OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) Technical Committee

The official charter for this Technical Committee is provided below. (For additional information, see the Call for Participation that was issued when this TC was formed.)

Contents

1. **Name of the TC**

OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) Technical Committee

2. **Statement of Purpose**

The goal of the Topology and Orchestration Specification for Cloud Applications (TOSCA) TC is to substantially enhance the portability of cloud applications and the IT services that comprise them running on complex software and hardware infrastructure. The IT application and service level of abstraction in TOSCA will also provide essential support to the continued evolution of cloud computing. For example, TOSCA would enable essential application and service lifecycle management support, e.g., deployment, scaling, patching, etc., in Software Defined Environments (SDE) such as Software Defined Data Centers (SDDC) and Software Defined Networks (SDN).

TOSCA will facilitate this goal by enabling the interoperable description of application and infrastructure cloud services, the relationships between parts of the service, and the operational behavior of these services (e.g., deploy, patch, shutdown) independent of the supplier creating the service, and any particular cloud provider or hosting technology. TOSCA will also enable the association of that higher-level operational behavior with cloud infrastructure management.

This capability will greatly facilitate much higher levels of cloud service/solution portability without lock-in, including:

- Portable deployment to any compliant cloud
- Easier migration of existing applications to the cloud
- Flexible bursting (consumer choice)
- Dynamic multi-cloud provider applications

Ultimately, this will benefit the consumers, developers, and providers of cloud-based solutions and provide an essential foundation for even higher-level TOSCA-based vocabularies that could be focused on specific solutions and domains.

3. Scope of Work

It is the goal of the TOSCA TC to complete a revision of the Topology and Orchestration Specification for Cloud Applications, submitted as a foundational specification by Capgemini, CA Technologies, Cisco, Citrix, EMC, IBM, NetApp, PwC, Red Hat, SAP, Software AG, Virtunomic, and WSO2 [1], and to complete the associated XML Schema plus conformance statements that meets the needs of members of the TOSCA TC and the IT industry as soon as possible, consistent with the OASIS process.

The TOSCA TC will leverage this submission as a foundation for further standardization of a basic set of ~~concrete~~ non-service specific and non-product specific components (for example, “database” or “OS”) as opposed to service specific and product specific components (such as “MySQL” or “Linux”),, relationships and properties ~~(~~with extension mechanisms to add additional components, relationships and properties~~)~~. Further ~~work~~ standardization on product -specific vocabularies, based on these extension mechanisms, is out of scope for ~~this specification~~this TC, ~~but could begin in parallel with this project, using the TOSCA naming syntax~~.

The scope of the TOSCA TC's work is to produce specifications that standardize the concepts as well as XML documents and XML Schema renderings of the areas described below by further refinement and finalization of the input document and any subsequent input documents accepted by the TOSCA TC. The following items are specifically in scope of the resulting TOSCA specification:

1. A language that provides the ability to specify a Service Template that can define the topology (or structure) of a service and that can utilize existing process modeling standards (especially BPMN 2.0) to define orchestration (via "plans") that can invoke the manageability behavior of cloud services.

2. A syntax for naming component types, components, relationship types, relationships, and properties, and for grouping of components.
3. Standardization of a basic set of non-service specific and non-product specific ~~non-vendor specific, concrete~~ component types, relationship types and properties and operations, which includes all type attributes ~~of the type~~ and all contained elements.
4. The ability to constrain the use of the various elements and their properties that define the topology of a service.
5. The ability to cross-reference Service Templates to enable composition of services and to enable the management of instantiations of a Service Template in heterogeneous environments.
6. The ability to use virtual images as implementation artifacts for parts of a Service Template.
7. The ability to use application artifacts (e.g. JEE, ABAP, etc) as deployment artifacts for parts of a Service Template.
8. The ability to use other artifacts (e.g. EAR files, OVF files, SCA components, etc) as deployment artifacts for parts of a Service Template.
9. The ability to annotate the various elements that define the topology of a Service Template with policies that influence use of instances of a Service Template. Such annotations could leverage a wide range of policy languages (e.g. WS-Policy [2], KaoS [3], etc.).

Compatibility

There are no formal requirements for upward compatibility. Nevertheless, the specification should be compatible with existing business process modeling standards like BPMN 2.0 [4] and WS-BPEL [5]. Furthermore, interfaces of component types should be able to be expressed in a proper REST-style based on HTTP and specified via WSDL 1.1, and allow for the use of scripts.

