Cloud transparency: the notion and the issues

Ernesto **Damiani**, Università degli Studi di Milano

My research lab

- Secure Software Architectures and Knowledge-based systems lab (SESAR) http://sesar.dti.unimi.it

• Located on the new campus in Crema, 40 km south-east of Milan
• Industry collaborations: SAP, British Telecom Nokia Siemens, Cisco, Telecom Italia

SEVENTH FRAMEWORK PROGRAMME

Aristotele

securescm

Outline

- The problem
- Virtualization
- Cloud assurance, SLA and certification
- A (meta-)model
- Some research objectives
- References

The problem

New paradigms (SOA, Cloud) -> new security problems…

- Breach of data integrity, confidentiality [1][2][3] and privacy [4]
- Spamming, cross-site scripting attacks [5]
- Denial-or-service (DoS) attacks [6][7]
- Reduced application and data availability [2]
- Authentication, authorization and accounting (AAA) vulnerabilities [2][1]

Source of the problem

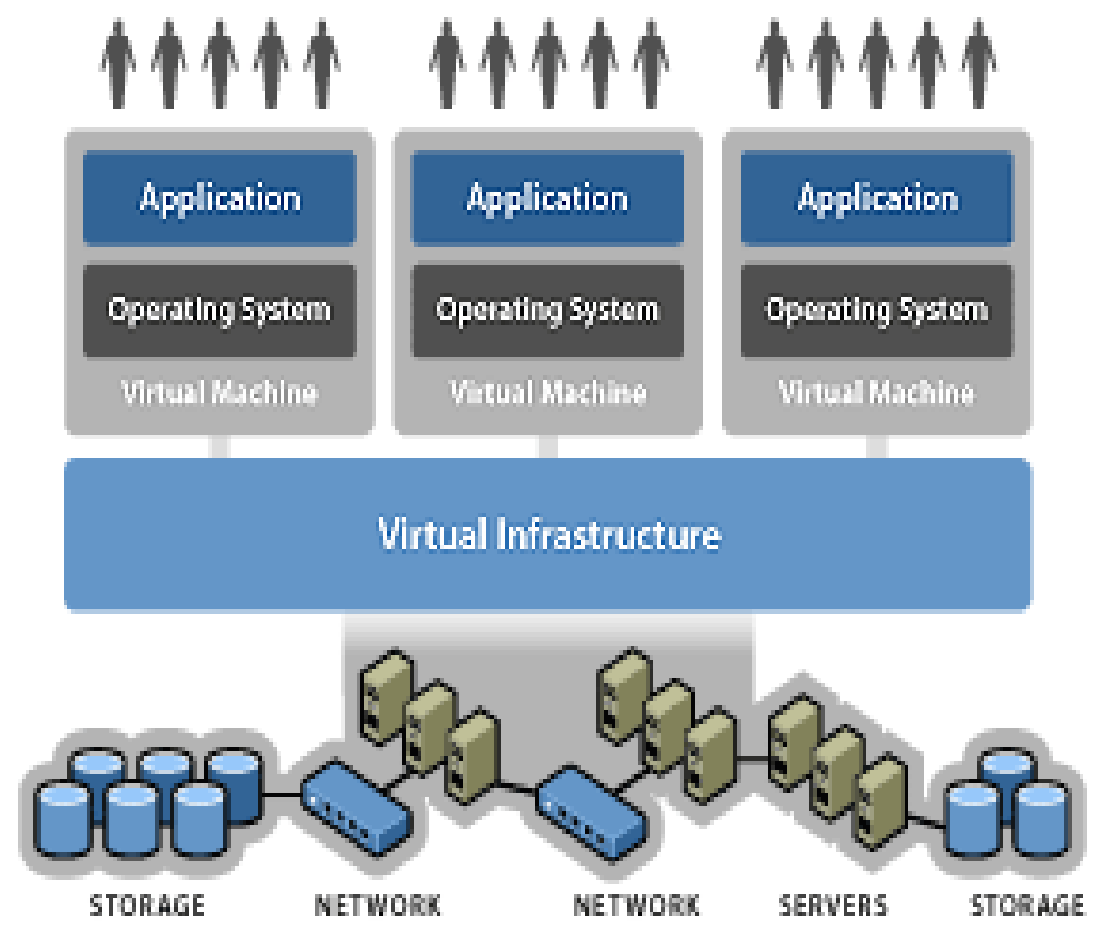
- Reduced control over software and data
  - Worse in the case of federated clouds as you do not know who is actually the cloud provider in the federation that has your software and data
- Multi-tenancy can lead to breaches of data integrity, confidentiality and privacy
- Interference between complex security mechanisms that might exist at different layers in a cloud (infrastructure, platform and software) → vulnerabilities
- Interference between security and cloud virtualisation/optimisation mechanisms,

What is a virtualized infrastructure?

Virtualized infrastructure (1)

- A virtualized infrastructure creates a *dynamic mapping* between (virtual) IT resources and IT requirements
- Ingredients:
  - **A physical IT supply infrastructure** with an access network
  - **Three suppliers**
    - COMPUTE
    - NETWORK
    - STORAGE
  - **Many users**
    - Requiring IT at different granularities: applications (SaaS), clients/servers (PaaS), networks/data centers (IaaS)

Virtual infrastructure

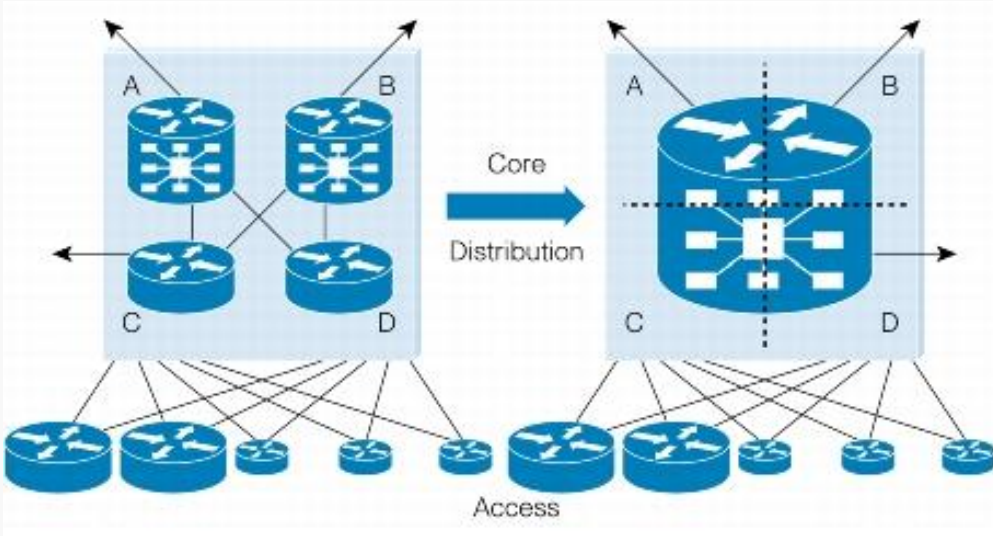


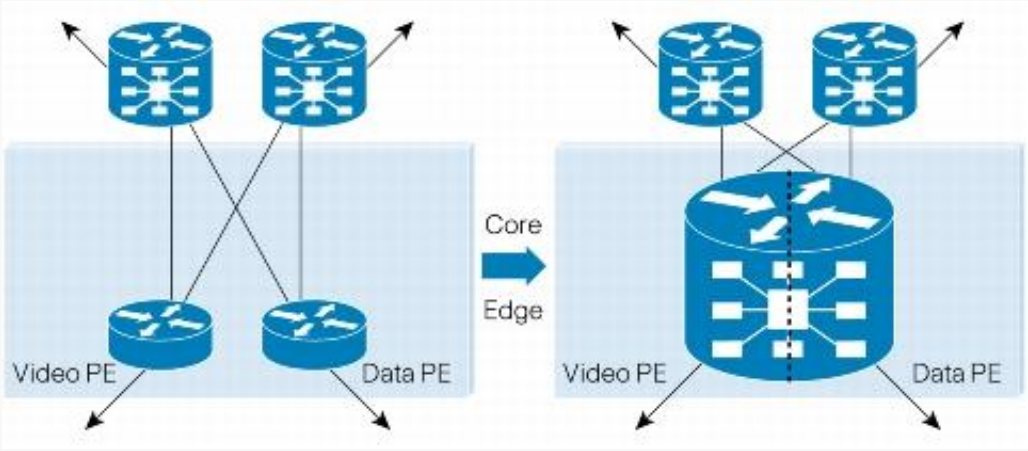
- De-couple software environment from hardware infrastructure
- Use virtual networking to aggregate virtual servers and storage in resource groups
- Allocate resource groups to application/processes /functions

- **No need to trunk**

Network Virtualization

- Objectives
  - "Vertical" consolidation
    - do all at layer 2
  - "Horizontal" consolidation
    - do all (data, voice, video) on the same network.
- Tools
  - (Complex and sophisticated) virtual appliances over (simple) commodity hardware





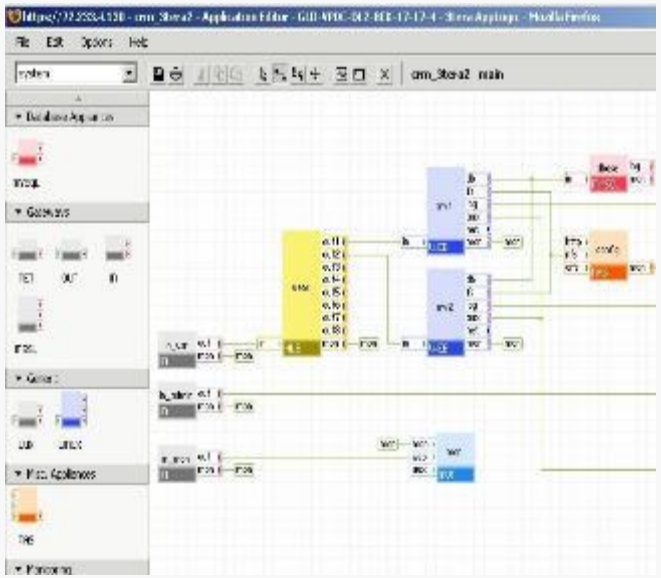
Where it is used

- *Network virtualization* is applied to provision, rapidly evolving, resource-intensive environments
  - Handle complexity both from a control plane and data plane perspective.
- Example: POPs and core network environments
  - Requirement: Aggregation point of all customers in a particular geographical region
    - Many routing adjacencies
    - full Internet routes to be exchanged among routing peers
    - High bandwidth demands (greater than 10 Gbps).
  - Answer: Use a simple physical infrastructure "on premises", with rack space and power, and create the environment on top of it

Evolution of Tools

- Hardware-Isolated Virtual Routers (HVR) have hardware-based resource isolation between routing entities
- Software-Isolated Virtual Routers (SVR) rely on software-based resource isolation between routing entities.
  - Problem: contention of resources.
  - Solution: overprovision resources on all SVRs so that no individual SVR is likely to affect the others.

