An Integrated Framework for Multi-layer Certification-based Assurance

ACSAC 2014 - Layered Assurance Workshop
New Orleans, Louisiana, USA. December 2014

Authors:

Rajesh Harjani, Marcos Arjona, Javier Espinar, Antonio Maña (speaker), Antonio Muñoz, Hristo Koshutanski

University of Malaga, Spain

Motivation

- **Current trends in computing share some characteristics that complicate the assessment and assurance of their non-functional properties (notably security and privacy)**

- ***Dynamism***
  - Adaptation
  - Evolution
- ***Composition***
  - Vertical & horizontal
  - Lack of a common owner/controller
  - Independent evolution cycles
- ***Complexity***
  - *Size*
  - *Connectivity*
  - *Context-awareness*

Approach

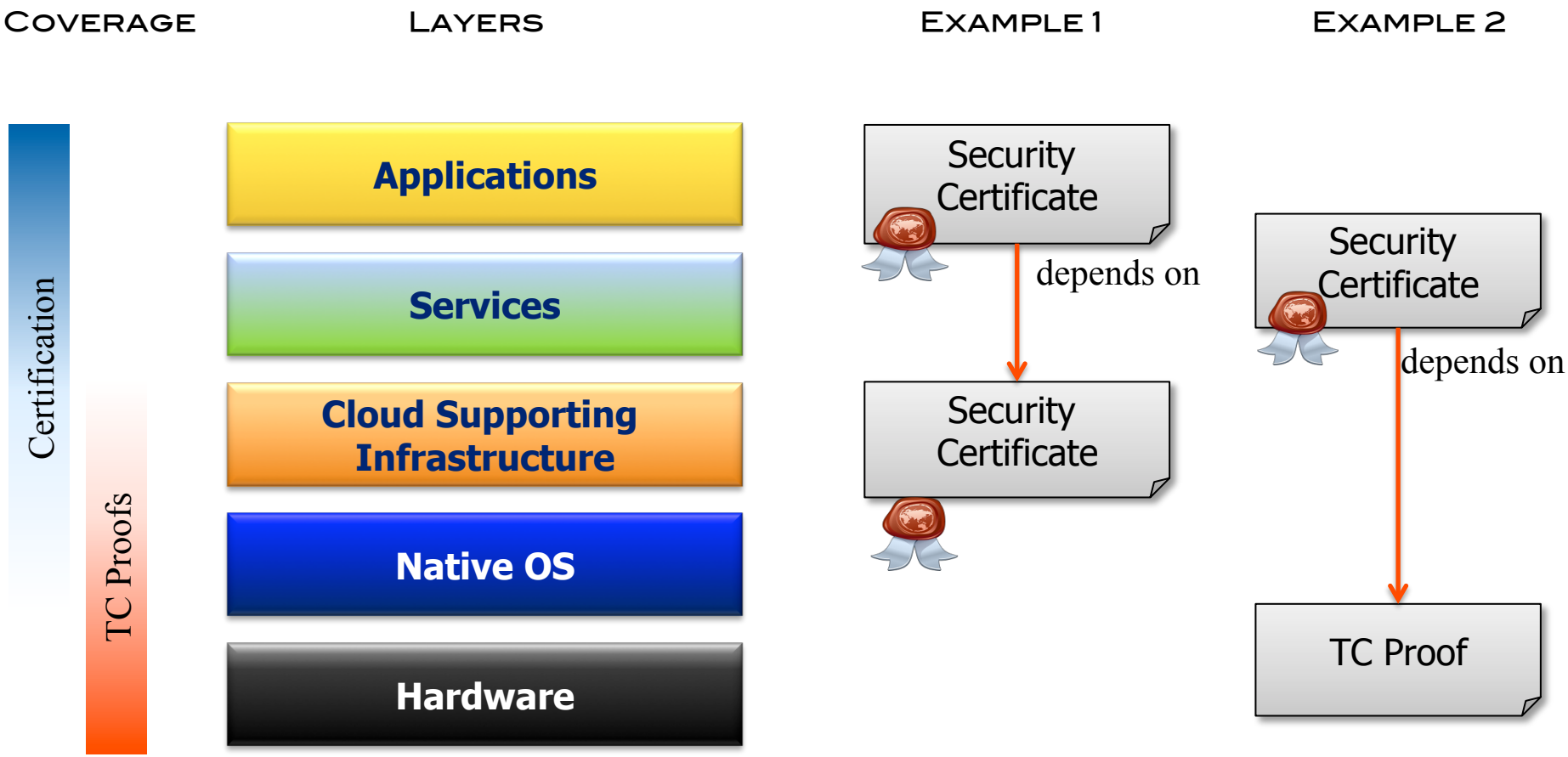
- **We believe that supporting assurance in these types of systems requires at least mechanisms:**
  - (i) to provide **static assurance** for the system components;
  - (ii) to **attest the dynamic state** of system components (including their supporting hardware infrastructure); and
  - (iii) to **derive properties** of the composite system based on properties offered by components and their state.