Out of Scope

The following is a non-exhaustive list. It is provided only for the sake of clarity. If some function, mechanism or feature is not mentioned here, and it is not mentioned in the Scope of Work section, then it will be deemed to be out of scope. The following items are specifically out of scope of the TOSCA specification:

~~10.~~1. The standardization or normative definition ~~definition~~ of product-specific or service-specific ~~concrete~~ cloud services, ~~i.e. the definition of concrete~~ component types, relationship types, and topology templates. ~~. However, standardization of a basic set of non-vendor specific concrete~~

~~component types, relationship types and properties, which includes all attributes of the type and all contained elements, is intended to be enabled by this work.~~

~~1.~~2. The definition of concrete plans, i.e. the definition of plans in any process modeling language like BPMN or BPEL.

~~2.~~3. The definition of a language for defining plans (i.e. a new process modeling language).

~~3.~~4. The definition of concrete policies influencing the management and use of instances of a Service Template.

~~4.~~5. The emphasis of any particular single technology (e.g. hypervisor virtualization) for the implementation of cloud services.

~~5.~~6. The emphasis of any particular policy definition language or mechanism.

~~6.~~7. The architecture of a service container used to instantiate service definitions and manage such instances.

~~7.~~8. The interface definitions of a service container.

~~8.~~9. A graphical notation for modeling Service Templates.

~~9.~~10. The definition of semantic models for cloud services.

~~10.~~11. The specification of functional behavior as well as functional composition of cloud services.

Subsequent specifications may provide the Service Templates of concrete cloud services. This will enable, for example, the creation of catalogues of Service Templates in various application domains.

Deliverables

The TOSCA TC will provide the following set of deliverables:

1. A revised Topology and Orchestration Specification for Cloud Applications, and associated XML Schema plus conformance statements will be approved and completed by the TC within nine months of the first TOSCA TC meeting.
2. A set of sample Cloud Service Templates and related artifacts will be approved and completed by the TC within nine months of the first TOSCA TC meeting. These examples are non-normative, but can be used as test cases for testing conformance of individual TOSCA implementations as well as interoperability between multiple TOSCA implementations.
3. Optionally, such other non-normative deliverables ~~within the scope listed in paragraphs 1-8~~ such as tutorials~~ or~~, presentations, best practices, or non-normative definitions to be used for testing,~~),~~ as the TC may elect, within nine months of the first TOSCA TC meeting.

Maintenance

Once the TC has completed work on a deliverable and it has become an OASIS Standard, the TC will enter "maintenance mode" for the deliverable. The purpose of maintenance mode is to provide minor revisions to previously adopted deliverables to clarify ambiguities, inconsistencies and obvious errors. Maintenance mode is not intended to enhance a deliverable or to extend its functionality.

The TC will collect issues raised against the deliverables and periodically process those issues. Issues that request or require new or enhanced functionality shall be marked as enhancement requests and set aside. Issues that result in the clarification or correction of the deliverables shall be processed. The TC shall maintain a list of these adopted clarifications and shall periodically create a new minor revision of the deliverables including these updates. Periodically, but at least once a year, the TC shall produce and vote upon a new minor revision of the deliverables.

IPR Mode

This TC will operate under the "RF (Royalty Free) on Limited Terms" IPR mode as defined in the OASIS Intellectual Property Rights (IPR) Policy.

Anticipated Audience

The anticipated audience for this work includes:

1. Vendors and service providers offering products and/or services designed to host or support cloud services, especially.
    - Solutions used to model and create cloud services
    - Solutions that support the execution of cloud services
    - Solutions that manage cloud services
    - Solutions designed to provide (parts of) cloud services as virtual images
    - Solutions designed to deploy or manage cloud services across multiple service providers
2. Other specification authors that require cloud Service Templates
3. Software architects who design, write, integrate and deploy cloud services in a cloud environment as well as in a mix of cloud environments and on-premise environments
4. End users implementing solutions that require an interoperable, composable solution using cloud services

Language

TC business will be conducted in (US) English. The output documents will be written in (US) English.

References:

[1] Topology and Orchestration Specification for Cloud Applications, Draft Specification, September 2011.
[2] Web Services Policy 1.5 - Framework, W3C Recommendation, available via http://www.w3.org/TR/ws-policy/
[3] KAoS, Florida Institute for Human and Machine Cognition, available via http://ontology.ihmc.us/index.html
[4] OMG Business Process Model and Notation (BPMN) Version 2.0, available via http://www.omg.org/spec/BPMN/2.0/
[5] OASIS Web Services Business Process Execution Language (WS-BPEL) 2.0, available via http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf