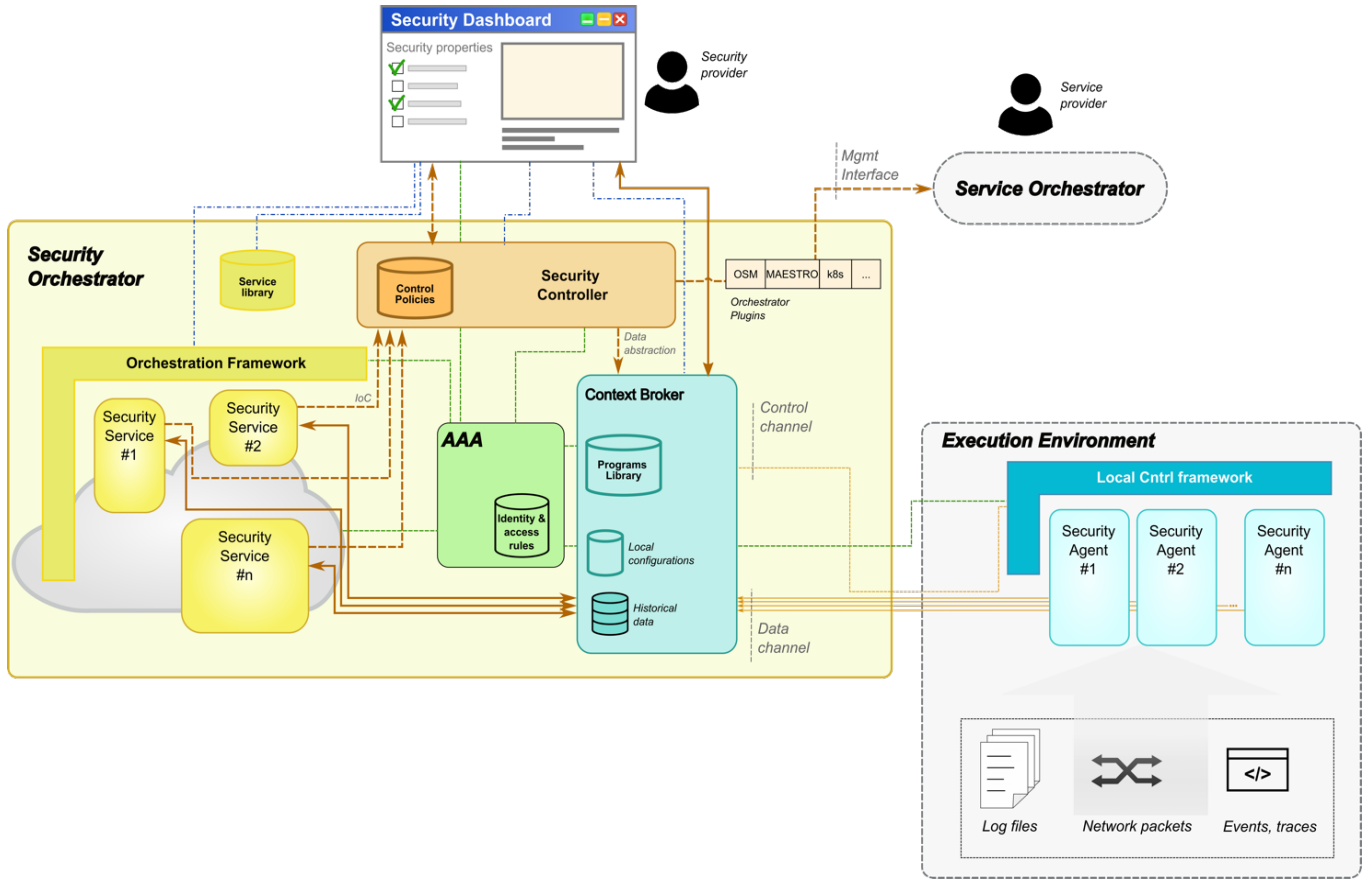
Apache Metron architecture

Towards new architectures

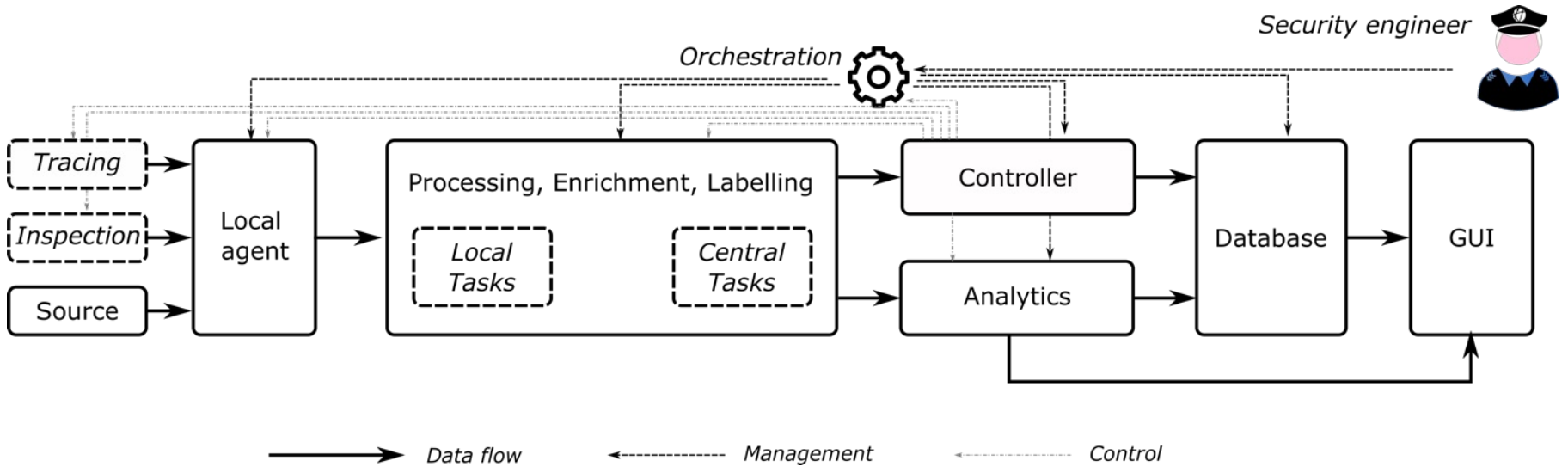
- Externalization of security processes
- Multi- and cross- cloud deployments
- Cloud-native applications
- Programmatic inspection and monitoring
- Portability
- Service orchestration



The ASTRID architecture



The ASTRID model



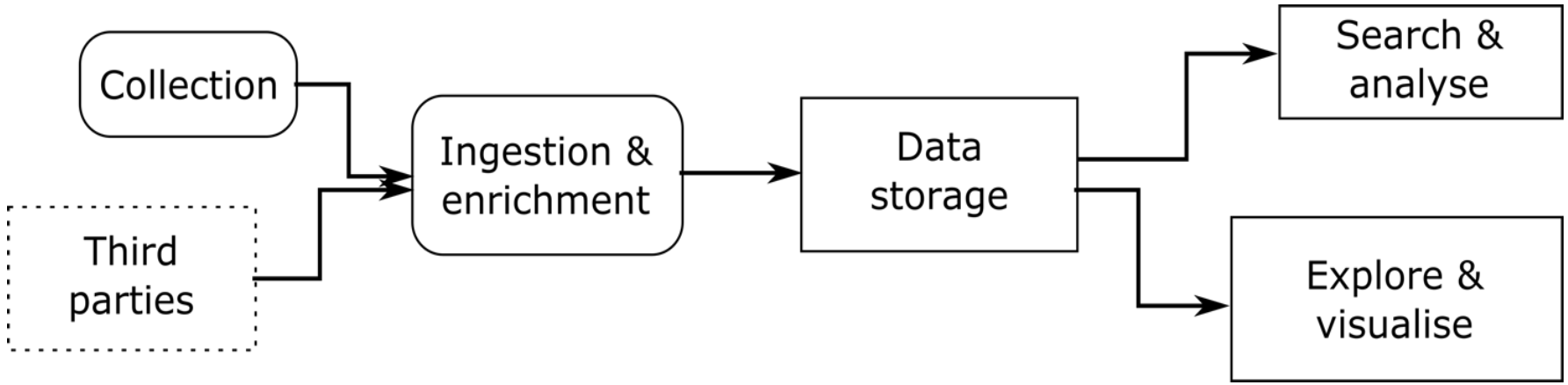
Event and log collection
Software tracing
Deep packet inspection

Aggregation
Tagging
Enrichment

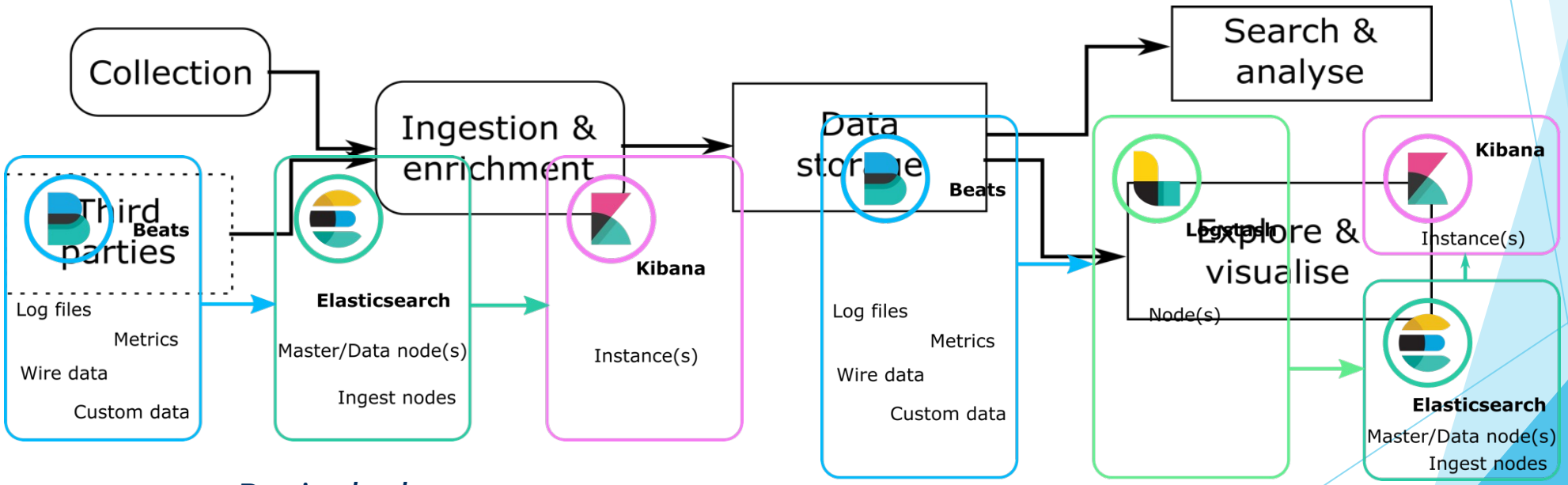
Delivery
Processing
Normalization
Correlation