  In addition to these, we also believe that any successful approach must be complemented with
  - (iv) **engineering** processes, methods and tools to support developers of such systems to take full advantage of the approach.

Approach

- Recent extensions and improvements to several existing technologies like certification, trusted computing and hardware-based security, dynamic testing and monitoring provide a solid basis to develop an integrated framework to support the assurance of complex, composite, layered and evolving systems **in practice.**

Layered certification approach



Security Certification As We Know It

- The concept of security certification provides an appropriate and robust mechanism for supporting assurance and compliance…

  but there are important problems:

  P1 certification has traditionally been represented **for humans** and has not been able to support automated processing (i.e. certification verification, selection based on certificates, etc.); and

  P2 certification can not provide **dynamic proofs of the status of a system at runtime**, which are extremely important in a dynamic, heterogeneous and unpredictable scenario such as cloud computing

  P3 existing applications are not prepared to take advantage of such machine-oriented assurance certificates

New Life for Security Certification

- We have developed an **integrated framework** that solves some of the aforementioned problems and provides:
  - A machine-understandable format for **assurance certificates**
  - 3 different mechanisms to provide **both static and dynamic proofs** of the status of the system based on:
    - Testing
    - Monitoring
    - Trusted Computing
  - Plus one based on formal proofs that we had developed in a previous project
  - A tool-supported **process** and to help developers to create
    - applications that can take advantage of assurance certificates; and
    - components that are "certification friendly" (i.e. are easy to evaluate and certify)

The CUMULUS Framework

- The CUMULUS **integrated framework** includes:
  - **Models** for certification-based assurance of security properties based on evidences drawn from different analyses
  - Dynamic **testing** techniques and tools
  - An assurance-oriented **monitoring** language and monitors
  - An interoperable **certification infrastructure** for generating, maintaining and using certificates according to the different types of the certification models developed, so that to make them available to cloud providers and cloud customers
    - Additionally, we have also developed a secure solution to automatically produce certificates based on runtime evidences
  - Development of an **engineering process and toolset** supporting the development of both components and applications

Our Engineering Approach

- **Model-based methodology**
- **Focused on supporting system engineers and certifiers**
  - Management of security aspects.
  - Making design decisions for finding the best solution for their systems
  - Provide solutions for security requirements
  - Provide communication artefacts
- **Separation of responsibilities based on real-world roles and functions**
- **Flexible modelling framework and tool**
  - Fits into existing development process

Keys of the Engineering Approach

- **Necessary knowledge**
  - The pencil analogy
  - **KEY 1:** Separate roles
- **Taming complexity**
  - Try doing this with just pen and paper
  - **KEY 2:** Provide advanced tool-support
- **At the end of the day, we must develop a system**
  - Only win-win approaches can succeed
  - **KEY 3:** Make security a part of system engineering
- **Necessary flexibility/specialization/reuse**
  - Security depends on precision and details
  - **KEY 4:** Capture security knowledge in different computer-processable artefacts

