BPaaS Adaptive Provisioning via CAMEL

K. Kritikos[1], F. Griesinger[2], D. Seybold[2] , J. Domaschka[2]

1: ICS-FORTH, Greece

2: University of Ulm, Germany

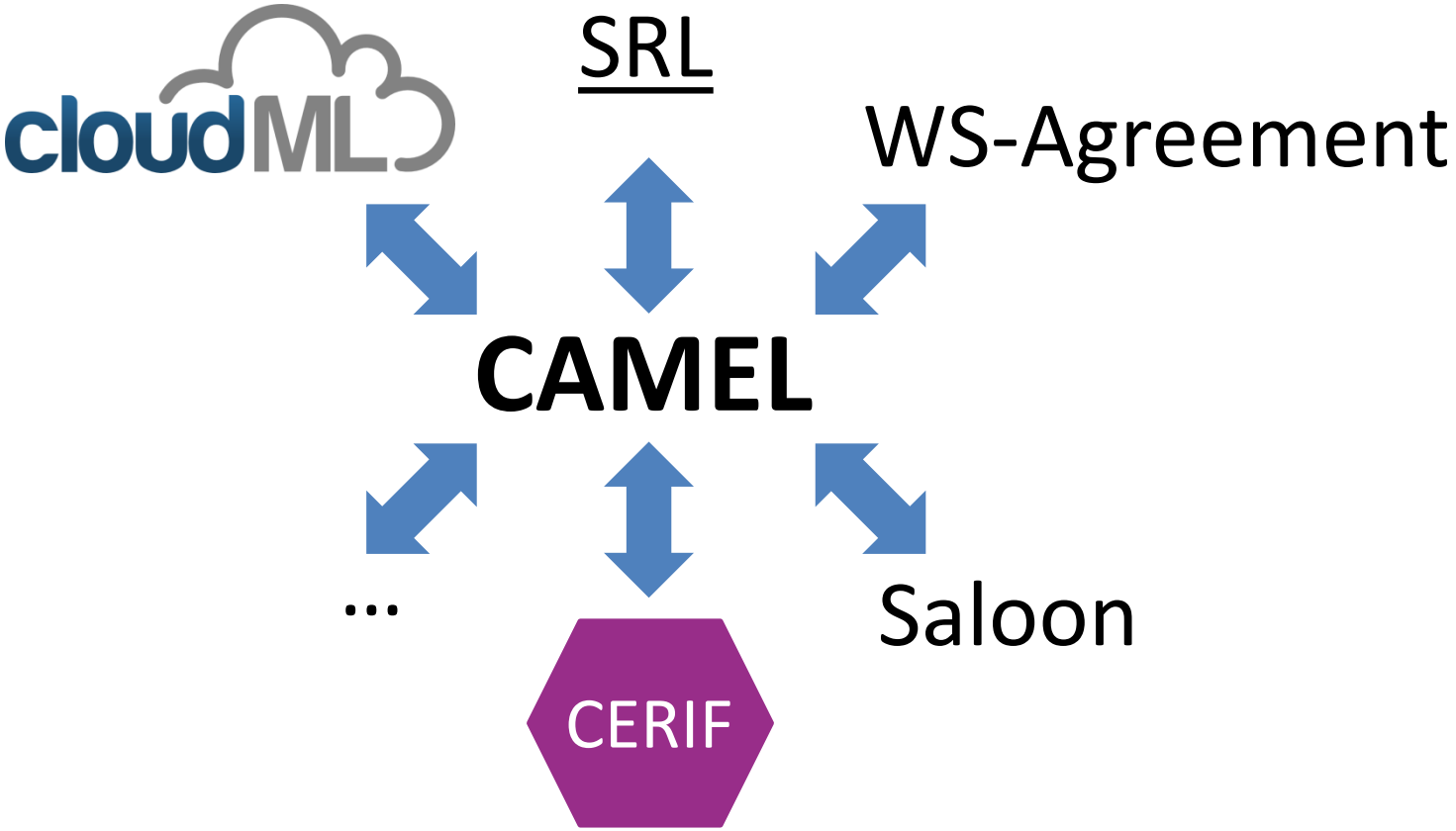
CloudSocket

Webinar Agenda

- CAMEL Features
- CAMEL Usage in CloudSocket
- Tool Support
- Conceptualisation
- Send Invoice Use Case
- Summary
- Questions and Answers

CAMEL Features

- Multi-purpose Multi-DSL
  - Integration of all DSLs into coherent whole
- Coverage of all relevant aspects: requirements, deployment, provider, organisation, security, metrics, adaptation, units & types
- Re-use existing languages (either developed in PaaSage project or standards)
- Cater for models@runtime support
  - Type-instance pattern
- Syntactic & semantic model validation