Detection
Attestation
Prediction
Assessment

Collecting data



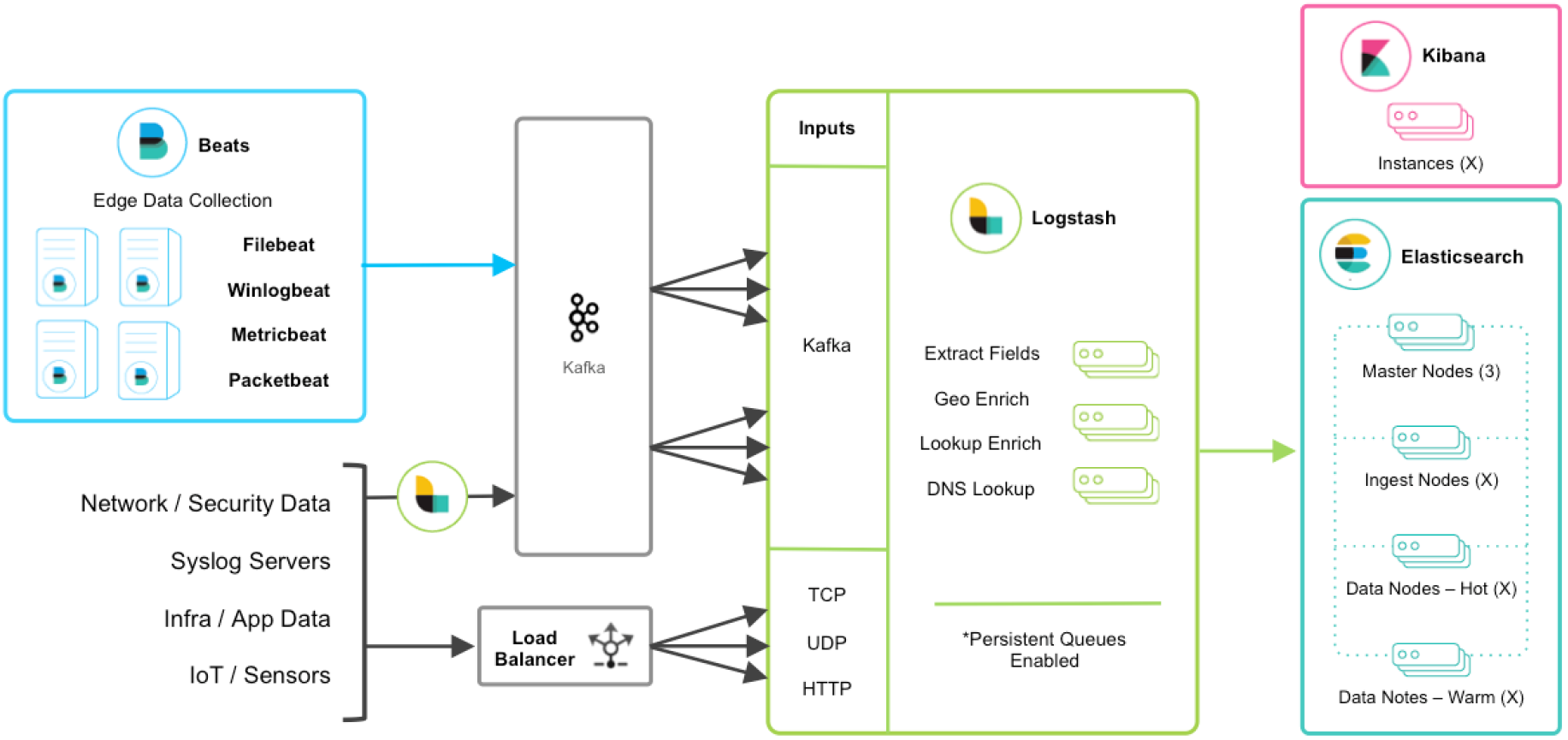
Collecting data: the Elastic Stack



Basic deployment

Basic deployment with enrichment

Collecting data: the Elastic Stack



Full deployment

Collecting data: alternative frameworks

Name	Use Cases
Elastic Stack	Log collection, metric collection, network packet collection
Prometheus	Metrics collection, monitoring
Zipkin	Trace Collection and Dashboarding
Jaeger	Trace Collection and Dashboarding
OpenCensus	Metrics and Trace Collection
Grafana	Analytics and Dashboarding

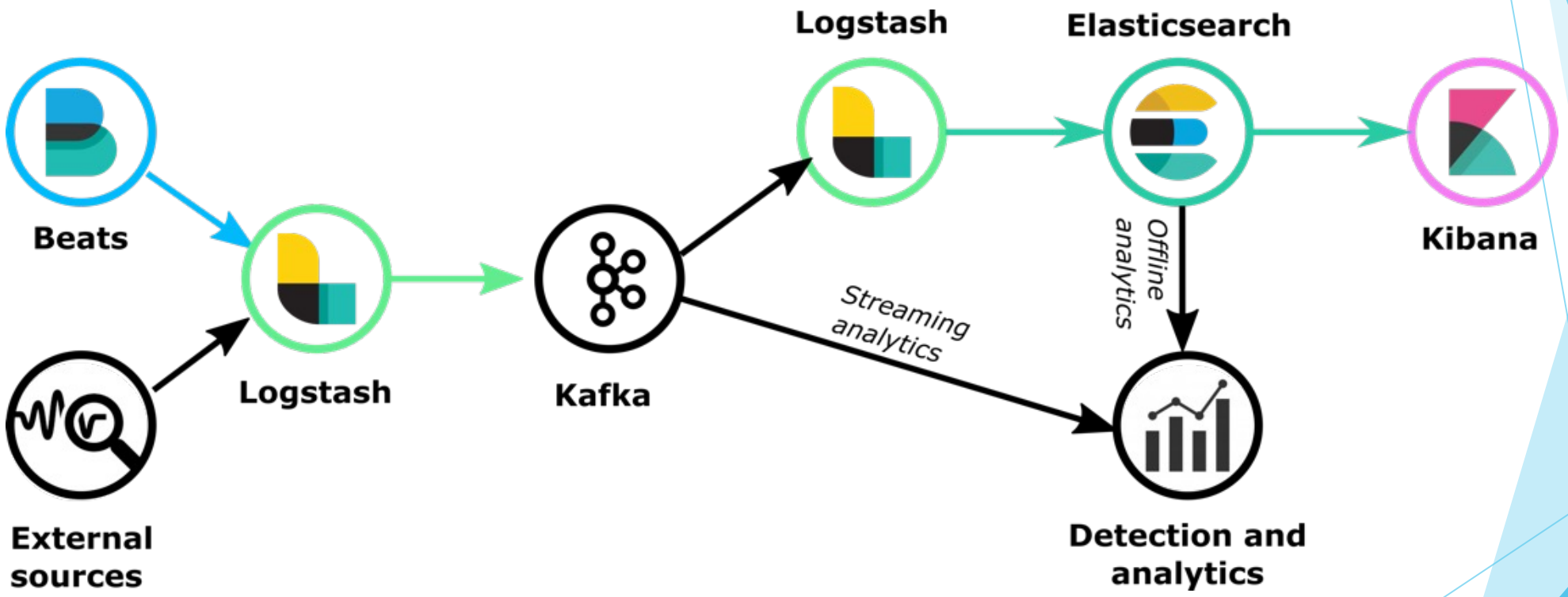
Elastic vs Prometheus and Splunk

Name	Elasticsearch	Prometheus	Splunk
Description	A distributed, RESTful modern search and analytics engine based on Apache Lucene	Open-source Time Series DBMS and monitoring system	Analytics Platform for Big Data
Primary database model	Search engine	Time Series DBMS	Search engine
Secondary database models	Document store		
Developer	Elastic		Splunk Inc.
License	Open Source	Open Source	Commercial
Cloud-based only	no	no	no
Implementation language	Java	Go	
Server operating systems	All OS with a Java VM	Linux, Windows	Linux, OS X, Solaris, Windows
Data scheme	schema-free	yes	yes
Typing info	yes	Numeric data only	yes
XML support	no	no	yes
Secondary indexes	yes	no	yes
SQL	SQL-like query language	no	no
APIs and other access methods	Java API; RESTful HTTP/JSON API	RESTful HTTP/JSON API	HTTP REST
Supported programming languages	.NET; Groovy; Community Contributed Clients; Java; JavaScript; Perl; PHP; Python; Ruby	.NET; C++; Go; Haskell; Java; JavaScript (Node.js); Python; Ruby	C#; Java; JavaScript; PHP; Python; Ruby

