



... and how it could fit into  
OpenStack Heat

OpenStack Design Summit, April 15<sup>th</sup> 2013

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

# Agenda

- TOSCA Overview and Examples
- TOSCA Concepts Explained
- About Encoding ...
- Discussion: TOSCA and Heat

# What is TOSCA?

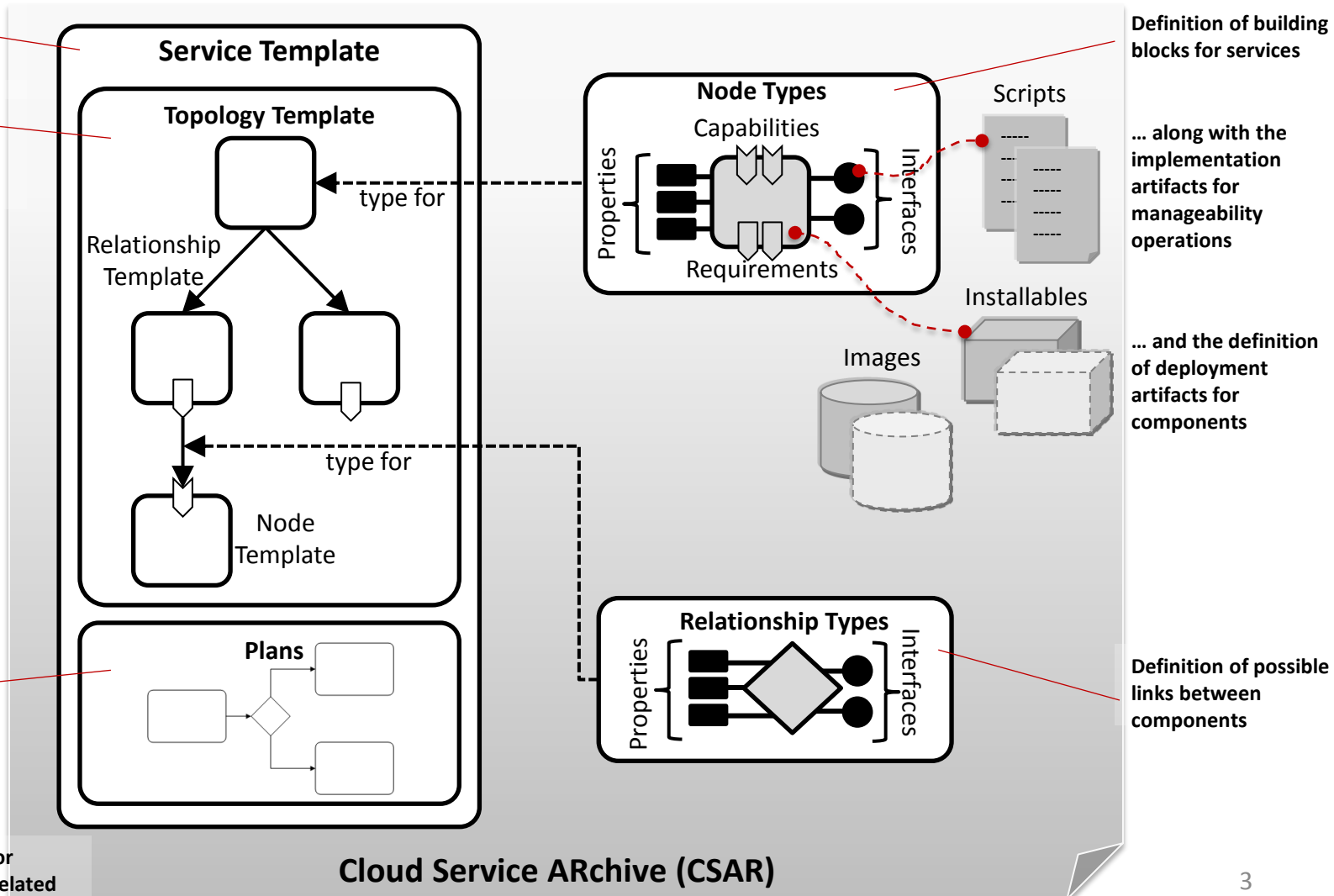
**OASIS**  **Topology and Orchestration Specification for Cloud Applications**

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.