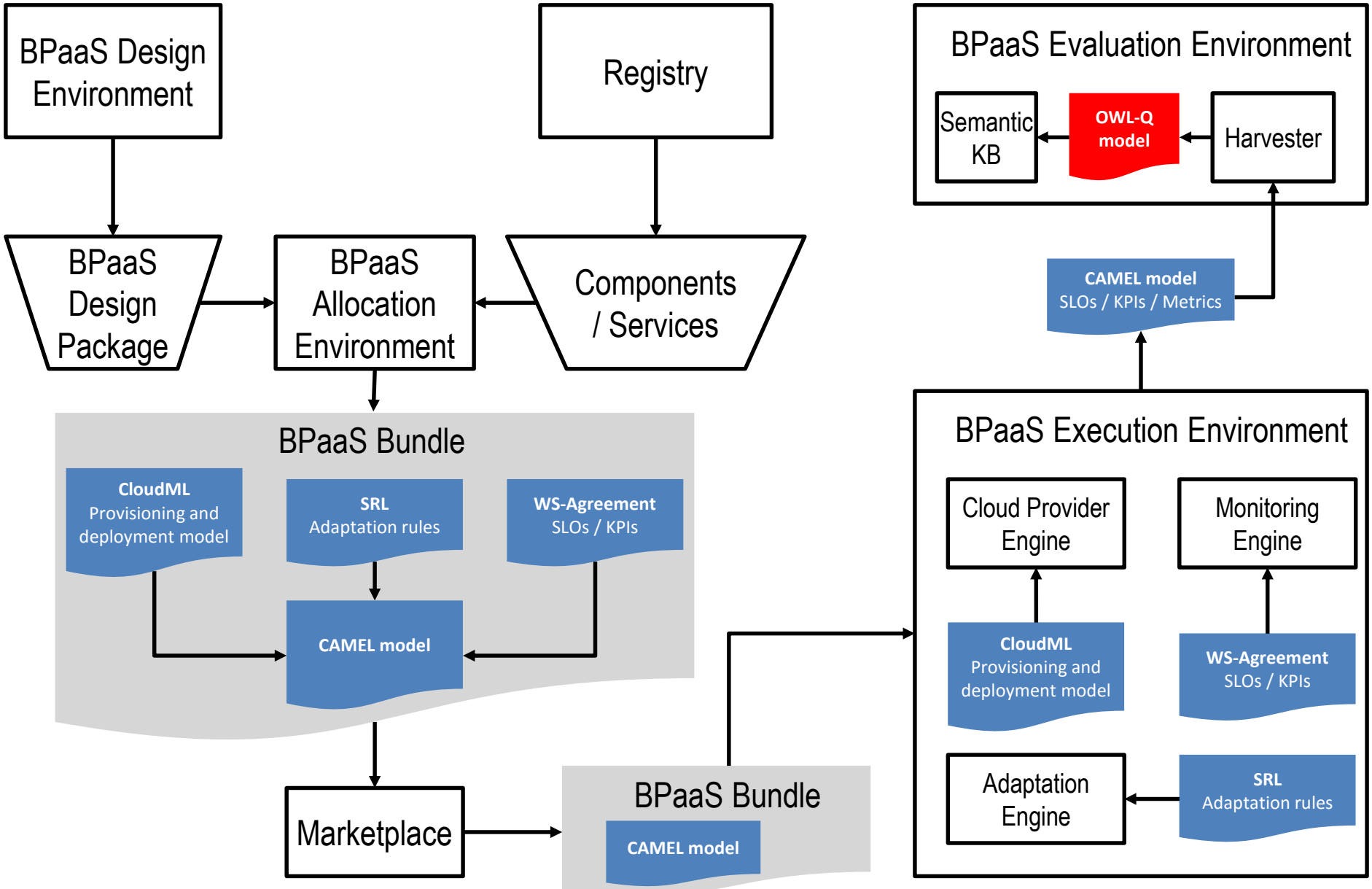


CloudSocket

CAMEL Models

Name	Coverage	Editor / Modeller	Design / Runtime
Camel	Top model, container of other models, application	Broker, System	Both
Deployment	Application topology	Broker, System	Both
Requirement	HW, PaaS, security, location, OS, provider, QoS requirements	Broker	Design
Metric	Metric, Sensors, Attributes, Conditions	Broker, System	Both
Adaptation	Adaptation rules, (composite) events, adaptation actions	Broker	Design
Location	Physical & cloud locations	Broker	Design
Unit	Units of measurement	Broker	Design
Types	Types & values	Broker	Design
Organisation	Organisation, users, cloud credentials, policies	Admin	Pre-Design
Provider	Provider offerings	Admin	Both*
Execution	Execution context, measurements, SLO assessments, adaptation actions	System	Runtime