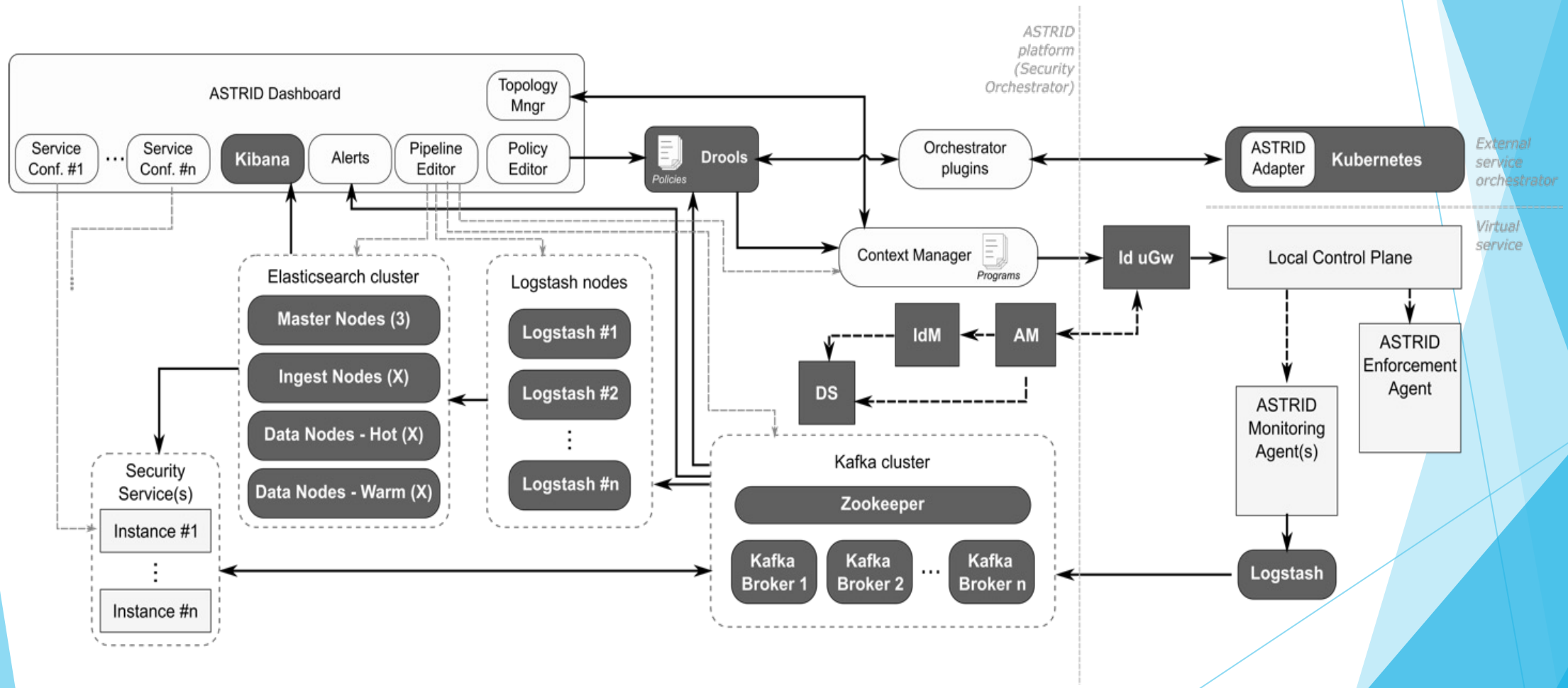
Elastic vs Prometheus and Splunk

Name	Elasticsearch	Prometheus	Splunk
Server-side scripts	yes	no	yes
Triggers	yes	no	yes
Partitioning methods	Sharding	Sharding	Sharding
Replication methods	yes	yes	Multi-source replication
MapReduce info	ES-Hadoop Connector	no	yes
Consistency concepts	Eventual Consistency	none	Eventual Consistency
Foreign keys	no	no	no
Transaction concepts	no	no	no
Concurrency	yes	yes	yes
Durability	yes	yes	yes
In-memory capabilities	Memcached and Redis integration	no	no
User concepts		no	Access rights for users and roles

Extending the Elastic Stack

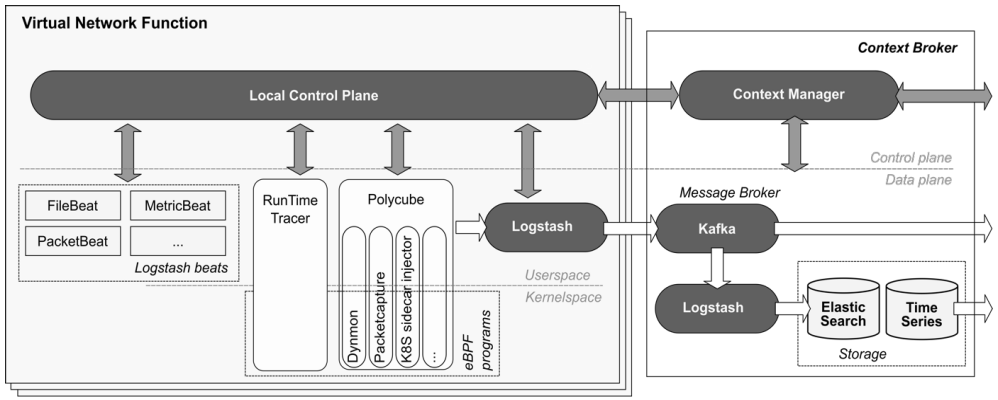


Software architecture



Programmable agents

- Elastic Beats
- eBPF-based tools
  - Polycube cubes: monitoring, inspection, firewalling
  - RunTimeTracer
- LCP



Security services

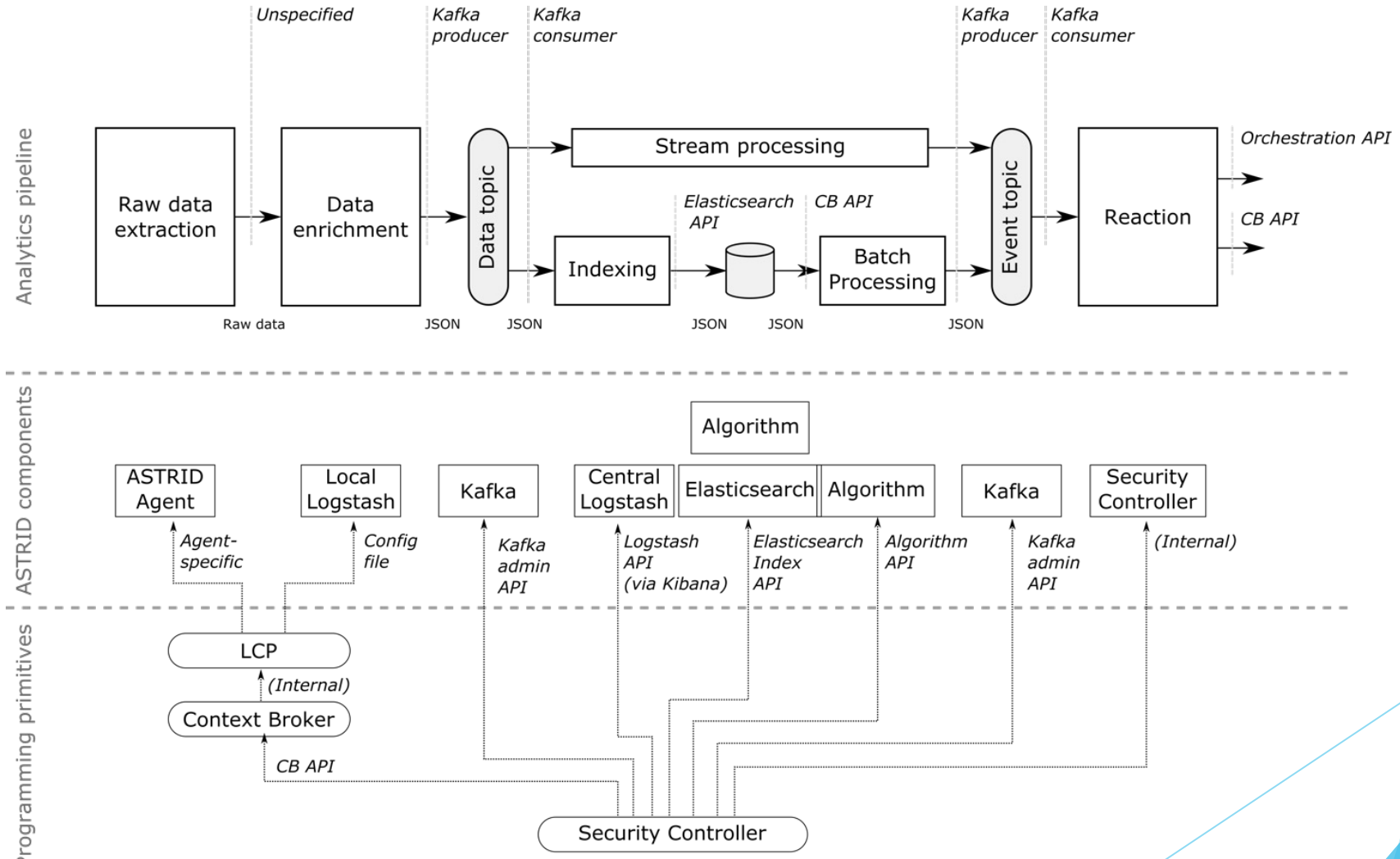
Network Analytics

- Learn periodic and seasonal trends
- Detect anomalies from the trend
- Switch between multiple states (BAU, warning)
- Change the behaviour of programmable agents
- Leverages eBPF programs for traffic measurements and deep packet inspection

