
The Open Group SOA Ontology Technical Standard

Clive Hatton

The Open Group Releases SOA Ontology Standard To Increase SOA Adoption and Success Rates

Ontology Fosters Common Understanding of SOA Concepts to Improve Alignment Between Business and IT

SAN FRANCISCO, December 8, 2010 – The Open Group today announced the availability of the Service Oriented Architecture (SOA) Ontology Technical Standard to develop and foster a common understanding between business and information technology (IT) communities regarding SOA concepts and terminology. Produced by the members of the Open Group's SOA Work Group, the ontology defines the concepts, terms and semantics of SOA in a common language that will allow for more precise and straightforward communications and facilitate SOA adoption without ambiguity.

Content

What is SOA?

Why an
Ontology for
SOA?

Why now?

How does it
relate to other
standards?

Who will use
it?

How will it be
used?

Where does it
apply?

What is an
Ontology?

What does it
describe?

SOA Principles – Thomas Erl

Standardised Service Contracts

- Services within the same service inventory are in compliance with the same contract design standards

Service Loose Coupling

- Service contracts impose low consumer coupling requirements and are themselves decoupled from their surrounding environment

Service Abstraction

- Service contracts only contain essential information and information about services is limited to what is published in service contracts

Service Re-usability

- Services contain and express agnostic logic and can be positioned as reusable enterprise resources

Service Autonomy

- Services exercise a high level of control over their underlying runtime execution environment

Service Statelessness

- Services minimize resource consumption by deferring the management of state information when necessary

Service Discoverability

- Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted

Service Composability

- Services are effective composition participants, regardless of the size and complexity of the composition

Key Features of SOA

Services
Service re-use
Service composition
Service discovery

Messaging
Message monitoring
Message control
Message transformation
Message security

Events
Complex Event Processing

Asset wrapping

Virtualisation

Model-driven implementation

Why an Ontology for SOA?

"A lack of mutually agreed-upon SOA terms, definitions and concepts can create interoperability issues that inhibit end-to-end business activities within an organization - as well as between vendor, customer, and partner organizations.

By providing common terminology and concept mapping that business and technical people may employ to discuss problems and opportunities, the ontology bridges different architecture, engineering, business and marketing domains.

It also creates a foundation for further work in domain-specific areas by supplying a consistent framework that can be reused and revised as SOA projects evolve."

Why Now?

The ontology is the result of years of implementation work and lessons learned.

"It is grounded in extensive real-world experience developing, deploying and communicating about SOA solutions over the past five years. The Ontology reflects the lessons learned about what terms NOT to use to avoid confusion, and how to best distinguish among some common and often overused concepts like service composition, process, service contracts, and policy and their roles in SOA."

SOA Standards Landscape

OASIS

Advancing open standards for the information society

OMG

WE SET THE STANDARD

THE *Open* GROUP

Making standards work®

WHITEPAPER

Explain & position standards for

- *SOA reference models;*
- *ontologies;*
- *maturity models;*
- *modelling languages;*
- *standards work on SOA governance*

Outlines
similarities

Highlights
Strengths

Complementary

User Guide

Audiences for SOA standards



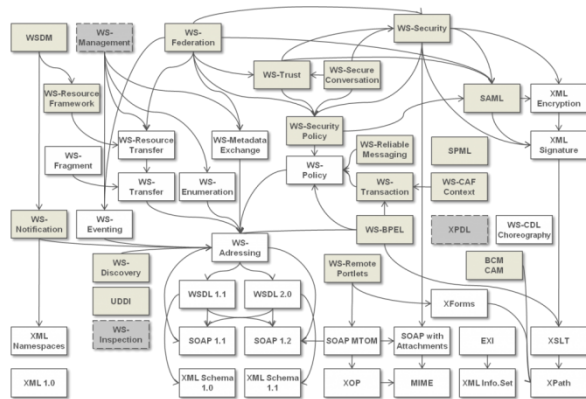
Business Architects & Analysts



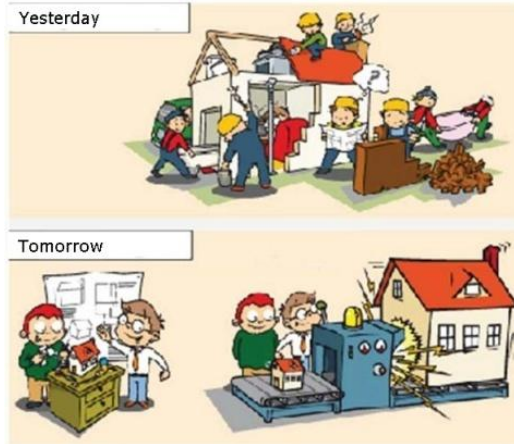
Developers / Practitioners



Customers / SOA Adoptors



Standards Organizations

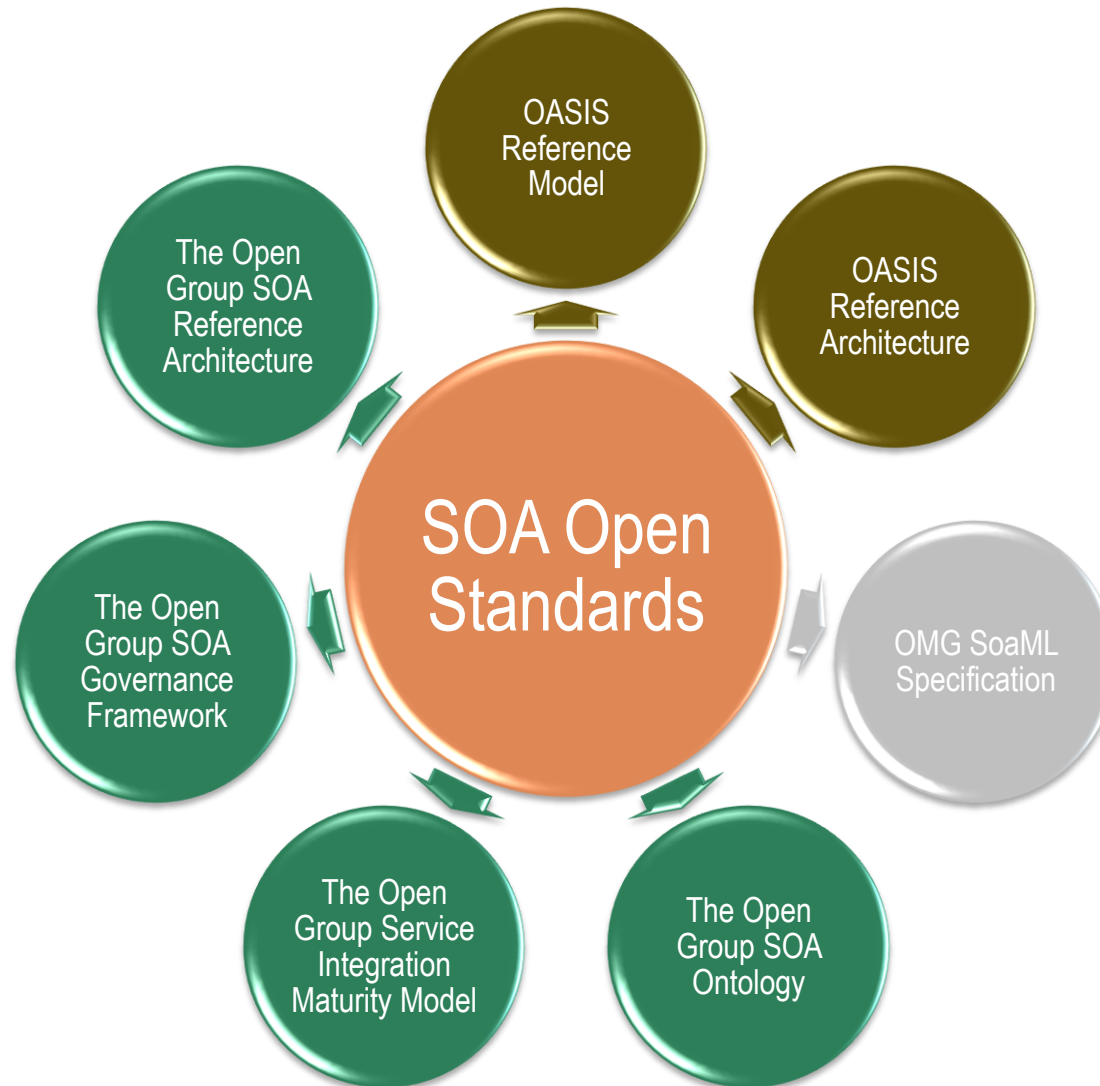


Analysts

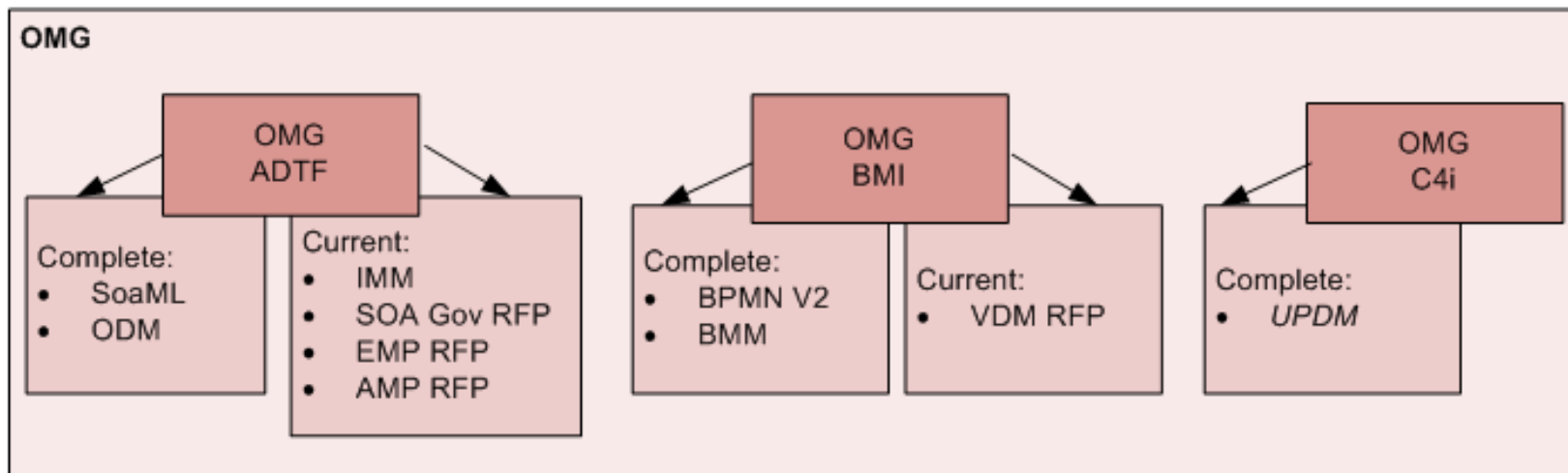
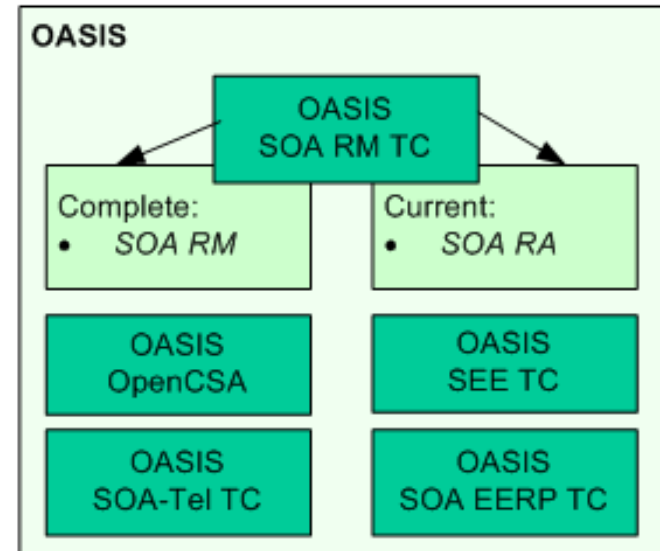
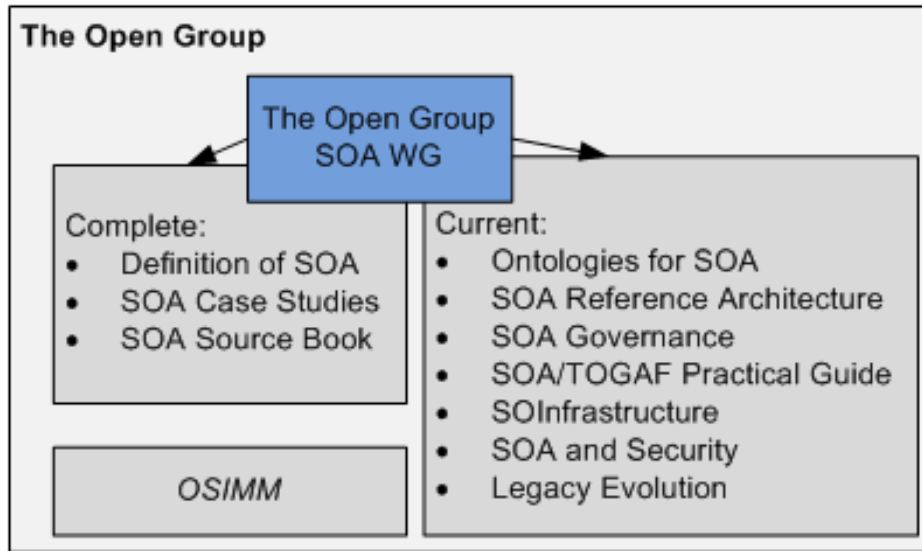