CAMEL Usage in CloudSocket



CloudSocket

Technical Underpinning

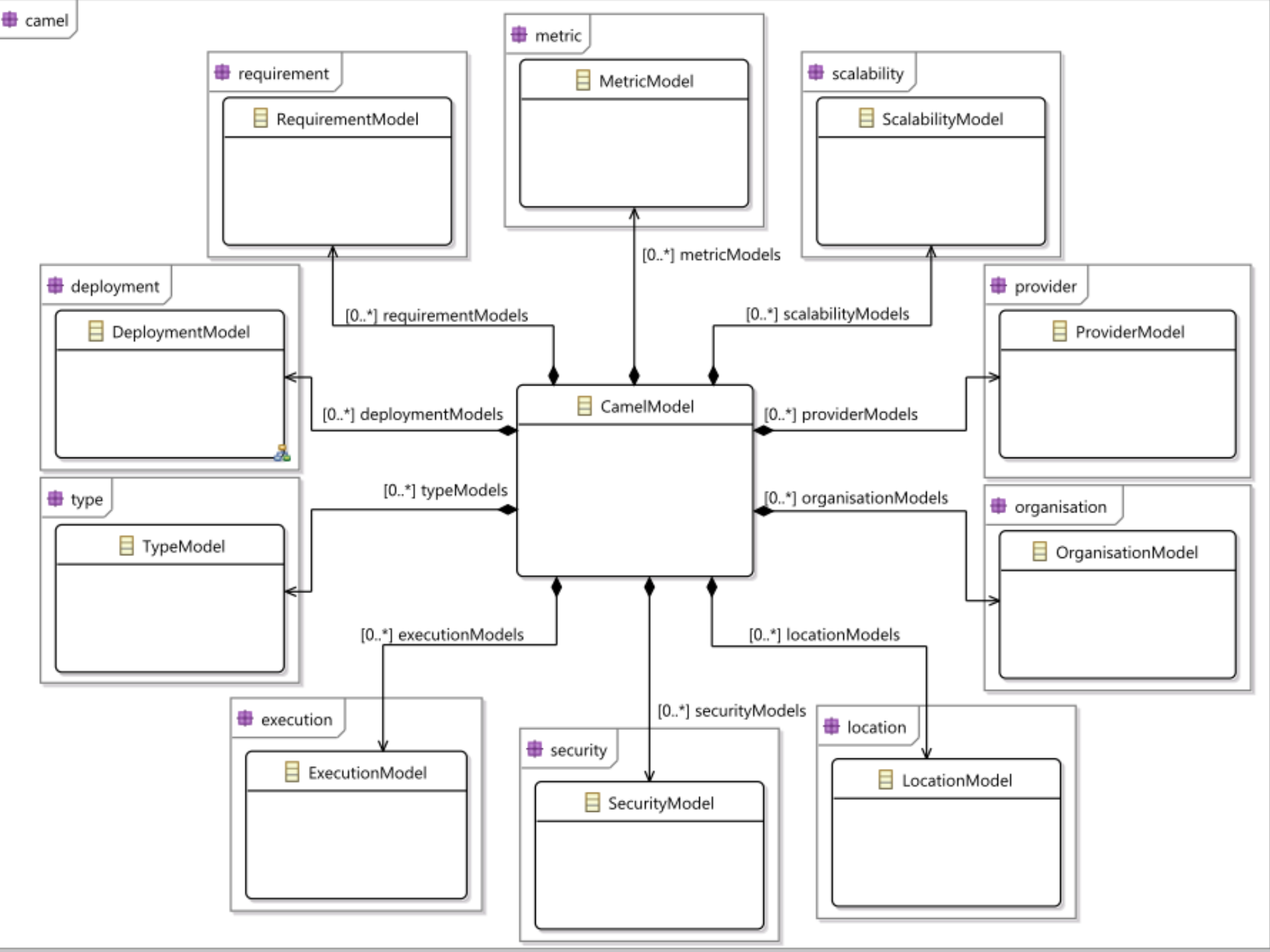
CloudSocket

Tool Support

Editor	Model Valid.	Aspects	Repos. Integr.	Control. Access	Format	Roles	Notes
Tree	✓	All	✓	~	XMI	any	Default
Textual	✓	All	-	-	Textual (ed.), XMI	devops	Xtext-based
Graphical (PaaSage)	✓	Req. & Org.	✓	✓	Both	any	Eclipse RAP
Graphical (Allocat. Environ.)	✓	Req., Metric, Depl.,	*	✓	Both	devops	Livebase, GWT, OSGI

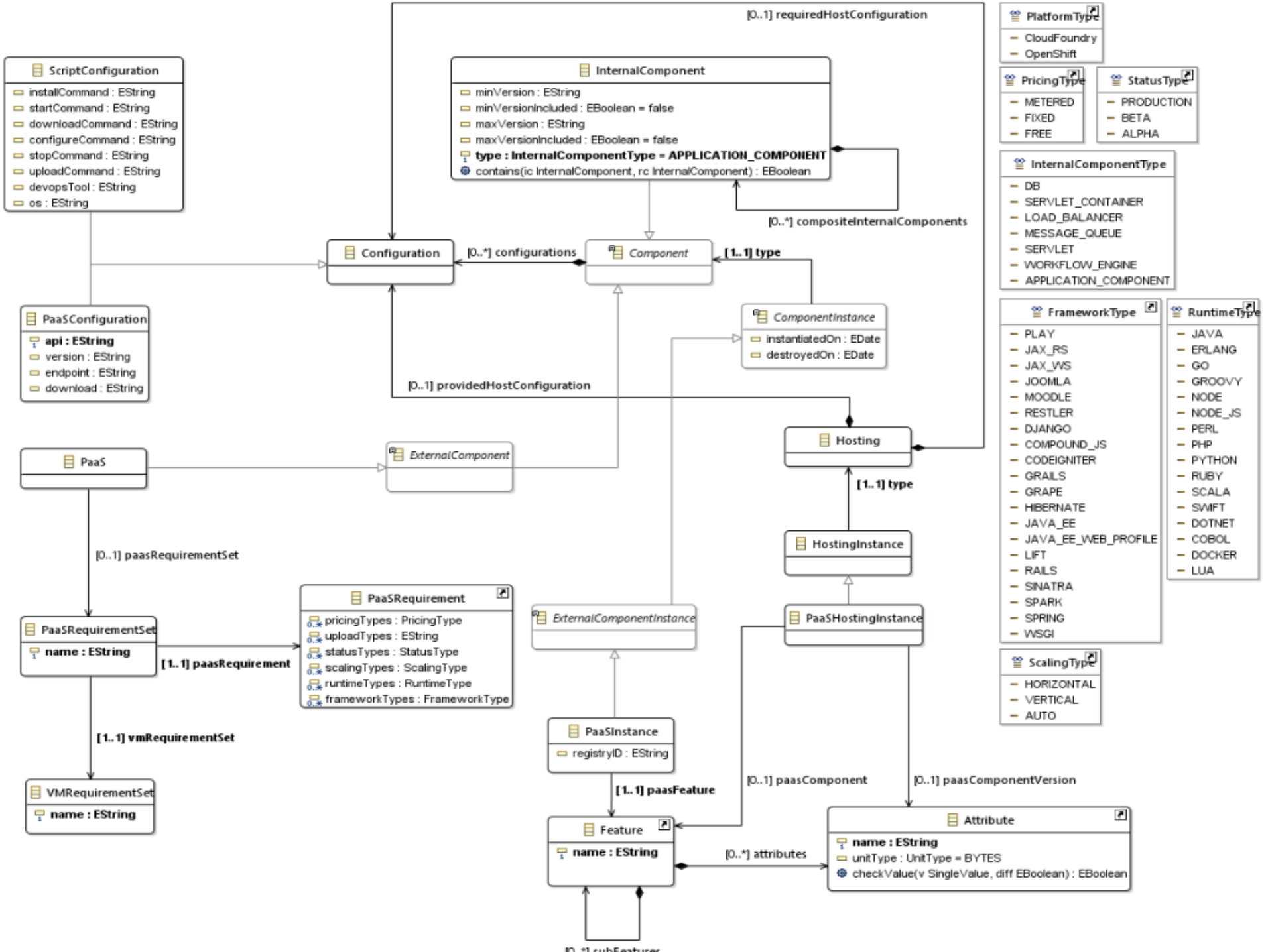
Conceptualisation – CAMEL Model

- Top Model
- Overall container of sub-models focusing on different aspects
- Cloud Application / BPaaS high-level details:
  - Name, version, owner, reference to deployment models


camel
metric
MetricModel
requirement
RequirementModel
scalability
ScalabilityModel
[0..*] metricModels
deployment
DeploymentModel
[0..*] requirementModels
[0..*] scalabilityModels
provider
ProviderModel
[0..*] deploymentModels
CamelModel
[0..*] providerModels
type
TypeModel
[0..*] typeModels
[0..*] organisationModels
organisation
OrganisationModel
[0..*] executionModels
[0..*] locationModels
execution
ExecutionModel
[0..*] securityModels
security
SecurityModel
location
LocationModel