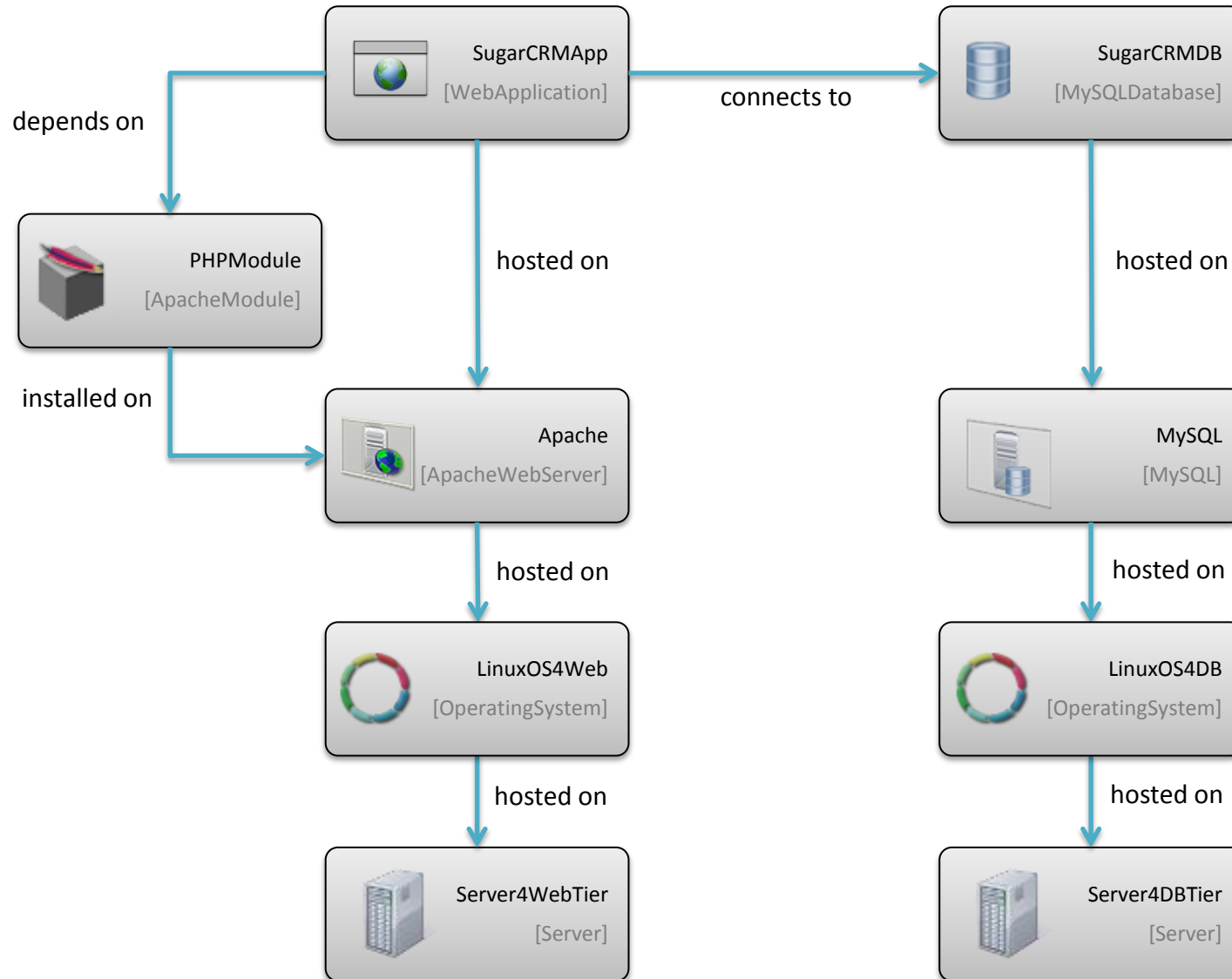


# OASIS TOSCA TC Members

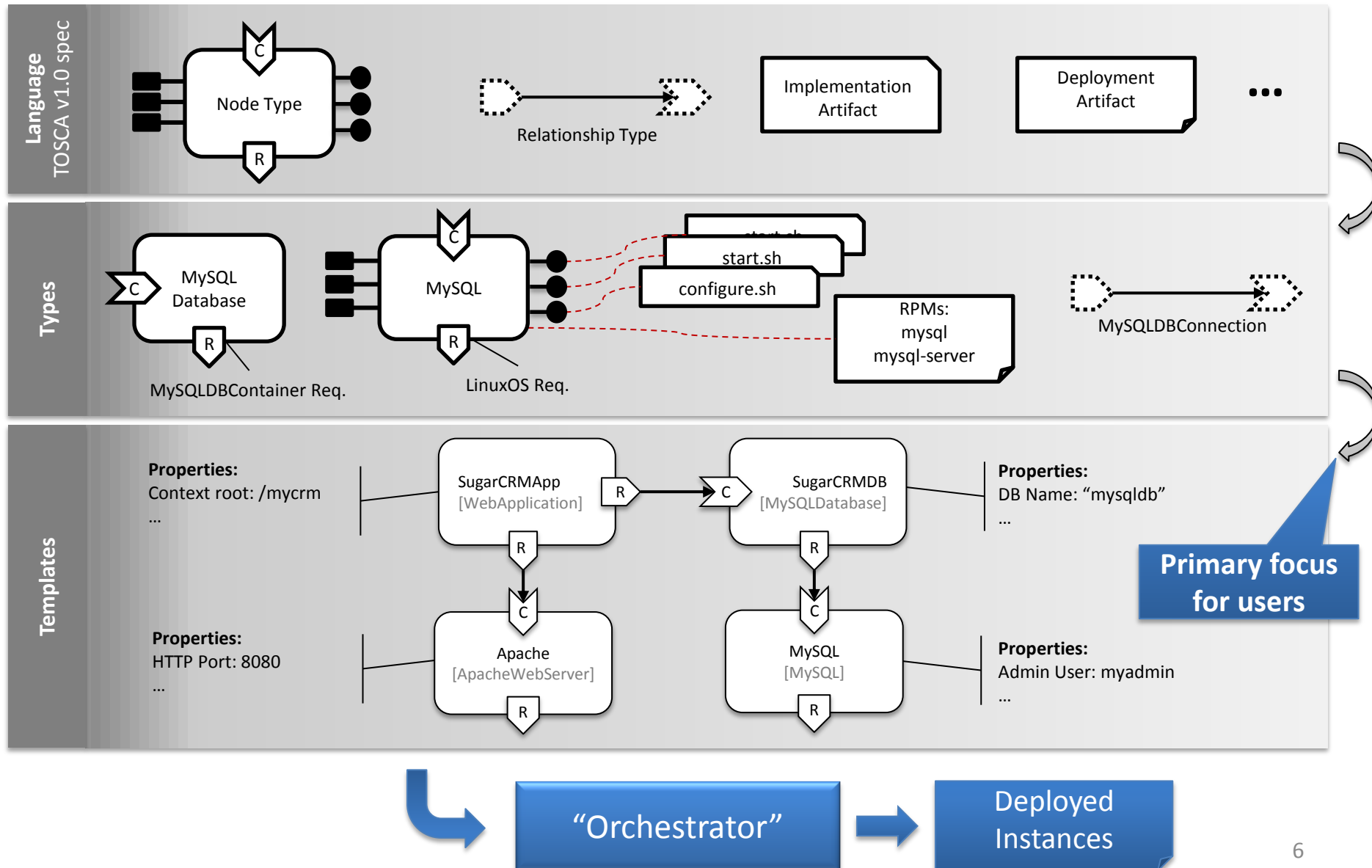
3M  
ActiveState  
Axway  
CA Technologies  
CenturyLink  
Cisco  
Citrix  
Cloudsoft  
EMC  
Fujitsu  
Google  
HP  
IBM  
Huawei  
Jericho Systems  
NetApp  
Nokia Siemens  
Pricewaterhouse  
Primeton  
Red Hat SAP  
Software AG  
VCE  
Vnomic  
WSO2  
Zenoss



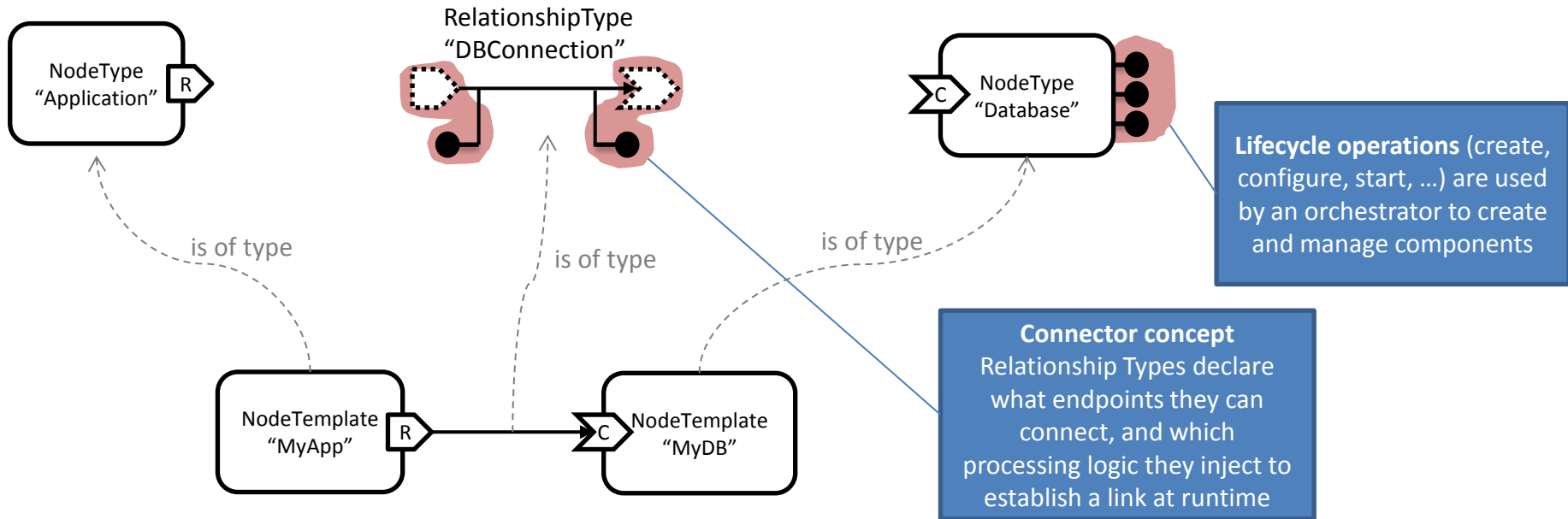
# 1<sup>st</sup> Interop Example: SugarCRM



# Using TOSCA to model Applications



# Role of Relationships in Model Processing



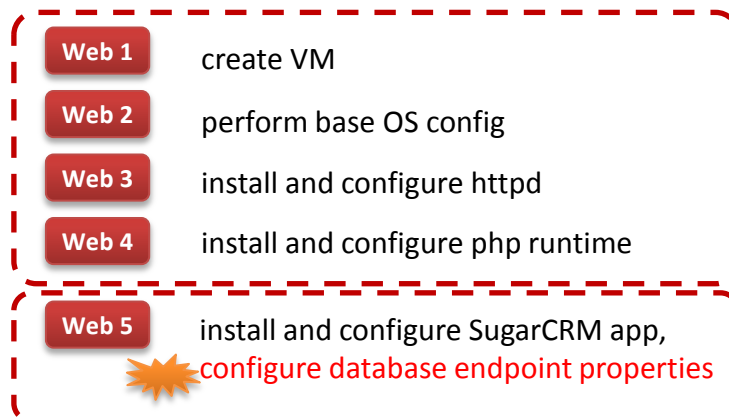
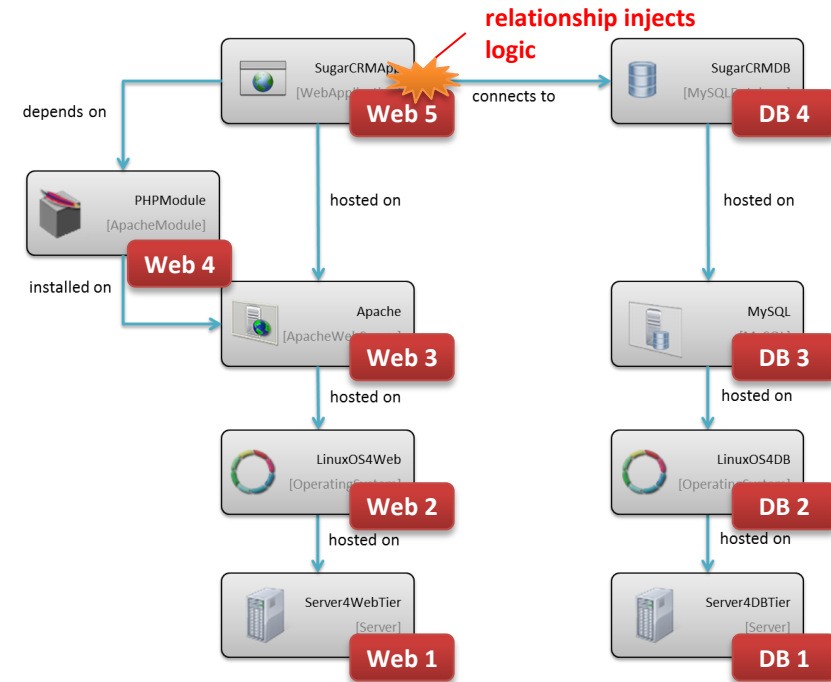
**Base Relationship Types**  
HostedOn, DependsOn and ConnectsTo define the base semantics for processing topology models

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

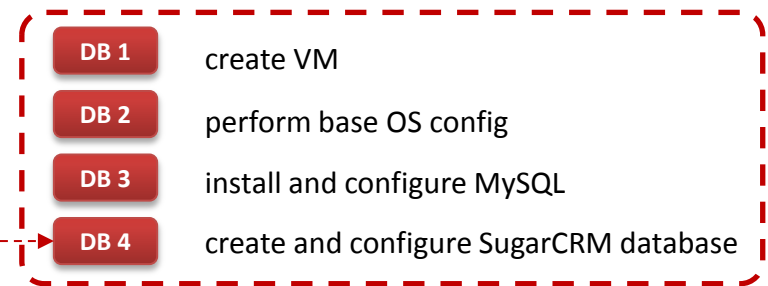
... a pretty mechanical process

# 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



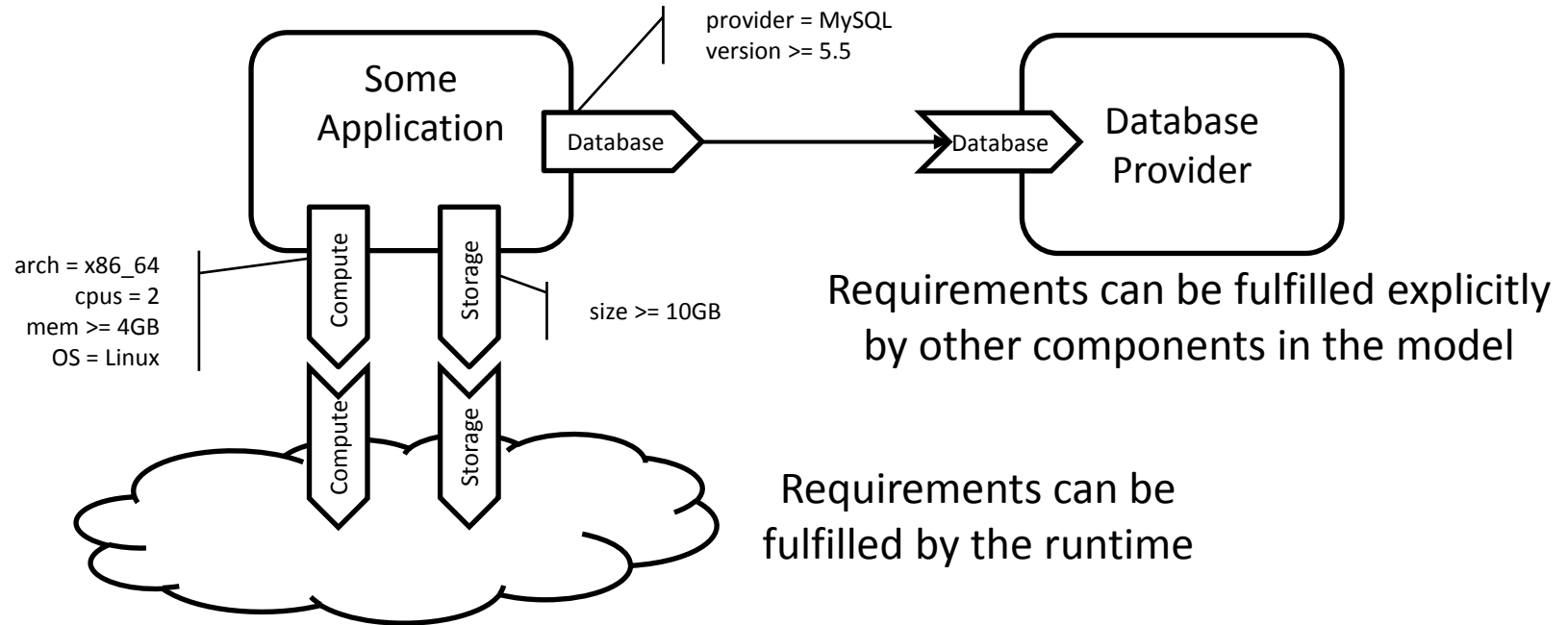
run in  
parallel



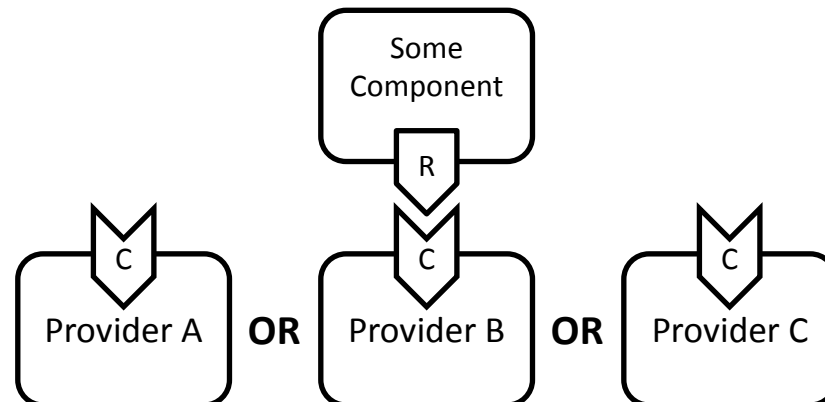
wait for step  
"DB 4" to complete



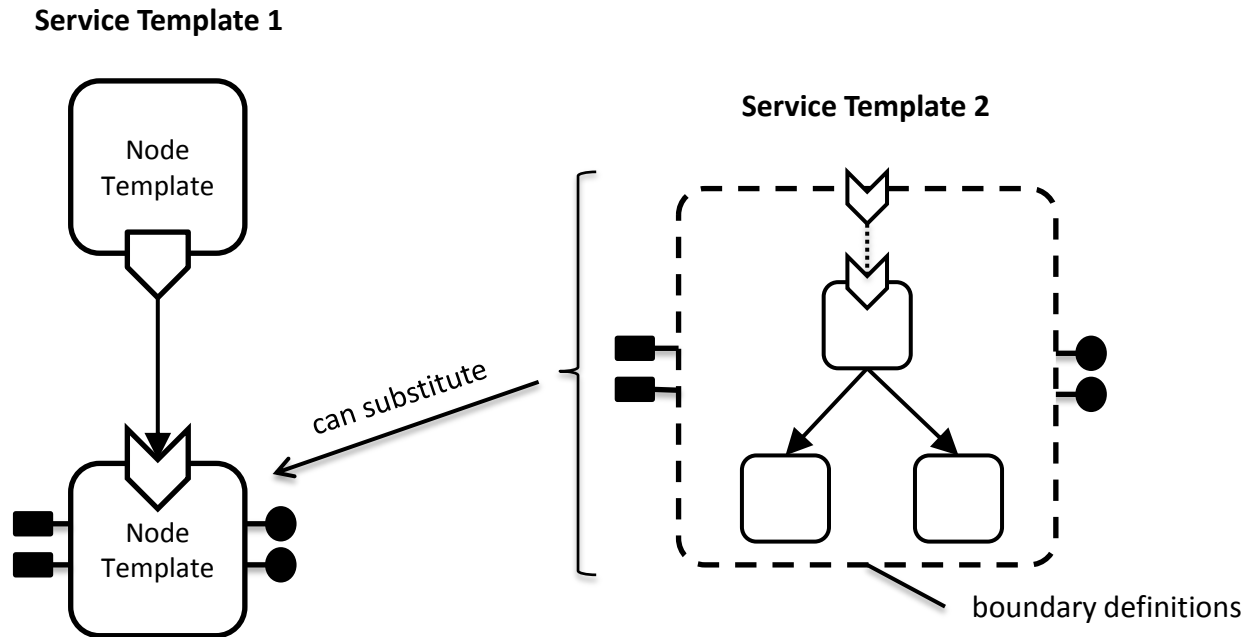
# Requirements & Capabilities



**Requirements/Capabilities  
are base for substitutability**



# Model Composition



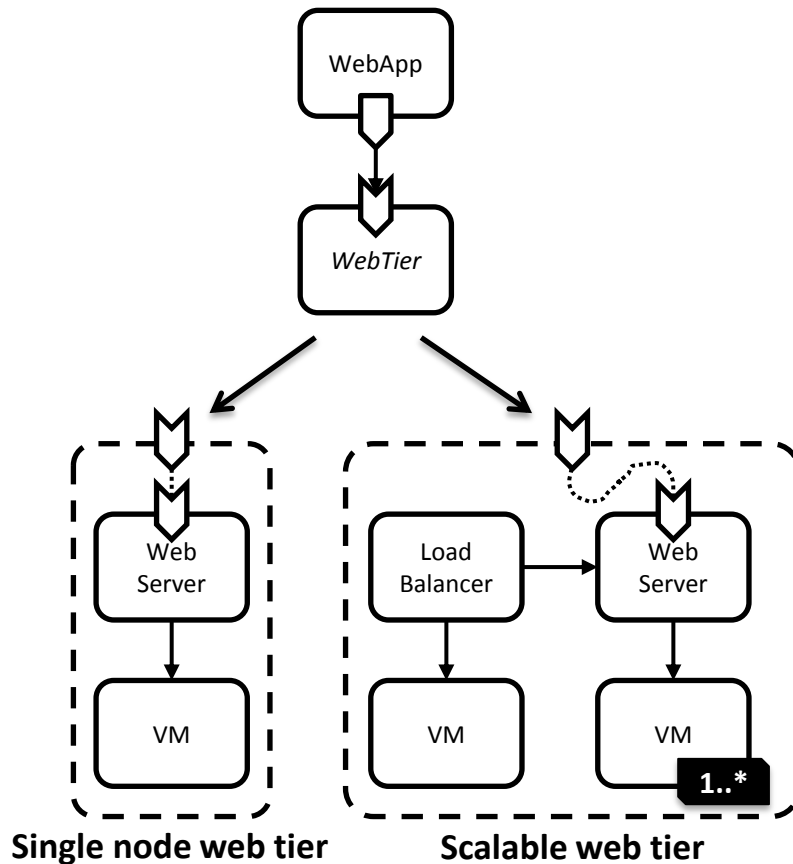
Subsystems can be abstracted in some models.

Other models can define details of subsystems.