Cooking up a Virtual Environment



Central notions:

RECIPE

Configuration information (e.g. in XML) defining an entire stack (OS/storage/application) to be launched on top of a virtualization infrastructure

COOKBOOK

A set of ready-to-cook recipes

KITCHEN

The environment where you do your cooking

Includes:

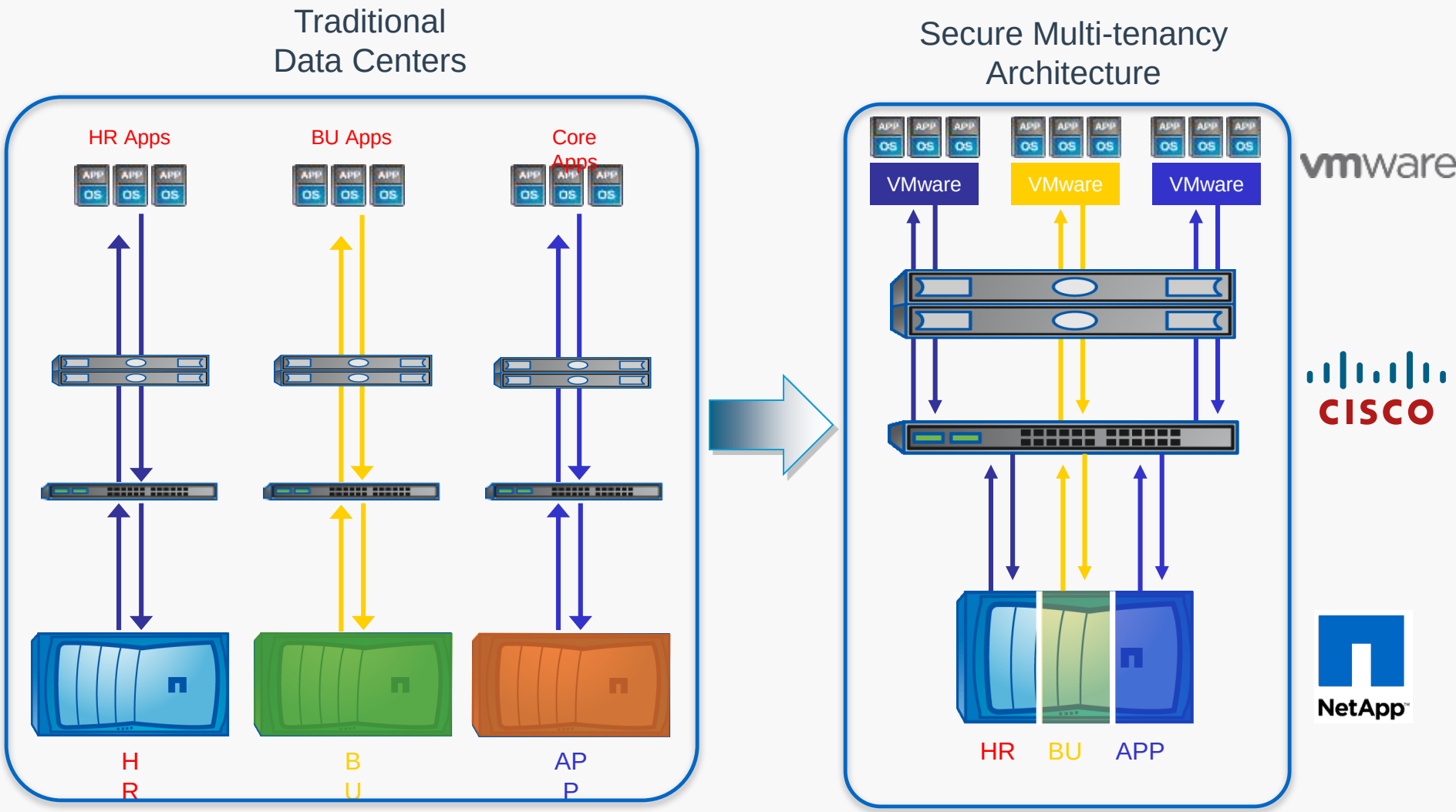
Stove

Where recipes are defined/created/tested

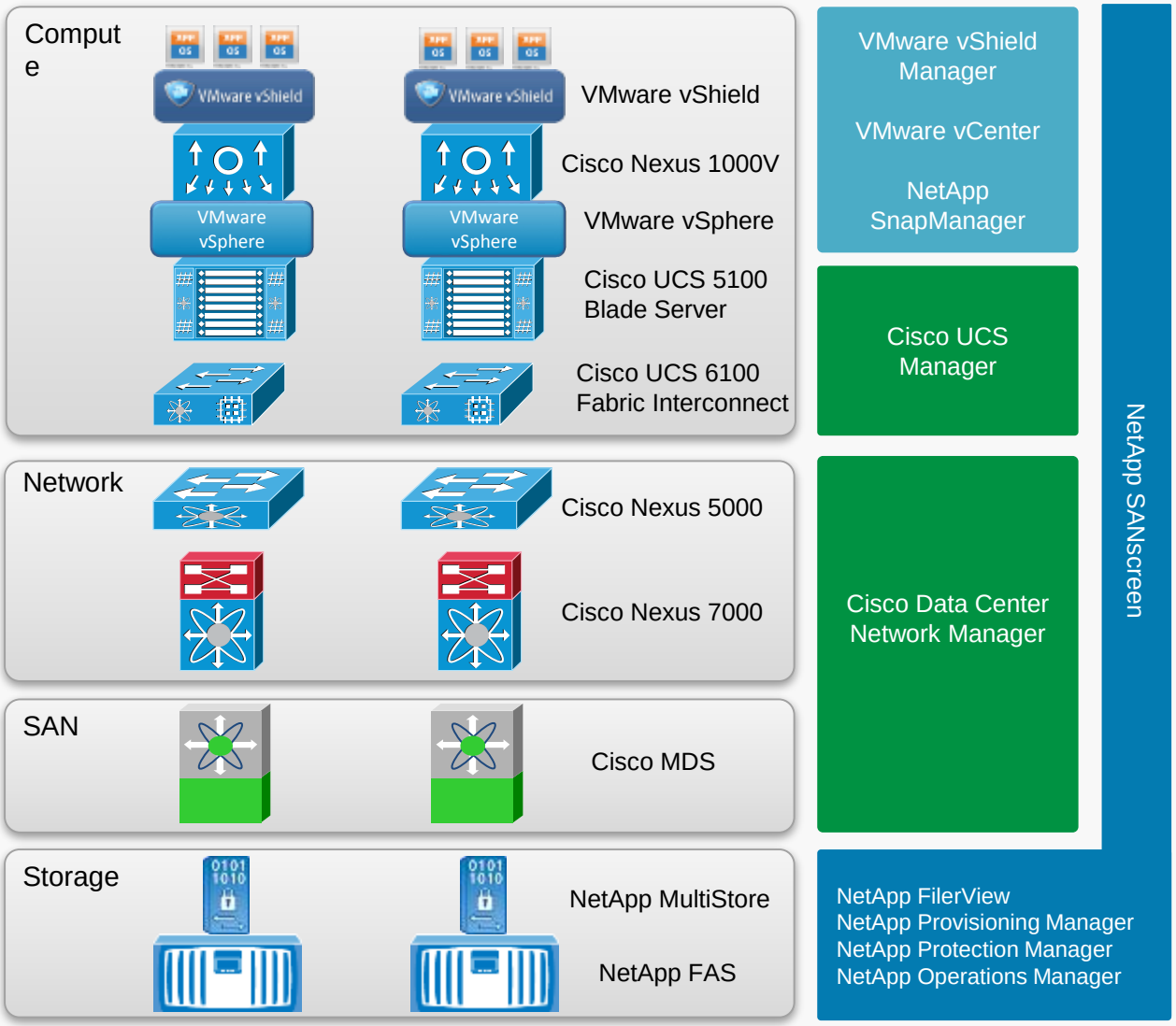
Storeroom

Where recipes and ingredientsare kept/shared

From Virtualization to Multi-tenancy



Sample Architecture



Compute

- VMware vShield
- VMware vSphere
- Cisco Unified Computing System

Network

- Cisco Nexus 1000V
- Cisco Nexus 5000
- Cisco Nexus 7000
- Cisco MDS

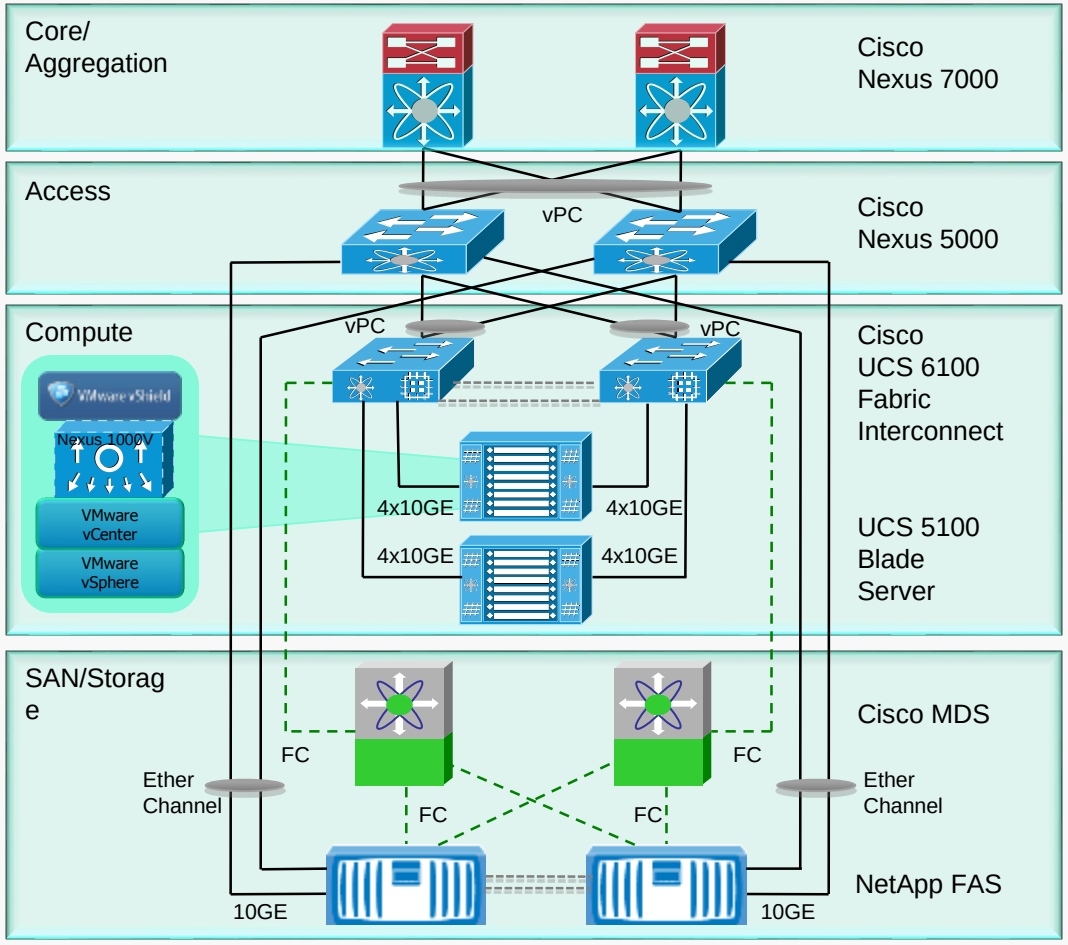
Storage

- NetApp FAS
- NetApp Multistore

Management

- VMware vShield Manager
- VMware vCenter
- Cisco UCS Manager
- Cisco DC Network Manager
- NetApp Operations Manager
- NetApp Provisioning Manager
- NetApp SANscreen & SnapManager

A closer look



Compute

- vCenter Heartbeat
- VMware HA
- vMotion/Storage vMotion
- UCS Fabric Redundancy