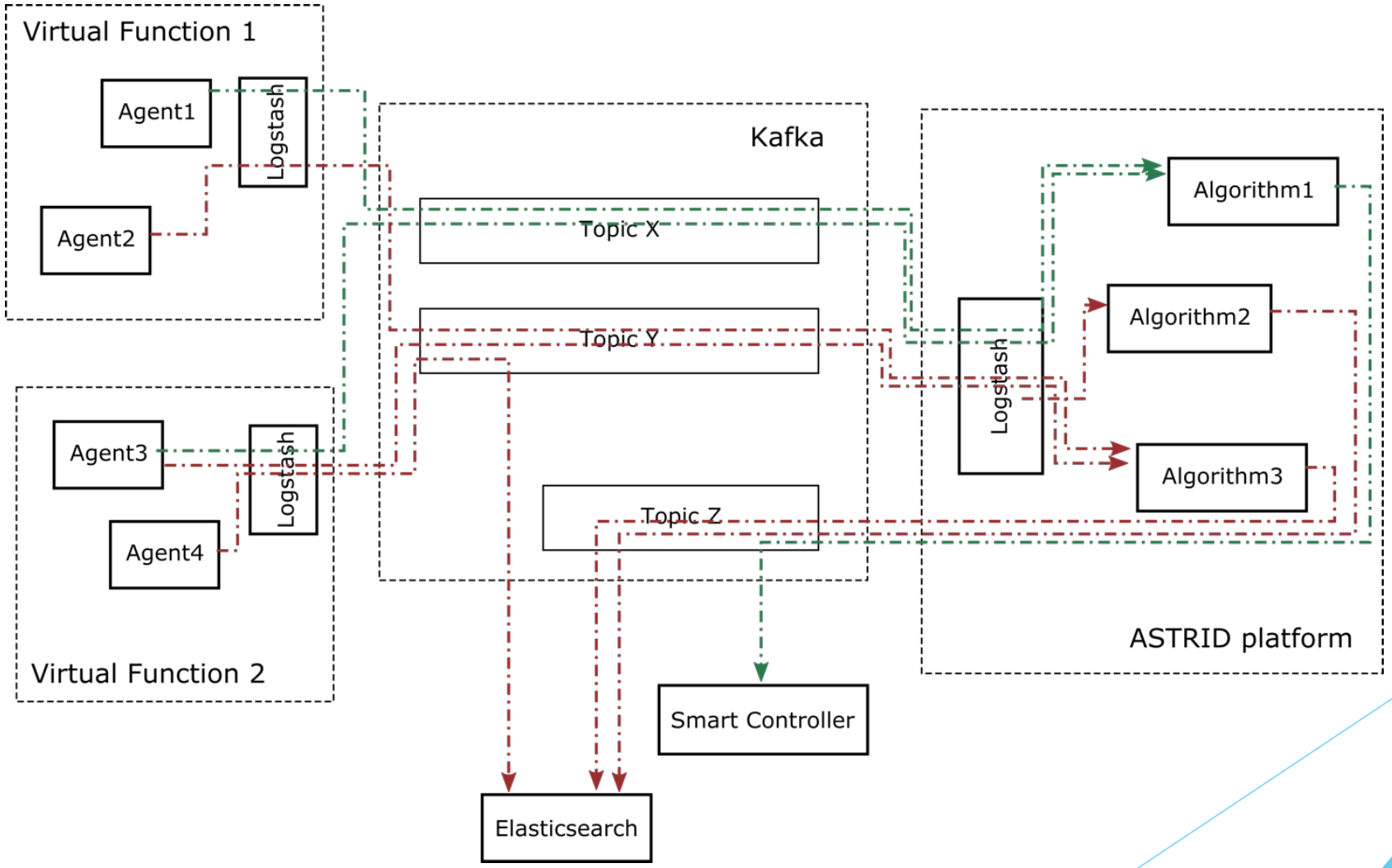
Remote attestation

- Verify the behaviour of the software at run-time
- Collect measurements validated by the TPM
- Attestation by proof/Attestation by quote
- Leverage eBPF tools for tracing the execution of kernel functions