Conceptualisation – Deployment Model

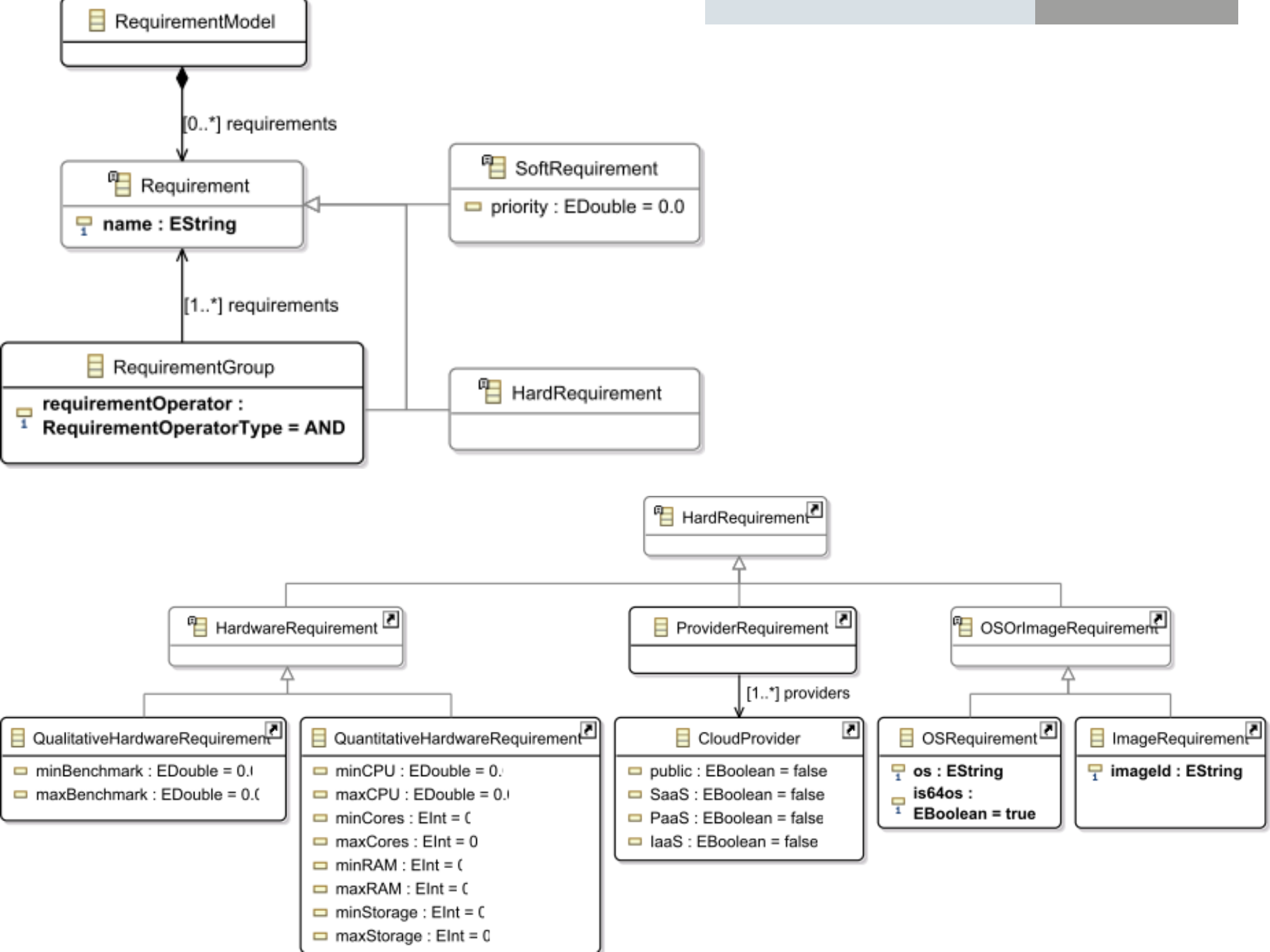
- Rationale:
  - Specify most information needed for deployment reasoning & execution
- Covers:
  - Application Topology
    - Components & their configuration
    - VM nodes & requirements on them
    - **PaaS nodes & requirements on them -> CloudSocket extension to CAMEL**
    - **SaaS nodes & requirements on them -> CloudSocket extension to CAMEL**
    - Hosting (component-to-VM & component-to-PaaS)
    - Communication (component-to-component)