Our focus

- **Engineering and developing CUMULUS-aware systems (services and applications) requires specific and novel considerations to be taken into account**
  - Trust is established by means of certificates
  - Systems will be composed not only based on functional aspects, but also on **security aspects (properties, threats, risks,…)**
  - Systems should be designed to use certified components
  - Components should be designed for certification
  - Components of a system may change without the knowledge or control of other components

The CUMULUS Engineering Process

- The CUMULUS **engineering process and toolset** includes:
  - A process based on the principle of separation of responsibilities (KEY 1)
  - Tools (KEY 2) to
    - Create, store and publish models and artefacts
    - Interact with the CUMULUS infrastructure
    - Assist system engineers during system design (KEY 3)
      - Assisted integration of elements (requirements, solutions, certified components,…)
      - Full traceability to support evaluation and certification
      - On-the-fly validation of designs
  - A modeling framework including different layers and artefacts (KEY 4)

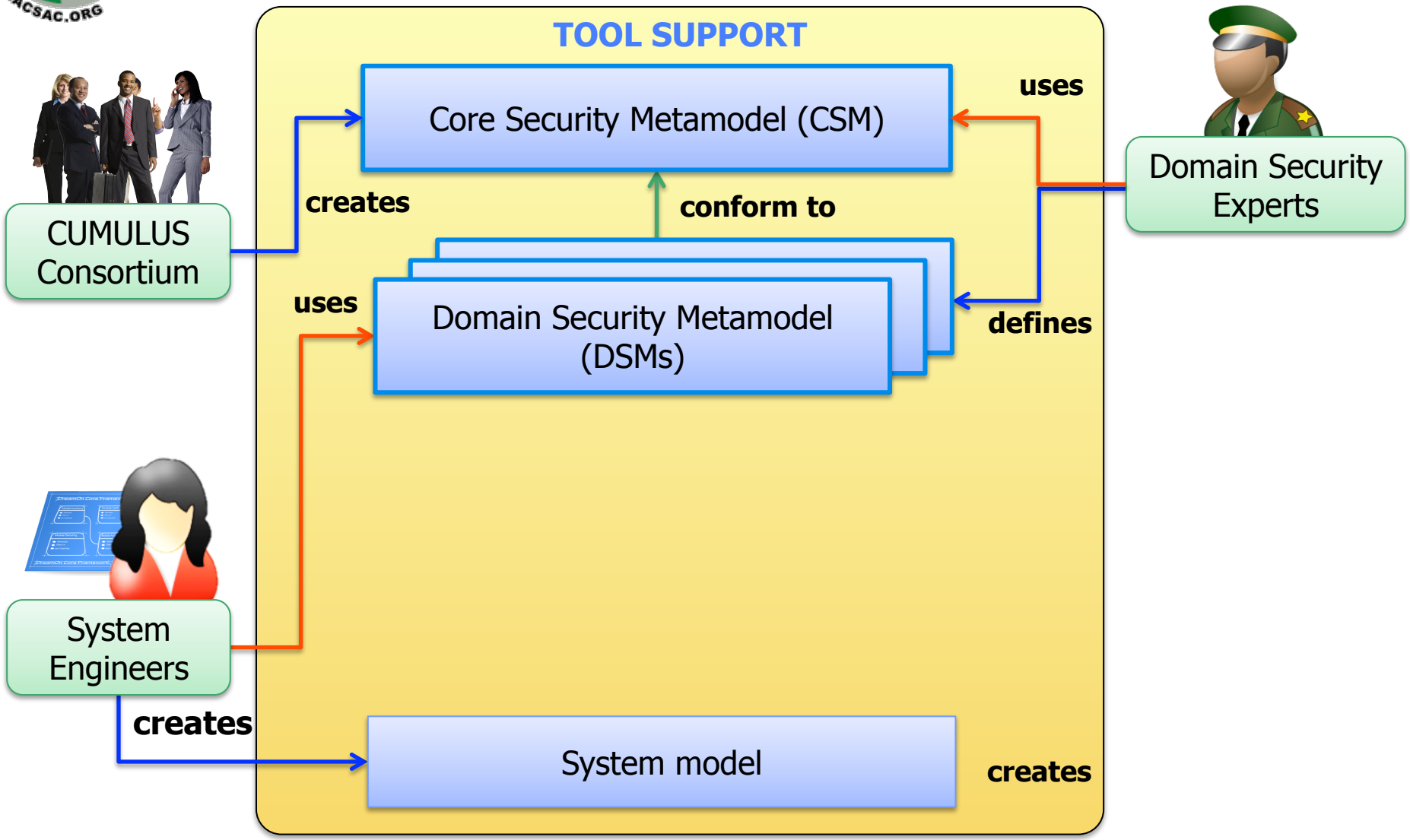
CUMULUS Engineering Process (Main artefacts)