Architects

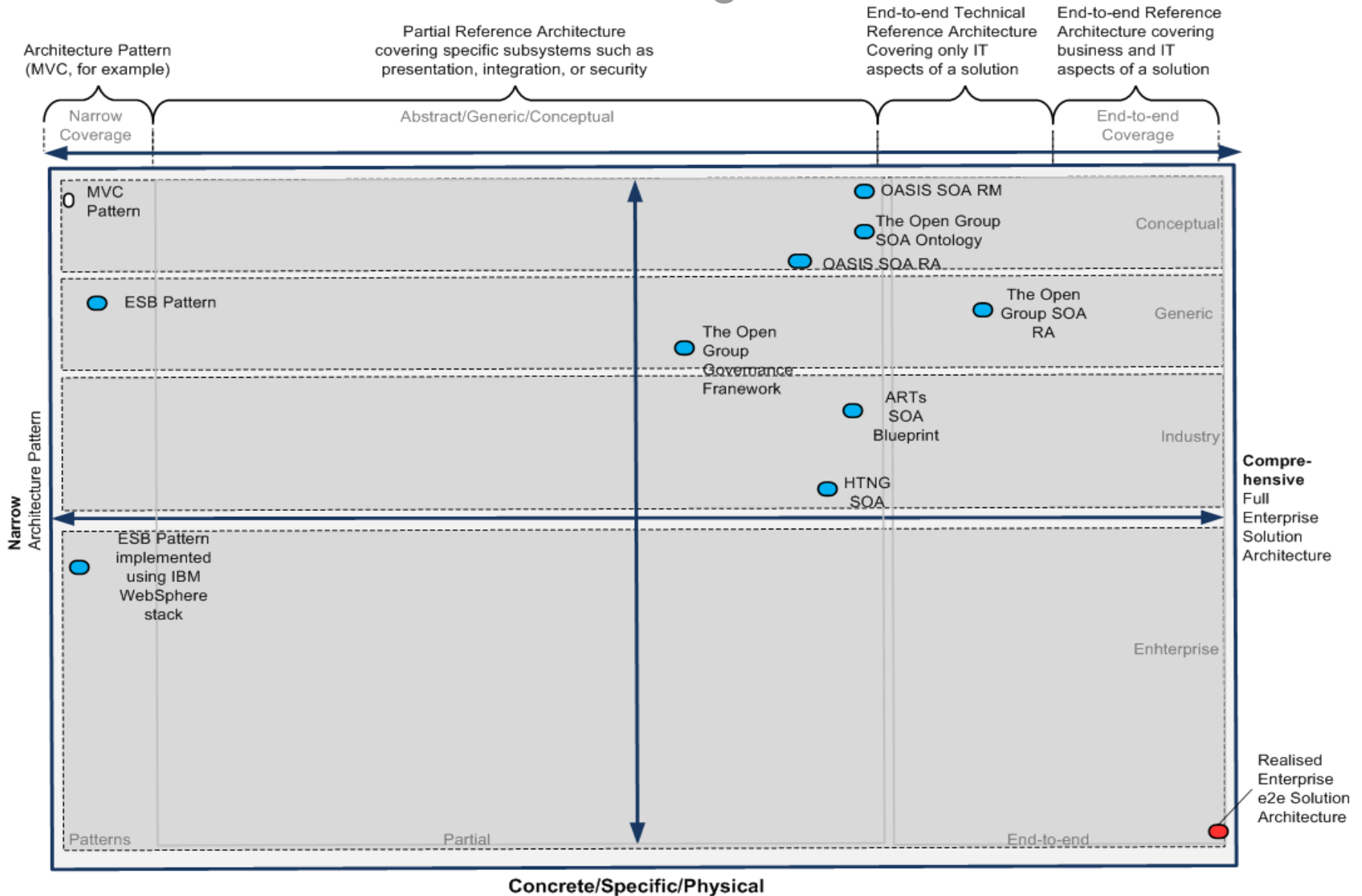
Specifications – White Paper



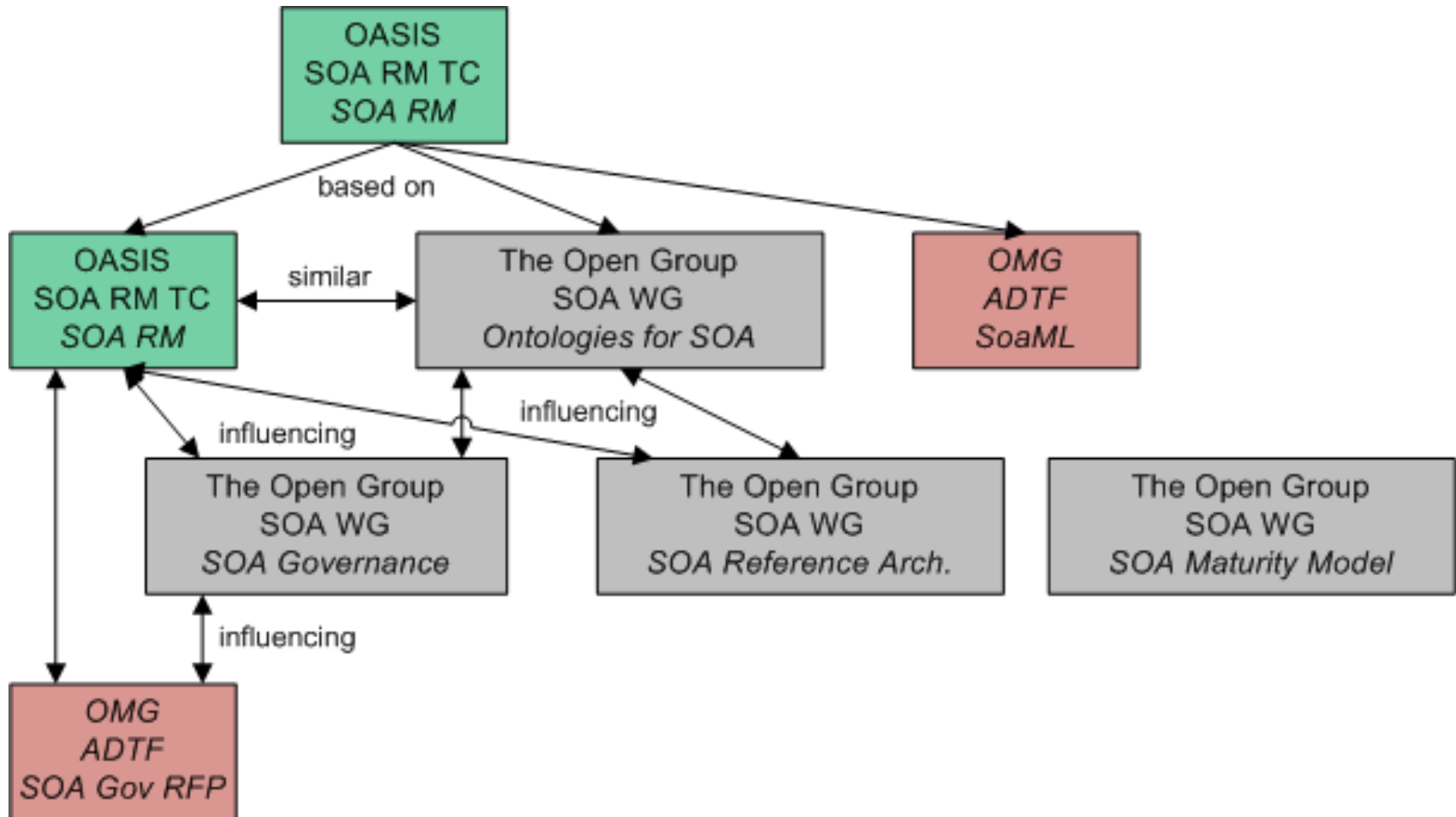
Working Groups



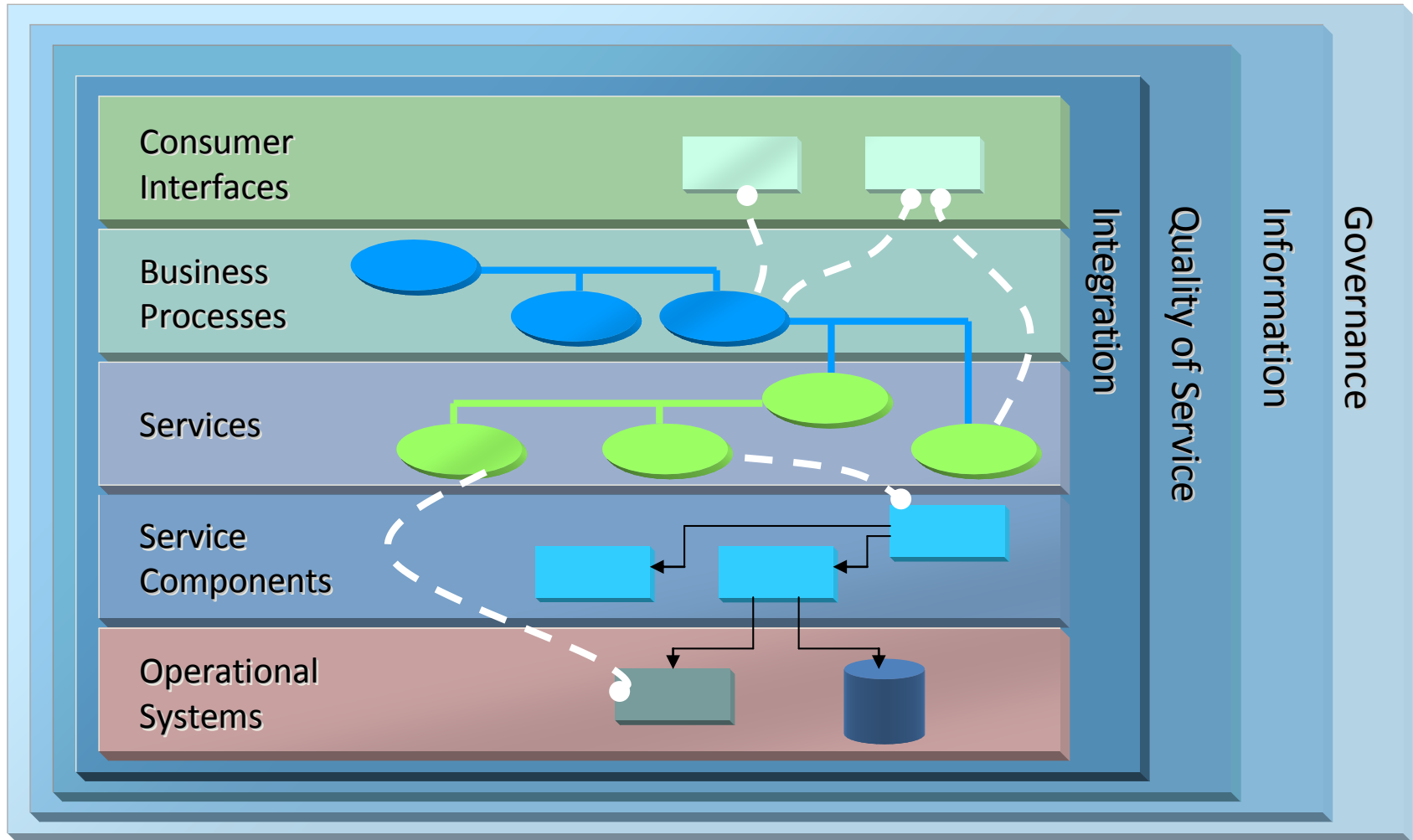
SOA Standards Positioning



Technical Products: Influence



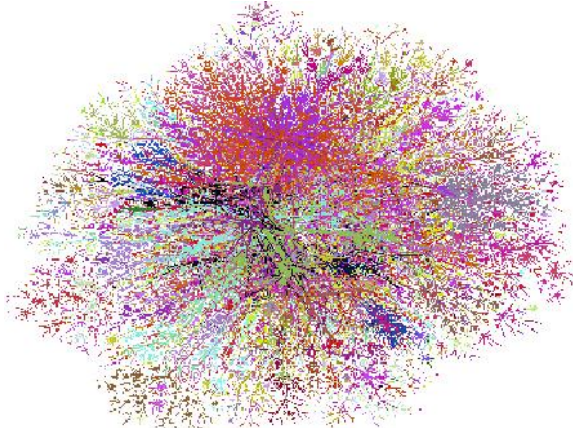
The Open Group SOA Reference Architecture



Ontologies

Gruber
Definition

Ontologies are useful to ensure that information items are defined in a standard and coherent manner, across teams. Ontologies formally describe the elements of and provide a language for both reference models and reference architectures.



In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members).

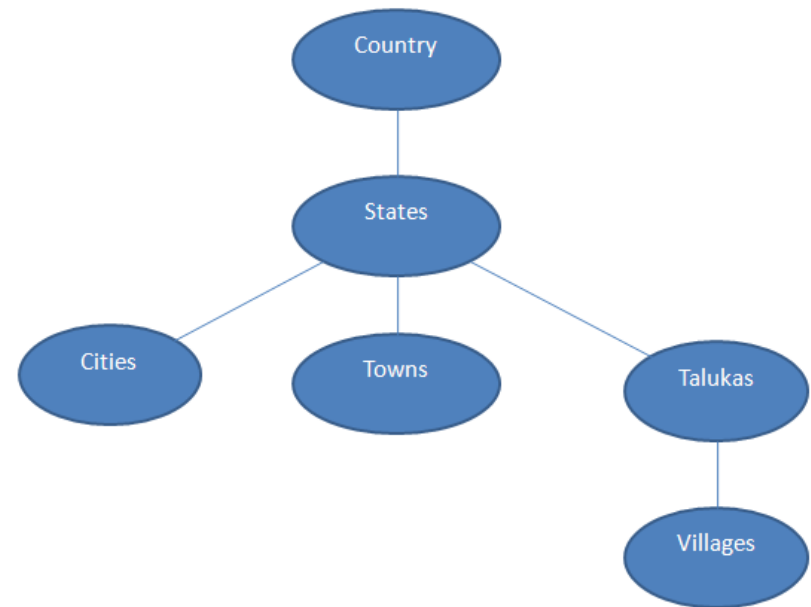
An explicit formal specification of the terms in the domain and relations among them.

Denoting an artefact that is *designed* for a purpose, which is to enable the modelling of knowledge about *some* domain, real or imagined.

Examples of Ontologies

	What	How	Where	Who	When	Why	
Scope							Scope
Business Model							Business Model
System Model							System Model
Technology Model							Technology Model
Detailed Representations							Detailed Representations
Functioning Enterprise							Functioning Enterprise
	What	How	Where	Who	When	Why	

Zachman Framework



Ontology for a geography of a country , showing hierarchy of classes

The Open Group SOA Ontology

Intended to:

- Facilitate understanding of terms & concepts
- Facilitate model-driven implementation

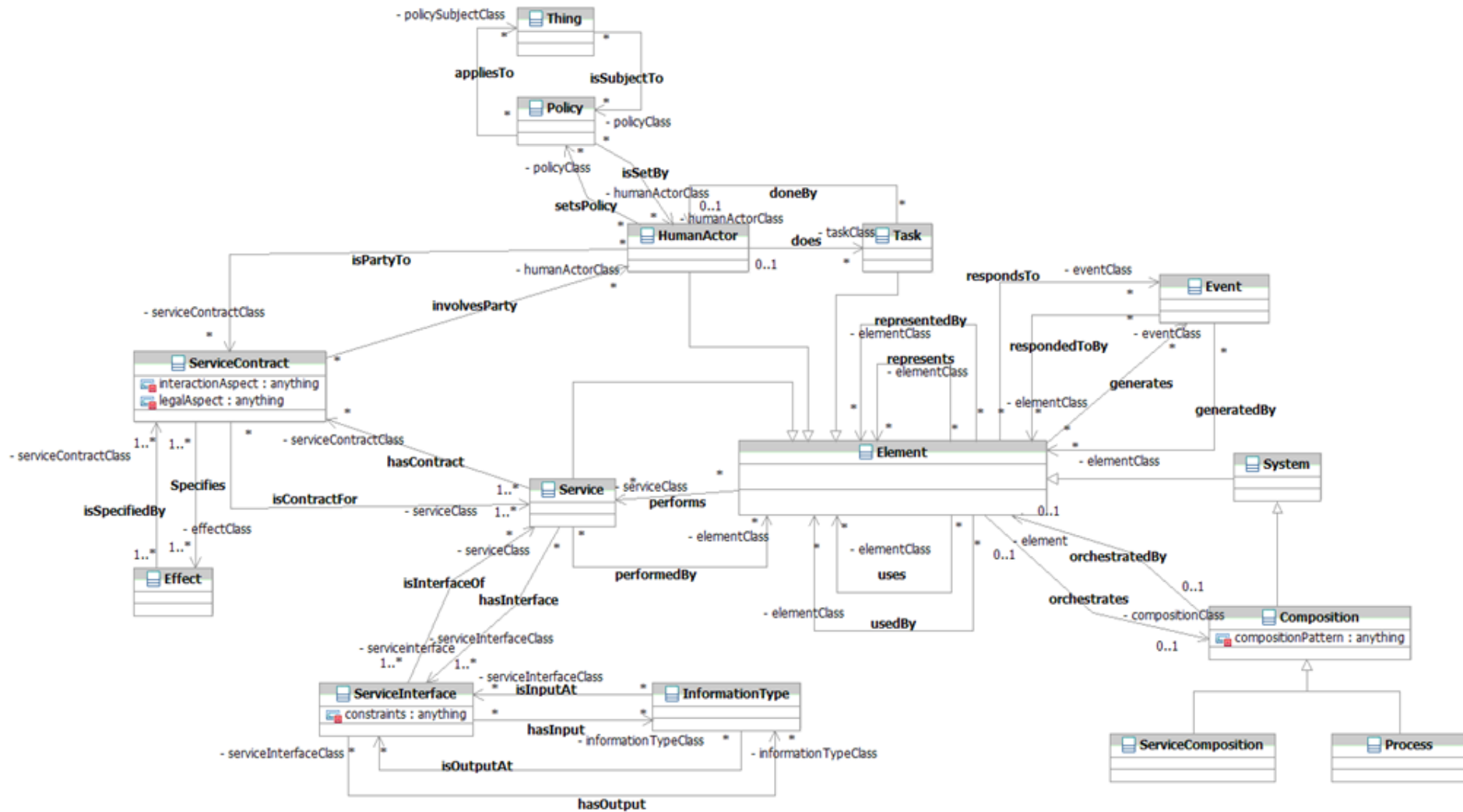
Enables:

- Automation & tools for processing
- Integration with other concerns

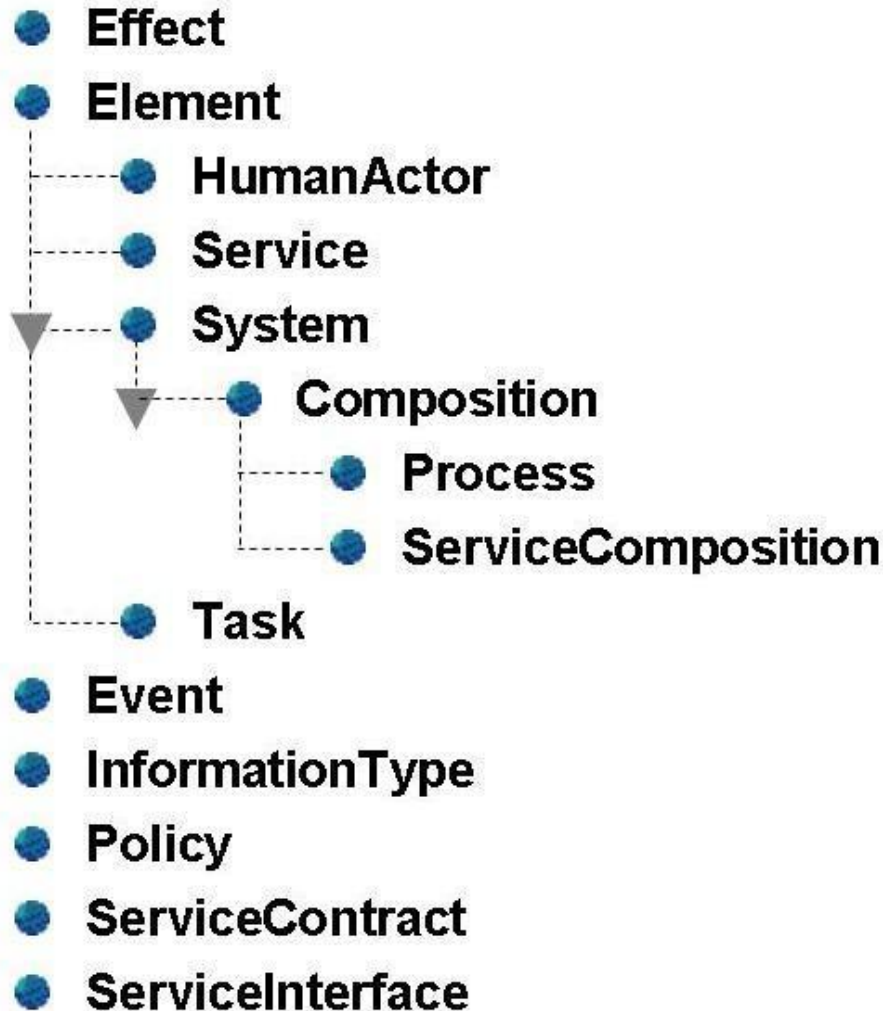
Represented by OWL



SOA Ontology – Graphical Overview



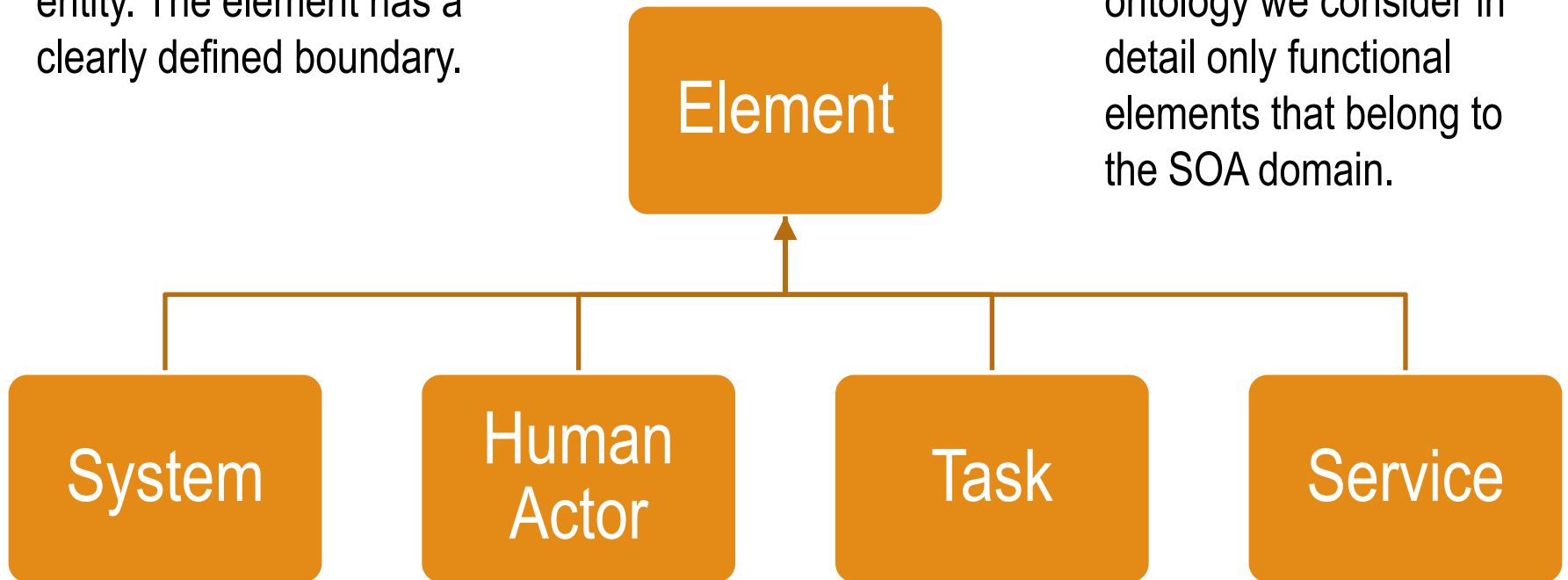
SOA Ontology Class Hierarchy



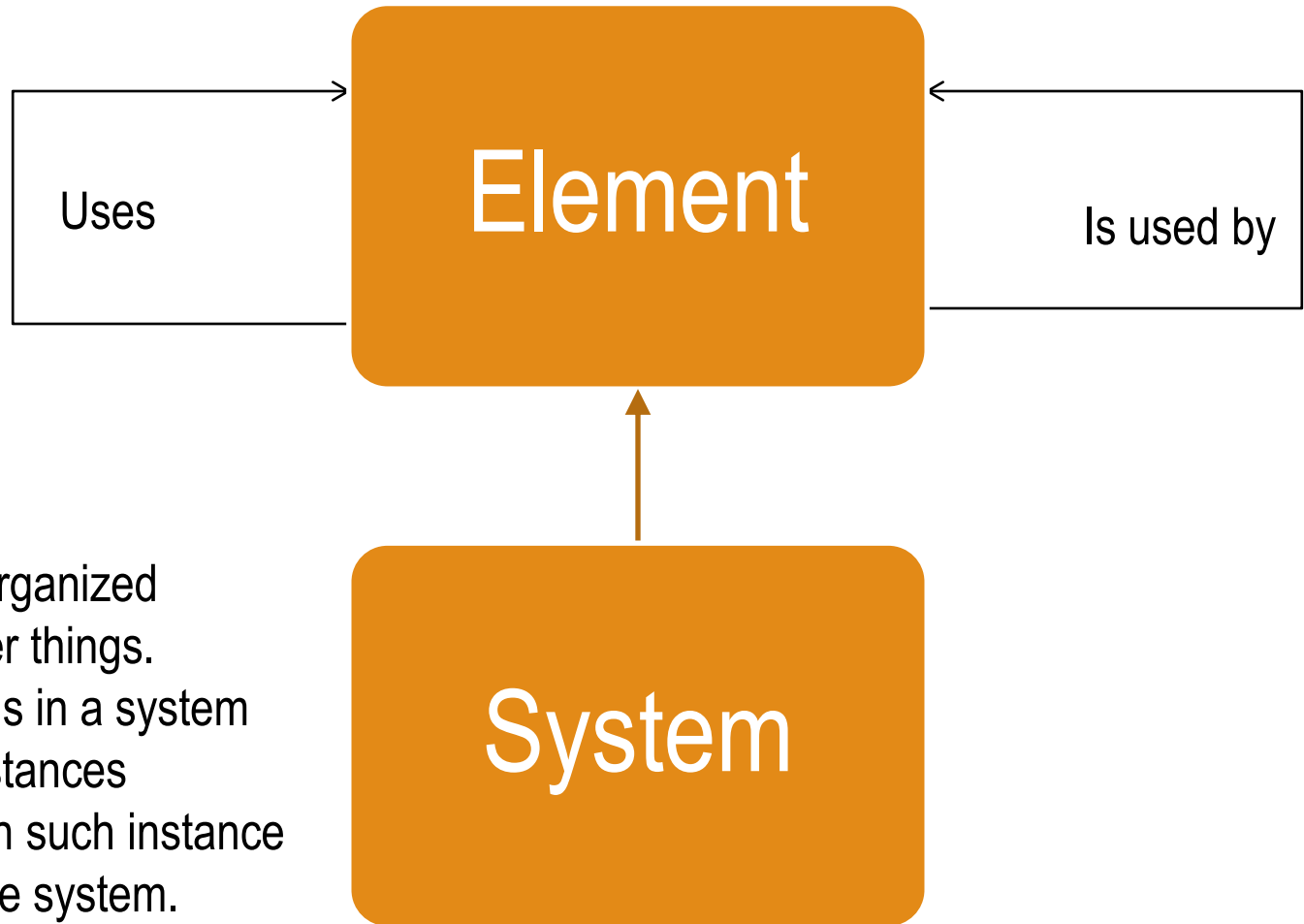
SOA Ontology – Element

An element is an opaque entity. The element has a clearly defined boundary.

In the context of the SOA ontology we consider in detail only functional elements that belong to the SOA domain.

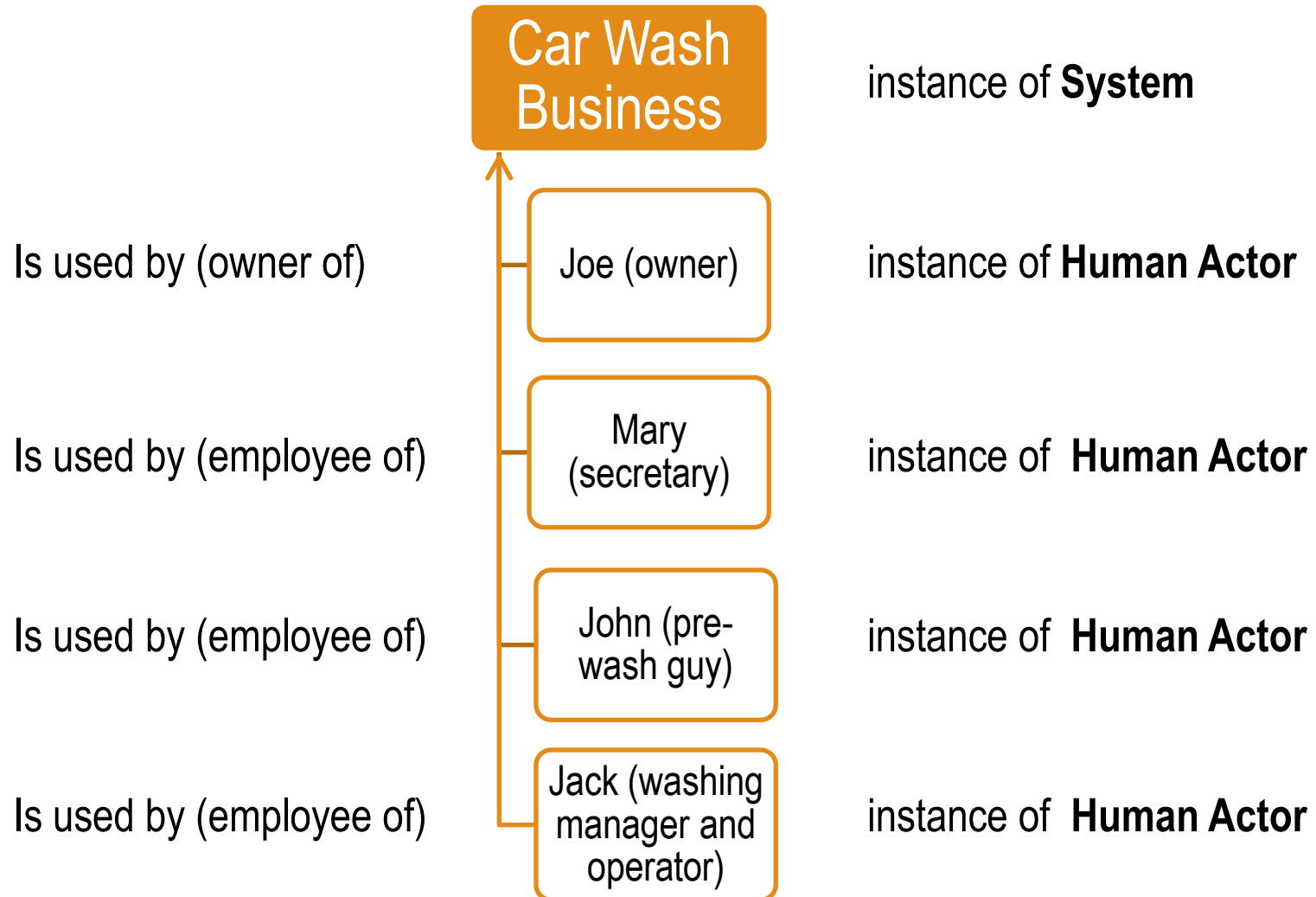


SOA Ontology – System and Element



A *system* is an organized collection of other things. Specifically things in a system collection are instances of **Element**, each such instance being used by the system.

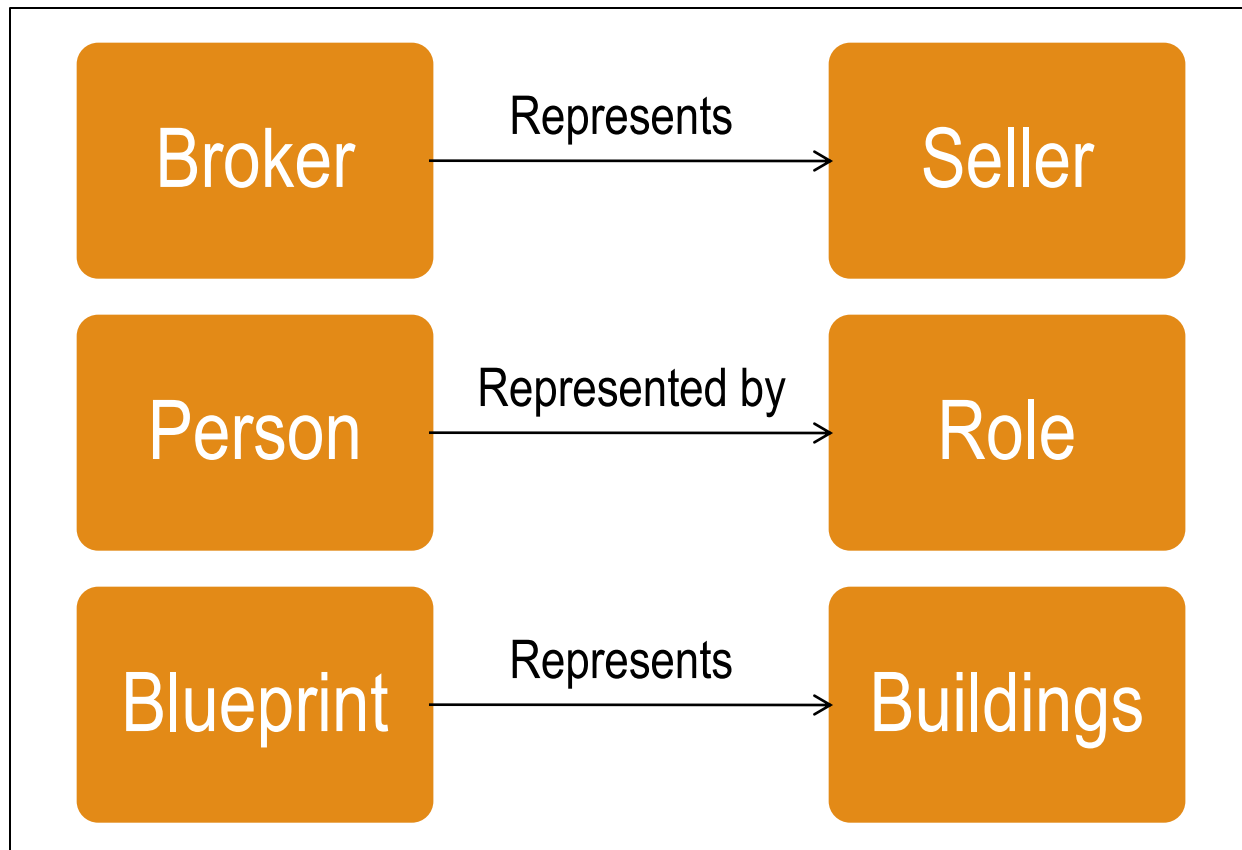
Car Wash Example – System and Element



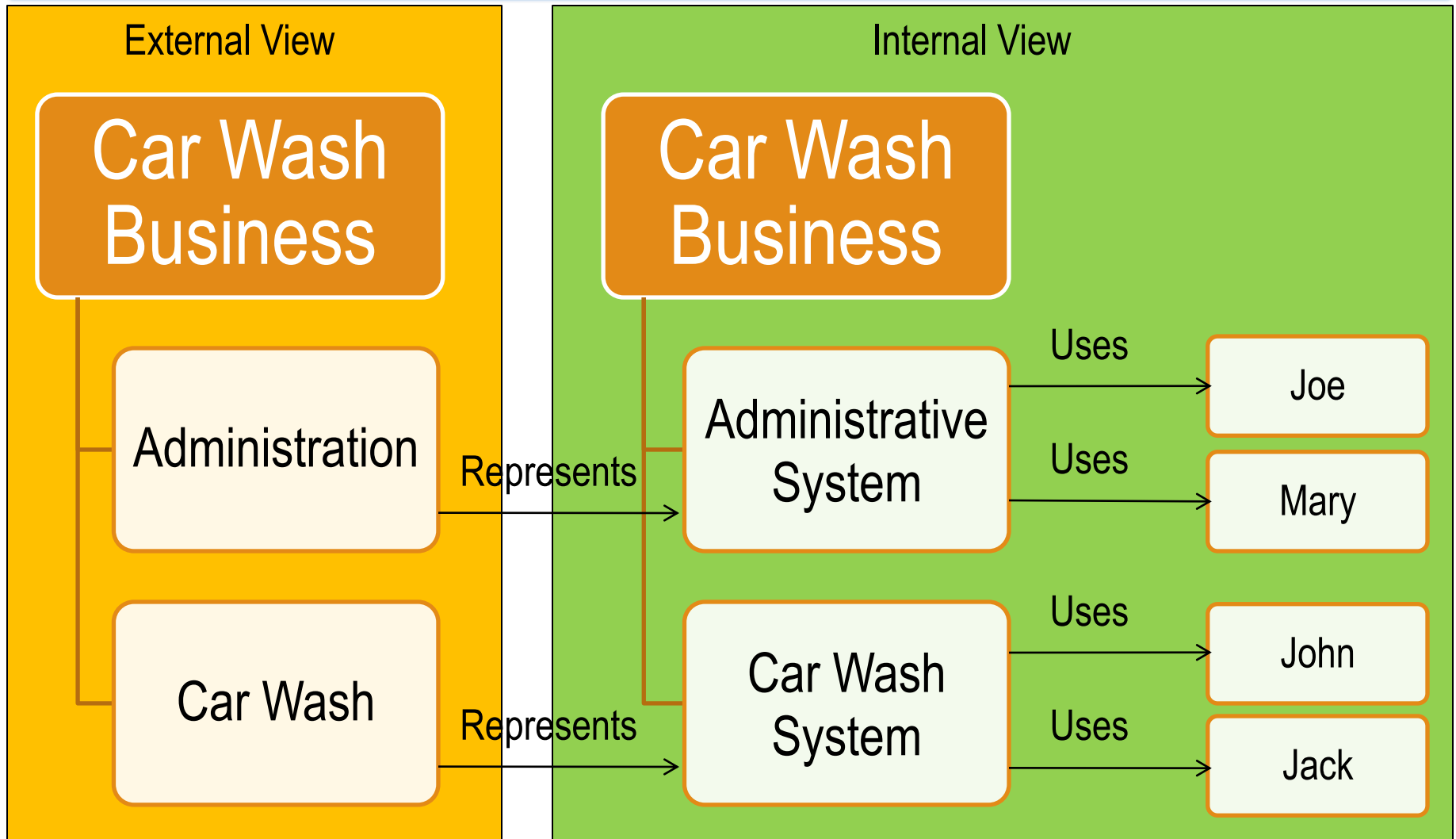
“Represents” Abstraction



Three examples (instances of element)

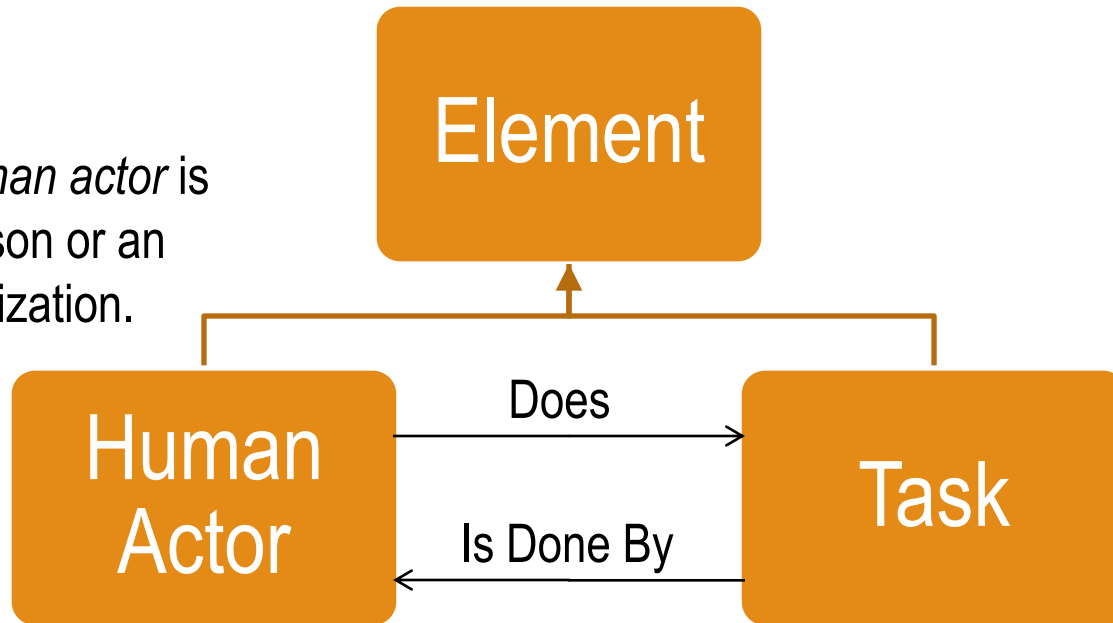


Car Wash Example – External and Internal Views



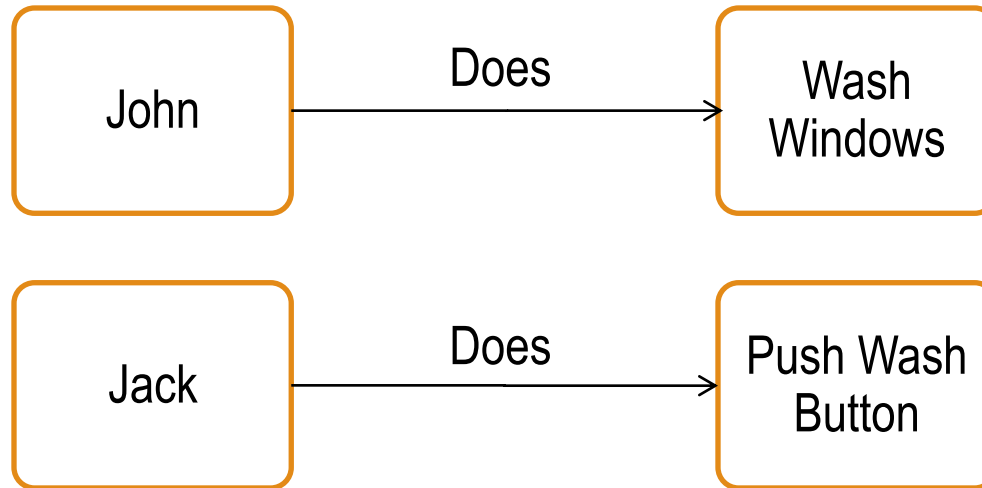
Human Actor and Task

A *human actor* is a person or an organization.

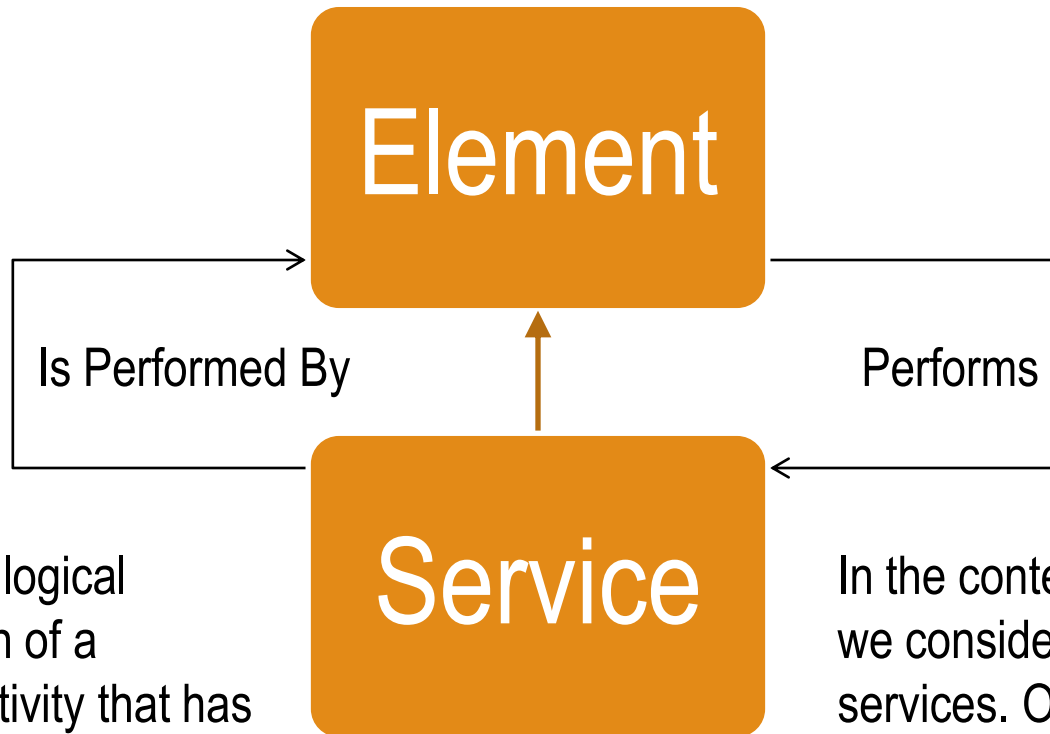


A *task* is an atomic action which accomplishes a defined result. Tasks are done by people or organizations, specifically by instances of **Human Actor**.

Car Wash Example – Human Actor and Task



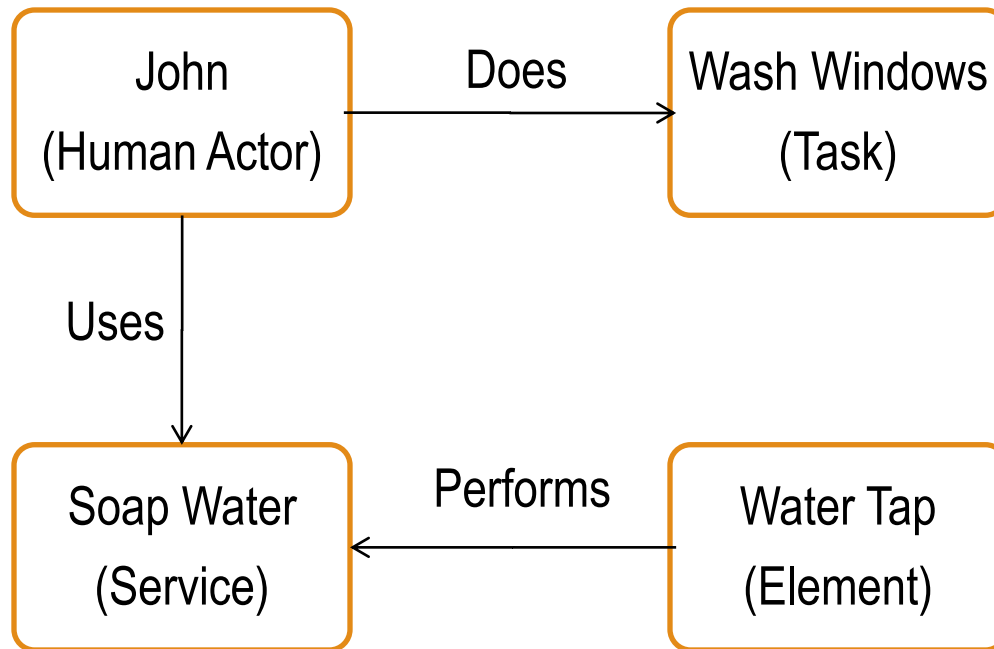
Service



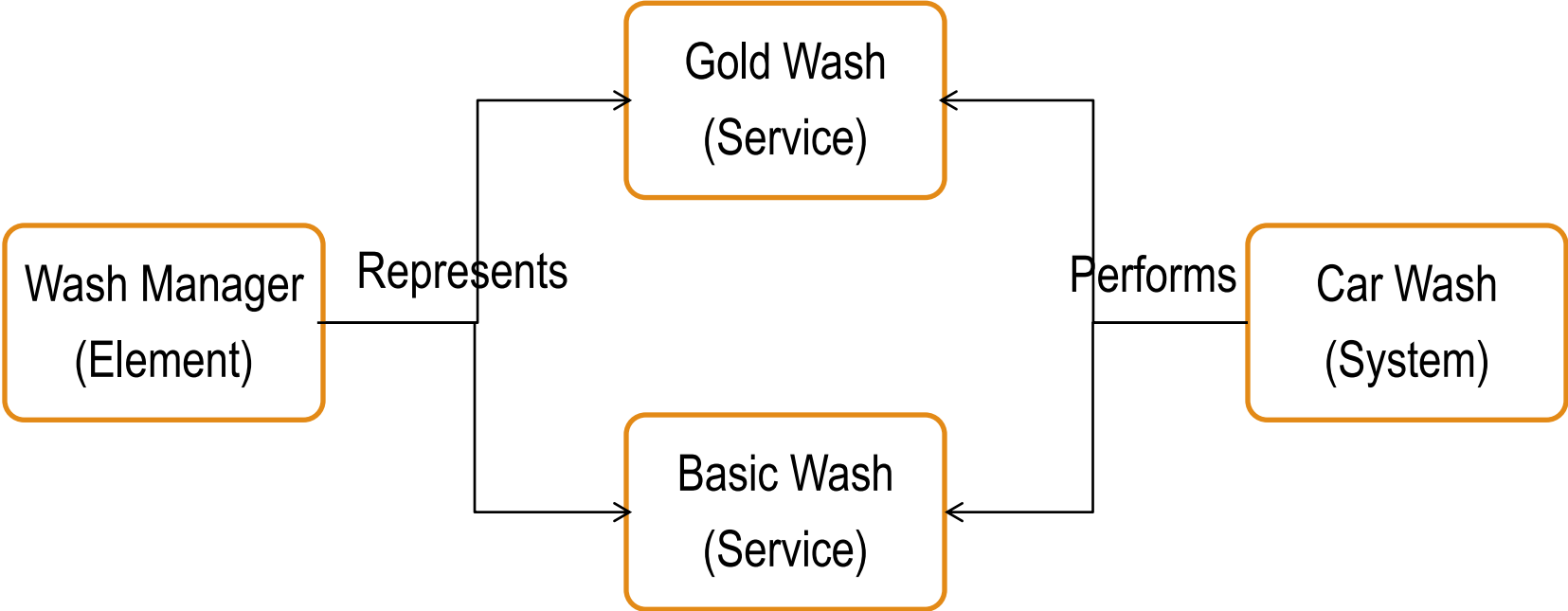
A service is a logical representation of a repeatable activity that has a specified outcome. It is self-contained and is a „black box“ to its consumers.

In the context of the SOA ontology we consider only SOA-based services. Other domains, such as Integrated Service Management, can have services that are not SOA-based and hence are outside the intended scope of the SOA ontology.

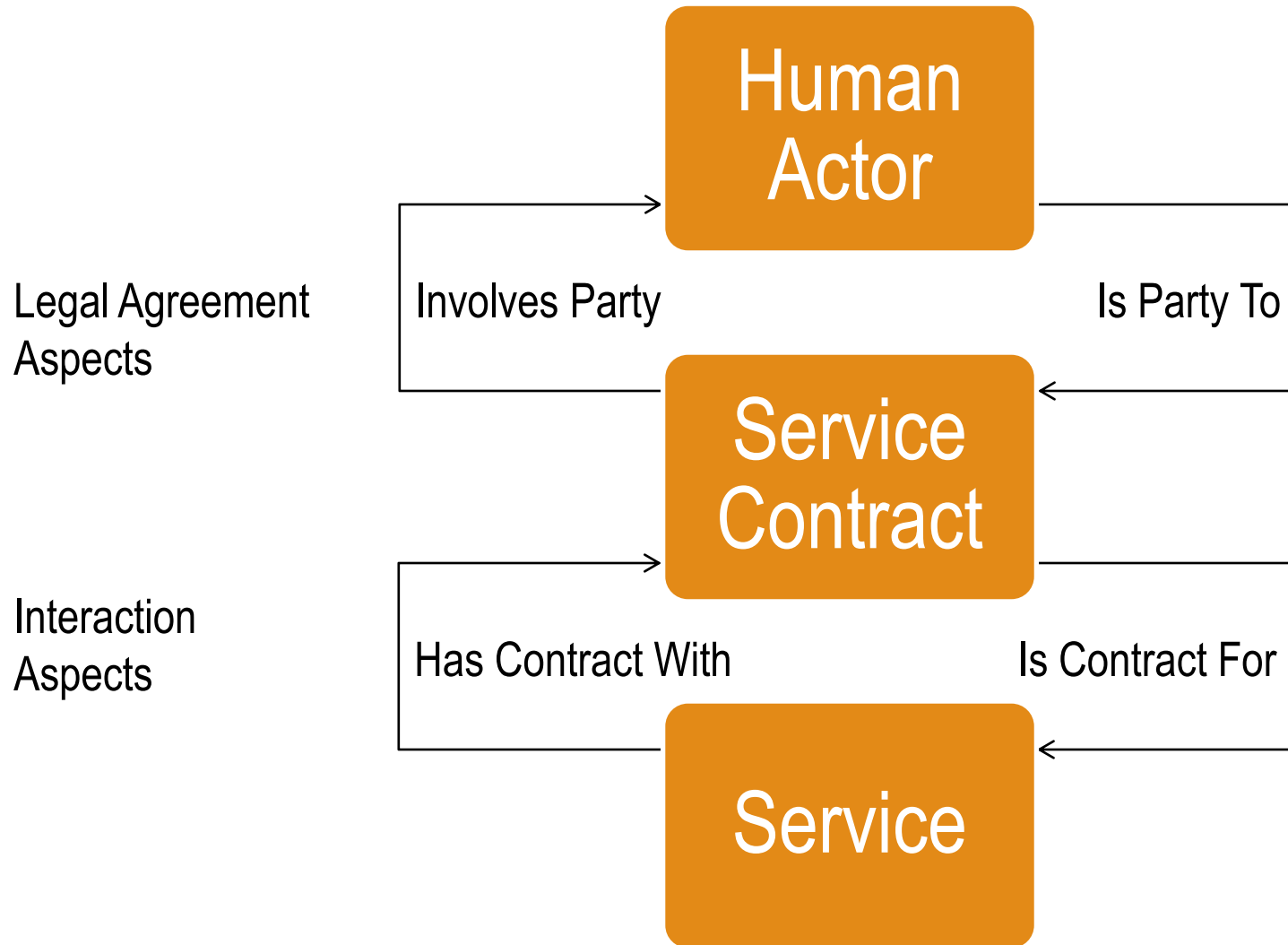
Doing a Task and Performing a Service



Car Wash Example – Represents and Performs



Service Contract and Service Interface

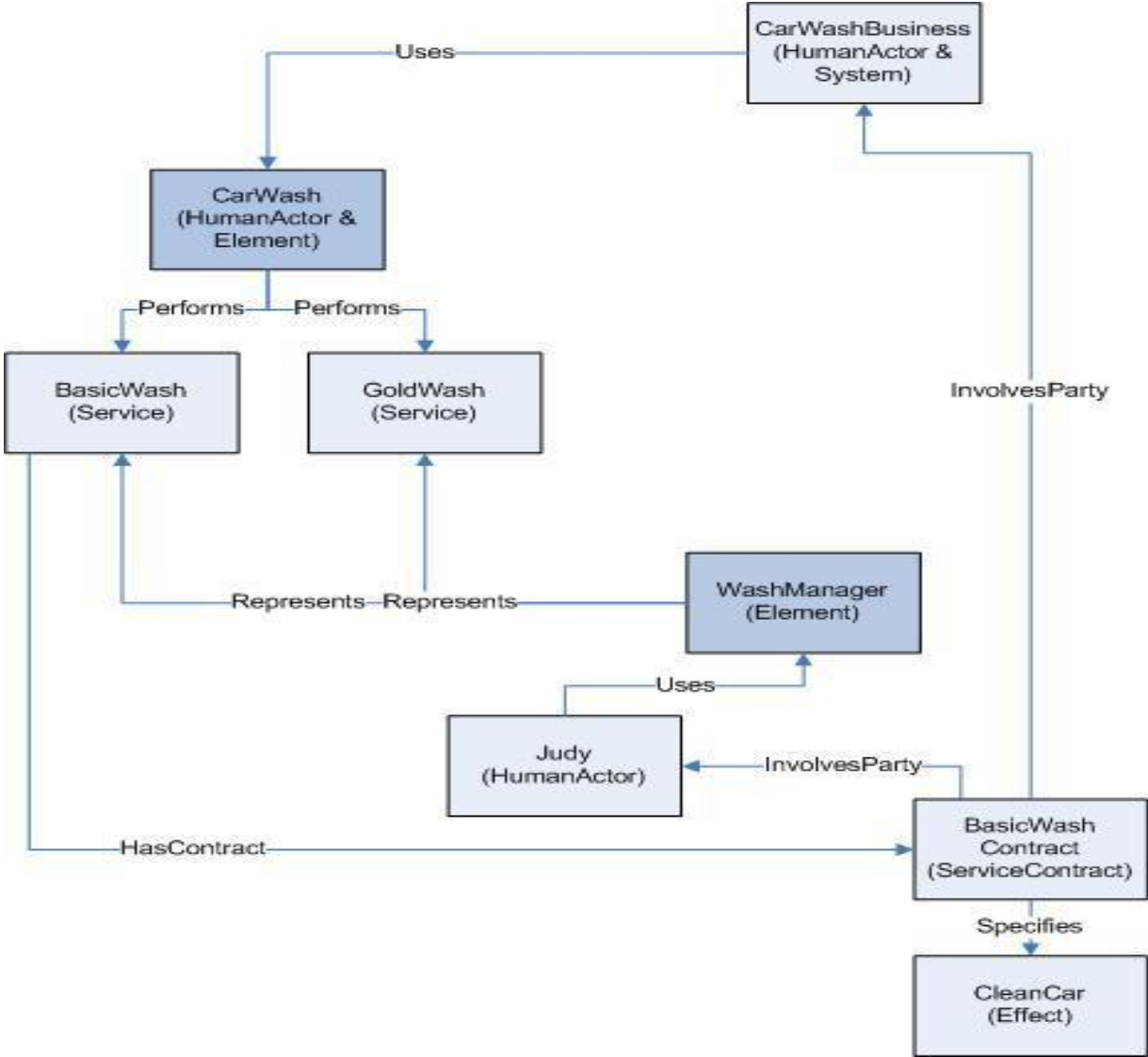


Effect

Interacting with something performing a service has *effects*. These comprise the outcome of that interaction, and are how a service (through the element that performs it) delivers value to its consumers.