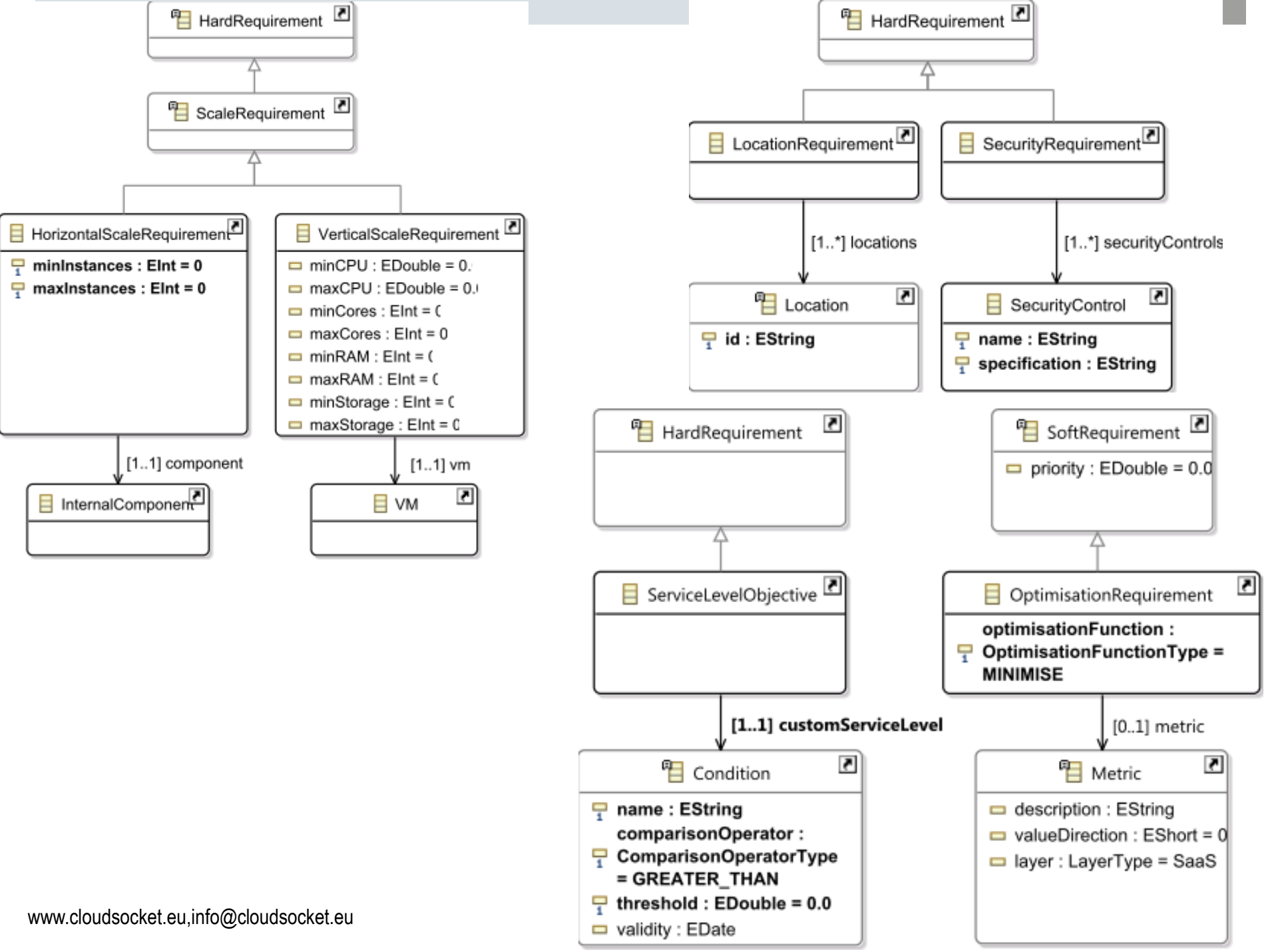

[0..1] requiredHostConfiguration
ScriptConfiguration
installCommand : EString
startCommand : EString
downloadCommand : EString
configureCommand : EString
stopCommand : EString
uploadCommand : EString
devopsTool : EString
os : EString
InternalComponent
minVersion : EString
minVersionIncluded : EBoolean = false
maxVersion : EString
maxVersionIncluded : EBoolean = false
type : InternalComponentType = APPLICATION_COMPONENT
contains(ic InternalComponent, rc InternalComponent) : EBoolean
[0..*] compositeInternalComponents
PlatformType
CloudFoundry
OpenShift
PricingType
METERED
FIXED
FREE
StatusType
PRODUCTION
BETA
ALPHA
InternalComponentType
DB
SERVLET_CONTAINER
LOAD_BALANCER
MESSAGE_QUEUE
SERVLET
WORKFLOW_ENGINE
APPLICATION_COMPONENT
Configuration
[0..*] configurations
Component
[1..1] type
PaaSConfiguration
api : EString
version : EString
endpoint : EString
download : EString
ComponentInstance
instantiatedOn : EDate
destroyedOn : EDate
FrameworkType
PLAY
JAX_RS
JAX_WS
JOOMLA
MOODLE
RESTLER
DJANGO
COMPOUND_JS
CODEIGNITER
GRAILS
GRAPE
HIBERNATE
JAVA_EE
JAVA_EE_WEB_PROFILE
LIFT
RAILS
SINATRA
SPARK
SPRING
WSGI
RuntimeType
JAVA
ERLANG
GO
GROOVY
NODE
NODE_JS
PERL
PHP
PYTHON
RUBY
SCALA
SWIFT
DOTNET
COBOL
DOCKER
LUA
[0..1] providedHostConfiguration
PaaS
ExternalComponent
Hosting
[1..1] type
HostingInstance
[0..1] paasRequirementSet
PaaSRequirement
pricingTypes : PricingType
uploadTypes : EString
statusTypes : StatusType
scalingTypes : ScalingType
runtimeTypes : RuntimeType
frameworkTypes : FrameworkType
ExternalComponentInstance
PaaSHostingInstance
PaaSRequirementSet
name : EString
[1..1] paasRequirement
ScalingType
HORIZONTAL
VERTICAL
AUTO
[1..1] vmRequirementSet
PaaSInstance
registryID : EString
[0..1] paasComponent
[0..1] paasComponentVersion
[1..1] paasFeature
VMRequirementSet
name : EString
Feature
name : EString
[0..*] attributes
Attribute
name : EString
unitType : UnitType = BYTES
checkValue(v SingleValue, diff EBoolean) : EBoolean
[0..*] subFeatures

Conceptualisation – Requirement Model

- Rationale: specify all kinds of user requirements
- Logical combination of requirements
  - Hard requirements over
    - HW (ranges over VM characteristics)
    - **PaaS (alternative options over PaaS characteristics) -> CloudSocket extension to CAMEL**
    - Location (set of allowed locations)
    - OS (set of allowed OS) / Image (set of allowed images)
    - Provider (set of allowed providers)
    - Security (set of required security controls)
    - QoS (SLOs on QoS properties / metrics)
  - Soft requirements (min/max) over
    - QoS properties / metrics

