Topology and Orchestration Specification for Cloud Applications (TOSCA) Standard

TOSCA Interoperability Demonstration

Participating Companies:

Join the TOSCA Technical Committee
www.oasis-open.org, join@oasis-open.org
Enable **portability** and **semi-automatic management** of cloud applications across clouds regardless of provider platform or infrastructure thus expanding customer **choice**, improving **reliability**, and reducing **cost** and time-to-value.

The TOSCA standard...

- **provides the Interoperable Description of:**
  - Applications, their component **Services** and **Artifacts**
  - Platform and **Infrastructure** services
  - Relationships between these services
  - Management and Operational behavior of these services

- **facilitates higher levels of Solution Portability:**
  - Portable deployment to any cloud that can orchestrate TOSCA service templates
  - Simplify migration of existing customer apps. to cloud
  - Dynamic, flexible scaling and bursting of multi-cloud applications

- **Enables Software Defined Environments (SDEs)**
  - Template contents provide the means to optimize the underlying cloud infrastructure

---

**Contributing Members**
Open Ecosystem for Cloud Services

- Vendor-independent definitions of complex Cloud services provide new marketing channel for solutions in the Cloud
- Decoupling of Cloud infrastructure and Cloud content helps focus on key aspects: Cloud Provider or Cloud Service Provider
- Ability to deploy services in any standards-compliant environment avoids vendor lock-in and eases migration

Interoperability and Composition

- Goes beyond VMs in describing the cloud application's components and their dependencies
- Composition of services defined independently by their domain experts into a higher-value service
- Key enabler for open hybrid Clouds

Easy Adoption of new Cloud Services

- Model-driven creation of Cloud Services
- Standardized deployment into various kinds of environments – from test to production, from cloud A to cloud B
- Process-driven Cloud Service Lifecycle Management
**Demonstrating:** different cloud orchestration tools from different vendors all interpreting and seamlessly running the same TOSCA service templates in the same way.

**Benefits:** Using TOSCA service templates, enterprise customers can easily move their applications from one cloud to another and orchestrate them using the expert knowledge the application developers have built into them.
Architects and Developers can choose from many open source and commercial tools to create, compose, manage and monitor TOSCA Cloud Application & services.

Reusable, Composable TOSCA Service Templates

- Companies & Software Providers can share TOSCA service templates which have encapsulated their expert knowledge using “marketplaces” for others to reuse and extend.

Orchestration of TOSCA Apps in any TOSCA enabled cloud

- Customers can seamlessly deploy, run and manage any TOSCA applications in any TOSCA enabled cloud.

Wide Range of Open Source & Commercial Tooling

- Architects and Developers can choose from many open source and commercial tools to create, compose, manage and monitor TOSCA Cloud Application & services.
Scenario Demonstration Elements

Demo participants demonstrating different parts of the TOSCA Ecosystem *

1. TOSCA Application Modeling
   - **Tooling Interop.** – Standardized modeling of cloud application services and relationships
   - Optionally, tools can directly deploy to clouds for testing or production

2. Publish TOSCA Service Template to a sample cloud marketplace
   - Using the TOSCA Cloud Service Archive (CSAR) file format

3. Share / Reuse TOSCA Templates
   - Customers can rapidly discover and compose cloud solutions from “ready made” templates

4. Deployment to Cloud Provider of Choice
   - **Runtime Interop.** – seamless deployment to TOSCA-enabled clouds

5. Monitor TOSCA Cloud Applications
   - **Tooling Interop.** – granular monitoring of application services described by TOSCA

*Many of the participating companies have products that apply to all parts of the ecosystem*
Architects and Developers can choose from many open source and commercial tools to create, compose, update and manage TOSCA Cloud Applications.

Demonstration shows a representative modeling tool able to quickly compose and publish the TOSCA SugarCRM sample application.

The 2-Tiers of the sample TOSCA SugarCRM app:

1. Web Application Tier
   - Linux, Apache, PHP, SugarCRM

2. Database Tier
   - Linux, MySQL

Demonstration shows an export of the TOSCA SugarCRM sample application to a representative cloud marketplace within a TOSCA CSAR package.
Companies & Software Providers can share TOSCA service templates which have encapsulated their expert knowledge using public or private “marketplaces” for others to **reuse and extend**.