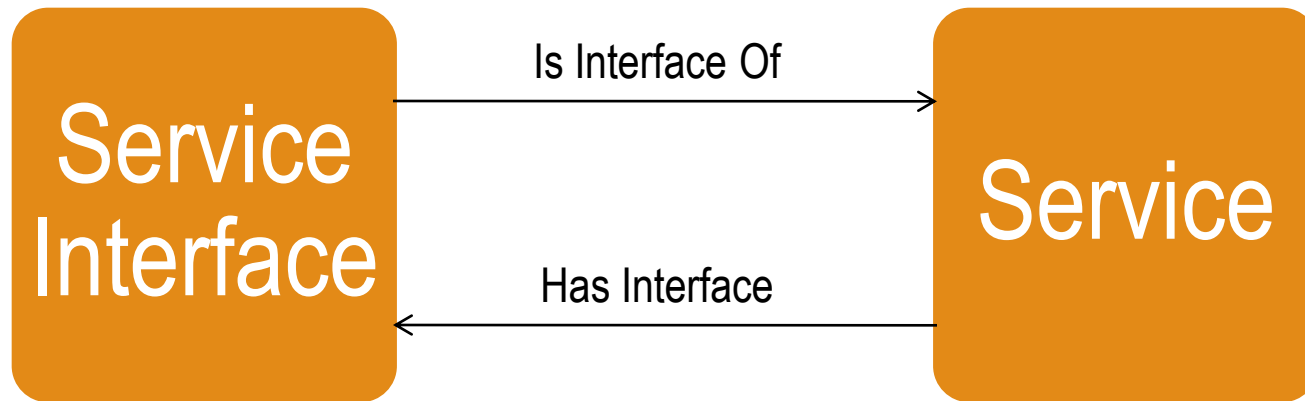


Car Wash Example – The Washing Services



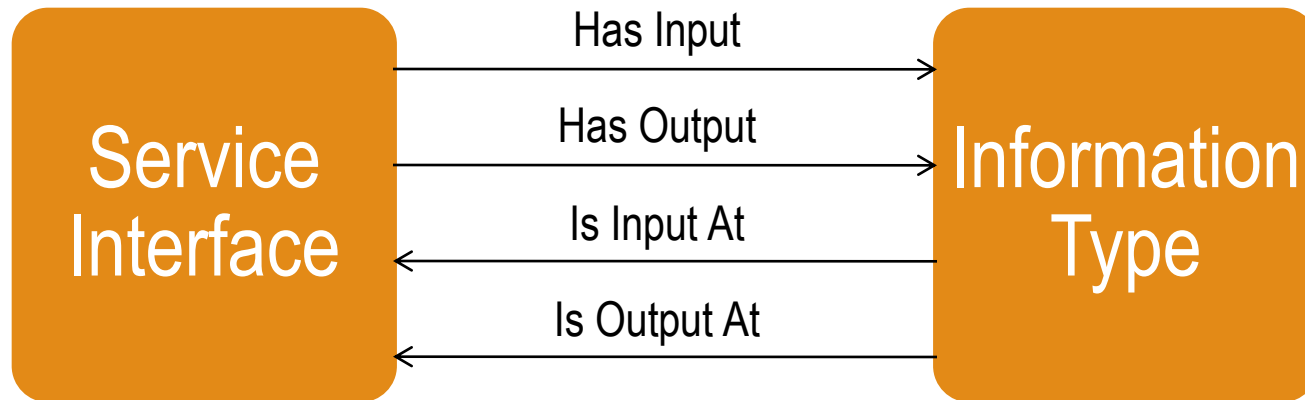
Service Interface

An important characteristic of services is that they have simple, well-defined interfaces. This makes it easy to interact with them, and enables other elements to use them in a structured manner. A *service interface* defines the way in which other elements can interact and exchange information with a service.



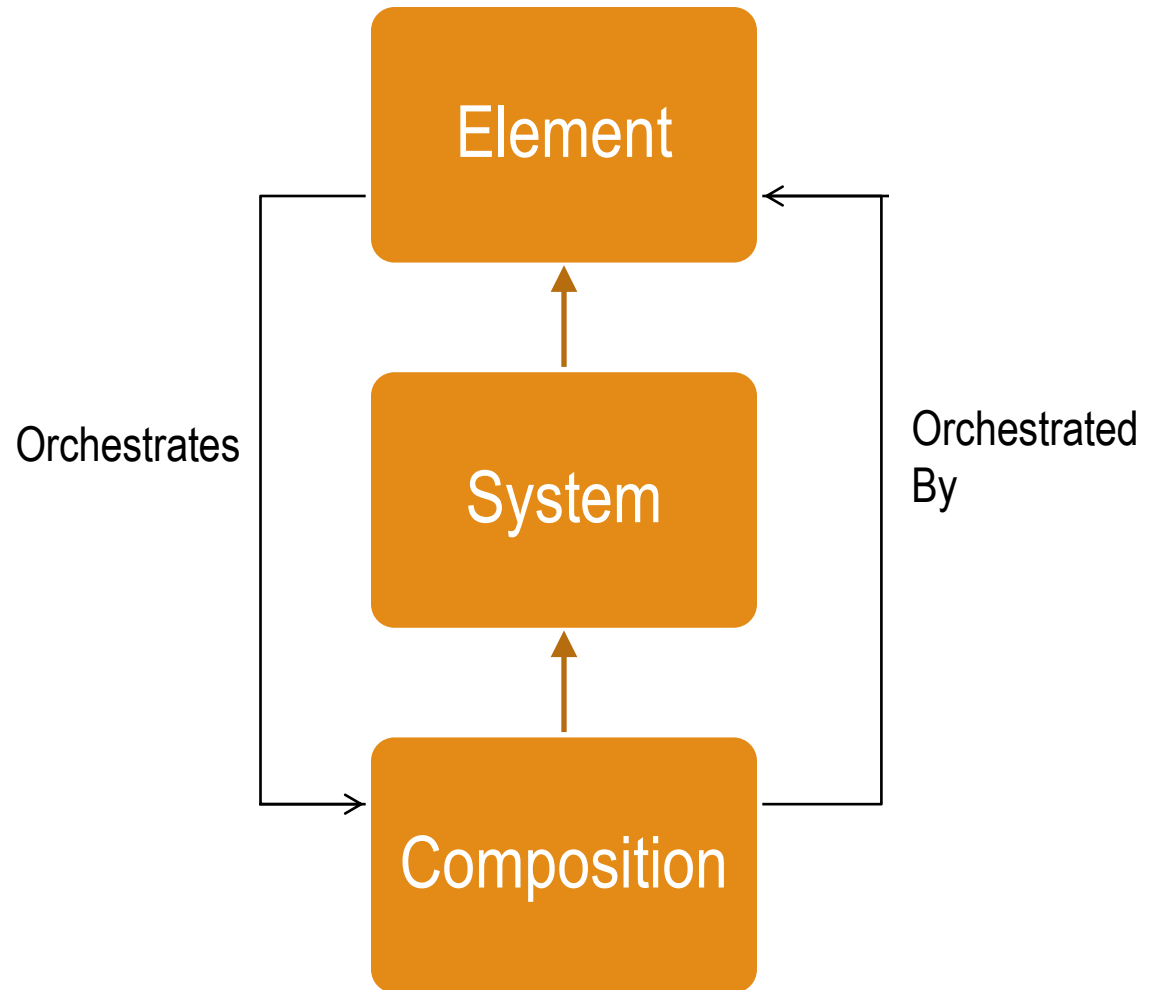
Information Type

A service interface can enable another element to give information to or receive information from a service (when it uses that service); specifically the types of information given or received.



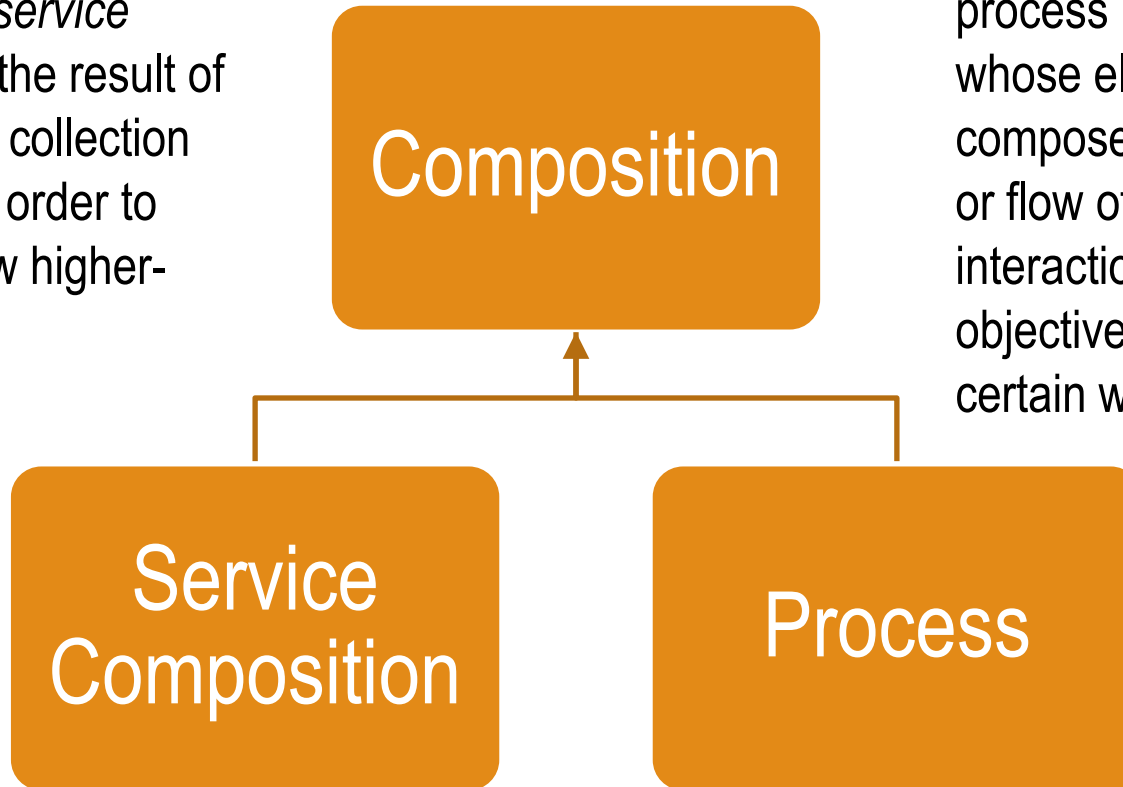
Composition

A *composition* is the result of assembling a collection of things for a particular purpose. Note in particular that we have purposefully distinguished between the act of composing and the resulting composition as a thing, and that it is in the latter sense we are using the concept of composition here.



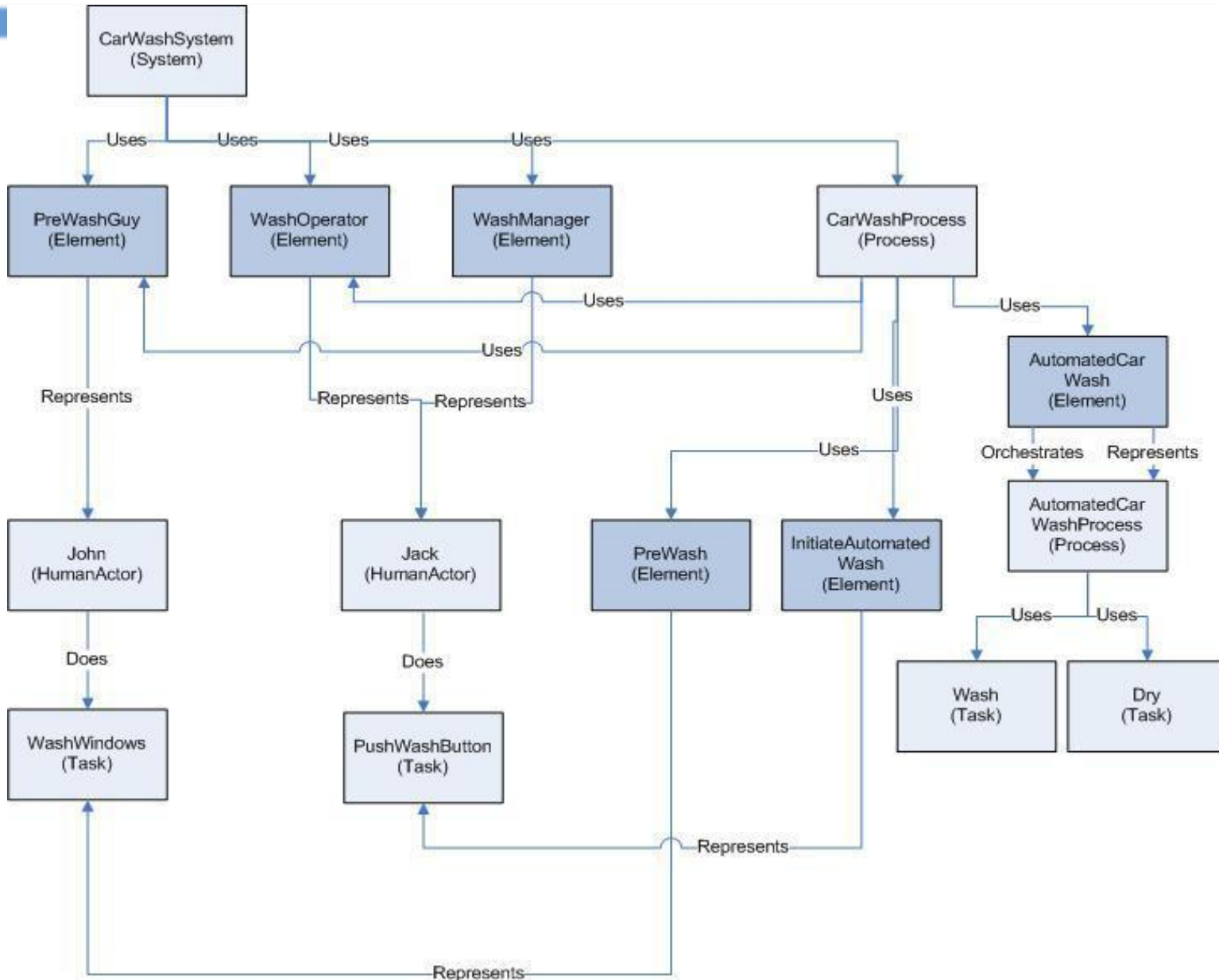
Service Composition and Process

A key SOA concept is the notion of *service composition*, the result of assembling a collection of services in order to perform a new higher-level service.



Another key SOA concept is the notion of *process*. A process is a composition whose elements are composed into a sequence or flow of activities and interactions with the objective of carrying out certain work.