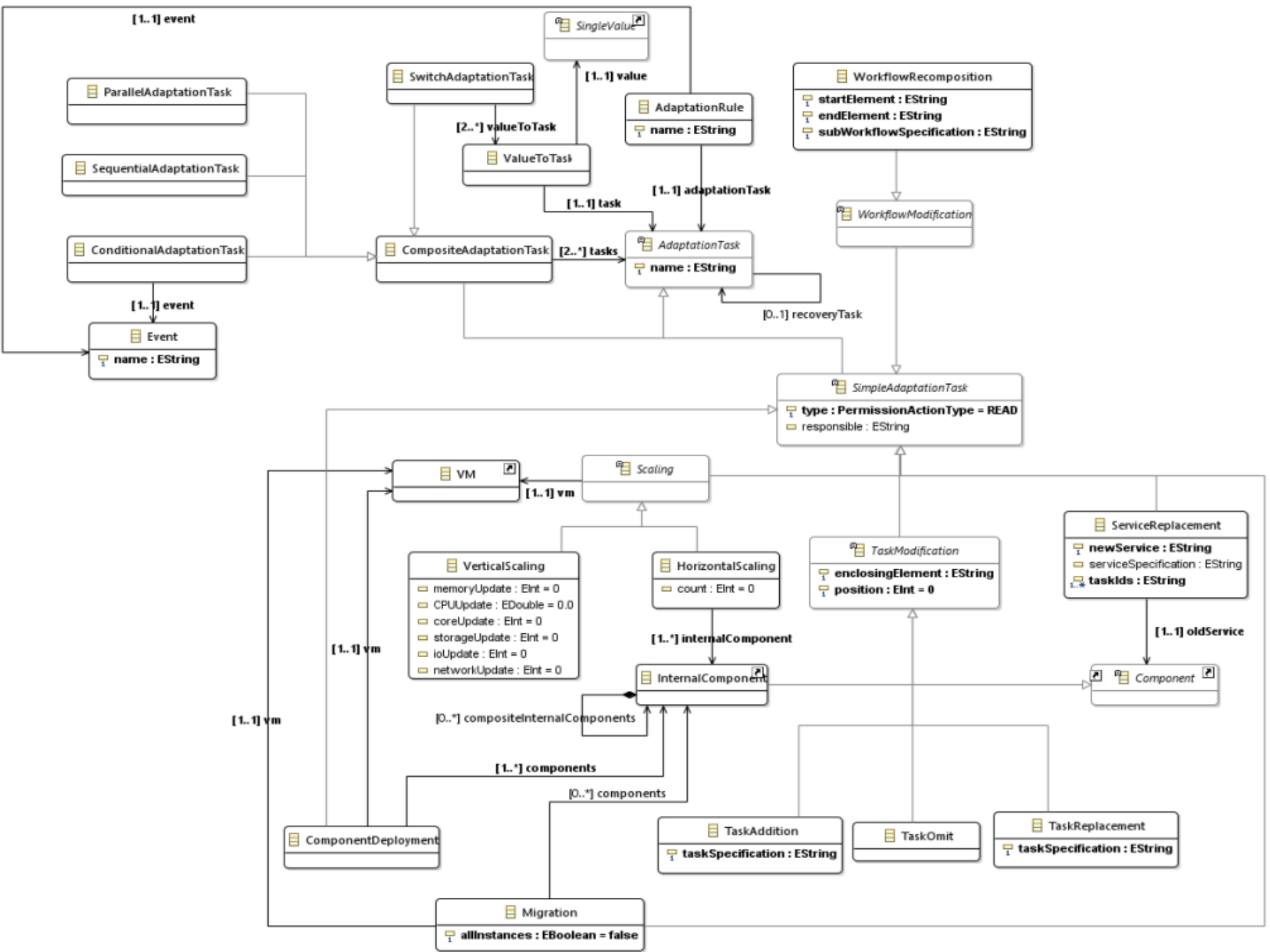
CloudSocket


RequirementModel
[0..*] requirements
Requirement
name : EString
SoftRequirement
priority : EDouble = 0.0
[1..*] requirements
RequirementGroup
requirementOperator :
RequirementOperatorType = AND
HardRequirement
HardRequirement
HardwareRequirement
ProviderRequirement
OSOrImageRequirement
[1..*] providers
QualitativeHardwareRequirement
minBenchmark : EDouble = 0.
maxBenchmark : EDouble = 0.
QuantitativeHardwareRequirement
minCPU : EDouble = 0.
maxCPU : EDouble = 0.
minCores : EInt = 0
maxCores : EInt = 0
minRAM : EInt = 0
maxRAM : EInt = 0
minStorage : EInt = 0
maxStorage : EInt = 0
CloudProvider
public : EBoolean = false
SaaS : EBoolean = false
PaaS : EBoolean = false
IaaS : EBoolean = false
OSRequirement
os : EString
is64os :
EBoolean = true
ImageRequirement
imageId : EString


HardRequirement
ScaleRequirement
HorizontalScaleRequirement
minInstances : EInt = 0
maxInstances : EInt = 0
VerticalScaleRequirement
minCPU : EDouble = 0.
maxCPU : EDouble = 0.
minCores : EInt = (
maxCores : EInt = 0
minRAM : EInt = (
maxRAM : EInt = (
minStorage : EInt = (
maxStorage : EInt = (
[1..1] component
[1..1] vm
InternalComponent
VM
HardRequirement
LocationRequirement
SecurityRequirement
[1..*] locations
[1..*] securityControls
Location
id : EString
SecurityControl
name : EString
specification : EString
HardRequirement
SoftRequirement
priority : EDouble = 0.0
ServiceLevelObjective
OptimisationRequirement
optimisationFunction :
OptimisationFunctionType =
MINIMISE
[1..1] customServiceLevel
[0..1] metric
Condition
name : EString
comparisonOperator :
ComparisonOperatorType
= GREATER_THAN
threshold : EDouble = 0.0
validity : EDate
Metric
description : EString
valueDirection : EShort = 0
layer : LayerType = SaaS

