



Introduction and Overview

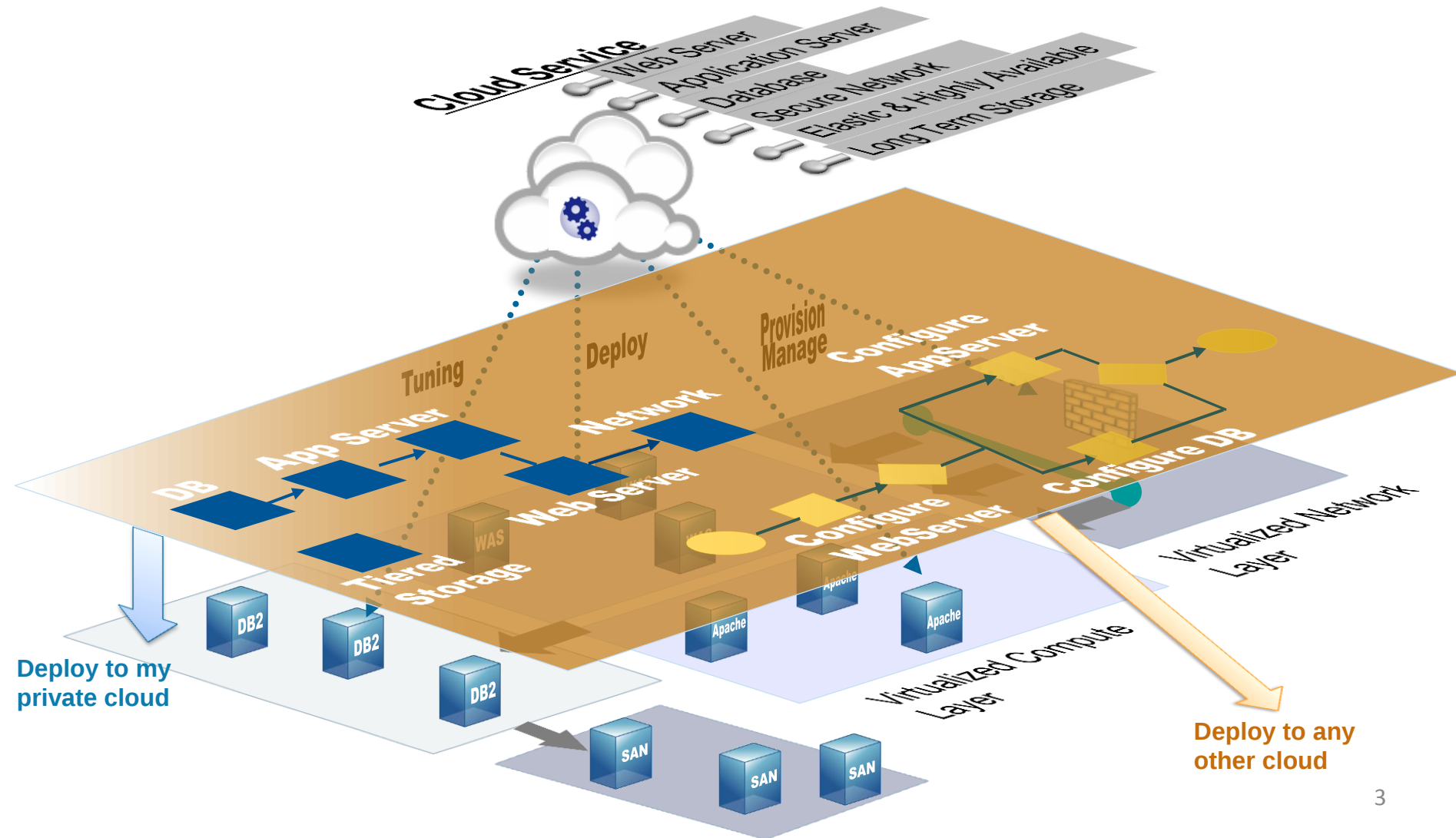
Thomas Spatzier, IBM (thomas.spatzier@de.ibm.com)

Agenda

- TOSCA Overview and Examples
- TOSCA Concepts Explained
- TOSCA advanced Features

Motivation for TOSCA

Even simple cloud services sometimes have complex software and hardware infrastructures backing them. So how do we ensure portability?



What is TOSCA?

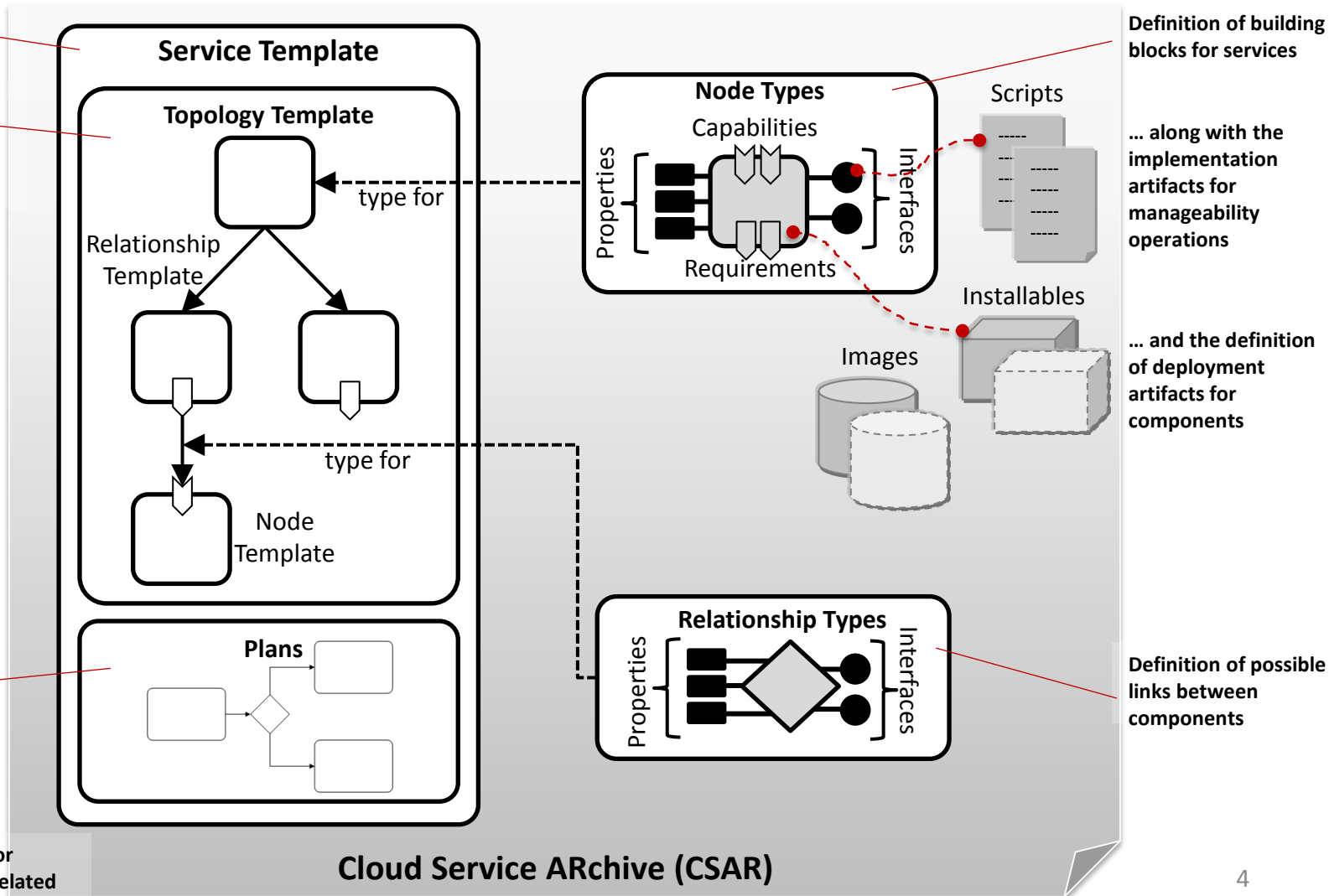
OASIS  **Topology and Orchestration Specification for Cloud Applications**

A language for defining Service Templates ...

... including a Topology Template describing the structure of a service

... including the definition of plans for orchestrating the application

Packaging format (CSAR) for packaging models and all related artifacts.

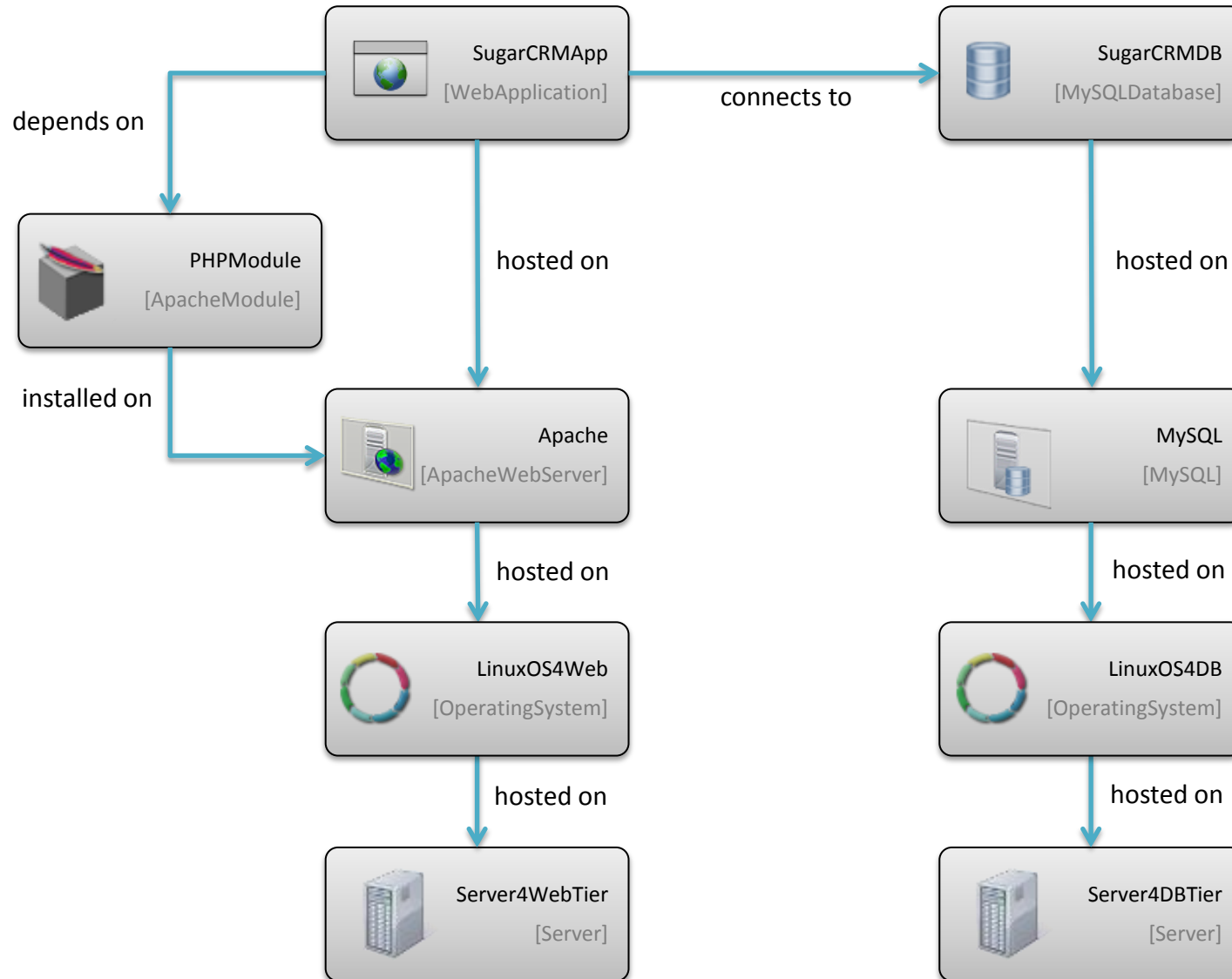


OASIS TOSCA TC Members

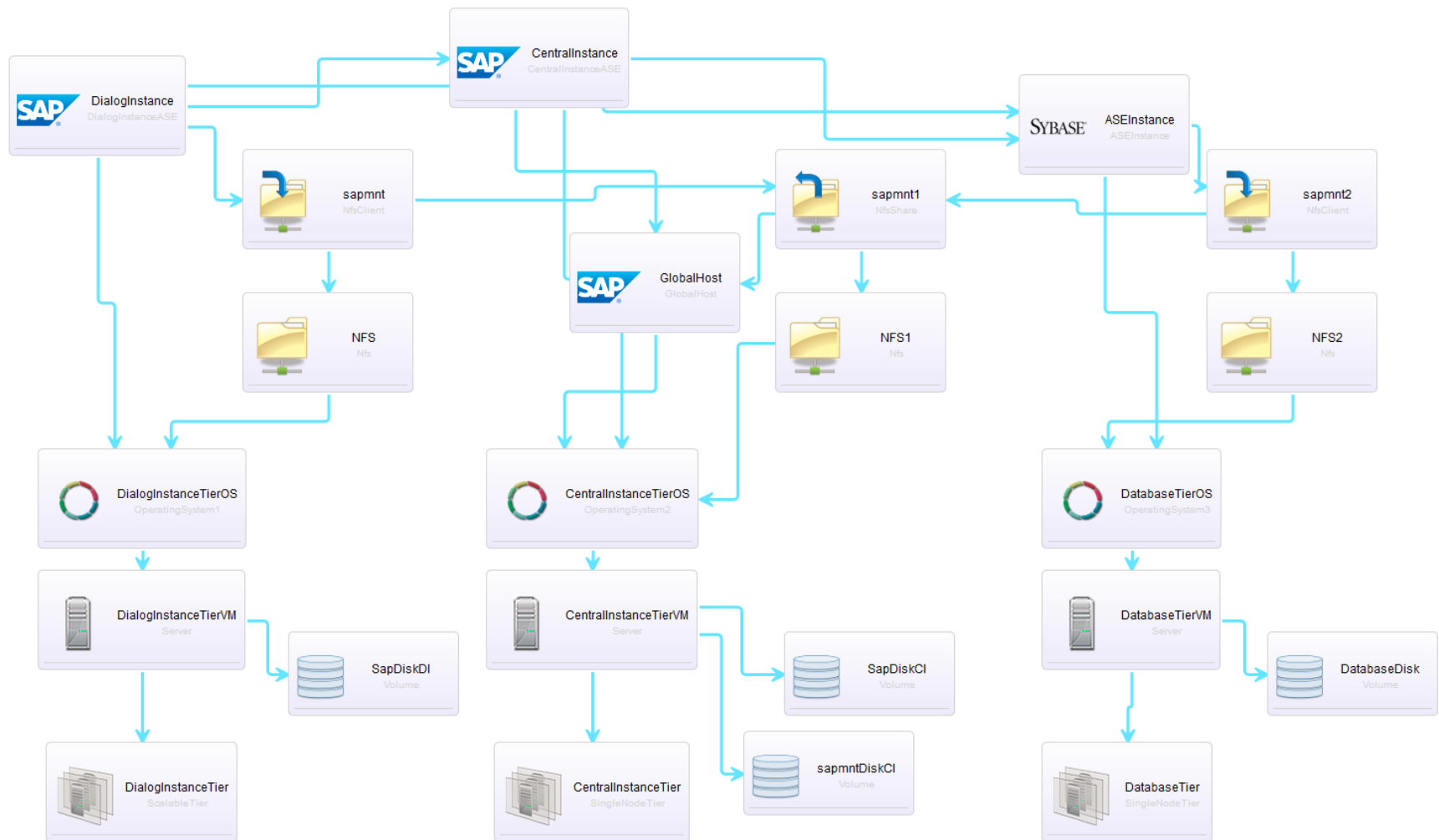
3M
ActiveState
Axway
CA Technologies
CenturyLink
Cisco
Citrix
Cloudsoft
EMC
Fujitsu
Google
HP
IBM
Huawei
Jericho Systems
NetApp
Nokia Siemens
Pricewaterhouse
Primeton
Red Hat SAP
Software AG
VCE
Vnomic
WSO2
Zenoss



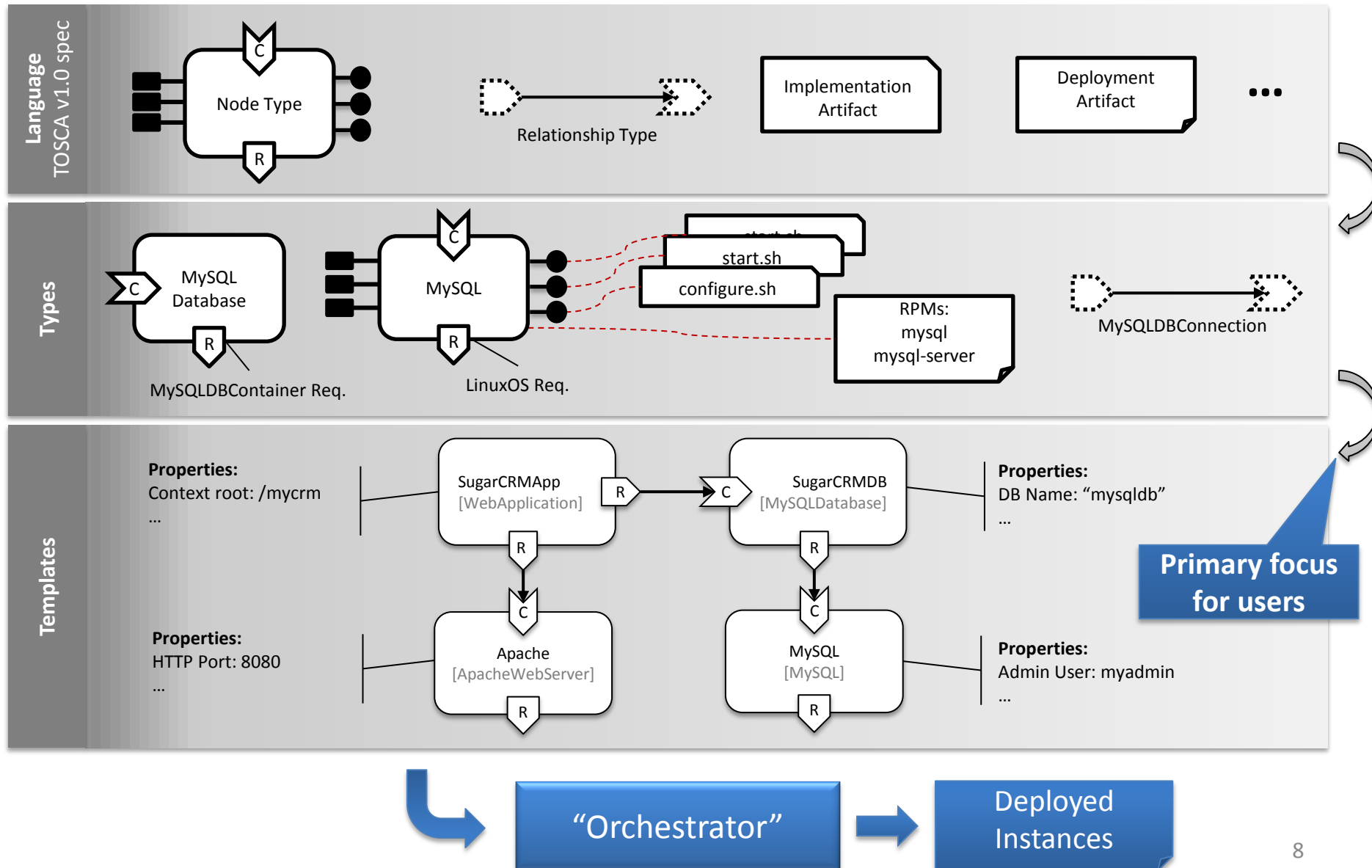
1st Interop Example: SugarCRM



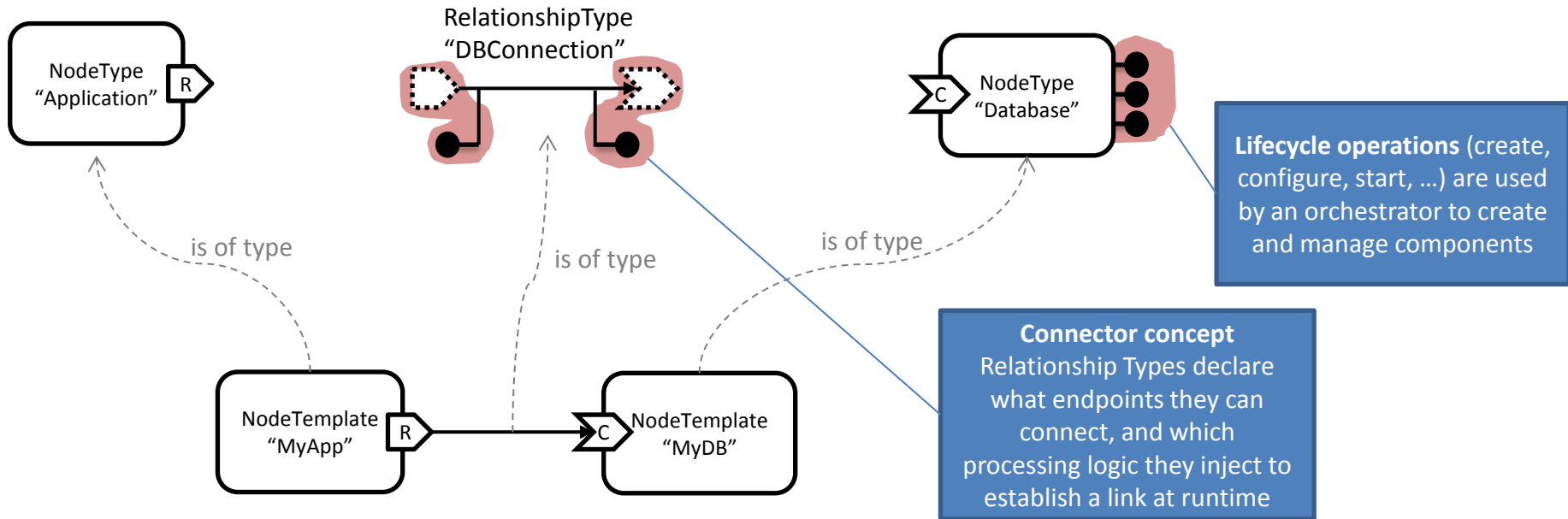
TOSCA for Enterprise Workloads: SAP CRM ASE Interop Example



Using TOSCA to model Applications



Role of Relationships in Model Processing



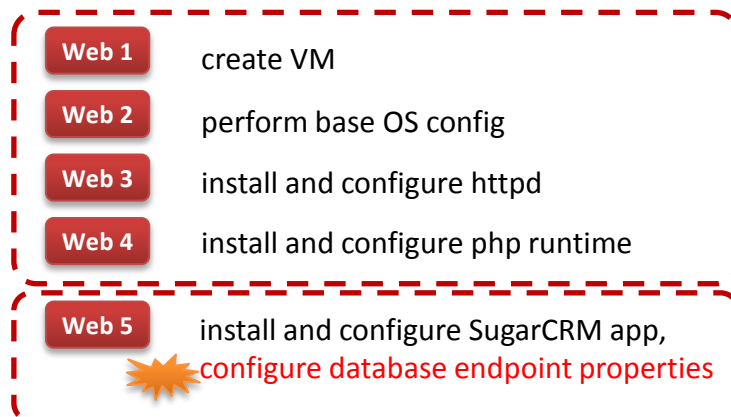
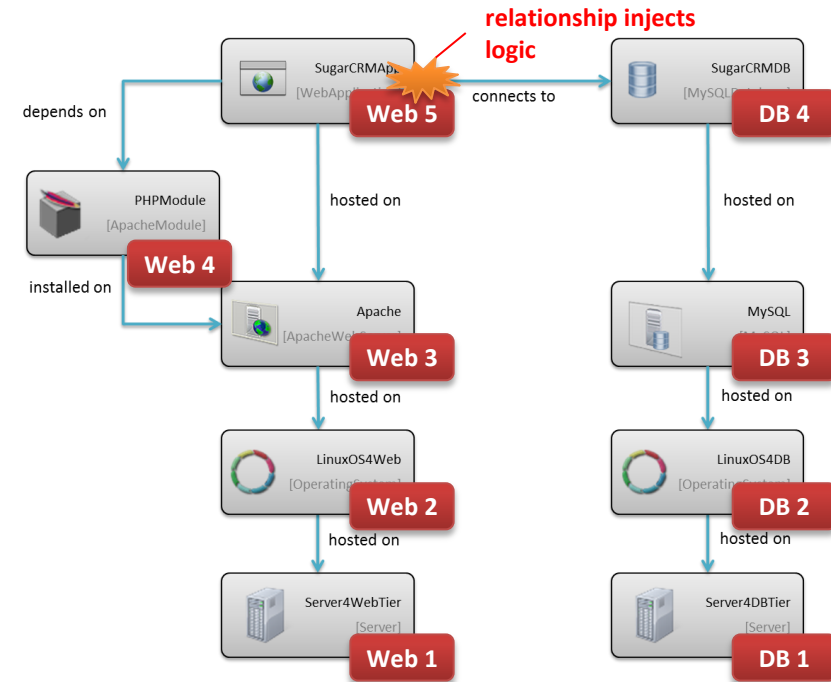
Base Relationship Types
HostedOn, DependsOn and ConnectsTo define the base semantics for processing topology models

- Use defined lifecycle operations to deploy and manage each component
- Use base relationship types to derive processing order
 - Process a host before a hosted component (HostedOn)
 - Process a provider before a client (DependsOn, ConnectsTo)
- If a component is related to another component, see if relationship injects additional processing logic (e.g. pre-configure endpoint)

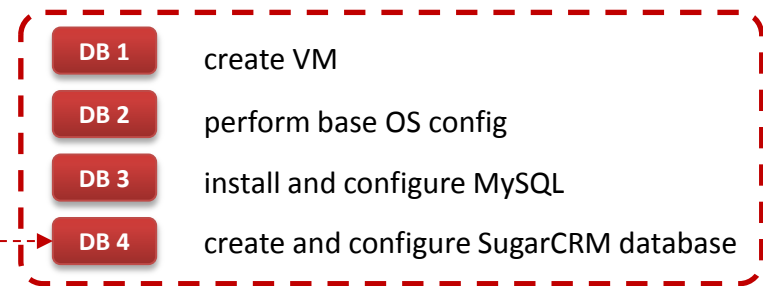
... a pretty mechanical process

How is a Topology processed?

- Use base relationship types to derive component processing order
 - First process a host, then process hosted component
 - First process a component that another component depends on, then process the dependent component
 - First process a component that another component connects to, then process the connecting component
- For each component
 - Deploy its Deployment Artifacts
 - Invoke lifecycle operations in right sequence (create, configure, start ...); their can be no-ops
- If a relationship contributes logic, inject it into component operation invocations

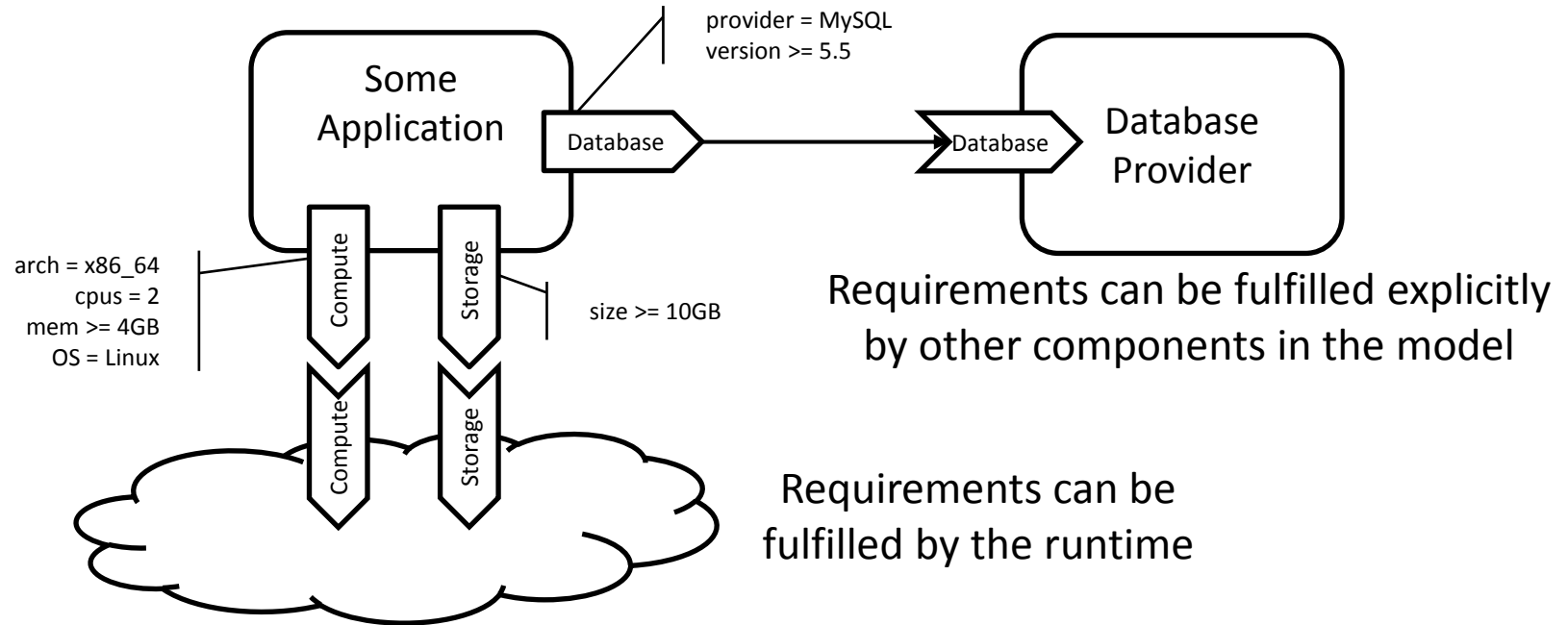


run in
parallel

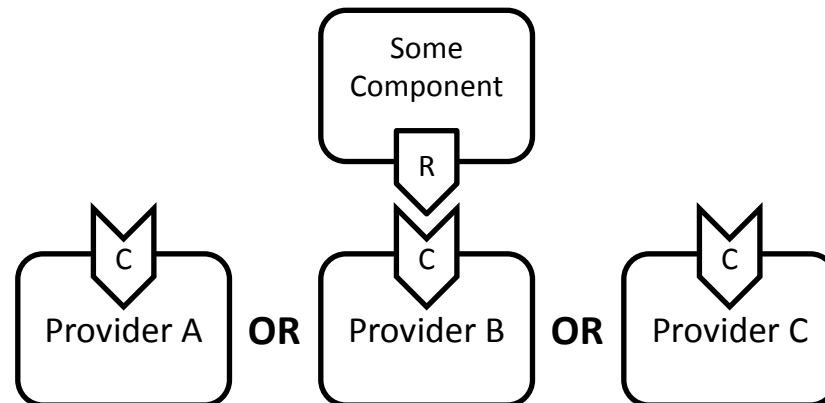


wait for step
"DB 4" to complete

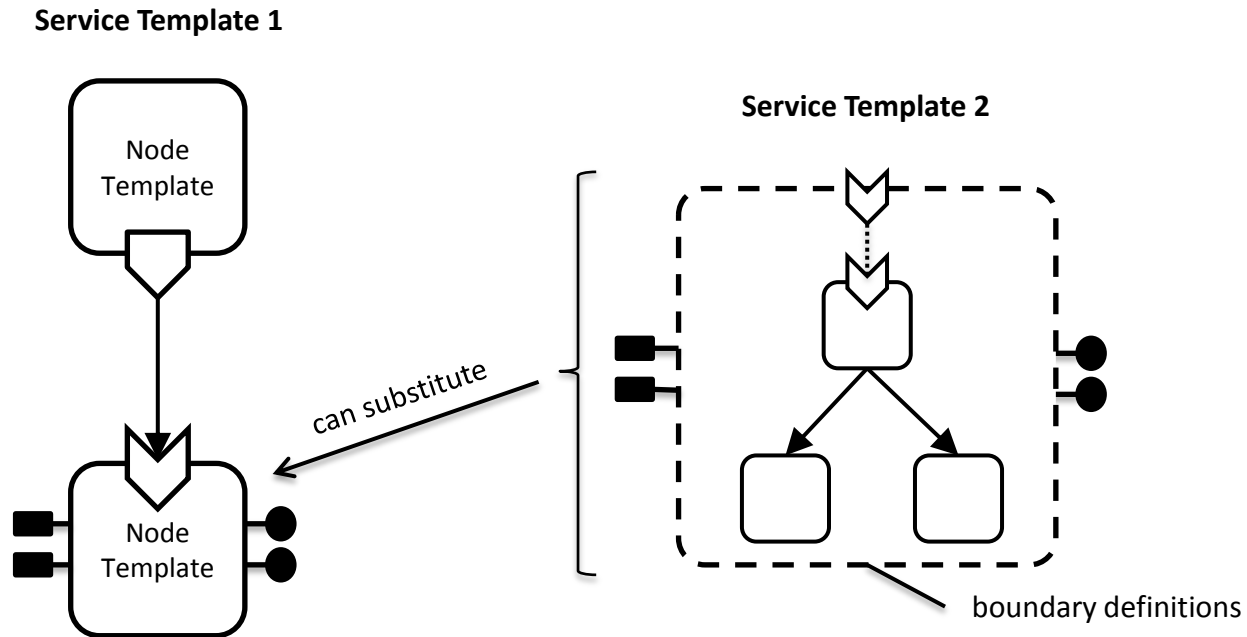
Requirements & Capabilities



**Requirements/Capabilities
are base for substitutability**



Model Composition



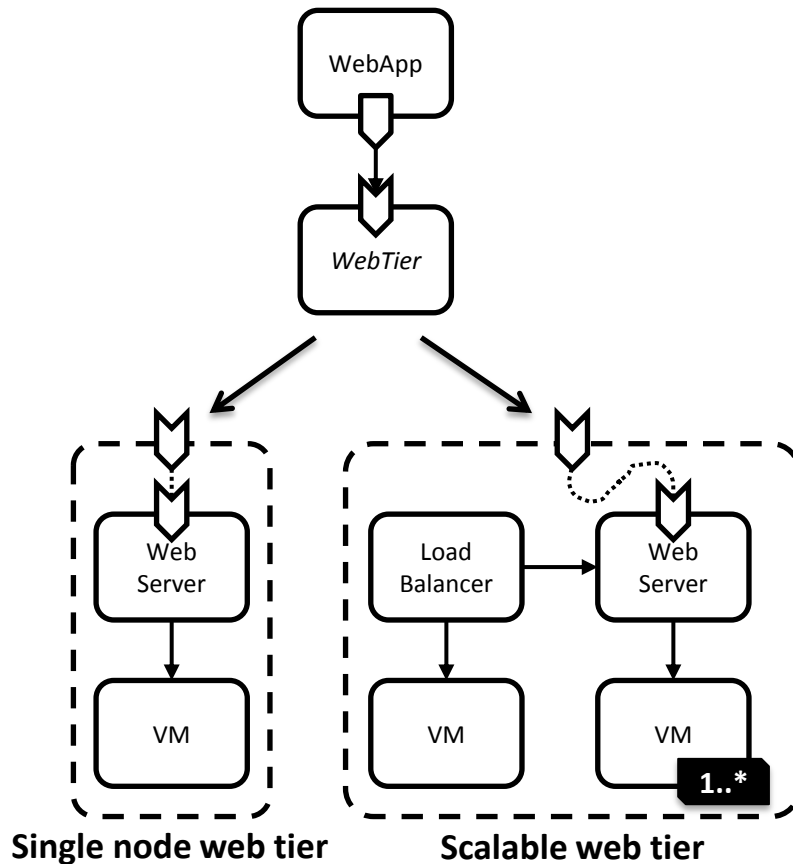
Subsystems can be abstracted in some models.

Other models can define details of subsystems.

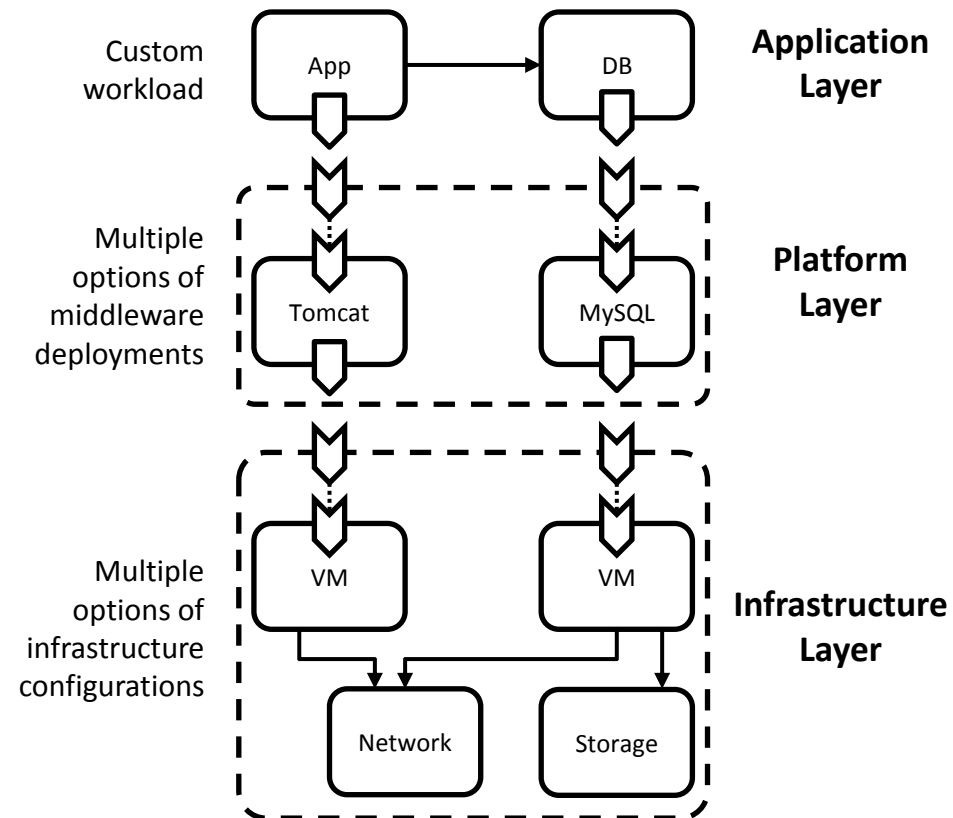
→ separation of concern, re-use

Usage Scenarios for Model Composition & Substitution

Varying deployment options



Layering of models



Learn more about TOSCA

- **TOSCA Specification, Version 1.0**
Committee Specification 02, 9 May 2013,
<http://docs.oasis-open.org/tosca/TOSCA/v1.0/cs02/TOSCA-v1.0-cs02.pdf>
- **TOSCA Primer, Version 1.0**
Committee Note Draft (CND) 01, Public Review Draft 01, 31 January 2013,
<http://docs.oasis-open.org/tosca/tosca-primer/v1.0/cnd01/tosca-primer-v1.0-cnd01.pdf>
- **TOSCA Implementer's Recommendations for Interoperable TOSCA Implementations, Version 1.0**
Interoperability Subcommittee, Working Draft 01, Rev. 02, 14 January 2013,
http://www.oasis-open.org/committees/document.php?document_id=47888&wg_abbrev=tosca-interop
- **TOSCA Interoperability Subcommittee, SugarCRM Scenario Sample CSAR**
Preliminary Draft CSAR for Interop. Testing against TOSCA v1.0 Specification,
http://www.oasis-open.org/committees/document.php?document_id=47585&wg_abbrev=tosca-interop

Thank you for your attention!