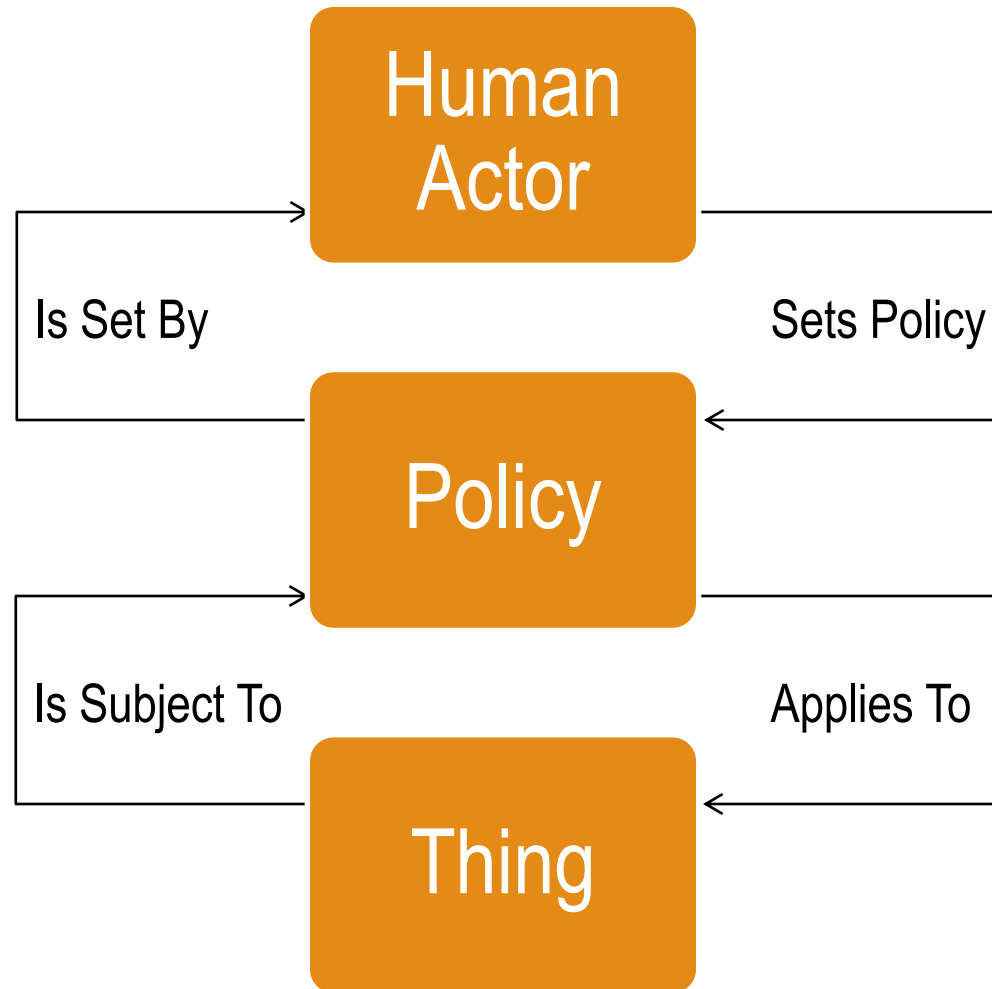
Car Wash Example – The Washing Processes



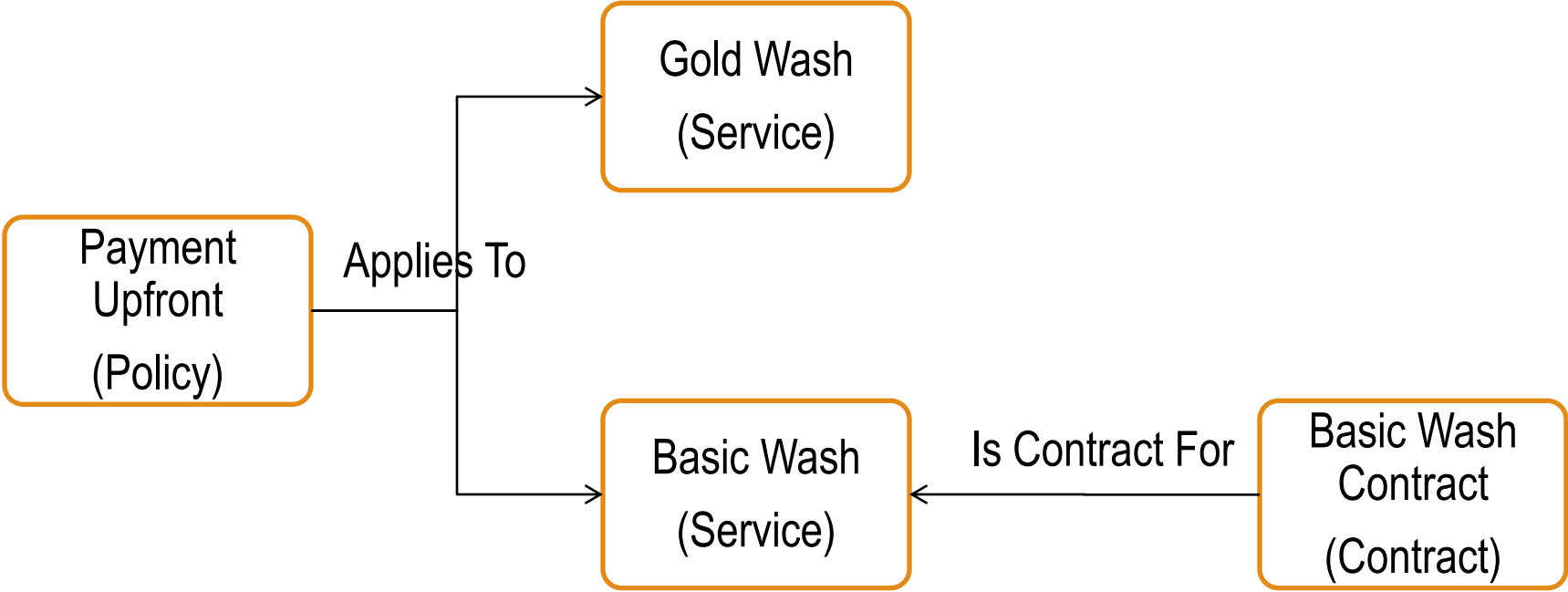
Policy

A *policy* is a statement of direction that a human actor may intend to follow or may intend that another human actor should follow.

Knowing the policies that apply to something makes it easier and more transparent to interact with that something.

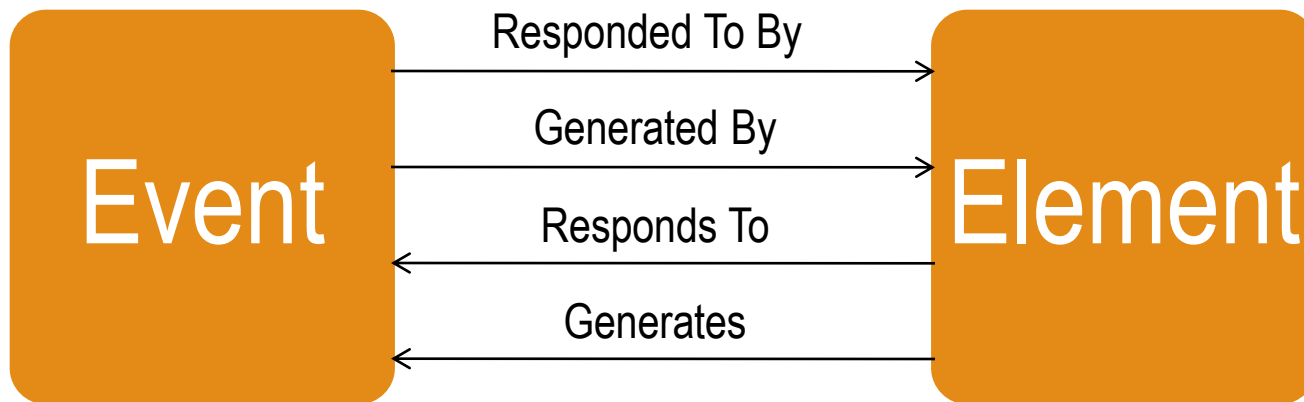


Car Wash Example – Policy and Contract

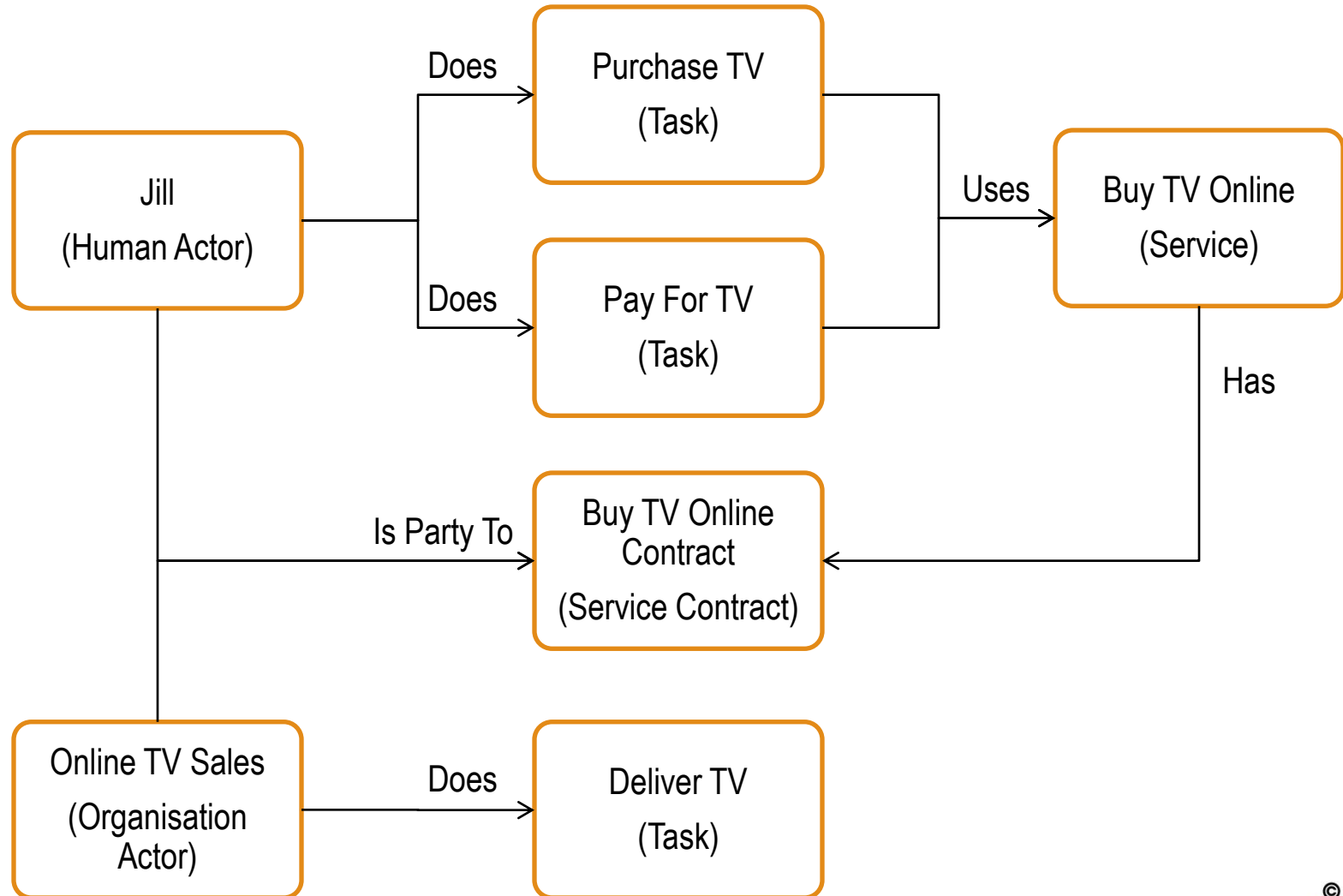


Event

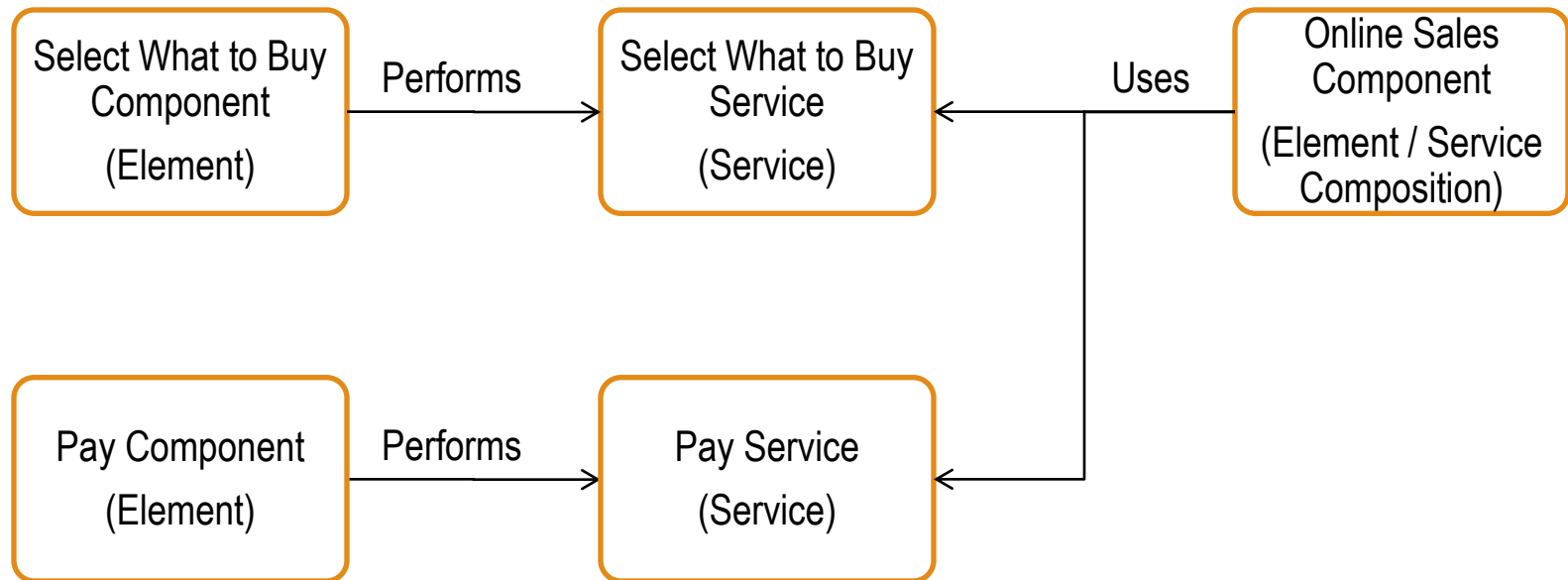
An *event* is something that happens, to which an element may choose to respond. Events can be responded to by any element. Similarly, events may be generated (emitted) by any element. Knowing the events generated or responded to by an element makes it easier and more transparent to interact with that element. Note that some events may occur whether generated or responded to by an element or not.



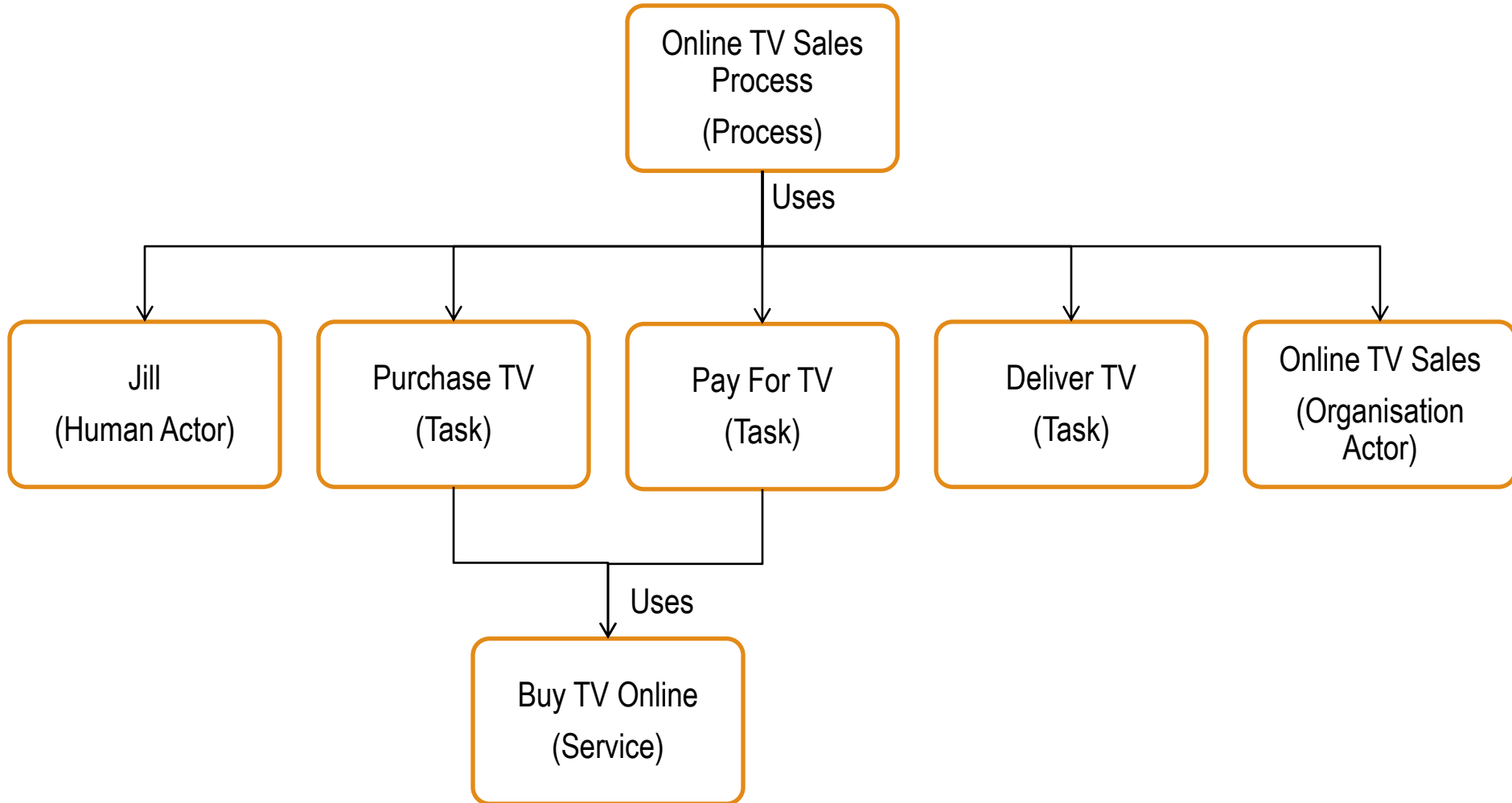
Internet Purchase Example – External View