Conceptualisation – Adaptation Model

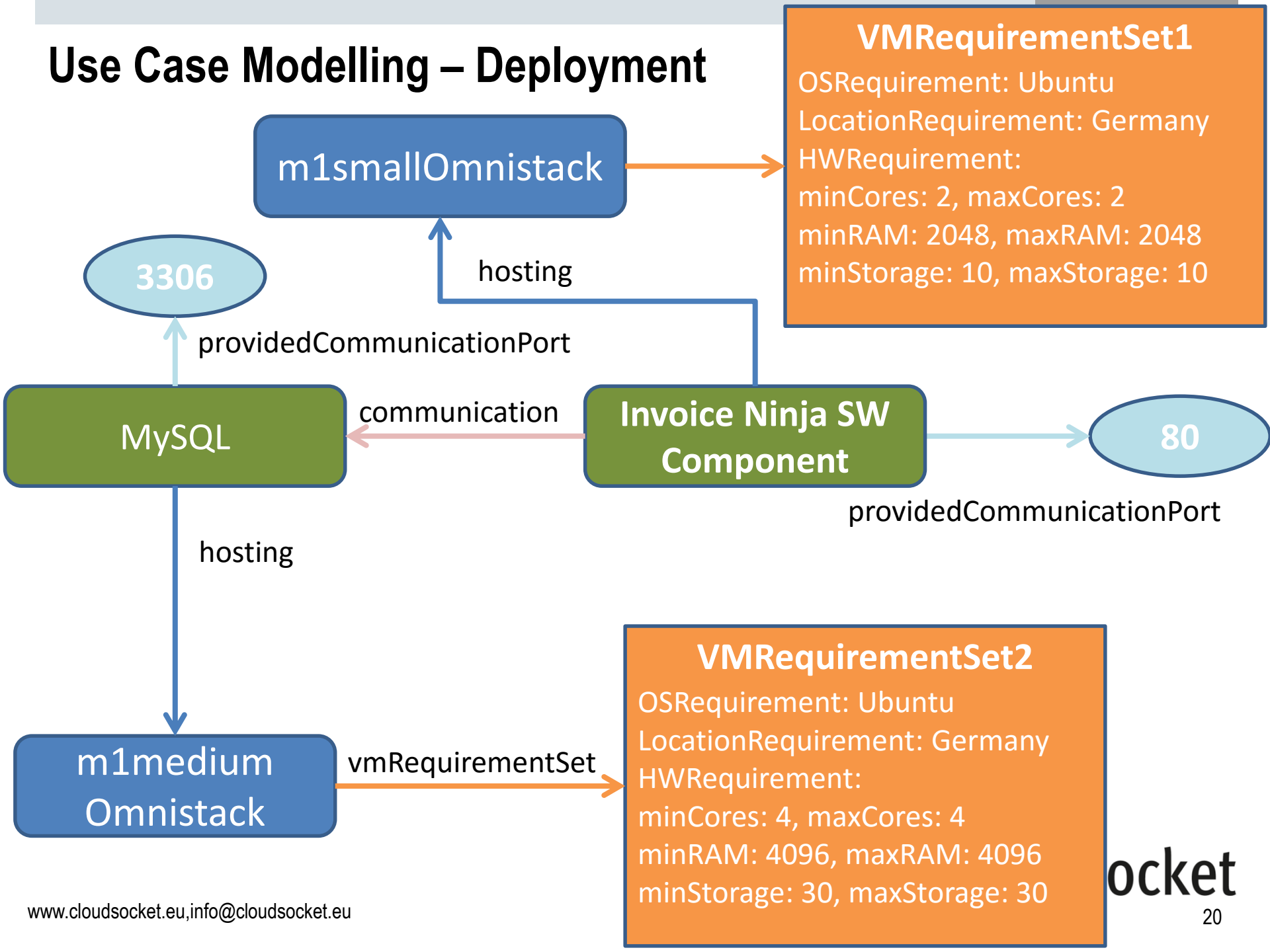
- Rationale: drive BPaaS reconfiguration
- New extension on CAMEL enhancing scalability model to become an adaptation one
- Covers:
  - Adaptation rules
    - Mapping of events to adaptation strategies / workflows
  - Events
    - Simple, non-functional events mappings to metric conditions
    - Composite events via composing events over logical and time-based operators
  - Adaptation tasks:
    - Various types of single adaptation tasks at different abstraction layers
    - Composite adaptation tasks representing adaptation strategies / workflows


[1..1] event
SingleValue
SwitchAdaptationTask
[1..1] value
ParallelAdaptationTask
[2..*] valueToTask
AdaptationRule
name : EString
WorkflowRecomposition
startElement : EString
endElement : EString
subWorkflowSpecification : EString
SequentialAdaptationTask
ValueToTask
[1..1] adaptationTask
[1..1] task
WorkflowModification
ConditionalAdaptationTask
CompositeAdaptationTask
[2..*] tasks
AdaptationTask
name : EString
[1..1] event
[0..1] recoveryTask
Event
name : EString
SimpleAdaptationTask
type : PermissionActionType = READ
responsible : EString
VM
Scaling
[1..1] vm
ServiceReplacement
newService : EString
serviceSpecification : EString
taskIds : EString
VerticalScaling
memoryUpdate : EInt = 0
CPUUpdate : EDouble = 0.0
coreUpdate : EInt = 0
storageUpdate : EInt = 0
ioUpdate : EInt = 0
networkUpdate : EInt = 0
HorizontalScaling
count : EInt = 0
TaskModification
enclosingElement : EString
position : EInt = 0
[1..1] oldService
[1..1] vm
[1..*] internalComponent
InternalComponent
Component
[0..*] compositeInternalComponents
[1..1] vm
[1..*] components
[0..*] components
ComponentDeployment
TaskAddition
taskSpecification : EString
TaskOmit
TaskReplacement
taskSpecification : EString
Migration
allInstances : EBoolean = false