Network

- vPC
- EtherChannel
- N1KV Active/Standby VSM
- Link/Device Redundancy

Storage

- RAID-DP
- NetApp HA
- Snapshot
- SnapMirror/SnapVault

Separating tenants

Compute

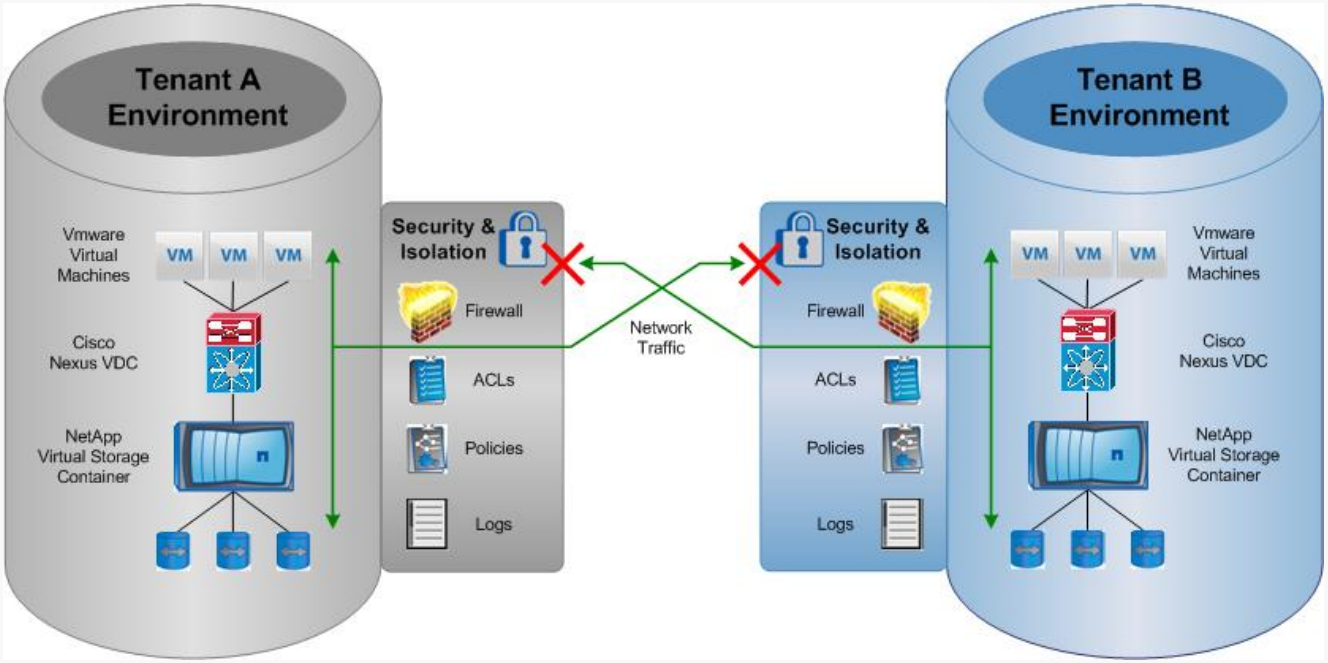
- UCS & vSphere RBAC
- VM Security with vShield and Nexus 1000V
- UCS Resource Pool Separation

Network

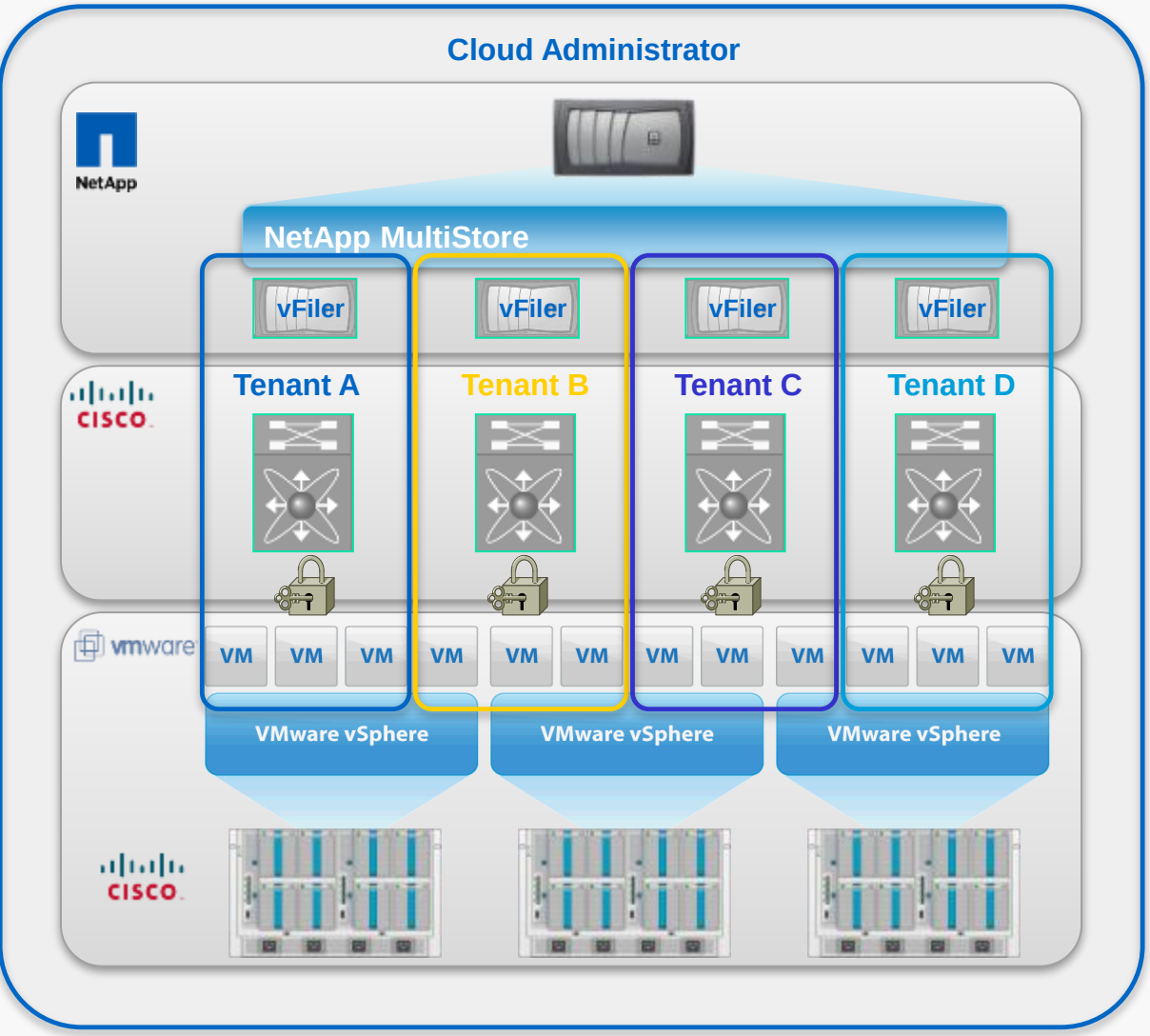
- Access Control List
- VLAN Segmentation
- QoS - Classification

Storage

- vFiler units
- IP Spaces
- VLAN Segmentation



Access control



Define Roles

- Cloud Administrator
- Tenant Administrator
- Tenant User

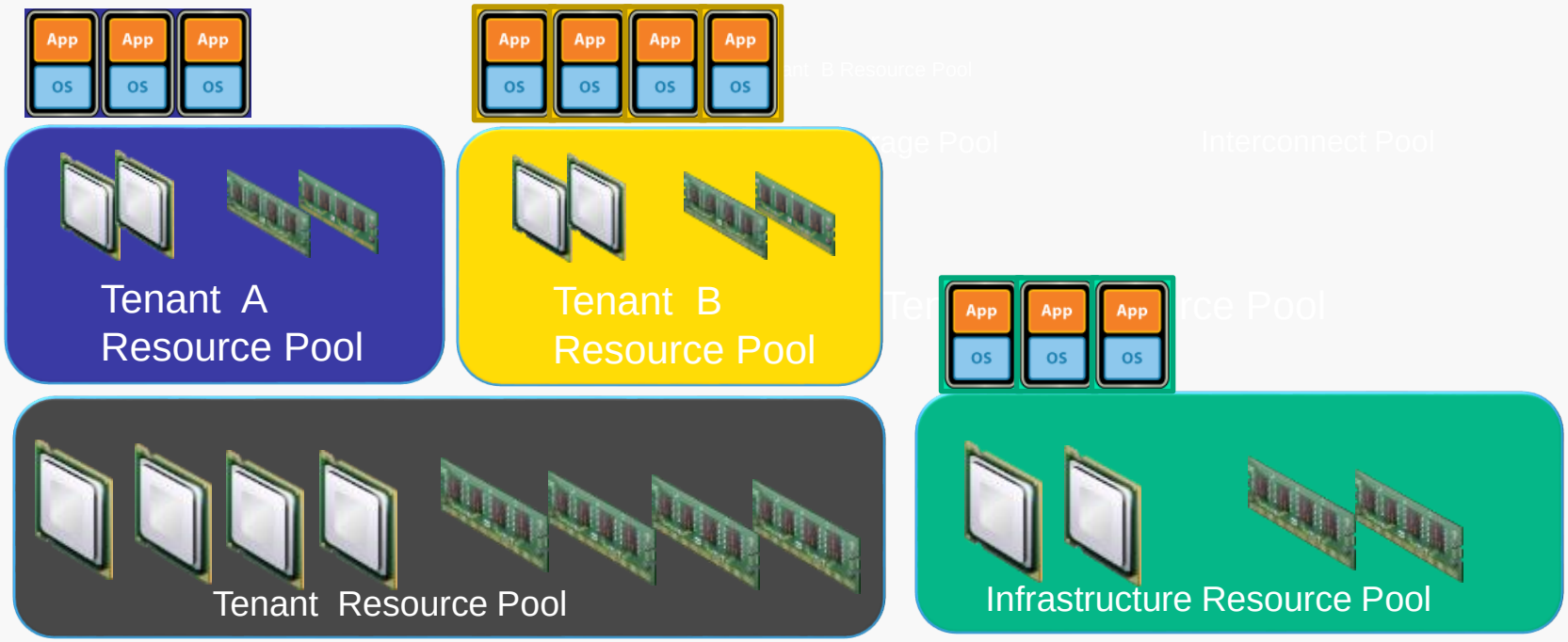
Role Based Access Control

- UCS Manager
  - Server Admin
  - Network Admin
  - Storage Admin
  - Customized Admin
- vCenter
  - Privilege Assignment
  - User Group Association
  - Permission Assignment

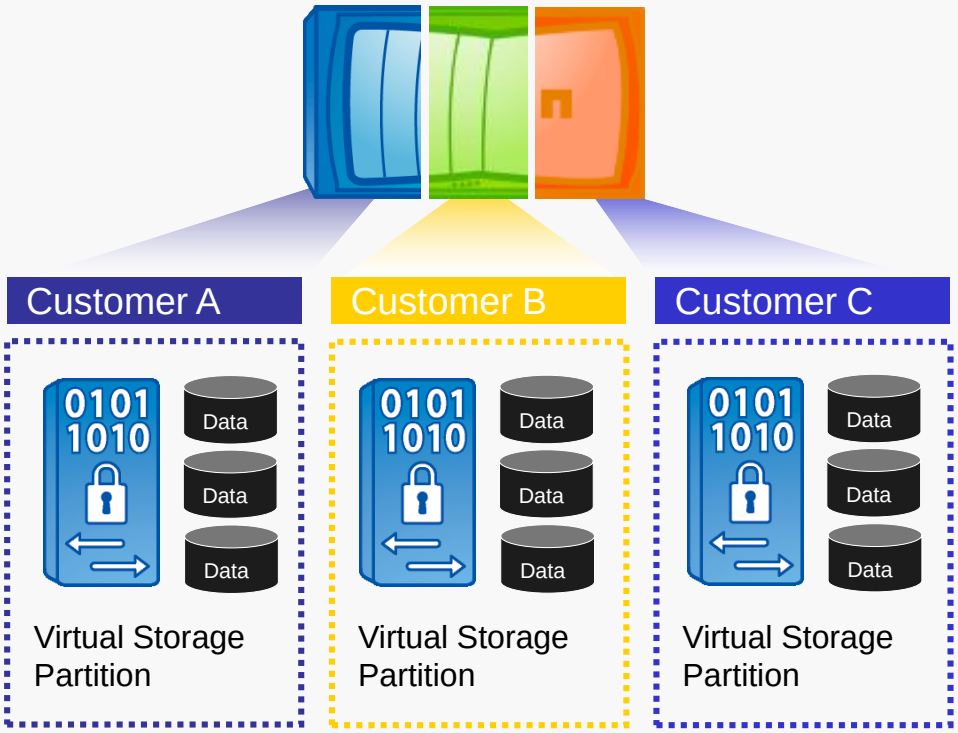
Separating tenants (2)

vSphere Resource Pool Design Best Practice

- Dedicated resource pools for infrastructure and tenants
- Separate sub-resource pool for individual tenants
- Combined with RBAC to securely isolate access between tenants