Demonstration shows a representative public cloud marketplace showing the newly published TOSCA SugarCRM sample application (i.e. a TOSCA Service Template, CSAR file)

- Publish TOSCA Cloud Service Archives (CSAR) on your choice of public or private application marketplaces.
- Benefit by **reusing** and customizing existing application templates which have expert architect and developer knowledge encapsulated.
Demo participants are demonstrating different parts of the TOSCA Ecosystem *

4 Demonstration shows choice of Cloud Providers, each able to seamlessly Import (from the marketplace), deploy and run the same TOSCA Service Template.

- Choose which participating company’s cloud you wish to see a TOSCA service template imported and deployed in.

Customers can choose to deploy and manage their TOSCA apps with many cloud providers that support the TOSCA standard.
Demonstration shows successful login to the SugarCRM application running at the chosen Cloud Provider

... using the address, user ID and password provided within the TOSCA SugarCRM application

Customers benefit from seamless TOSCA Run-time portability
Demonstration shows service level, service impact and the granular monitoring of application services described by TOSCA Service Templates.

“TOSCA-aware” tooling can enable monitoring of TOSCA service instances running in the cloud.

Customers can manage lifecycle and make adjustments to TOSCA defined services.
The following slides provide details on each participant’s interoperability demonstrations.
Demo import and deployment of SugarCRM and SAP CRM to a private cloud using a TOSCA Service Template

FlexFrame® Orchestrator

- interprets the TOSCA Service Template,
- orchestrates the environment for an application instance and
- deploys it into a private cloud
Import and deploy a TOSCA SugarCRM model as well show how to deploy a TOSCA SAP model.

HP’s comprehensive management solution for heterogeneous clouds provides you with all the management and governance capabilities you need to automate service delivery for a successful hybrid cloud.
Deploy the SugarCRM Based on Tosca CSAR Package

Step 1: Login

Step 2: Upload

Step 3: Deploying

Step 4: Deployment Successful

Topology of the Application based on the CSAR Package and the Deploying Flow

Status of each nodes descript in the CSAR Package
Demo TOSCA Service Templates import and deployment using IBM SmartCloud Orchestrator

- **Integrated tooling for TOSCA template creation and editing**
- **Import and export of TOSCA v1.0 compliant service templates**
- **Deploy-time composition of service templates based on policies**
- **Integrated monitoring and scaling of deployed services**

Import of the SugarCRM or SAP TOSCA applications into our integrated application builder tool.

SugarCRM or SAP CRM services offered in a service catalog.

Deployed instances of SugarCRM or SAP services with public IP addresses.
Demo designing, publishing and deploying SugarCRM and SAP CRM TOSCA Service Templates with Vnomic Suite

Vnomic Service Designer

Create TOSCA Service Templates using modeled Components and Artifacts

Vnomic Suite: Declarative Desired State Service Delivery and Governance for the most complex applications and infrastructures

Publish to Marketplaces

Import and Compose new Services

Compose Services from existing deployments and component libraries

Marketplaces

Deploy from Marketplaces

Vnomic Desired State Controller

Orchestrate TOSCA Service Templates across diverse clouds and infrastructures

Model existing deployments

Cloud X
Service Model Monitoring via TOSCA

• SAP service deployed with TOSCA orchestration
• Realized deployment is instrumented for *monitoring* using TOSCA template along with deployment information
• Results in deployment and *operational* support of services based on IT policies
TOSCA Resources - Learn More & Participate!