- **CSM (Core Security Metamodel): The language**
  - contains the definition of core elements (represented in UML) and their relations to represent the most relevant security concepts, such as of security properties, security requirements, threats, attacks, certification, etc.
  - Work in progress but already usable
  - We have focused on requirements, properties and solution modeling (through patterns)
  - But also cover threat, attack and attacker modeling, certification, verification and validation, risks…
    - We still need contributions in those aspects

CUMULUS Engineering Process (Main artefacts)

- **DSM (Domain Security Metamodel): The knowledge**
  - allow experts to capture specific security knowledge, including the definition of additional rules (company policies, standards, etc.) for a particular domain
  - Include domain "asset stereotypes" and "domain security requirements" to facilitate requirement elicitation, coherence and management
  - Include specific requirements, properties, solutions, attacks, certification schemes, …
  - Include domain rules to ensure coherence among the mentioned elements

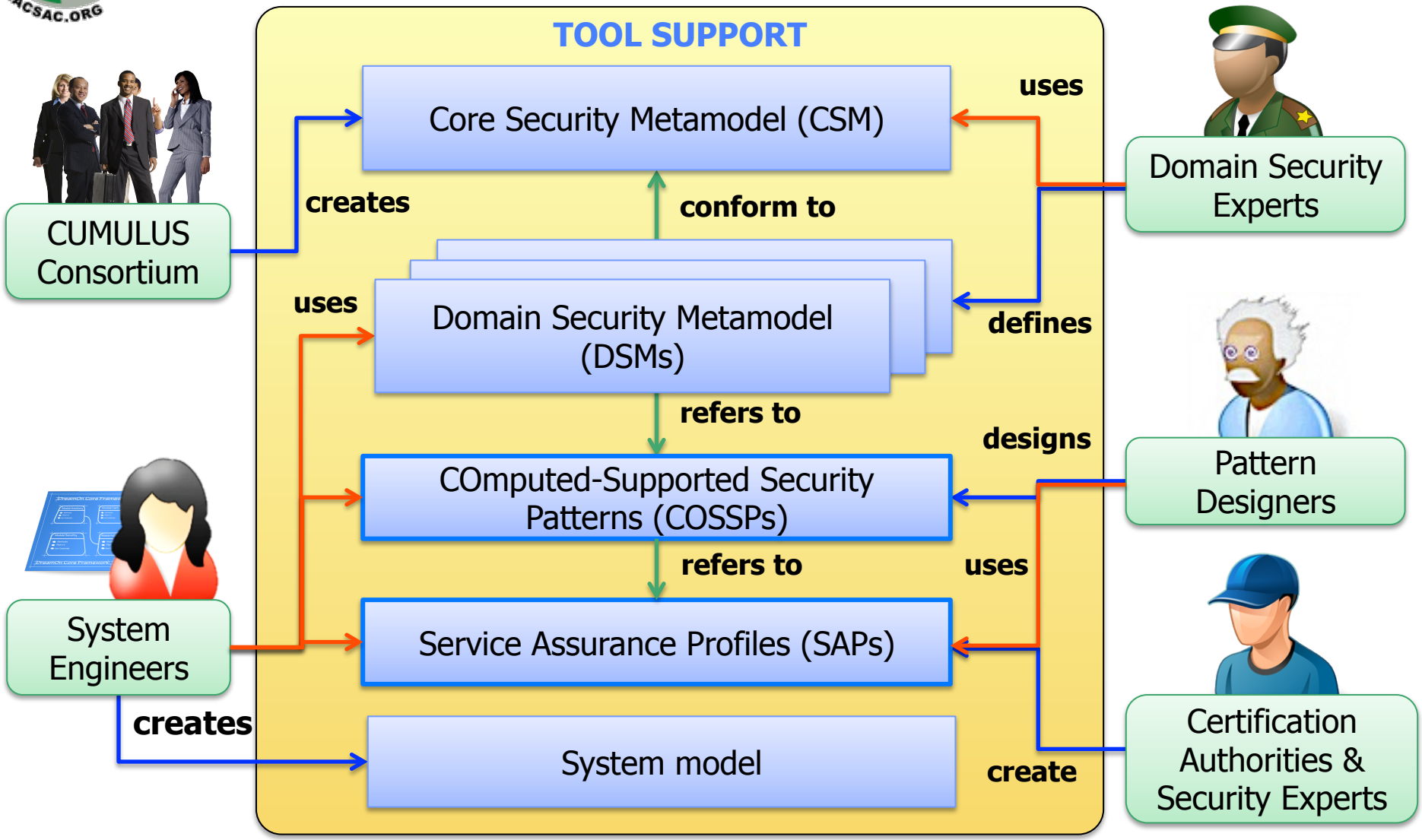
Layered Modeling Framework: Artifacts



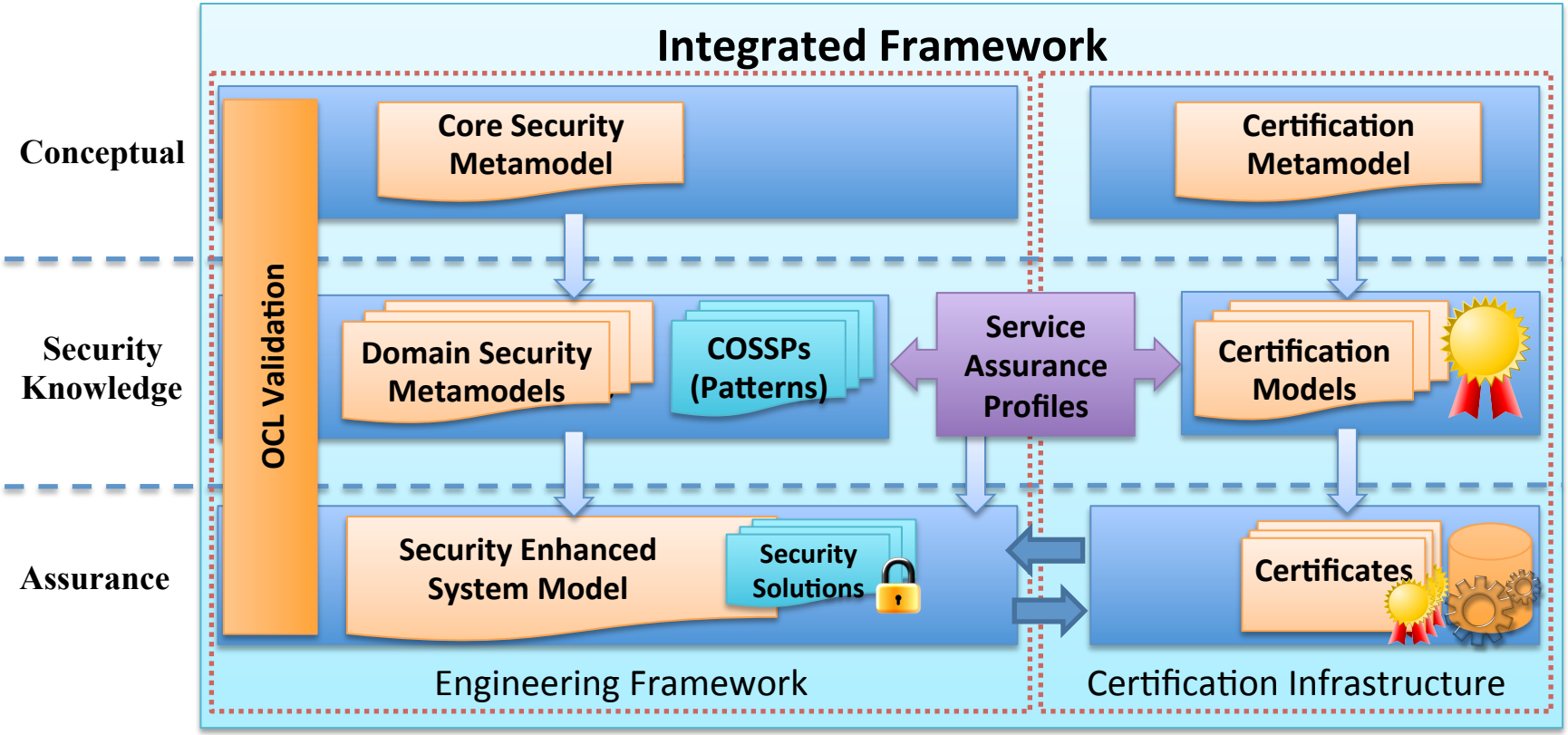
CUMULUS Engineering Process (Main artifacts)

- **COSSP (COmputer-Supported Security Pattern): The solutions**
  - contains precise, machine-readable, integration-oriented, descriptions of validated solutions to recurrent security problems
- **SAP (Security Assurance Profiles): The certification requirements**
  - allows expressing which certificates are acceptable for the system in order to guarantee that an external component fulfills a given security requirement

Layered Modeling Framework: Artifacts



Model relations with the CUMULUS Fw



Engineering Tool Relevance

- The **CUMULUS Engineering Tool (CET), based on the relevant artifacts,** provides rich support for Cloud System Developers to make **informed security design decisions** in the **application development** process.
  - *Applications are engineered to use certified cloud services*
- CET supports the **development of certification-friendly cloud services**.
  - *Services are engineered to facilitate certification*
- The **certification-centric approach** of CET addresses the current concerns regarding assurance, security, privacy, governance and compliance of the data and services offered through the cloud.
  - *This approach –based on the certification of security properties– increases trust in cloud services and their providers, and reduces uncertainty of cloud customers regarding the security of their services once they are deployed in the cloud*.

What CET can do now

- Import appropriate **DSMs** for our SUD
- Apply **asset stereotypes** to our model elements
- Automatically derive **requirements** based on the asset stereotypes
- Automatically attach relevant **security knowledge** to the security-sensitive elements
- Automatically on-the-fly check **coherence rules**
- Assist engineers in selecting the right **solution** for their requirements
- Assist engineers in integrating the selected **solutions** (in the form of **COOSPs**) in their SUDs
- Transform security requirements into **certification requirements** (via **SAPs**)
- Select cloud services based on the selected SAPs (via the CUMULUS infrastructure)
- Automatically on-the-fly check **solution constraints**

What we are working on

- Offer personalized **views** of the SUD
- Develop a trust infrastructure for **Security Knowledge**
- Automatically on-the-fly check of **advanced coherence rules**
- Automatic documentation generation including
  - Traceability
  - Security design decisions
- Assist engineers in implementing the selected **solutions** (in the form of **COOSPs**) in their SUDs
- Supporting system re-engineering for legacy systems

Like to see it working?

- Very brief demo for a a cloud-supported e-health system

Thanks for your attention! Questions?

ABOUT CUMULUS

- **CUMULUS:** **C**ertification infrastr**U**cture for **MU**lti-**L**ayer clo**U**d **S**ervices

  - EU-Funded Research Project
  - 8 Partners from across Europe
  - Oct. 2012 – Sept. 2015

http://proteus.lcc.uma.es/cumulus

http://www.cumulus-project.eu