Separating tenants (3)



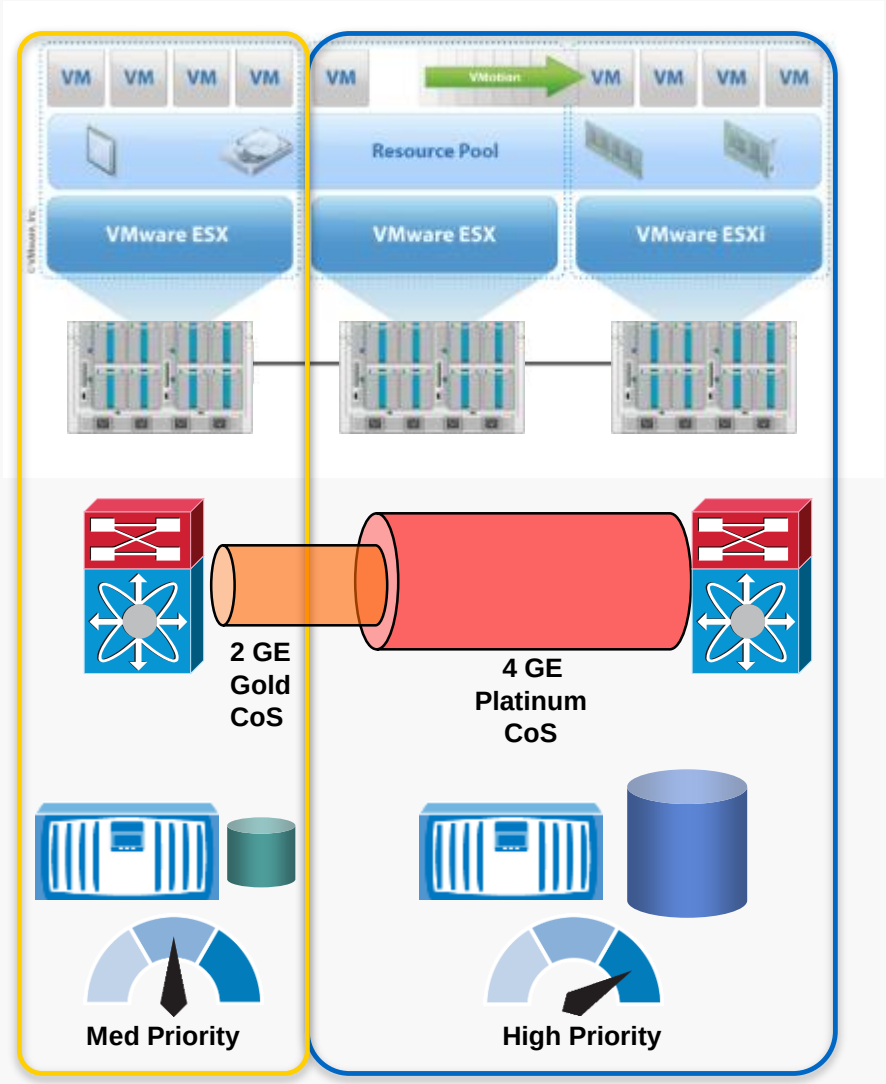
Secure multi-tenancy MultiStore

- Secure partition of storage and networking
- Proven technology: 16,000 licenses
- Third-party valid security testing

What is Virtualized Infrastructure's Assurance?

First of all, SLA….

Managing SLA



Compute

- Expandable Reservation
- Dynamic Resource Scheduler
- UCS QoS System Classes for Resource Reservation and Limit

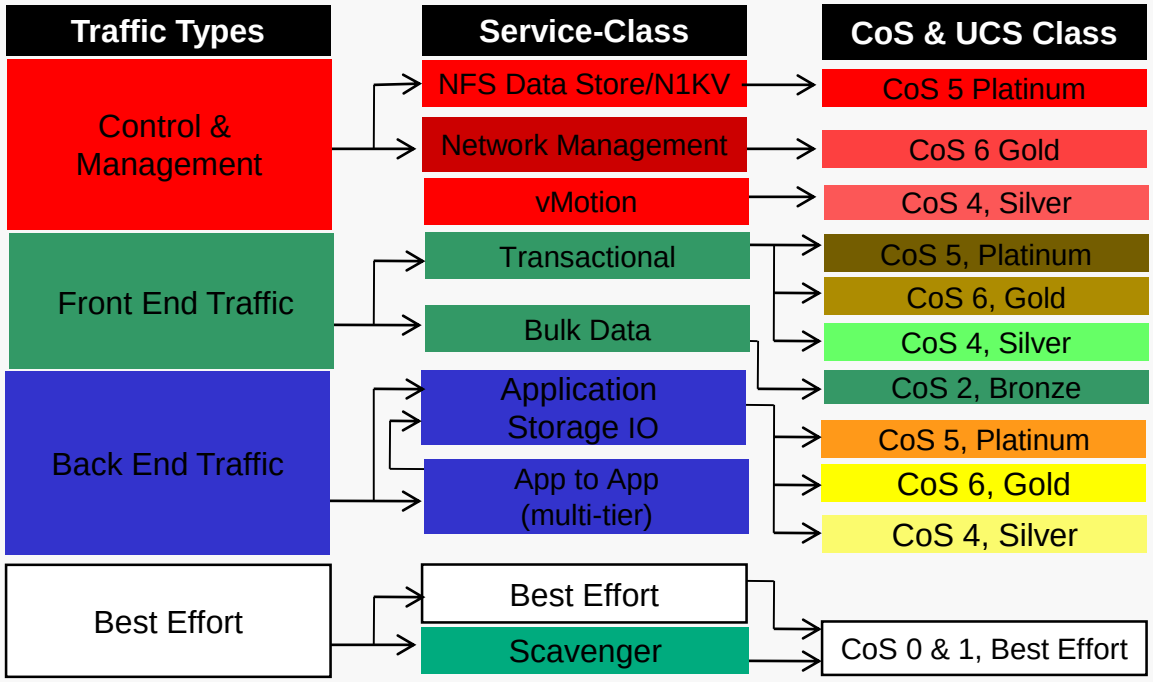
Network

- QoS - Classification
- QoS - Queuing
- QoS - Bandwidth control
- QoS - Rate Limiting

Storage

- FlexShare
- Storage Reservations
- Thin Provisioning

Network Service SLA



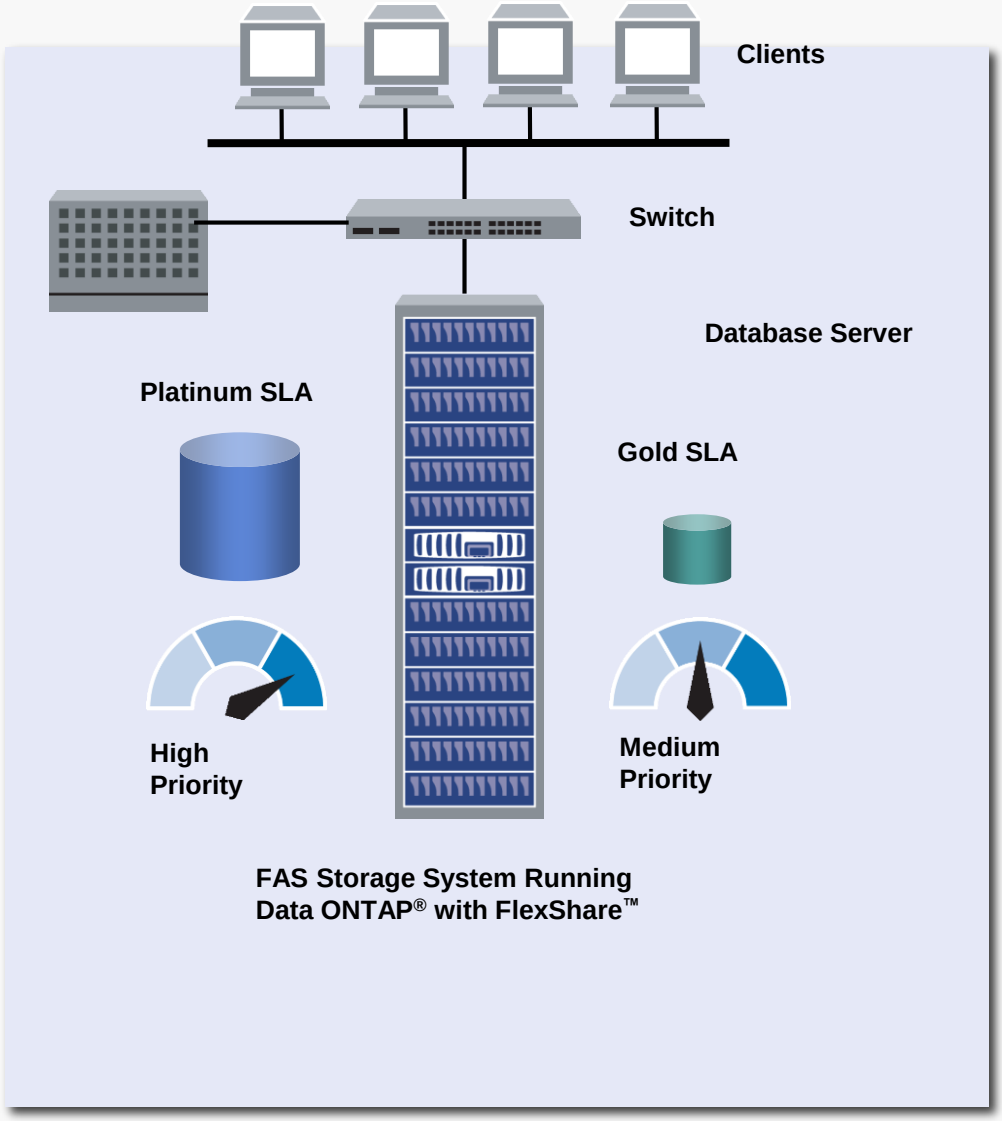
- QoS – Classification
  - Classification Capability
  - Identify Traffic Types
  - Classify at Source of Origin
- QoS – Queuing
  - Packet Delivery Schedule
- QoS - Bandwidth Control
- QoS – Rate Limiting

Computing Service SLA

Resource Pool Settings	Platinum Tenant	Gold Tenant	Silver Tenant
Reservation	Reserved	Reserved	No reservation
Limits	Unlimited	Limited	Limited
Shares	High	Medium	Low
Expandable Reservation	Enabled	Disabled	Disabled

- Built-in vCenter Resource Pool settings
  - Resource guarantee for infrastructure and tenant services
- Resource pool settings to be set based on tenant SLA
- For example, VMware DRS provides automated load distribution across all blades in the ESX Cluster

Storage SLA



- Set high priority for database (or Platinum) SLA
- Multiple levels of prioritization available
- Isolates tenant performance
- .

What about security assurance?