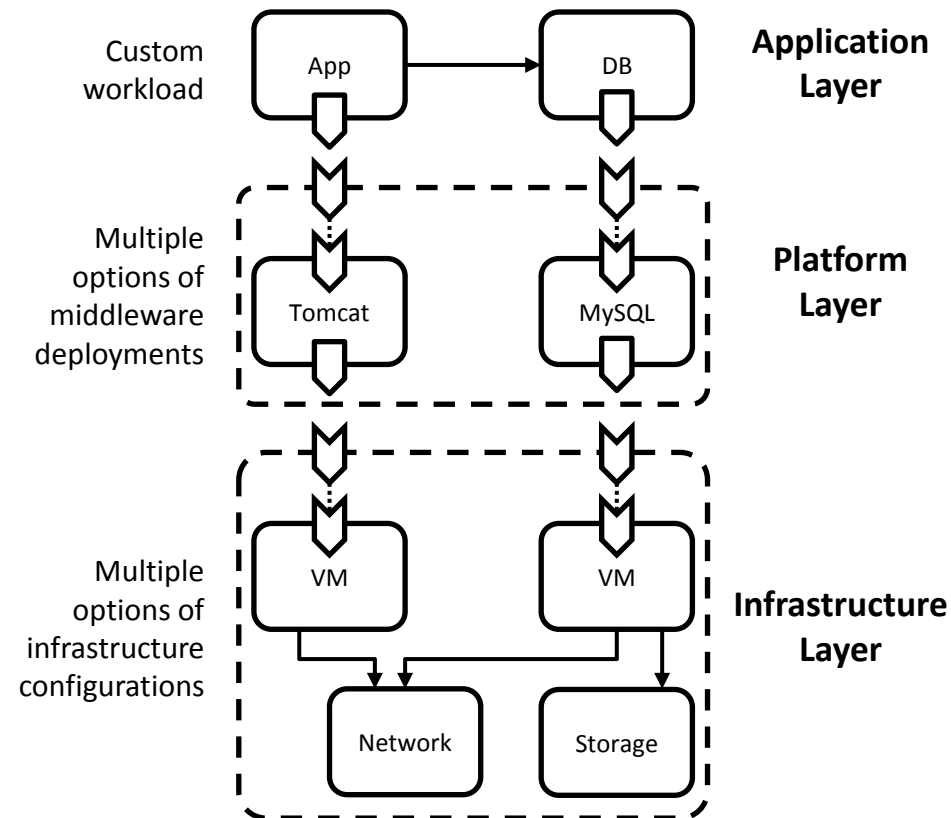
→ separation of concern, re-use

# Usage Scenarios for Model Composition & Substitution

## Varying deployment options



## Layering of models



# About Encoding ...



The TOSCA TC decided to use XML and XML Schema as the normative way for defining the TOSCA language.

**But: The important thing about TOSCA are the concepts, not the encoding.**

Alternative encodings (e.g. JSON) can be defined as part of TOSCA vNext work.

# Alternative Encoding Example

```
<NodeType name="Server">
  <PropertiesDefinition element="tbase:ServerProperties"/>
  ...
</NodeType>
```

```
<xs:element name="ServerProperties" type="tServerProperties"/>
<xs:complexType name="tServerProperties">
  <xs:sequence>
    <xs:element name="NumCpus" default="1">
      <xs:simpleType>
        <xs:restriction base="xs:int">
          <xs:enumeration value="1"/>
          <xs:enumeration value="2"/>
          <xs:enumeration value="4"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Memory" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```



```
"NodeType" : {
  "name" : "Server",
  "properties" : [
    {
      "name" : "NumCpus",
      "type" : "integer",
      "options" : [1, 2, 4]
    },
    {
      "name" : "Memory",
      "type" : "integer",
    }
  ]
  ...
}
```

```
<TopologyTemplate>
  <NodeTemplate id="MyServer" type="tbase:Server">
    <Properties>
      <tbase:ServerProperties>
        <NumCpus>2</NumCpus>
        <Memory>4096</Memory>
      </tbase:ServerProperties>
    </Properties>
    ...
  </NodeTemplate>
  ...
</TopologyTemplate>
```

```
"TopologyTemplate" : {
  "NodeTemplates" : [
    {
      "id" : "MyServer",
      "type" : "Server",
      "properties" : [
        "NumCpus" : 2,
        "Memory" : 4096
      ]
    },
    ...
  ],
  ...
}
```

# Now, how could it fit into Heat?

- It's all about concepts: as long as a pattern engine's concepts and TOSCA are aligned, mapping to TOSCA as an external format is straight forward
  - No need to use TOSCA as Heat's core format
- Define a sub-set profile of TOSCA that is appropriate for use cases targeted by Heat
- Define an alternative JSON rendering for TOSCA sub-set profile
- Define a set of base Node- and Relationship Types for core OpenStack resources: Compute (Nova), Network (Quantum), Block Storage (Cinder), Object Storage (Swift)
  - As natively supported types
  - Users do not have to care about defining TOSCA types, but can just start defining templates
- Use implementation to improve and refine the TOSCA standard; use standardized concepts to shape implementation

# Learn more about TOSCA

- **TOSCA Specification, Version 1.0**  
*Committee Specification 01, 18 March November 2013,*  
<http://docs.oasis-open.org/tosca/TOSCA/v1.0/cs01/TOSCA-v1.0-cs01.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](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](http://www.oasis-open.org/committees/document.php?document_id=47585&wg_abbrev=tosca-interop)