TOSCA Technical Committee – Public Website


TOSCA Specification, Version 1.0, Candidate Spec. 01, June 11, 2013

- http://docs.oasis-open.org/tosca/TOSCA/v1.0/cos01/TOSCA-v1.0-cos01.pdf

TOSCA Primer, Version 1.0, Committee Note Draft 01, 31 January 31, 2013

- http://docs.oasis-open.org/tosca/tosca-primer/v1.0/cnd01/tosca-primer-v1.0-cnd01.pdf


TOSCA Interop. Demo, SugarCRM Scenario Sample CSAR, August, 2013

More on TOSCA Modeling...
Modeling Topologies with TOSCA

Service Topologies are described using the TOSCA Meta-model:

Nodes
- Represent **Components** of an application or service and their **Properties**. Example nodes include:
  - Infrastructure: Compute, Network, Storage, etc.
  - Platform: OS, VM, DB, Web Server, etc.
  - Granular: functional Libraries, Modules, etc.
- Include **Operations** which are the management functions for the node
  - e.g. `deploy()`, `start()`, `stop()`, `connect()`, etc.
- Export their dependencies on other nodes as **Requirement** and **Capabilities**

Artifacts
- Describe **Installables** and **Executables** required to instantiate and manage a service. Currently, they include:
- **Implementation Artifacts**:
  - Executables or **Plans** that implement a Node’s or Relationship’s Operations (e.g. a Bash script)
- **Deployment Artifacts**:
  - Installables of the components (e.g. a TAR file)

Relationships
- Represent the logical **Relationships** between nodes
  - e.g. “hostedOn”, “connectsTo”, etc.
- Describes the valid **Source** and **Target nodes** they are designed to couple
  - e.g. source “**web application**” node is designed to connect to a target “**database**” node
- Have their own **Properties** and **Constraints**

Service Templates
- **Group** the nodes and relationships that make up a service’s topology
  - Allowing modeling of sub-topologies
- Service Templates **look like nodes** enabling:
  - **Composition** of applications from one or more service templates
  - **Substitution** of abstract Node types with available service templates of the same type

A service’s Topology Model is included in a **TOSCA Service Template** which is packaged and shared, along with all dependent artifacts, as a **TOSCA Cloud Service Archive (CSAR)**
TOSCA service templates can model any cloud application or infrastructure pattern.

**Value:** enables rapid and continuous delivery of diverse set of workloads with **agility** and **optimization** on programmable heterogeneous infrastructure leveraging **reusable** building blocks.
**TOSCA Service Templates support ...**

**Complete Topology Modeling**
Allow developers to describe the topology of their applications and *encapsulate their expert knowledge, including service configurations, policies and dependencies.*

**Full Lifecycle Orchestration**
Go beyond simple deployment; services can provide instructions for any *lifecycle operations* enabling *precise orchestration and control* of application management tasks.

**Service Composability**
Supports the ability to substitute logical parts of applications through *composable service templates* providing choice in both service vendor and implementation.
TOSCA supports **Containment** via the “**HostedOn**” relationship

**Containment**

Nodes can **host** or contain other Nodes of specified types

- **Nodes can export the types of nodes they are capable of hosting**, 
- **These are matched to other nodes that export their specific host container requirements**

In this example:

- **A MySQL Database node is “**hostedOn**“ a MySQL Database Management System (DMBS) node**
- **The MySQL DBMS node, in turn, is “**hostedOn**“ a Linux Operating System, and so on...**

**Tier** is a topological concept used to describe sets of nodes (or sub-topologies) that can be deployed and managed as a single group.
**TOSCA expresses Connectivity relationships between service nodes**

Connectivity

Nodes can **connect** to other specified node types

- **Nodes export the types of nodes they require connectivity to**
- **These are matched to nodes that export they are capable of accepting specific connections**

*In this example:*

- The SugarCRM Application node **connectsTo** a database node in another tier
- Note that the “DB Tier” components are packaged into a separate service template permitting **substitution**

**Network Connectivity**

Relationship Type is named “connectsTo”

*Components grouped into composable service templates.*
**TOSCA Supports Scalability via the “Tier” Grouping Node**

**Scalability**

“Tier” Node Types convey scalability

- The “Web Application Tier” is declared **scalable** with upper bounds ‘n’ instances
  - Note: the database tier remains a single instance
- A Load Balancer node is added to the previous template to route requests among “Web Application Tier” instances
- Both tiers are packaged into their own service templates permitting **substitution** of either

The range of instances would be a property of the “Tier” Node Type

Components grouped into composable service templates.