Related work

Security risks assessment

- QUIRC: Quantitative impact and risk assessment framework

  $R_O = 1/n \, \Sigma_{e=1,\ldots,n} P_e \times I_e$ (Risk = Likelihood × Impact)

- Security risk assessment (without an explicit cloud focus) :

  CRAC++ [19], COBRA [20], CORAS [21]

- Governance, Risk management and Compliance Stack (GRC stack; by *Cloud Security Alliance*):
  - Cloud Controls Matrix: principles and guidelines to assess the overall security of a cloud provider [14]
  - Consensus Assessments Initiative Questionnaire (CAIQ [15]): questions designed to help cloud customers and auditors to identify gaps in CCM controls in specific cloud providers
  - CloudAudit: common interface and namespace to enable the audit and assessment of the security of cloud services [12]
  - Cloud Trust Protocol: protocol for obtaining evidence for cloud operations
- IT audit practices and standards: industry driven (Service Organisation Controls (SOC), ISO27001); labour intensive and static

Certification

- Software certification is not new (e.g., Common Criteria model) BUT
  i. Covers monolithic systems
  ii. Targets humans → certificates not amenable to automated processing, e.g.,
    - cannot be used for automated (and possibly on-fly) system component selection/replacement, verification etc)
  iii. Cannot cope with changes to system structures and the operational environment
- Recent work on SOA certification (Assert4SOA project [22]) covers (i)-(iii) in some circumstances
  - Schema for specifying machine processable service certificates
  - Ontologies for annotating certificates
  - Certificates aware software service discovery and SaaS level composition [23]

The idea

Development of an integrated framework of models, processes, and tools supporting the ***dynamic certification of assurance*** *related to security/privacy/dependability properties.*

Suitable for infrastructure (IaaS), platform (PaaS) and software application services (SaaS) in clouds.

The framework will use multiple types of assurance evidence including

- *testing (evidence),*
- *monitoring (evidence) and*
- *trusted computing proofs,*

and models for

- *hybrid,*
- *incremental and*
- *multi-layer security certification.*

Objectives

- **Objective 1:** Development of advanced service certification models based on service testing data, service monitoring data, and trusted computing platforms proofs and supporting hybrid, incremental and multi-layer certification.
- **Objective 2:** Development of an interoperable certification infrastructure for generating, maintaining and using certificates according to the different types of certification models.
- **Objective 3:** Delivery of an interoperable certification solution and contribution to standards.

Objective 1

- **Objective 1:** Development of advanced service certification models based on service testing data, service monitoring data, and trusted computing platforms proofs and supporting hybrid, incremental and multi-layer certification for clouds.
- **Objective 2:** Development of an interoperable certification infrastructure for generating, maintaining and using certificates according to the different types of certification models.
- **Objective 3:** Delivery of an interoperable certification solution and contribution to standards.

OBJ 1: hybrid certification

- What?

  Certification of assurance based on a combination of different types of evidence

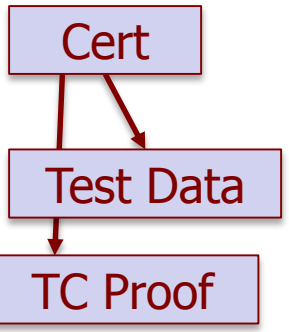
  - testing data
  - monitoring data
  - trusted computing proofs for the hardware elements of cloud infrastructures

- Why?

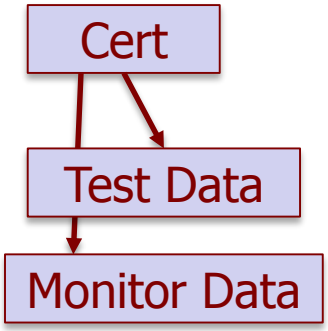
  Some properties might be certifiable using a combination of evidence types

OBJ 1: hybrid certification – examples

- The availability of a service S may be certified by a certificate that is based on test data for the service as well as a TC proof for the configuration of the hosting cloud infrastructure (to ensure that the infrastructure where the service is deployed is the same as that for which test data were obtained)



- Hybrid certificate for software service availability based on test data and continuous monitoring in real operating conditions



OBJ 1: multi-layer certification

What?

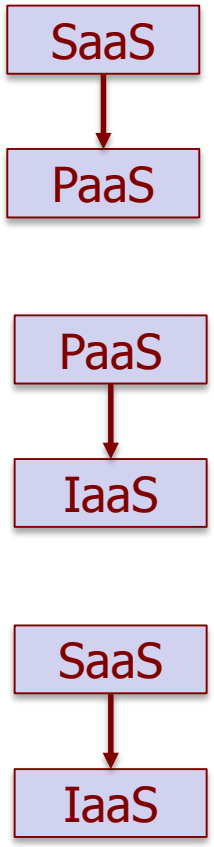
- Certification based on a combination of certificates of inter-dependent services (as opposed to simply “evidence”) at different layers of the cloud stack

Why?

- “Recipes” security properties are affected by such dependencies
- Inability to obtain the direct evidence required for property assessment) require making assessments on the basis of certificates rather than direct evidence

OBJ 1: multi-layer certification – examples

- The *integrity of data-at-rest* of a software service S_1 using a cloud storage service S_2 could under certain circumstances be certified on the basis of a certificate regarding the correct implementation of a *"proof-of-storage" protocol* by S_2
- The *availability* of a *messaging service* in a cloud federation may be certified on the basis of certificates regarding *DoS-resilience* of the hosting node(s) in the federation
- A *data-in-process integrity* certificate of a SaaS layer service requires *TCP based certificate for hypervisor* as the latter can ensure correct monitoring of security conditions of infrastructure services that are necessary for data-in-process integrity, and avoidance of data leaks of relevant monitoring data



OBJ 1: incremental certification

- What?

  Ability to cover changes that may affect certified properties of cloud services without having to re-certify properties from scratch

- Why?

  - Operational conditions within a cloud infrastructure may change
  - Cloud services and data may migrate to different cloud infrastructures within a cloud federation
  - Constituent services of composite services may be substituted (whether co-tenant or not)

OBJ 1: incremental certification – examples

- Re-validation of certificate due to changing operational conditions, e.g.:

  the certificate C for data integrity of a software service requires a certificate C' for the data isolation scheme operated by the cloud storage service;

  the software service migrates to a different node in a cloud federation →

  C needs to be revalidated by considering whether the new hosting cloud has a certificate equivalent to (or appropriate substitute for) C'

- Use continuous monitoring to create new certificates or "strengthen" existing certificates with increased operational evidence, e.g.,

  The certificate of data-isolation for software service in a given infrastructure requires isolation of co-tenant services in the infrastructure; the certificate is continually validated through continuous monitoring of the infrastructure

OBJ 1: Certification models

- Purpose:

  To determine the evidence (type and extent) that needs to be considered to be able to certify a security property and how it will be used to assess the property

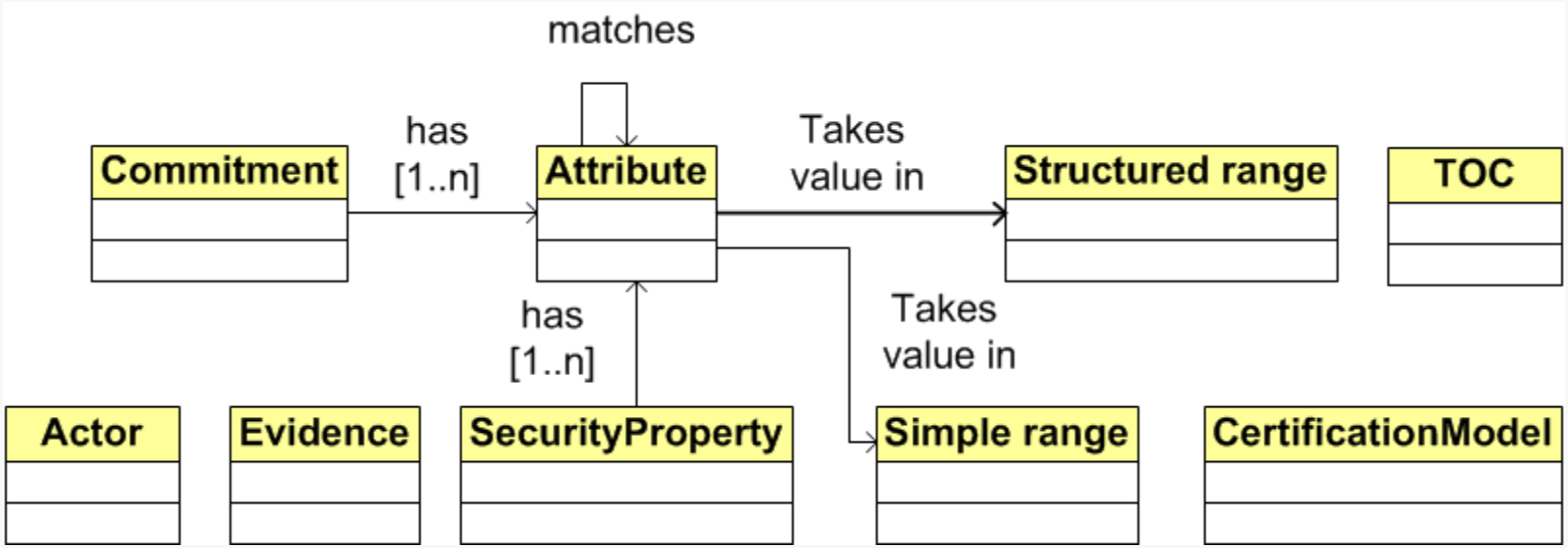
- Address questions of the form
  - When two distinct pieces of evidence can be considered equivalent for a given security property?
  - If conflicting evidence arises what happens to the certificate?
  - Should a certificate be revalidated/revoked when:
    - The composition of a service changes
    - The deployment configuration of a service changes (e.g., code or data migration to another node in a federation)
    - The configuration of an infrastructure changes
  - How certificate re-validation should be carried out? for example:
    - Could equivalent security properties be considered sufficient?
    - Could alternative equivalent pieces of evidence be used?

Some modeling…

Cloud Certification Meta-Model

- **Meta-classes: specify shared concepts, elements, and relationships**
  - ***Security properties* and *commitments***
  - ***Target of certification* (service-unit, resource-groups, resources in CSA document)**
  - ***Actors***
  - ***Models of certification***
  - ***Evidence***

CUMULUS Meta-Model

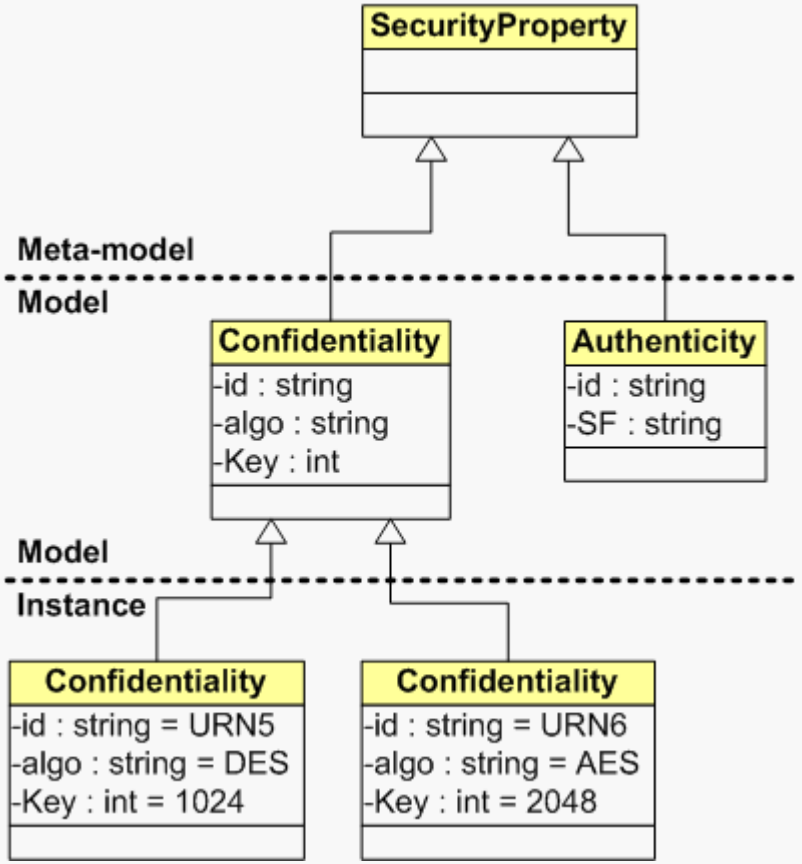


Security Property: Model

- **Security properties (*security attributes fully qualified type* in the Cloud Security Alliance terminology)**
  - **Express abstract security properties**
    - **E.g., confidentiality, integrity, authenticity**
  - **May have a set of attributes that refine the abstract property (*attribute parameter template* and *measurement parameter* in CSA document)**
    - **Refer to security functionalities (e.g., encr-algo=DES)**
    - **Refer to threats (e.g., attack=MIM)**
    - **Refer to contextual information (e.g., OS=Linux)**