The ASTRID programming model

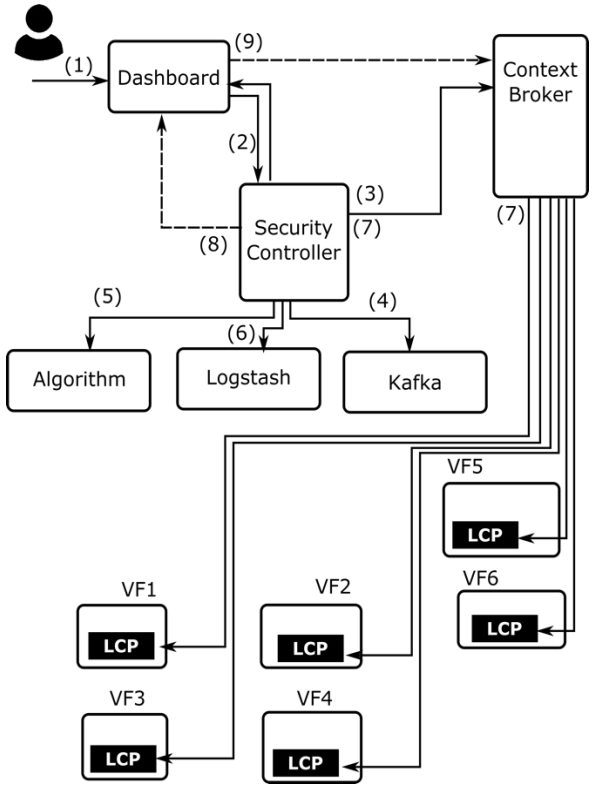


Building processing pipelines

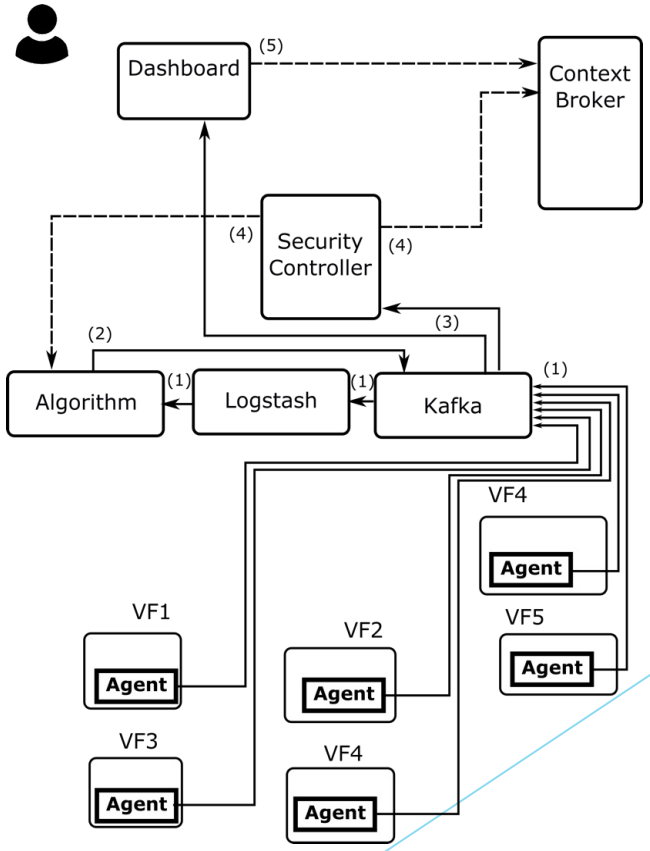


Building processing pipelines at run-time

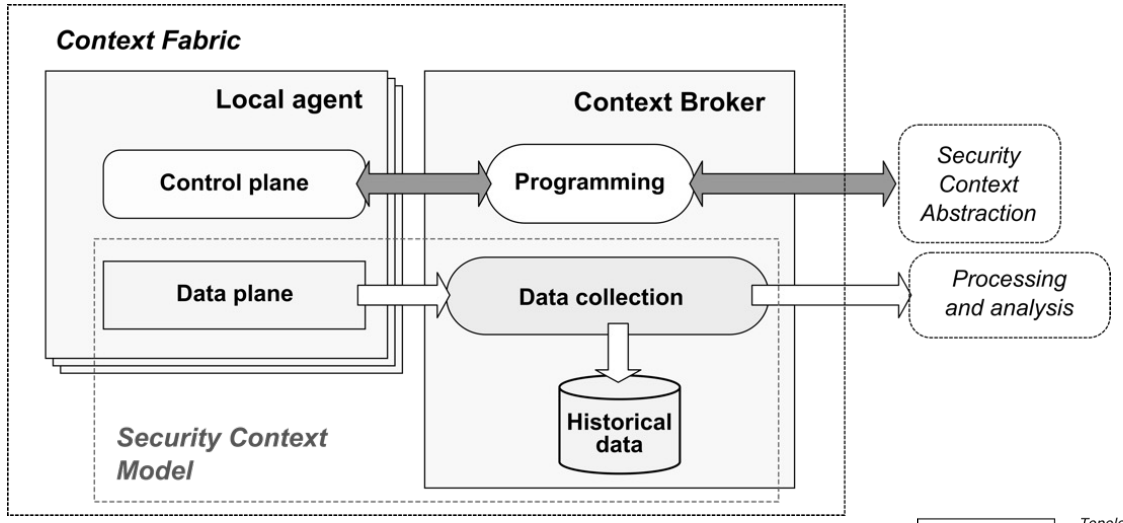
Activation



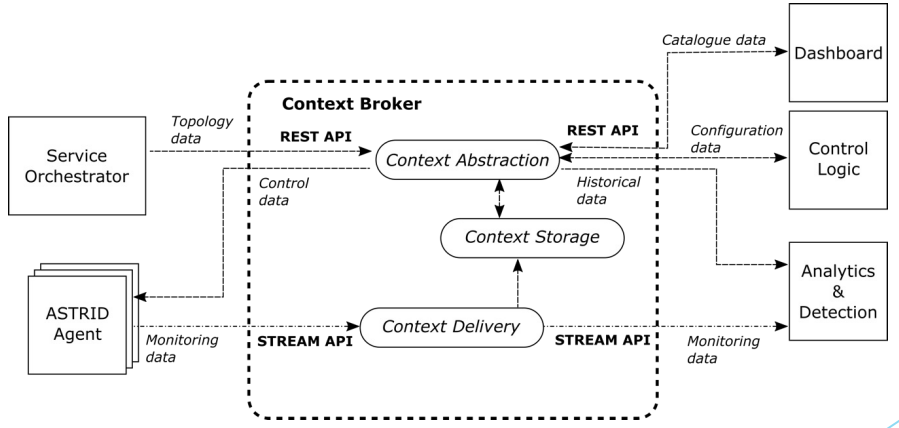
Mitigation and reaction



Abstracting the context: the Context Broker

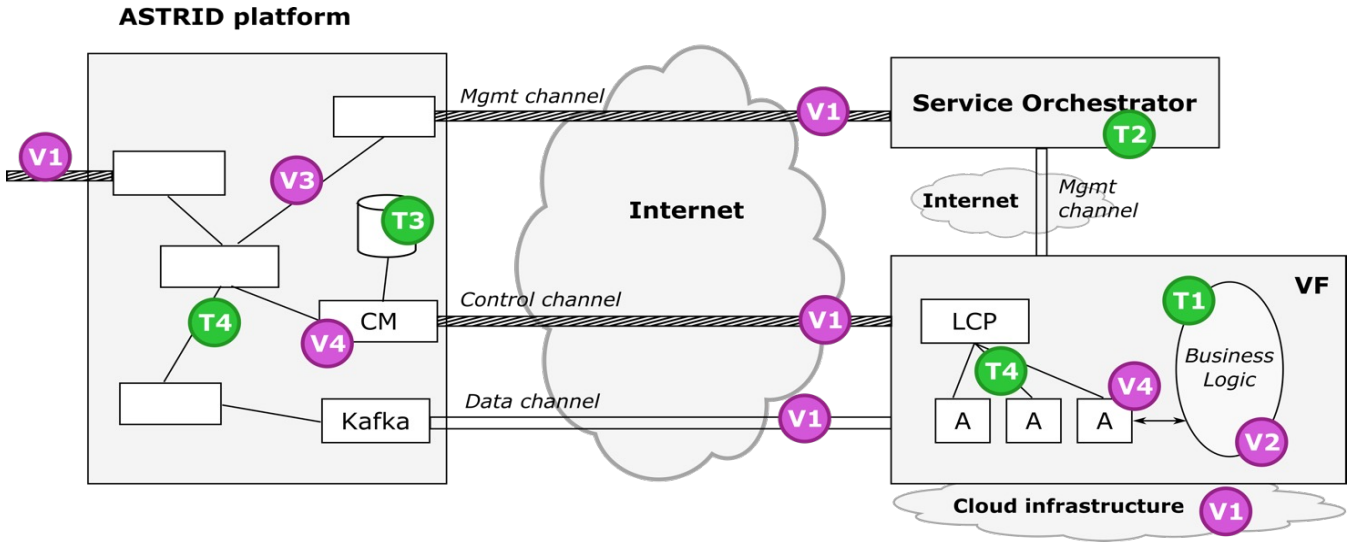


The Context Fabric



The model and interfaces

Security considerations



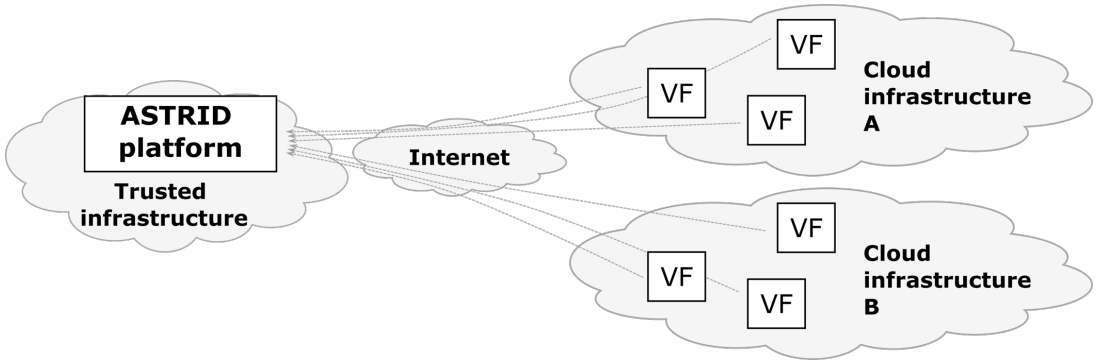
Targets:

- **T1**: The Business Logic
- **T2**: The service orchestrator
- **T3**: Security-related data
- **T4**: Security processes

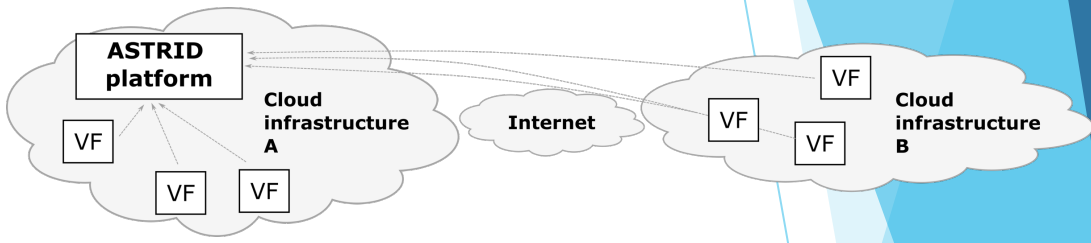
Threat vectors:

- **V1**: APIs exposed by micro-services
- **V2**: Privilege escalation in the VF
- **V3**: Micro-services architecture
- **V4**: Programmable components

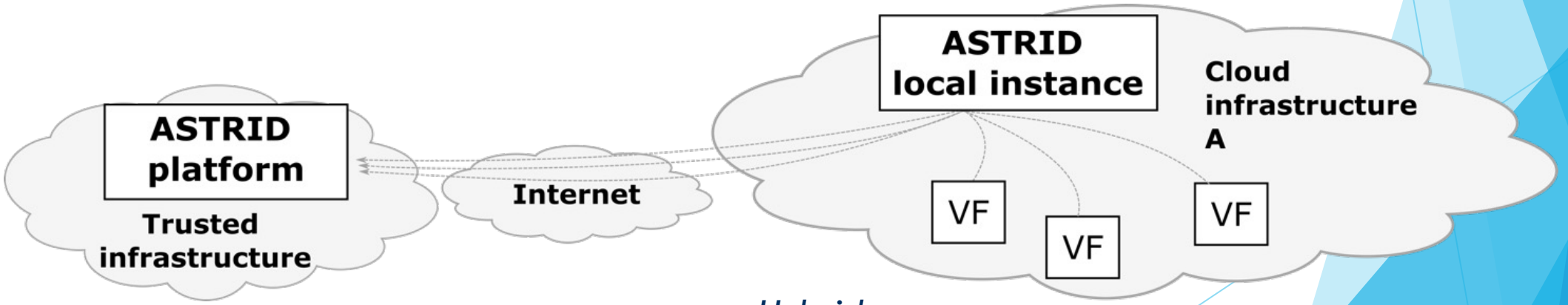
Deployments



Centralized standalone

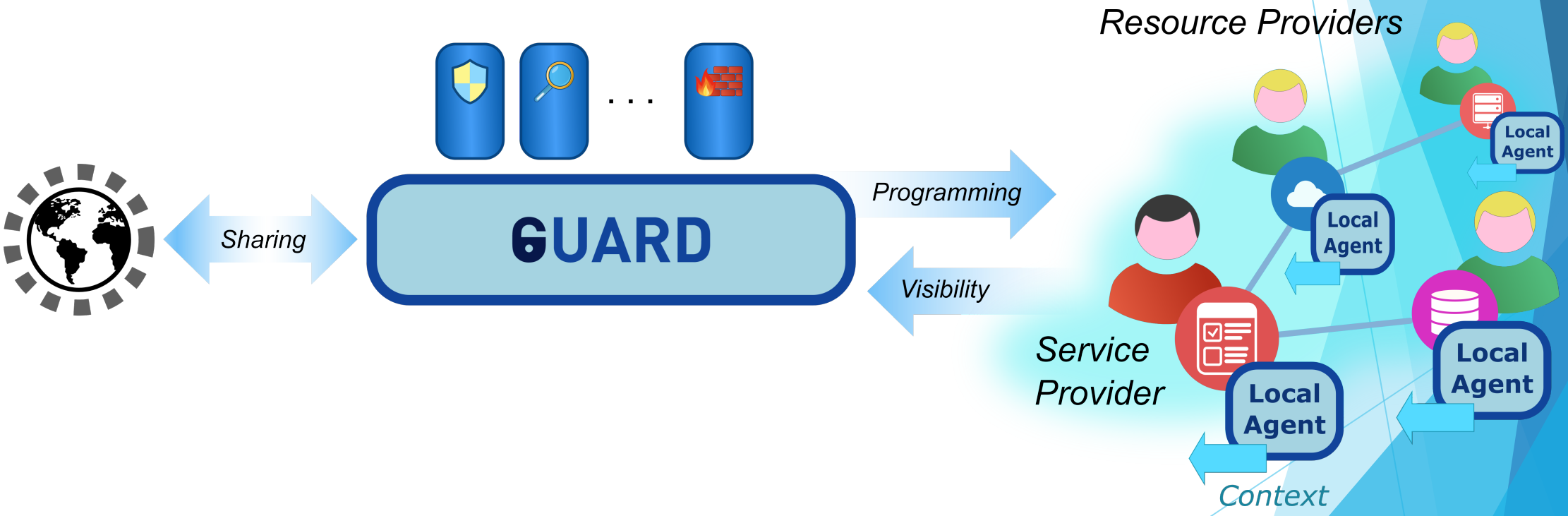


Co-located

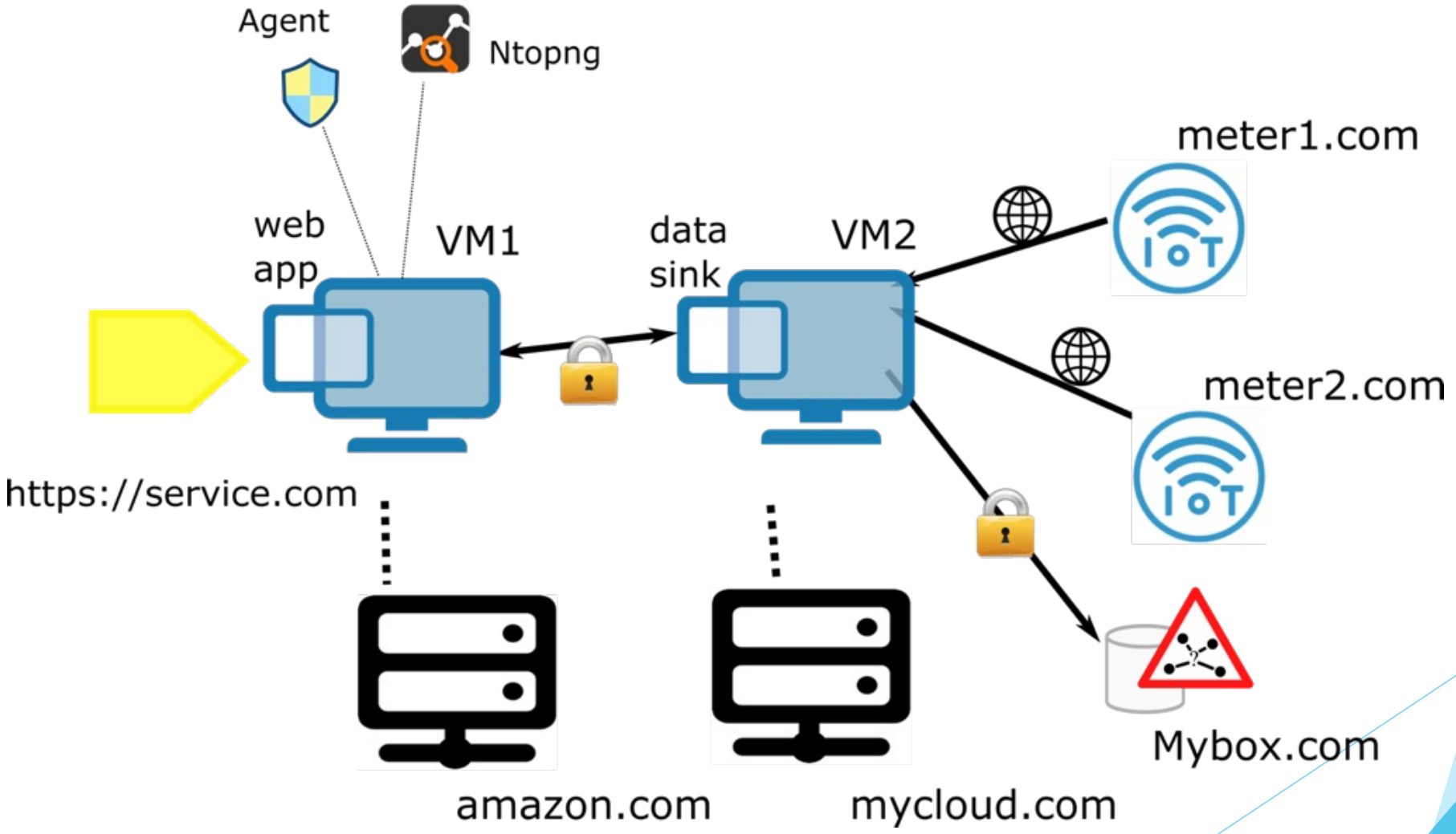


Hybrid

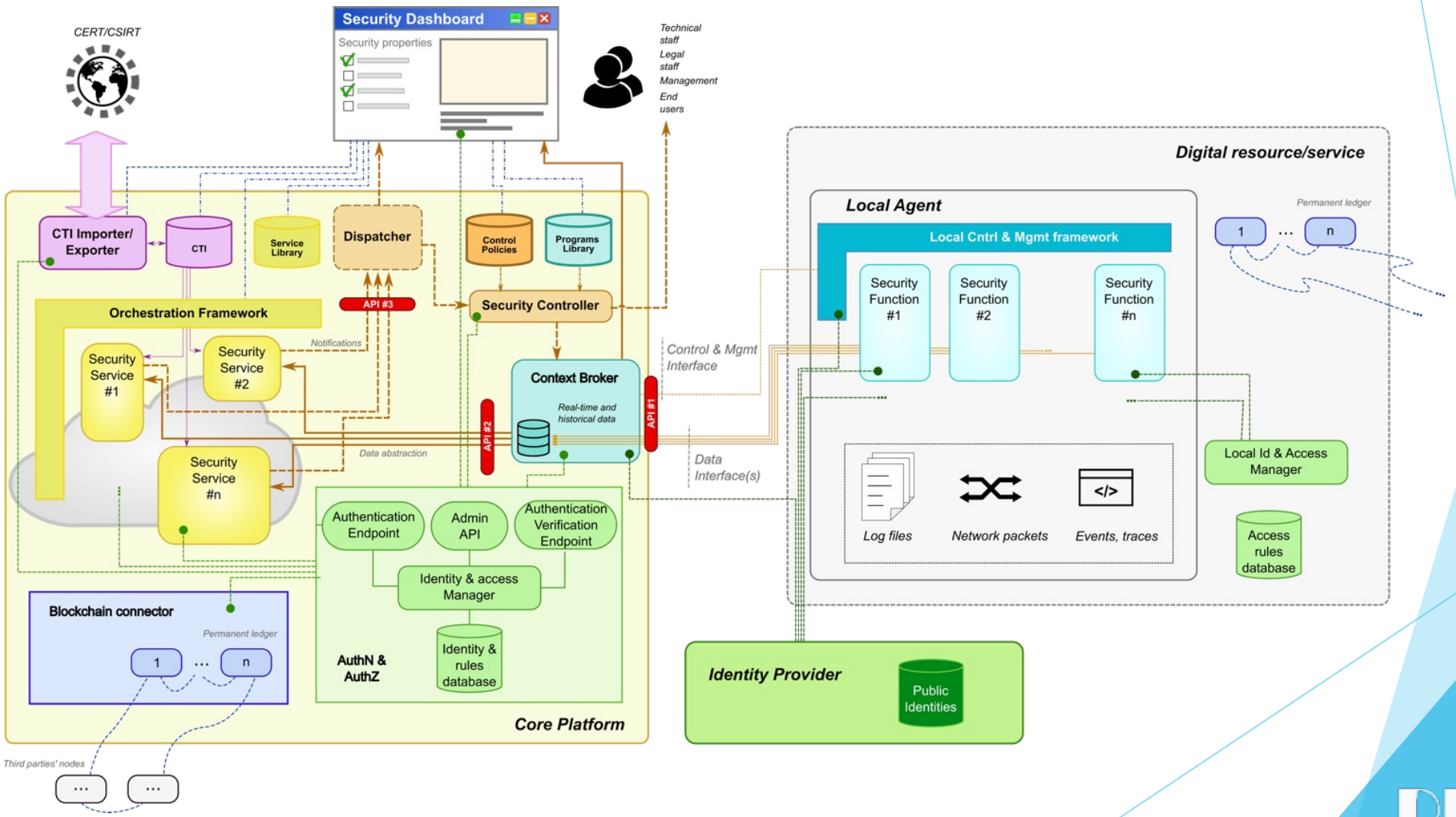
Beyond the cloud: digital services



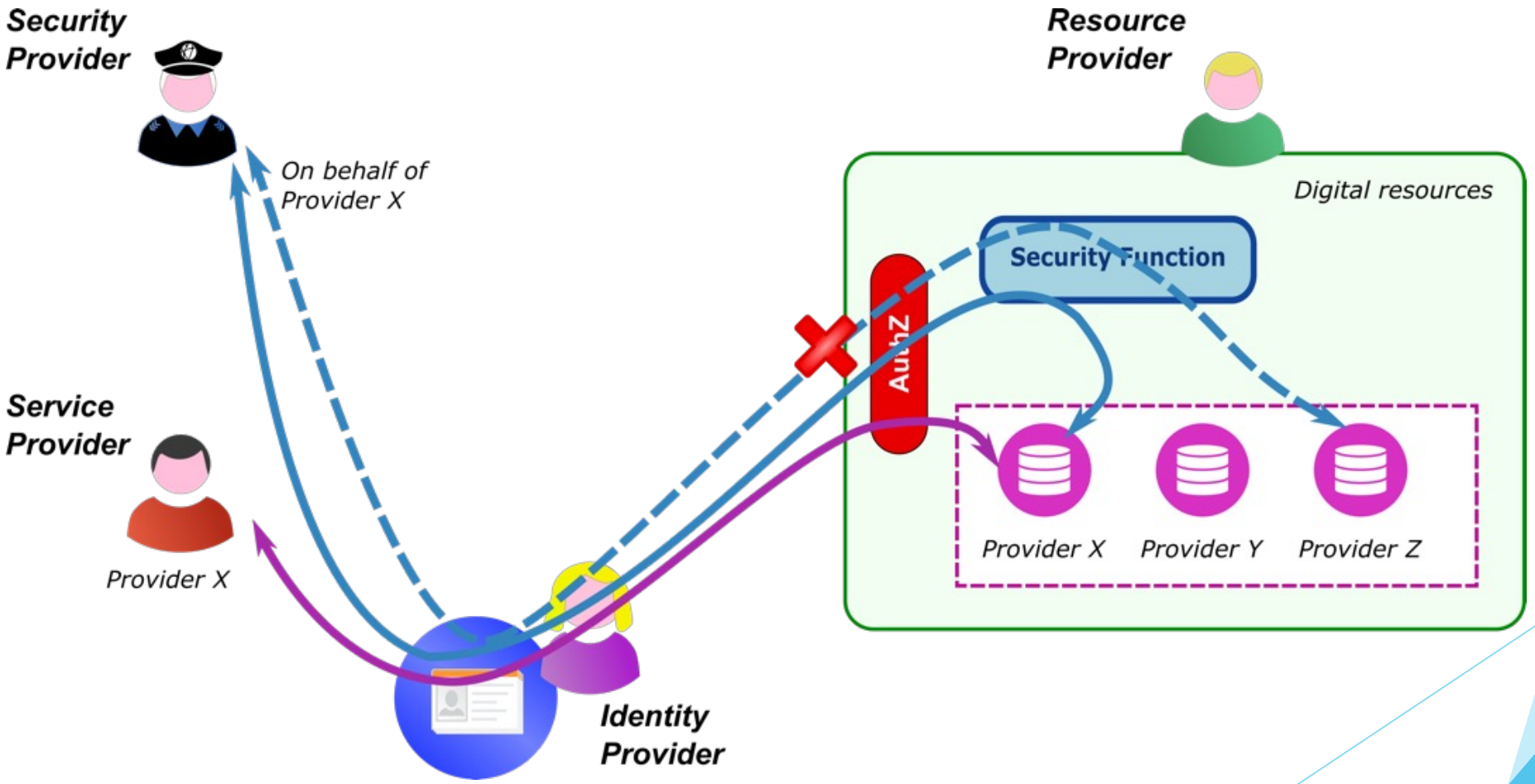
Digital service chains: example



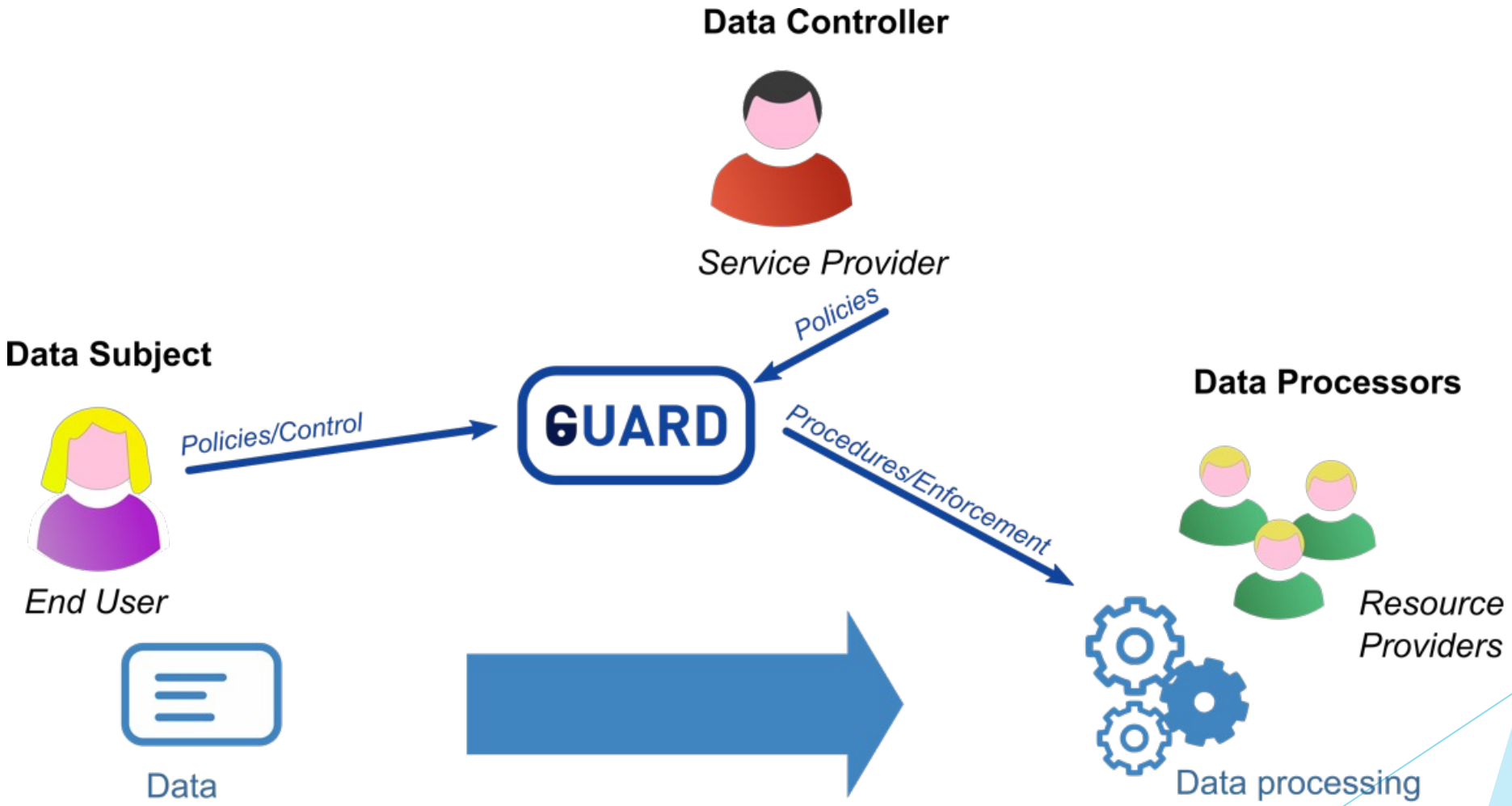
The GUARD architecture



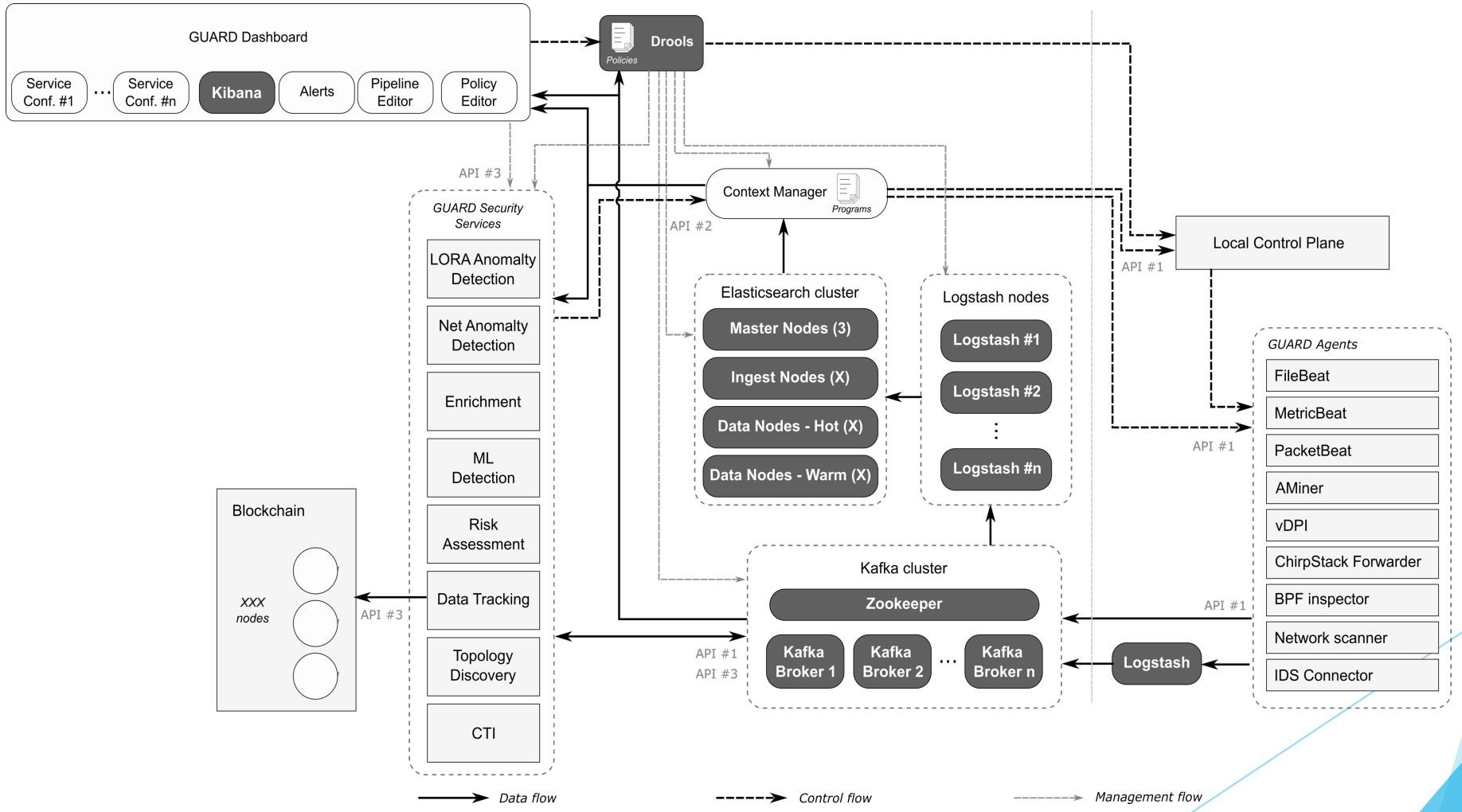
AuthN & AuthZ


Security Provider
On behalf of Provider X
Service Provider
Provider X
Identity Provider
Resource Provider
Digital resources
Security Function
AuthZ
Provider X
Provider Y
Provider Z

GDPR aspects

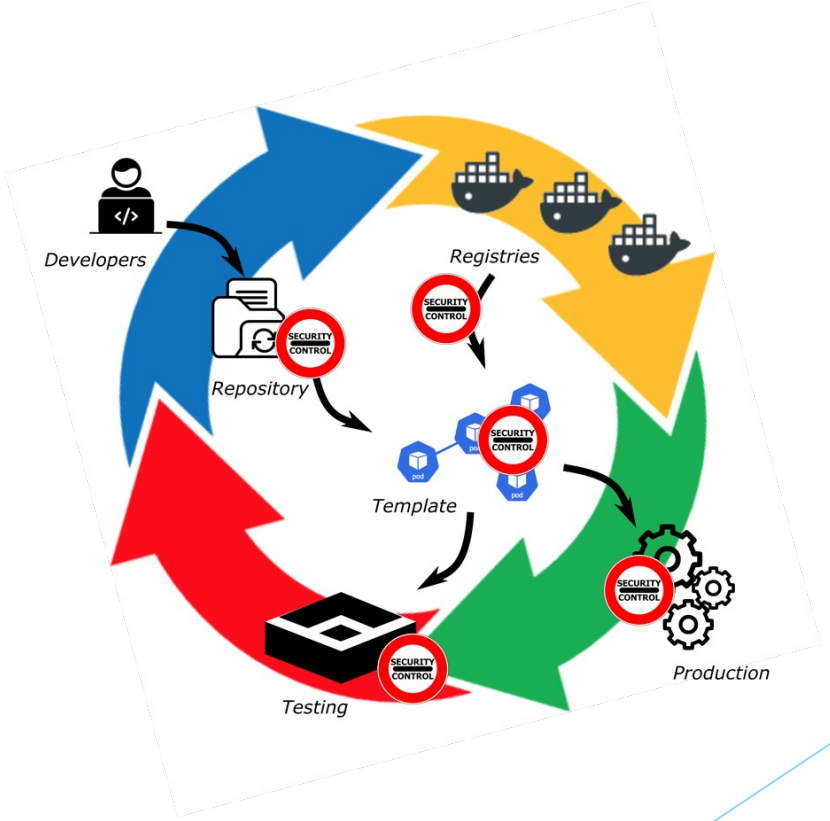


The GUARD software architecture

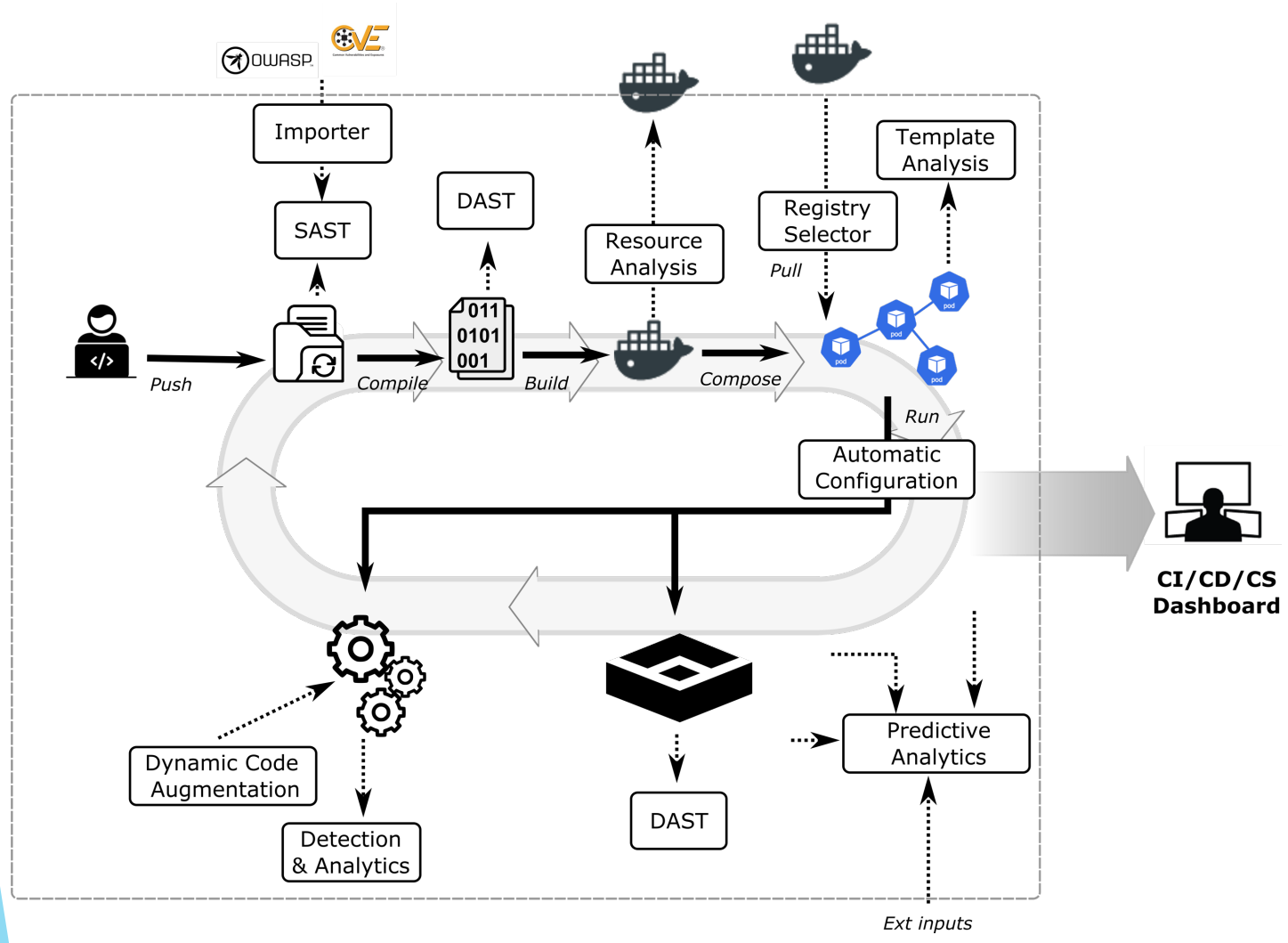


The road away: secure CI/CD pipelines

- Security vs quick code delivery
- Templates and software orchestration
- Unknown topologies
- Lack of visibility into the cloud



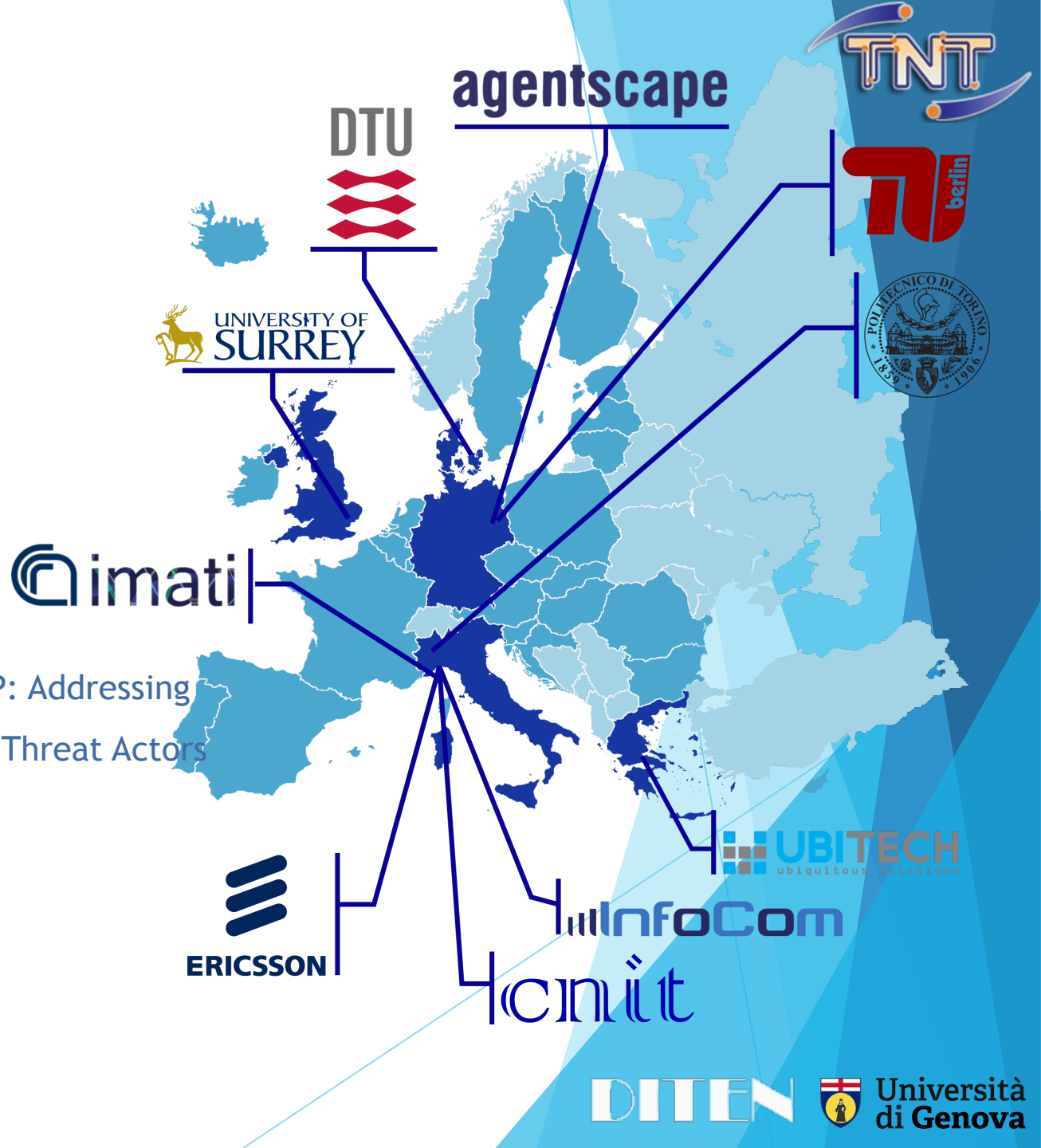
CI/CD/CS



- Source code assessment
- Container images
- Kubernetes templates
- Offline testing
- Run-time monitoring and analysis

ASTRID factsheet

Project acronym	ASTRID
Project ID	786922
Starting Date	1st May 2018
Ending Date	30th April 2021
Call	H2020-DS-2016-2017
Topic	H2020-DS-07-2017 – Cybersecurity PPP: Addressing Advanced Cyber Security Threats and Threat Actors
Total Cost	EUR 2,932,297.50
EC Contribution	EUR 2,932,297.50
Funding scheme	RIA



GUARD factsheet

Project acronym	GUARD
Project ID	833456
Starting Date	1st May 2019
Ending Date	30th April 2022
Call	H2020-DS-2016-2017
Topic	SU-ICT-01-2018 – Dynamic countering of cyber-attacks
Total Cost	€ 5,443,250.00
EC Contribution	€ 4,684,700.00
Funding scheme	IA

DITEN Università di Genova

Thesis/Research grants

- Deep packet inspection through eBPF programs
- Kernel tracing through eBPF programs
- Integration/extension to open-source frameworks (Cilium)
- Detection of stegomalware and covert channels

Suggested readings

- M. Repetto, A. Carrega, R. Rapuzzi. An architecture to manage security operations for digital service chains. *Future Generation Computer Systems,* Volume 115, February 2021, Pages 251-266. DOI: 10.1016/j.future.2020.08.044
- S. Covaci, R. Rapuzzi, M. Repetto, Fulvio Risso. A New Paradigm to Address Threats for Virtualized Services. In *IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Tokyo, Japan, July 23rd-27th, 2018, pp. 689-694.
- R. Rapuzzi, M. Repetto. Building situational awareness for network threats in fog/edge computing: Emerging paradigms beyond the security perimeter model, *Future Generation Computer Systems*, Volume 85, August 2018, pp. 235-249. DOI: 10.1016/j.future.2018.04.007.

Suggested readings

- Gábor Pék, Levente Buttyán, Boldizsár Bencsáth. A survey of security issues in hardware virtualization, *ACM Computing Surveys*, July 2013, Article No.: 40, https://doi.org/10.1145/2480741.2480757.
- Cloud Security Alliance. Security Guidance v4.0 – CSA Security Guidance for Critical Areas of Focus in Cloud Computing v4.0, 07/26/2017.
- ASTRID website: https://www.astrid-project.eu/
- GUARD website: https://guard-project.eu/