Internet Purchase Example – Internal View



Internet Purchase Example – Process View



Conformance

OWL Application

- ❑ Must conform to the OWL standard
- ❑ Must include (in the OWL sense) the whole of the ontology
- ❑ Can add other OWL constructs, including class and property definitions
- ❑ Can import other ontologies in addition to the SOA ontology

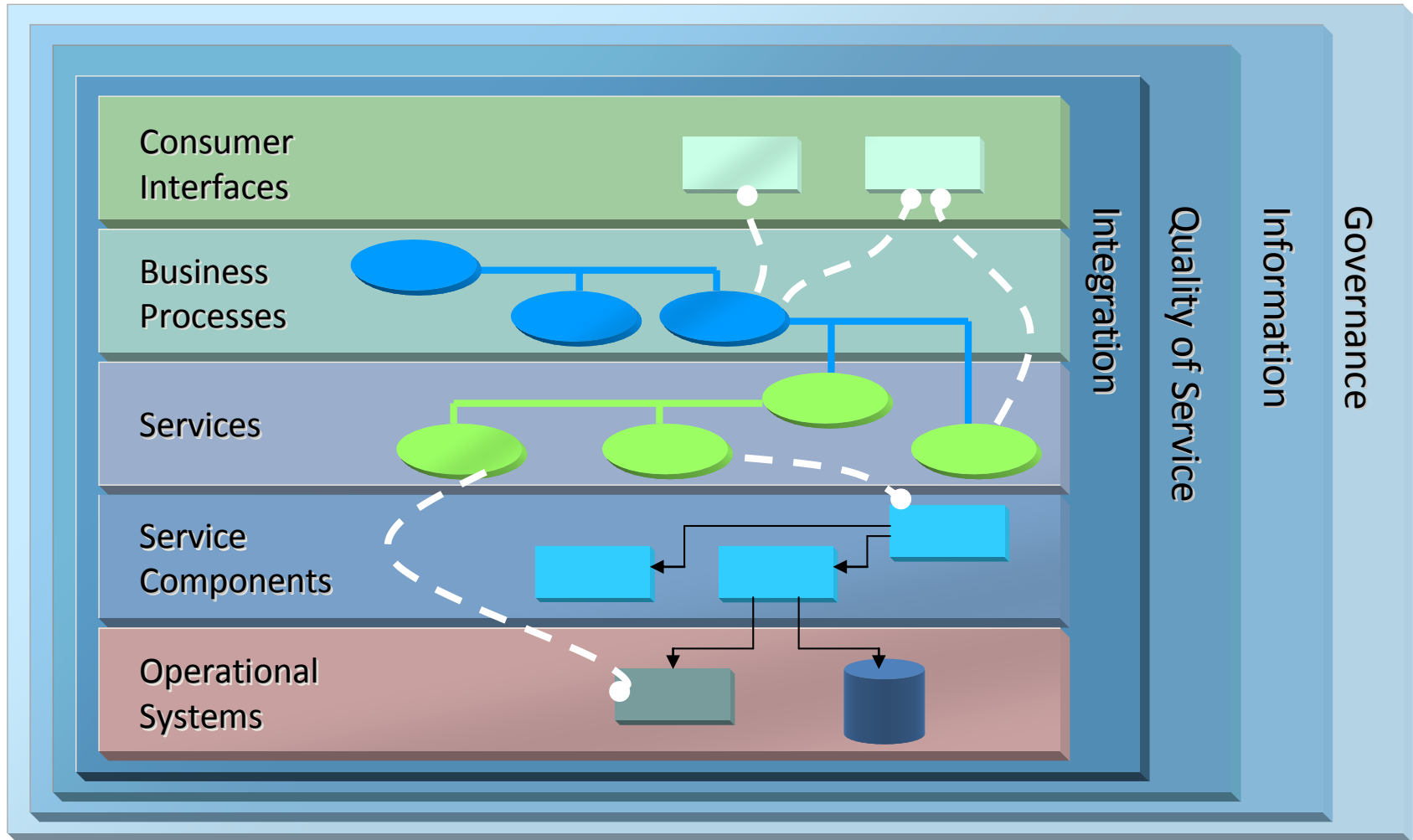
Non-OWL (Meta-model or Software)

- ❑ Must include a defined and consistent transform to a non-trivial subset of the ontology contained in Appendix A of this Technical Standard
- ❑ Can add other constructs, including class and property definitions
- ❑ Can leverage other ontologies in addition to the SOA ontology

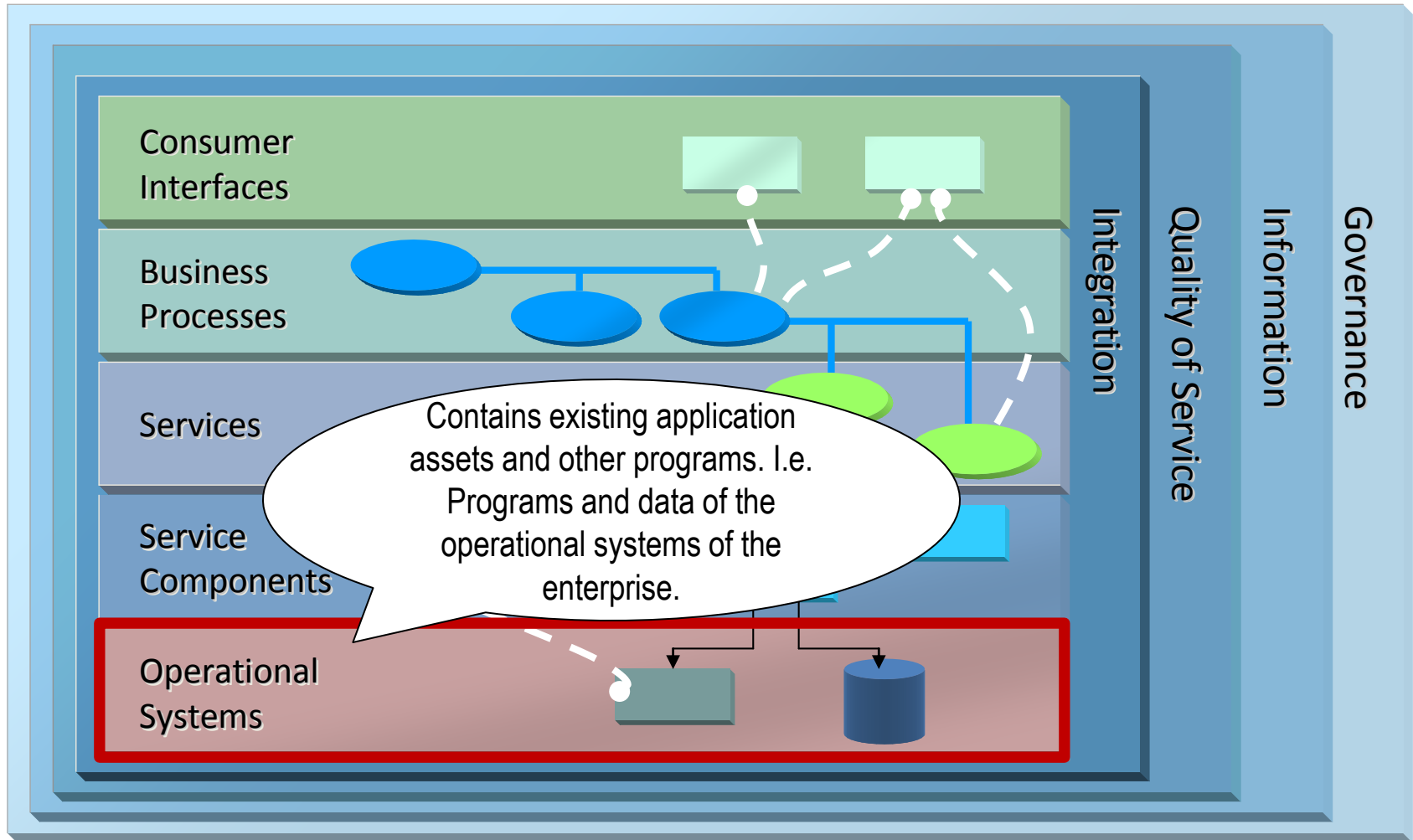
References

- ❑ www.opengroup.org/soa/source-book/ontology/
- ❑ www2.opengroup.org/ogsys/jsp/publications/PublicationDetails.jsp?catalogno=c104
- ❑ Catalog number C104
US ISBN 1931624887
Oct 2010
- ❑ 90 pages

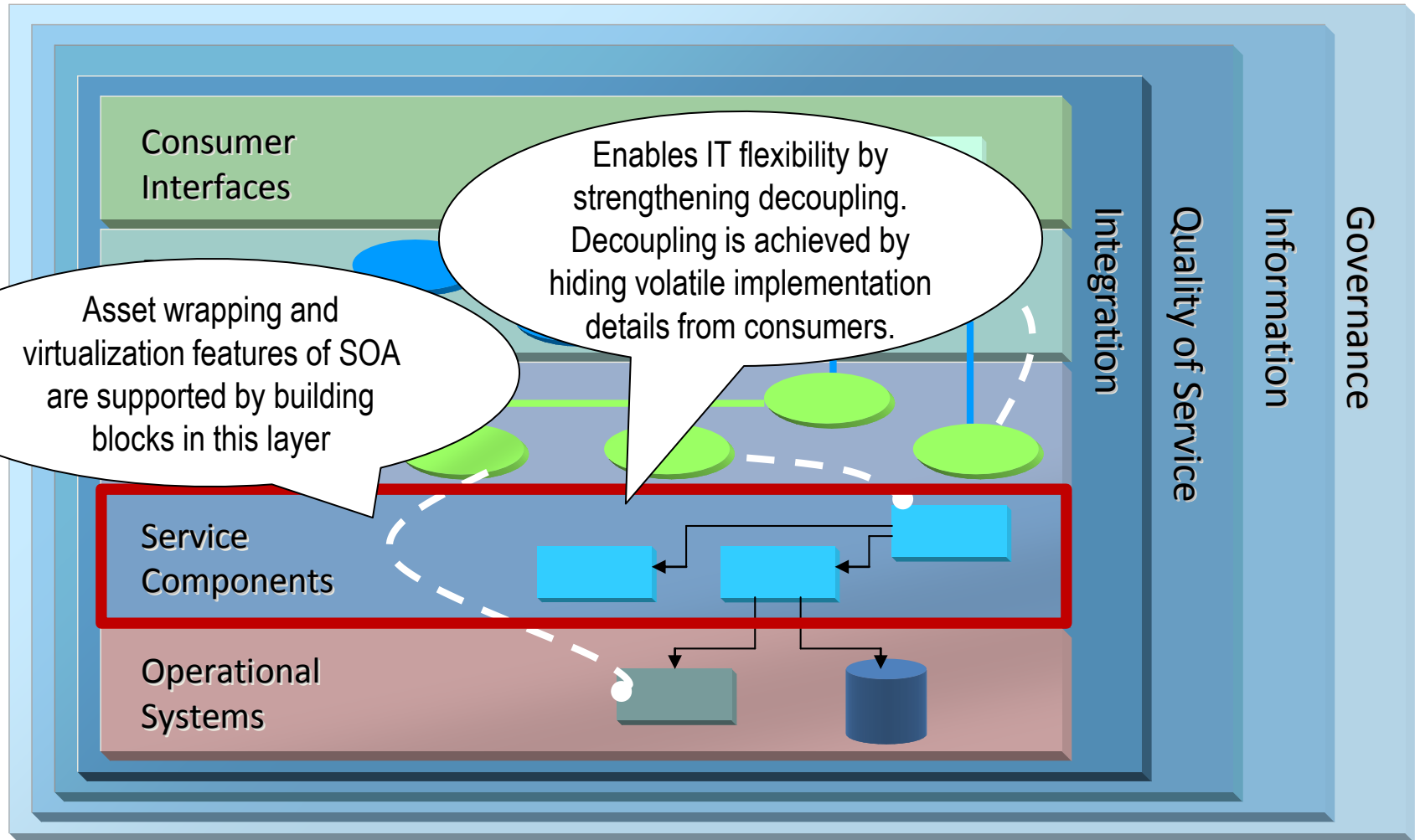
The Open Group SOA Reference Architecture



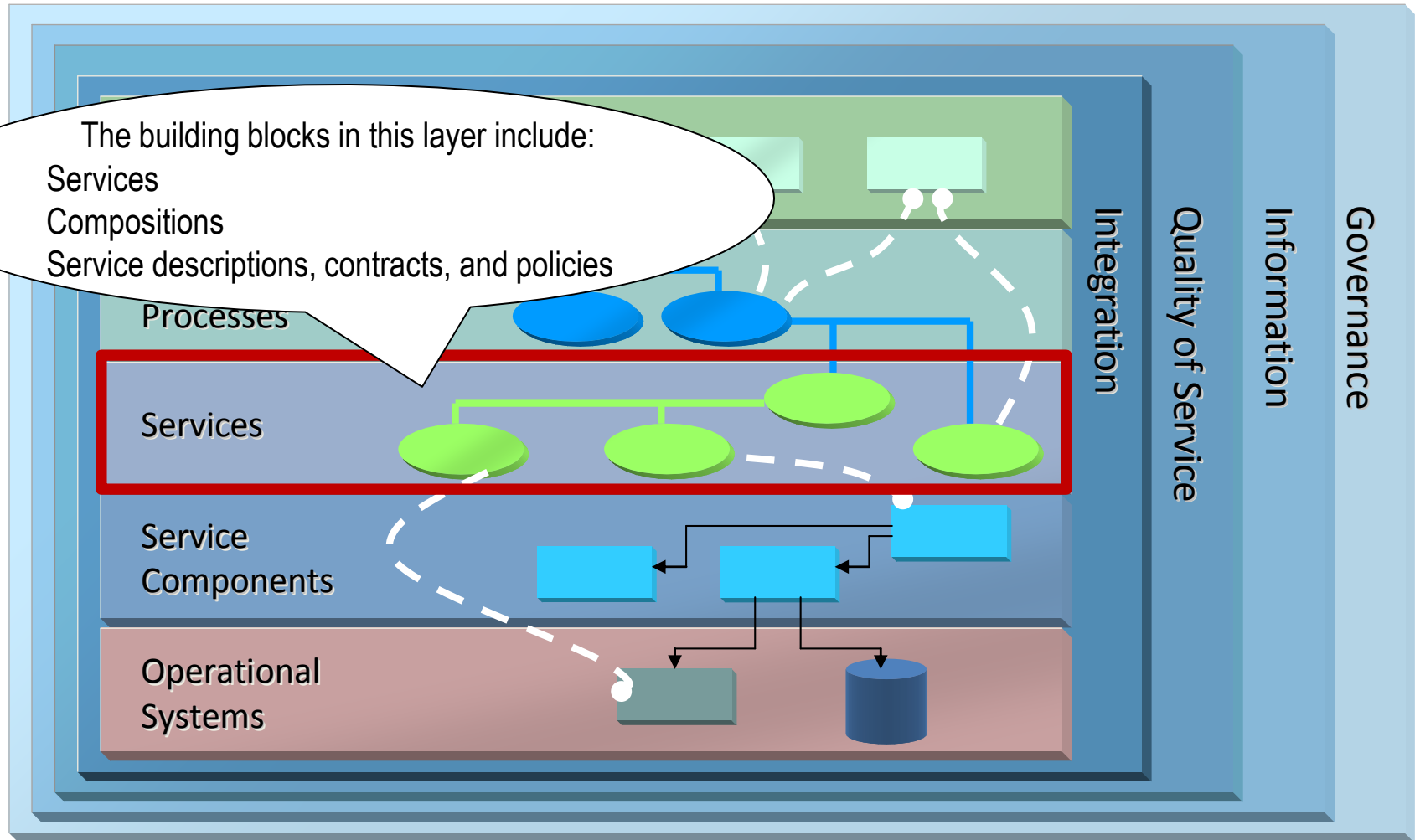
SOA RA – Operational Systems Layer



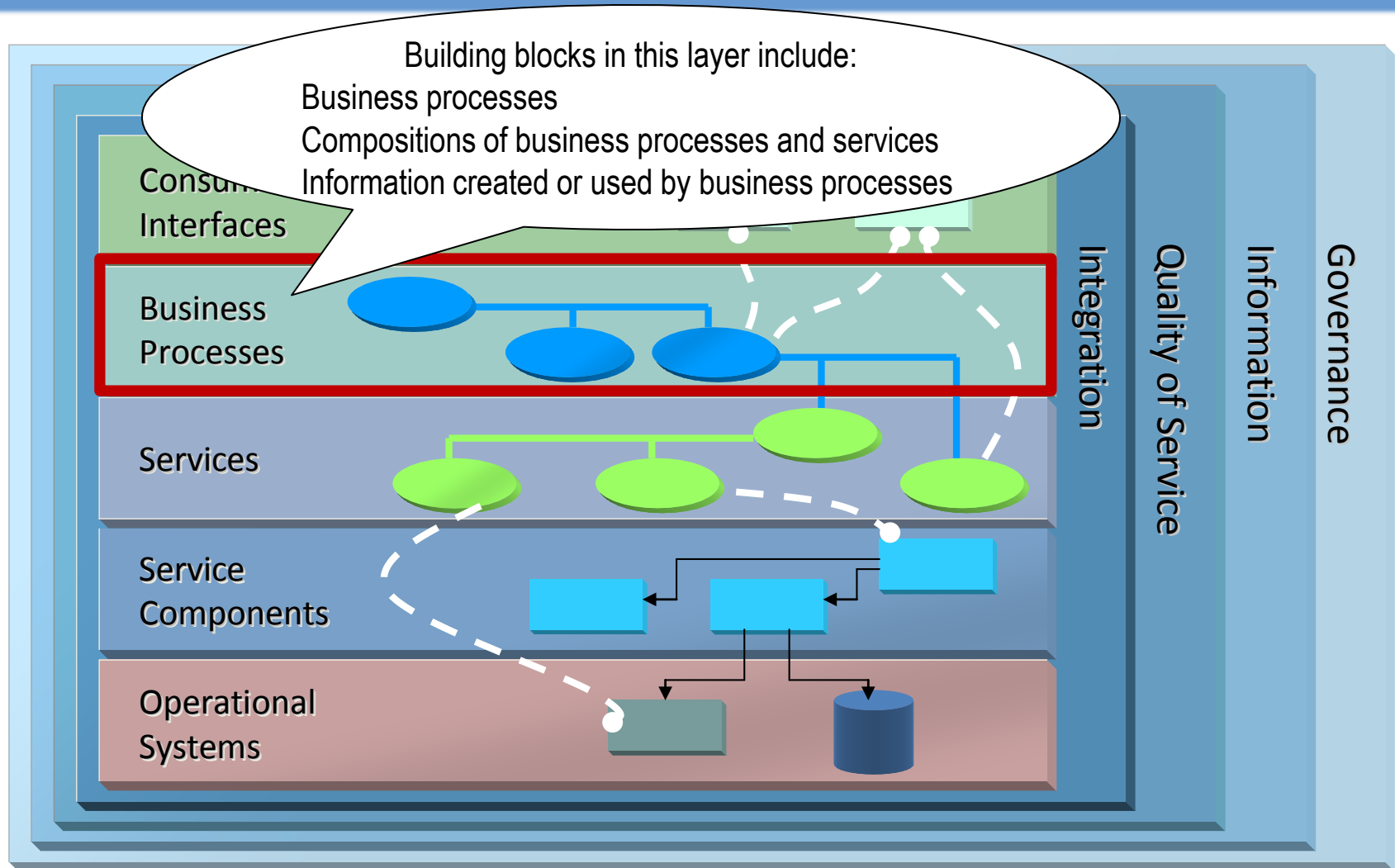
SOA RA – Service Components Layer



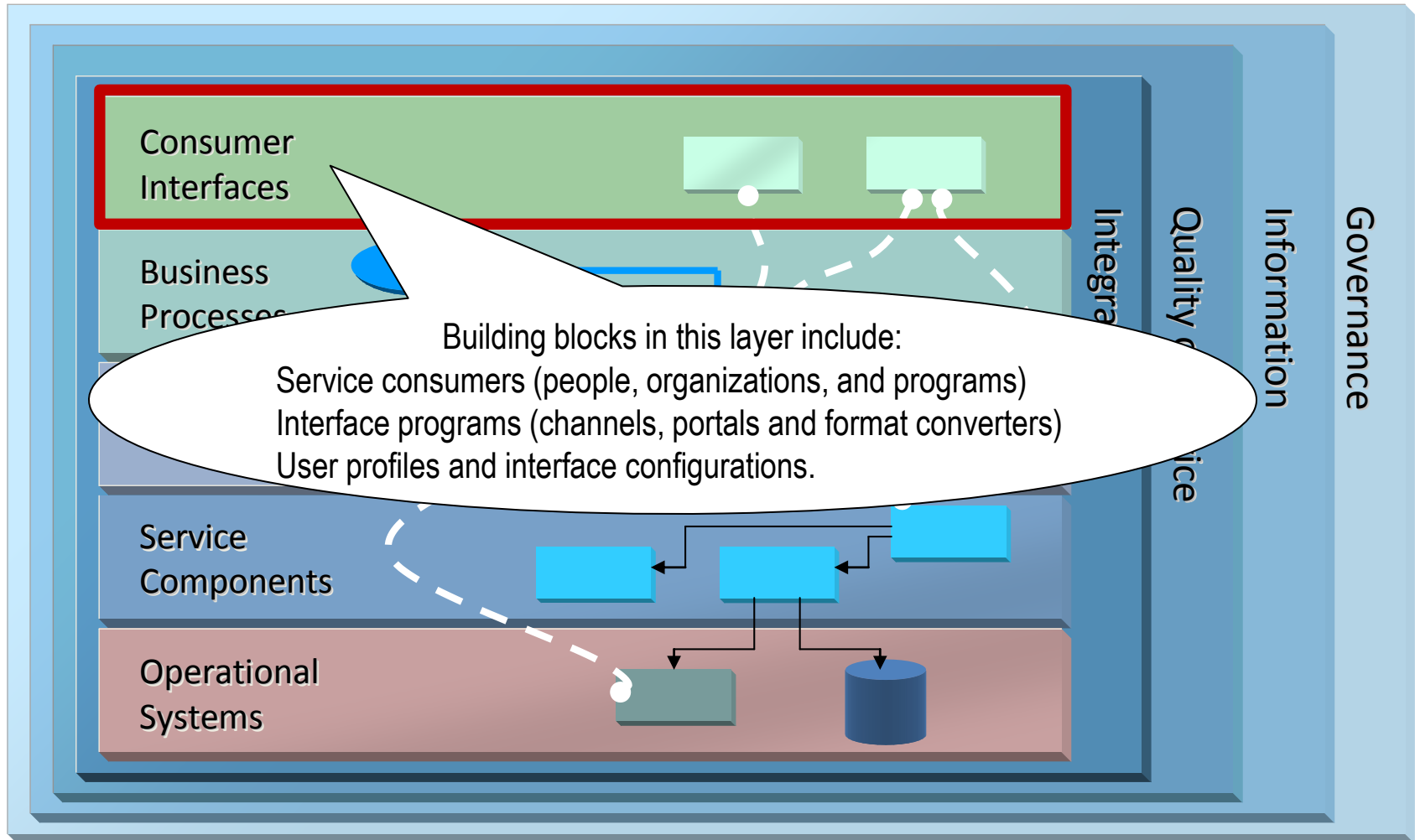
SOA RA – Services Layer



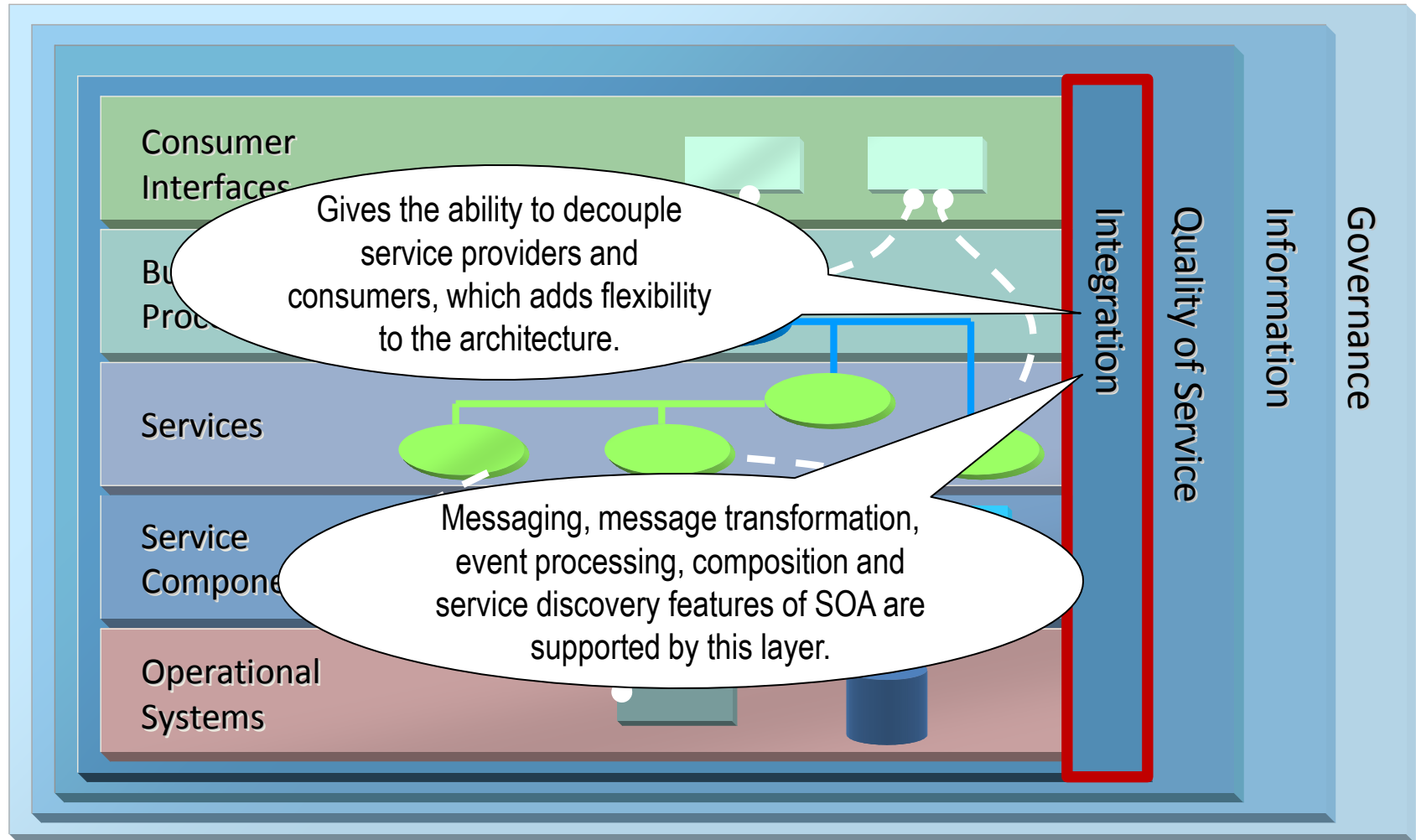
SOA RA – Business Processes Layer



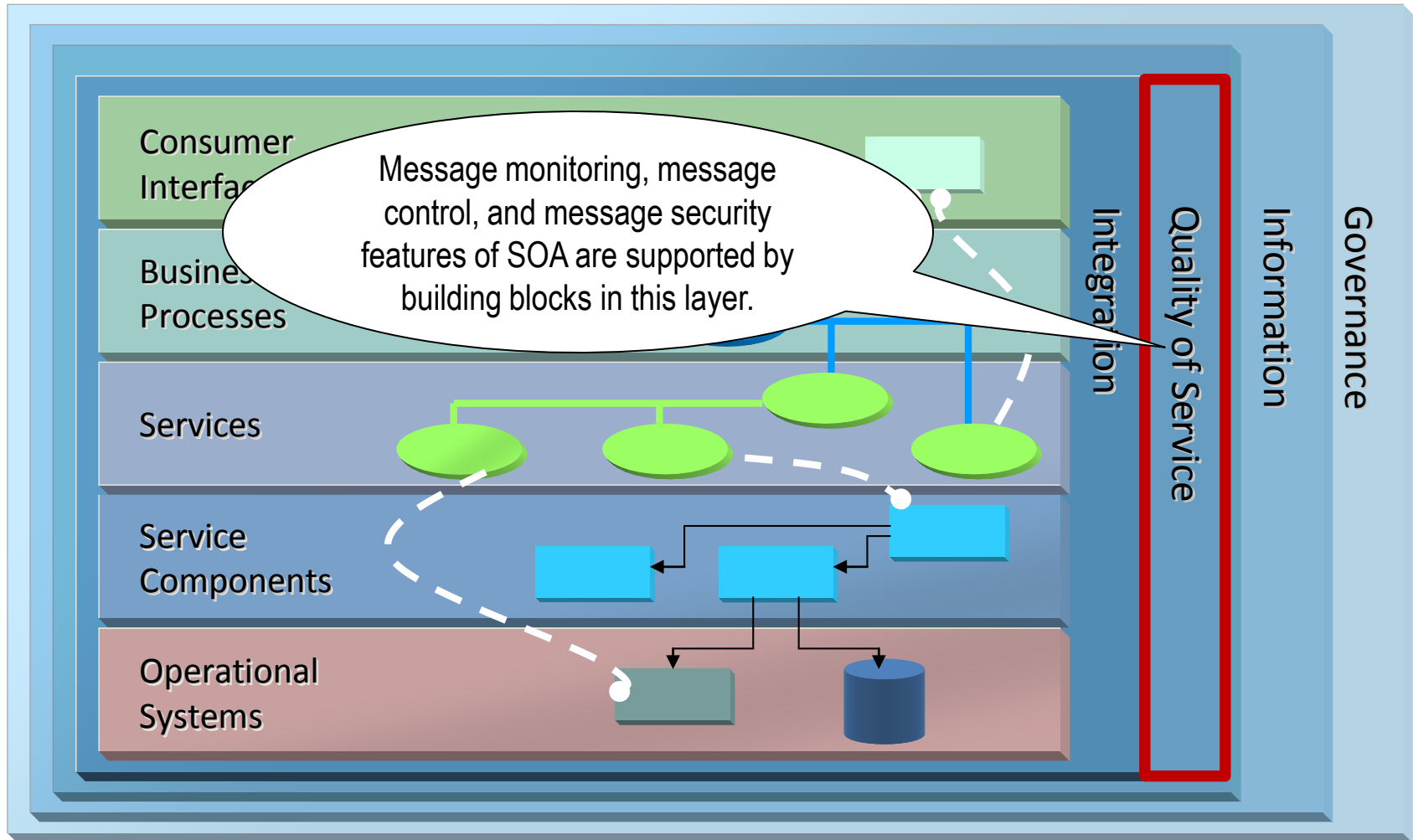
SOA RA – Consumer Interfaces Layer



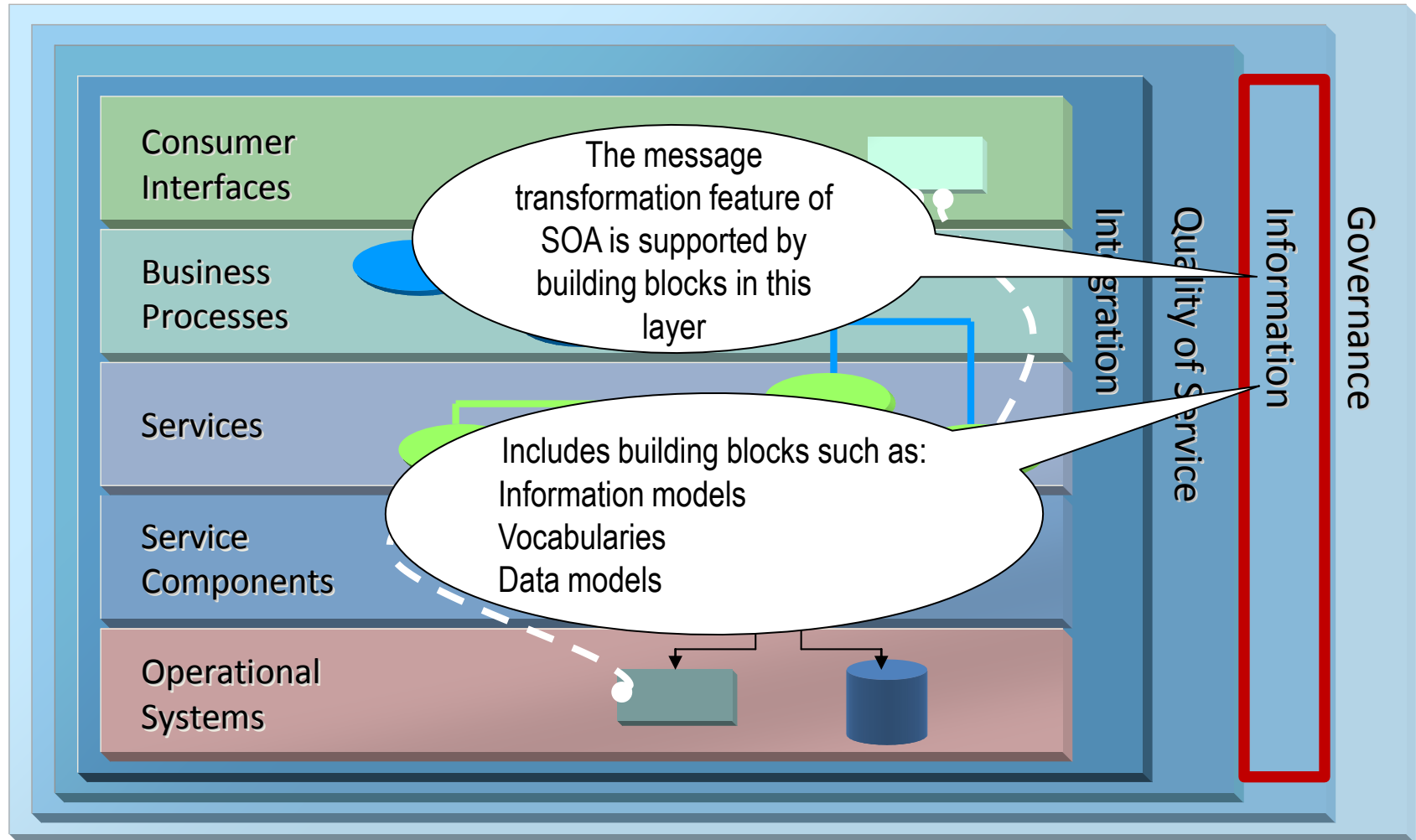
SOA RA – Integration Layer



SOA RA – Quality of Service Layer



SOA RA – Information Layer



SOA RA – Governance Layer