Security Property: Example

- Meta-Class: SecurityProperty
- Class
  - Confidentiality
    - Att1: id [String]
    - Att2: algo [String]
    - Att3: key [Int]
  - Authenticity
    - Att1: id [String]
    - Att2: SF [String]
- Instance
  - Confidentiality
    - id=URN5
    - algo=DES
    - key=1024
  - Confidentiality
    - id=URN6
    - algo=AES
    - key=2048

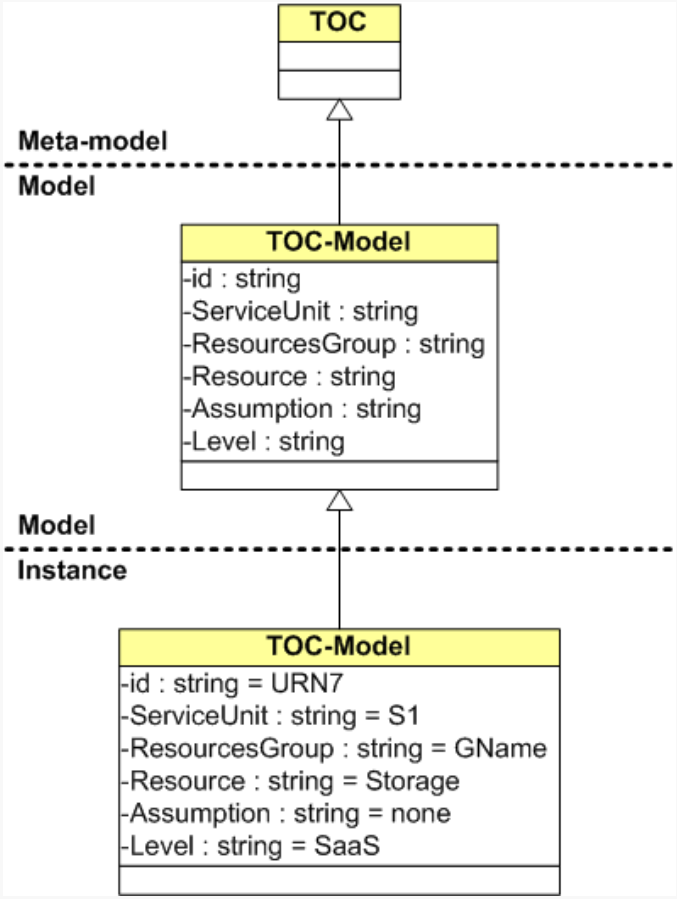


Target of Certification (TOC): Model

- **Target of certification**
  - **Service-unit, resource-groups, resources in CSA document**
  - **Assumptions on the TOC (e.g., HW in EU)**
    - **Possibly part of the security property**
  - **It can be the service under certification (SaaS), the platform deploying services (PaaS), the infrastructure hosting platforms and services (IaaS) or any combination of the above**

Target of Certification (TOC): Example

- Meta-Class: TOC
- Class
  - TOC-Model
    - Att1: id [String]
    - Att2: ServiceUnit [string]
    - Att3: ResourceGroup [string]
    - Att4: Resource [string]
    - Att5: Assumption [string]
    - Att6: Level [string]
- Instance
  - TOC-Model
    - id=URN7
    - ServiceUnit=S1
    - ResourceGroup=GName
    - Resource=Storage
    - Assumption=None
    - Level=SaaS

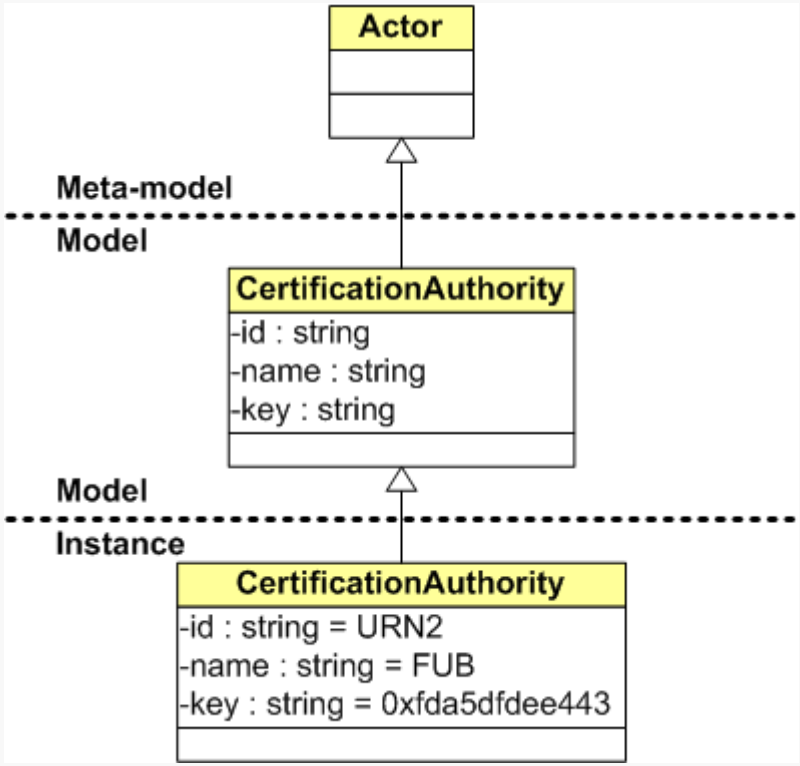


Actors: Model

- "Actor" models
  - CUMULUS Clients (searching certified resources)
  - Service Providers (providing services/platforms)
  - Cloud Providers (providing the infrastructure)
  - Certification Authority
  - CUMULUS Certification Infrastructure
  - Attacker
- Compliance with other cloud actors models (e.g., NIST)

Actors: Example

- **Meta-Class: Actor**
- **Class**
  - **CertificationAuthority**
    - **Att1: id [String]**
    - **Att2: name [String]**
    - **Att3: key [String]**
- **Instance**
  - **CertificationAuthority**
    - **id=URN2**
    - **name=FUB**
    - **key=0xfda5dfdee443**

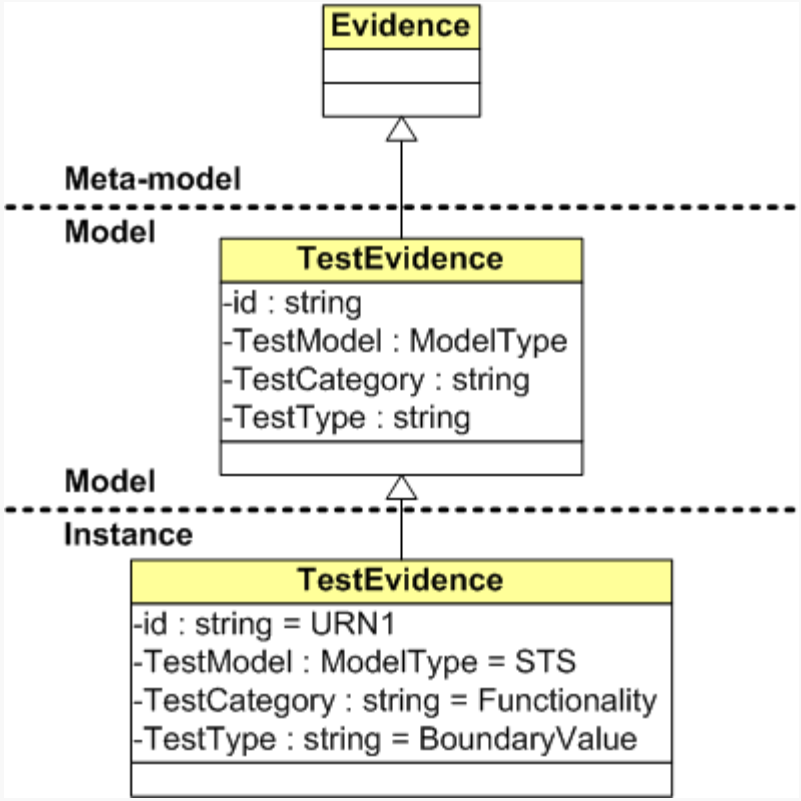


Evidence: Model

- A set of artifacts supporting a given property for the TOC
  - Verification model: a model used to produce the evidence
  - Verification mechanism: the mechanism used to produce the evidence
- Verification model and mechanism depend on the selected model of certification

Evidence: Example

- Meta-Class: Evidence
- Class
  - TestEvidence
    - Att1: id [String]
    - Att2: TestModel [ModelType]
    - Att3: TestCategory [String]
    - Att4: TestType [String]
    - ...
    - Attn
- Instance
  - TestEvidence
    - id=URN1
    - TestModel=STS
    - TestCategory=Functionality
    - TestType=BoundaryValue

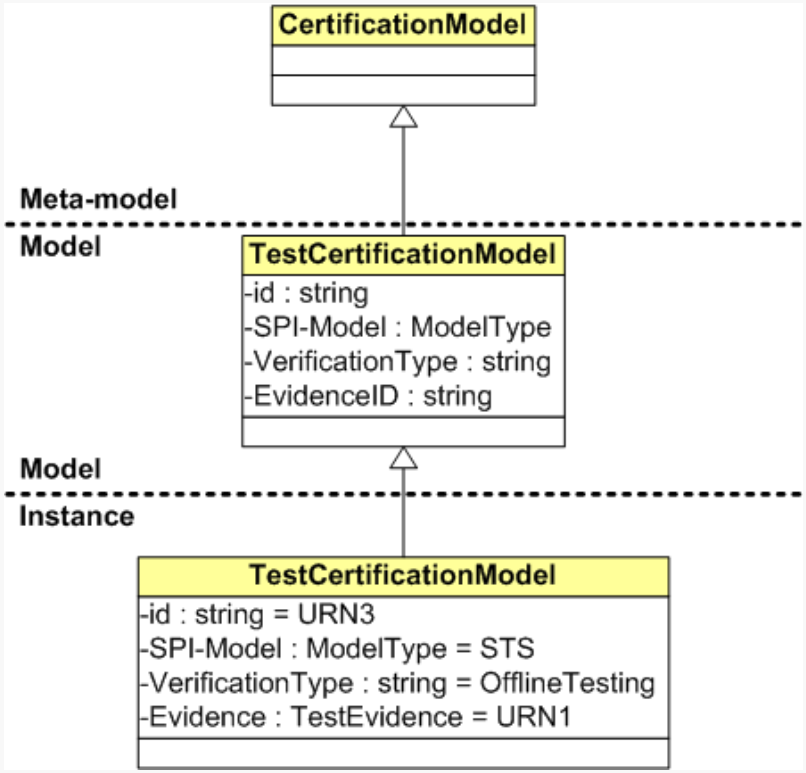


Models of Certification: Model

- **Each model of certification includes the elements needed for a given class of certification**
  - **Service/Platform/Infrastructure (S/P/I) model**
  - **Verification type**
    - **Test, Monitoring, TPM, hybrid, incremental**
    - **Offline (Static), Online (Dynamic)**
  - **Evidence (instance of the evidence meta-class)**
  - **Others**

Model of Certification: Example

- Meta-Class: CertificationModel
- Class
  - TestCertificationModel
    - Att1: id [String]
    - Att2: S/P/I-Model [ModelType]
    - Att3: VerificationType [String]
    - Att4: Evidence [TestEvidence]
    - …
- Instance
  - TestCertificationModel
    - id=URN3
    - S/P/I-Model=STS
    - VerificationType=OfflineTesting
    - Evidence=URN1

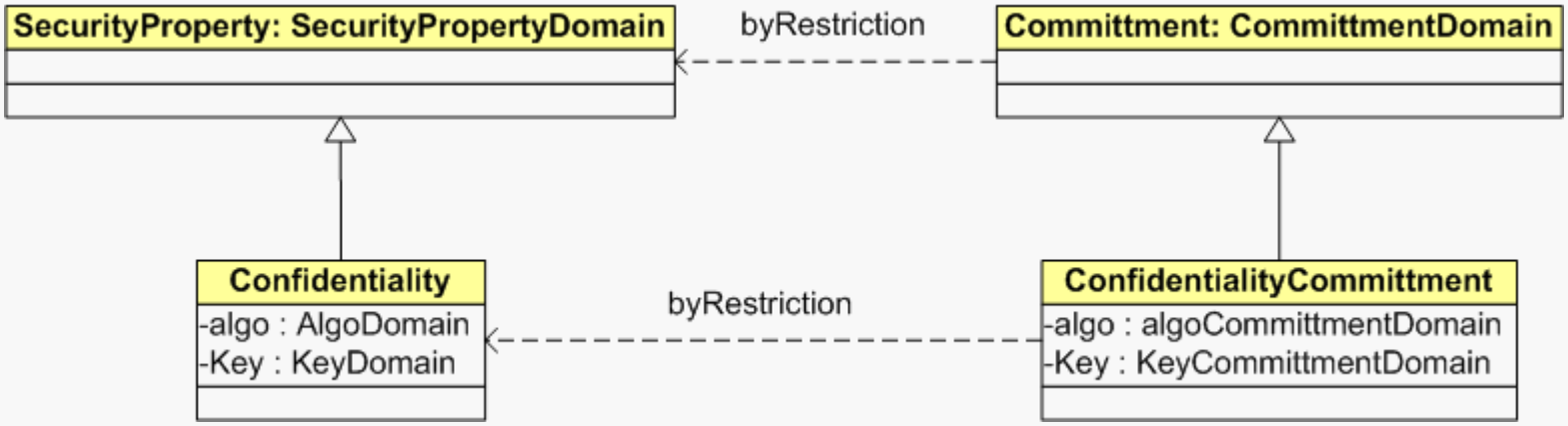


Authenticity Example

- **Complete example from meta-model to instance**
- **Consider complex types including formulas**

Security SLAs - Security Property Food for Discussion

- **SLA are based on commitments**
- **At the meta-model level, define commitments by restriction, that is, as a sub-class of security properties**
  - **Security properties defined on *security property domain***
  - **Commitments defined on *commitment domain***
- **Commitment domain is a restriction of security property domain**



Security SLAs - Security Property Food for Discussion

- The MOST IMPORTANT attribute slot of a property is the one corresponding to the mechanism.
  - This is the reason why this attribute is mandated (or at least suggested) by the meta-model to any modeler wishing to set up a model.
- The main slots of any property are the name, a subject, a TOC and a mechanism

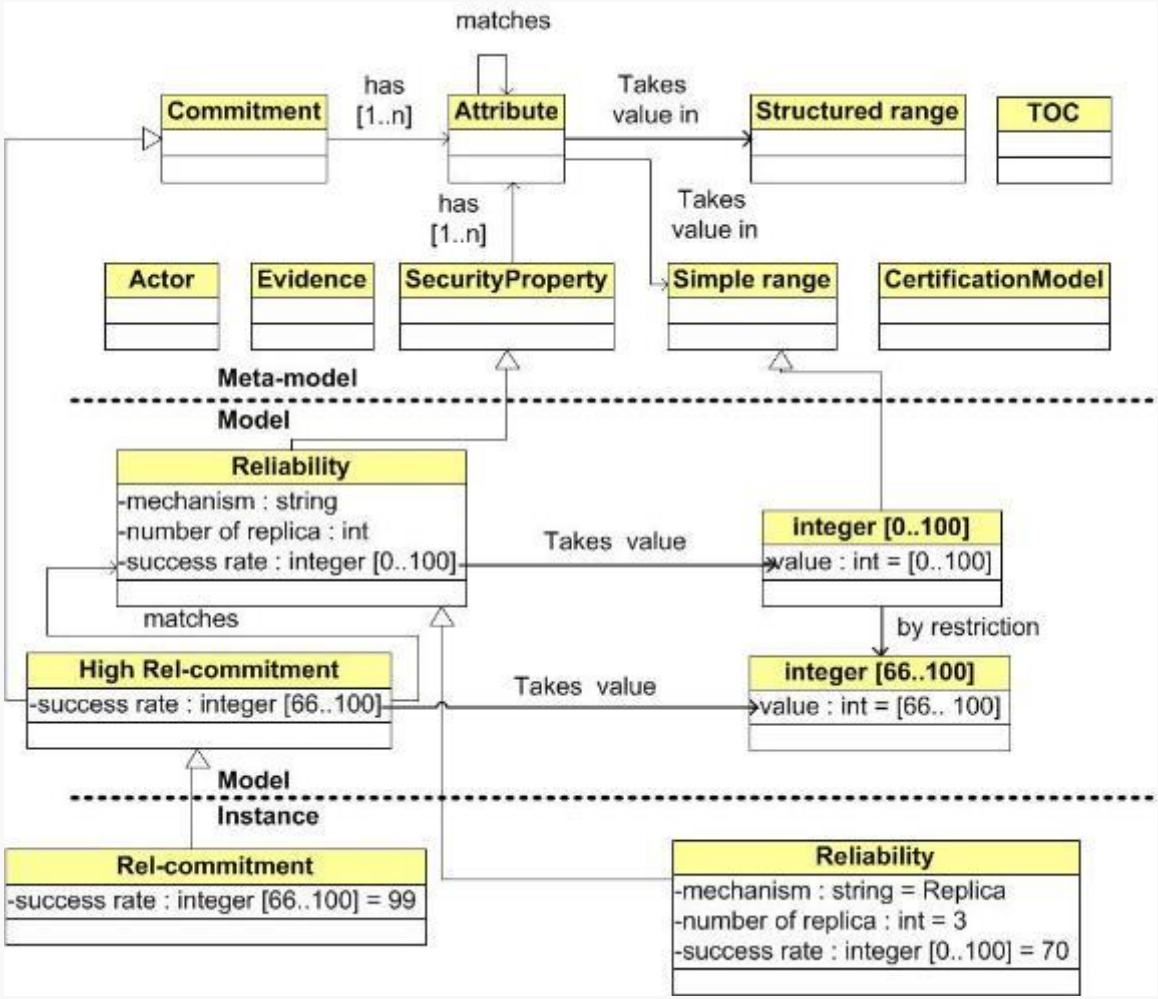
Value-related properties

- The meta-model puts a (soft) constraint on the types that slots will be allowed to have in models
  - Whatever the modeler comes up with as the mechanism slot, it must take values in a domain which is a RESTRICTION of the generic domain mentioned in the meta model
- The slot typing constraints also affect the relation between a property and a commitment on that property: all slots in the commitment must belong to types that are restrictions of the types of the corresponding property slots.

Performance-related properties

- For "performance-related" properties, the "mechanism" slot will not point to a value (be it a simple type or a structured type), but to a **typed monitor**.
  - Example: in the case of some dependability-related properties, say redundancy, asserting the number of replicas as an integer value is just not useful.
- The meta-model will say that the slot must belong to a procedural type; thus the modeler will be advised to assign to that slot a specific procedural type, e.g. the endpoint of a monitor that returns an integer, plus an expected return value of that endpoint (say, 3).
- In an availability SLA, a commitment on redundancy will be a restriction, e.g. an interval over the procedural type domain (say [2-3])

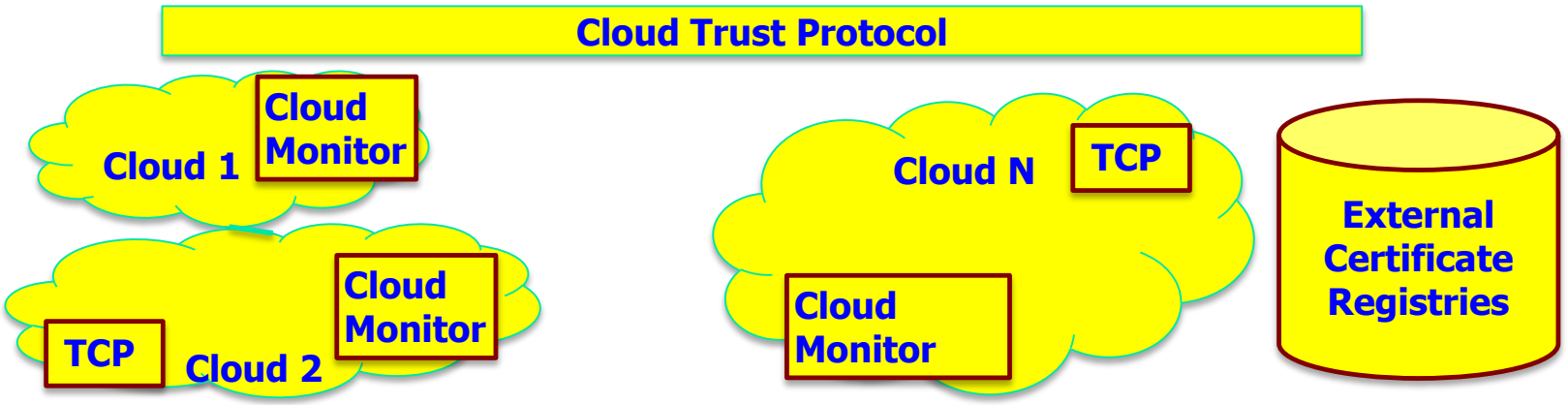
Reliability Example



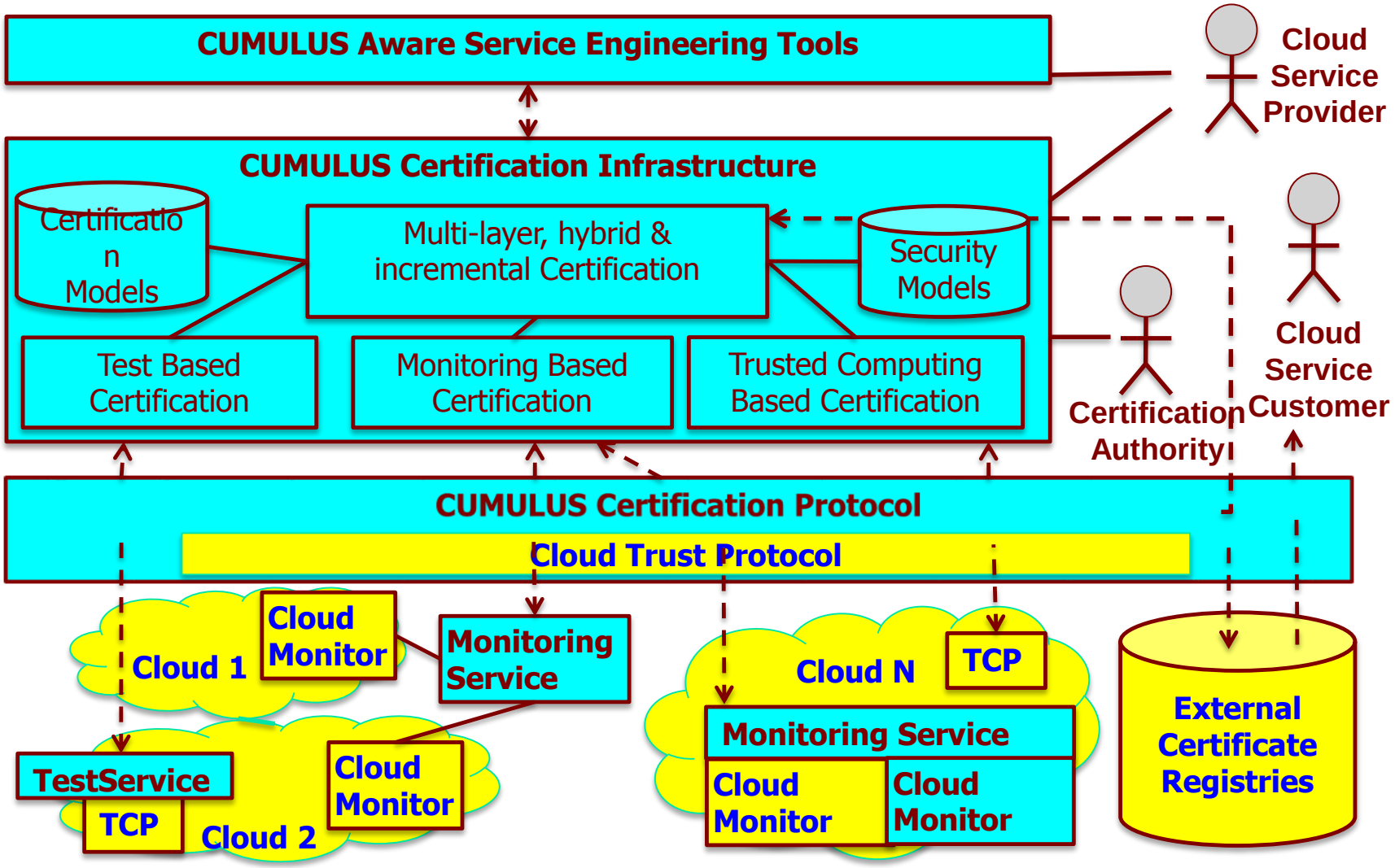
Objective 2

- **Objective 1:** Development of advanced cloud service certification models based on service testing data, service monitoring data, and trusted computing platforms proofs and supporting hybrid, incremental and multi-layer certification.
- **Objective 2:** Development of an interoperable certification infrastructure for generating, maintaining and using certificates according to the different types of certification models.
- **Objective 3:** Delivery of an interoperable certification solution and contribution to standards.

OBJ 2: CUMULUS Infrastructure



OBJ 2: CUMULUS Assurance Infrastructure

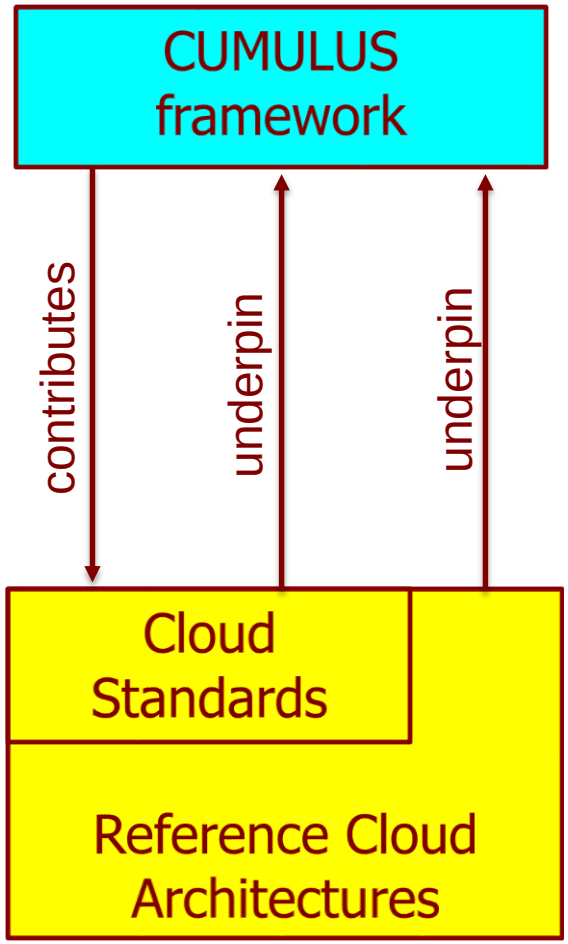


Objectives

- **Objective 1:** Development of advanced cloud service certification models based on service testing data, service monitoring data, and trusted computing platforms proofs and supporting hybrid, incremental and multi-layer certification.
- **Objective 2:** Development of an interoperable certification infrastructure for generating, maintaining and using certificates according to the different types of the certification models developed in CUMULUS..
- **Objective 3:** Delivery of an interoperable certification solution and contribution to standards.

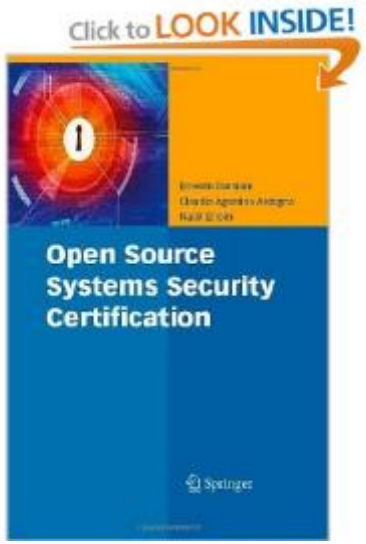
OBJ 3: interoperability & standards

- Interoperability with
  - emerging standards (e.g., GRC stack, STAR Registry) for cloud audit
  - reference cloud architectures (e.g., Nebula, CSA's reference architecture)
- Contribution to standards, e.g.:
  - OCF (CSA; ongoing)
  - ISO 27017 (Cloud controls; ongoing)
  - ISO 27018 (Privacy in public clouds; ongoing)
- Key challenge/opportunity
  - Most of these standards are under development (e.g., OCF, ISO27017)



Five readings:

- Ernesto Damiani, Claudio Ardagna, Nabil El-Ioini "**Open Source Systems Security Certification**", Springer 2009
- Jean Christophe Pazzaglia, et al., Advanced Security **Service cERTificate for SOA: Certified Services go Digital!,** Proc. of Information Security Solutions for Europe, 2011
- Marco Anisetti, Claudio Ardagna, Ernesto Damiani: **A Low-Cost Security Certification Scheme for Evolving Services**. ICWS 2012: 122-129
- Marco Anisetti, Claudio Ardagna, Ernesto Damiani, Fulvio Frati, Hausi A. Müller, Atousa Pahlevan: **Web Service Assurance: The Notion and the Issues**. Future Internet 4(1): 92-109 (2012)
- Marco Anisetti, Claudio Ardagna, Ernesto Damiani, F. Saonara, **A Test-based Security Certification Scheme for Web Services** ACM Trans. On the Web 12-0040, to appear



Other References

[1] J. Heiser and M. Nicolett, Assessing the Security Risks of Cloud Computing, Gartner Report G00157782, June 2008

[2] D. Catteddu and G. Hogben, “Cloud Computing: Benefits, Risks and Recommendations for Information Security.”, European Network and Information Security Agency (ENISA), 2009

[3] L. Kaufman. Data Security in the World of Cloud Computing. IEEE Security and Privacy 7, 4: 61- 64, July 2009.

[4] R. Austin, et al., “Domain 5: Information Lifecycle Management.”, In Security Guidance for Critical Areas of Focus in Cloud Computing V2.1, CSA Cloud Security Alliance, December 2009.

[5] T. Forsheit, et al., “Domain 3: Legal and Electronic Discovery.”, In Security Guidance for Critical Areas of Focus in Cloud Computing V2.1, CSA Cloud Security Alliance, December 2009.

[6] A. Haeberlen. “A case for the accountable cloud.”, SIGOPS Oper. Syst. Rev. 44(2): 52-57, April 2010.

[7] M. Jensen, et al., On Technical Security Issues in Cloud Computing. In Proceedings of the 2009 IEEE International Conference on Cloud Computing (CLOUD '09). IEEE Computer Society, Washington, DC, USA, 109-116., 2009

[8] T. Ristenpart, et al., Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In Proceedings of the 16th ACM conference on Computer and communications security. ACM, USA, 199-212. 2009

[9] Y. Chen, V. Paxson, R. Katz, What’s new about cloud security?, Technical Report No. UCB/EECS-2010-5, University of California at Berkeley, 2010

[10] Song, Z., Molina, J., Lee, S., Lee, H., Kotani, S., Masuoka, R. “Trustcube: An infrastructure that builds trust in client”. In: Future of Trust in Computing, Proceedings of the First International Conference, 2009

[11] Okuhara, M., Shiozaki, T., Suzuki, T., Security Architectures for Cloud Computing, Fujitsu scientific and technical journal, 46 (4): 397-402, 2010.

[12] Cloud Security Alliance, Security Guidance for Critical Areas of Focus in Cloud Computing v2.1, available from: http://www.cloudsecurityalliance.org/guidance/csaguide.v2.1.pdf

[13] NIST, Recommended Security Controls for Federal Information Systems and Organisations, NIST SP 800-53

[14] Cloud Security Alliance, Cloud Controls Matrix, Available from: https://cloudsecurityalliance.org/research/ccm/ (last accessed on 8/1/2012)

Other References

[15] Cloud Security Alliance, Consensus Assessments Initiative Questionnaire, https://cloudsecurityalliance.org/research/cai/)

[16] ISO/IEC 27001:2005

[17] Saripalli, P. and Walters, B., QUIRC: A Quantitative Impact and Risk Assessment Framework For Cloud Security, IEEE 3rd International Conference on Cloud Computing, IEEE, pp. 280 – 288, 2010.

[18] Kiran, M., Jiang, M., Armstrong, D. J., Djemame, K., Towards a Service Lifecycle based Methodology for Risk Assessment in Cloud Computing, In Proceedings of the 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC '11), pp. 449-456, 2011

[19] Morali, A. and Wieringa, R. J., Risk-Based Confidentiality Requirements Specification for Outsourced IT Systems, Proceedings of the 18th IEEE International Requirements Engineering Conference, pp. 199-208, 2010.

[20] Visintine, V., An Introduction to Information Risk Assessment, GSEC Practical, Global Information Assurance Certification Paper, Version 1.4b, 2003, http://www.giac.org/paper/gsec/3156/introduction-information-risk-assessment/105258

[21] Lund, M.S., Solhaug, B., Stolen, K., Model-Driven Risk Analysis -The CORAS Approach. Springer,2011.

[22] J.C. Pazzaglia, et al., Advanced Security Service cERTificate for SOA: Certified Services go Digital!, Proc. of Information Security Solutions for Europe, 2011

[23] Pino L., Spanoudakis G.: Finding Secure Compositions of Software Services: Towards A Pattern Based Approach, 5th IFIP Int. Conf. on New Technologies, Mobility and Security (Track on Security), 2012

[24] Spanoudakis G., Damiani E., Mana A.: Certifying Services in Cloud: The Case for a Hybrid, Incremental and Multi-Layer Approach , 14th IEEE Inte. Symp. on High Assurance Systems Engineering, Oct2012

Thanks !

Any questions?