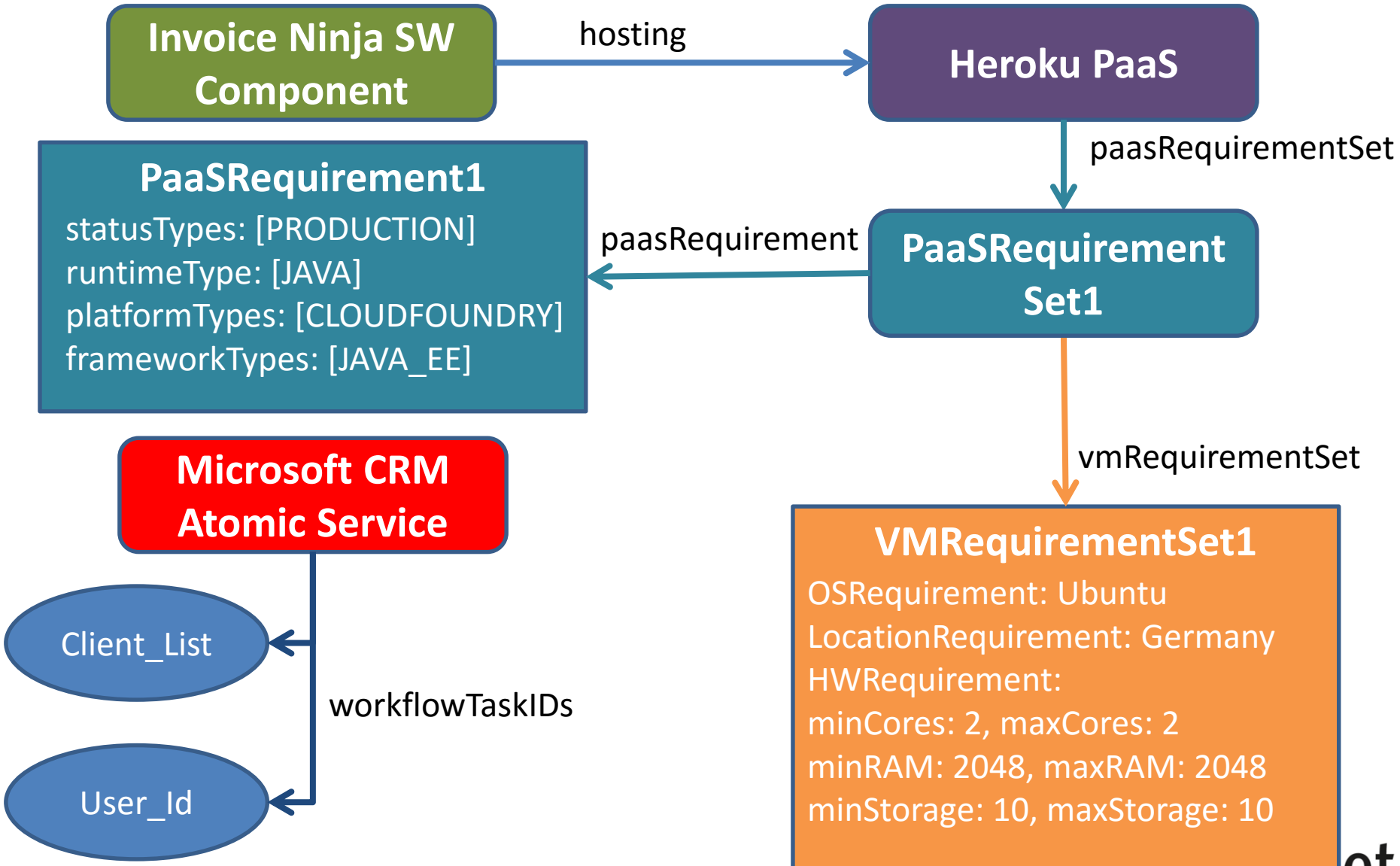
Use Case

- Send Invoice
  - Support BP realised as a BPaaS
  - Requires two main SaaS: CRM & Invoicing
  - Different deployment options where CRM is always external SaaS:
    - Option1: InvoiceNinja as external Invoicing SaaS
    - Option2: InvoiceNinja exposed as internal SaaS component offered by broker
    - Option3: Invoice Ninja exposed as internal SaaS component dedicated solely to the client

Use Case Modelling – Deployment



Use Case Modelling – Deployment



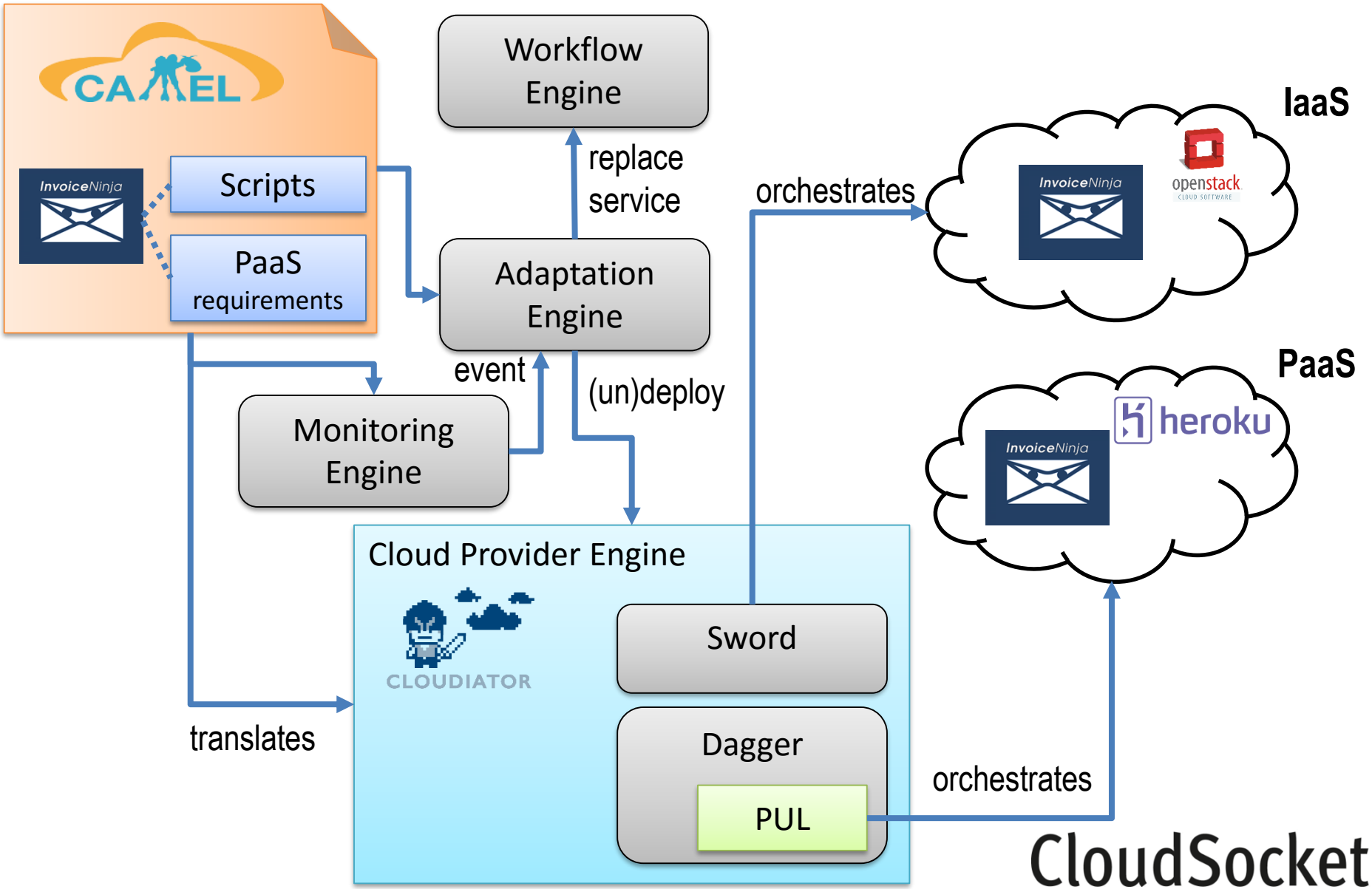
Use Case Modelling – Requirements

- Apart from HW, PaaS, OS & location reqs for VMs, also SLOs:
  - SLO1: Mean response time less than 10 seconds
  - SLO2: Mean availability greater than 70%

Use Case Modelling – Adaptation Rules

- One rule:
  - Event: When Invoice Ninja SaaS component availability is very low (< 70%) or response time is very big (> 10 seconds), then
  - Action: Deploy Invoice Ninja SaaS component via a PaaS to a new VM, update service endpoint in BPaaS workflow (service replacement), delete old Invoice Ninja VM
- Rationale:
  - Something wrong with current VM
  - Need to rapidly move Invoice Ninja SaaS component to new VM
    - PaaS-based deployment is in general faster than IaaS-based deployment

BPaaS Adaptation Architecture

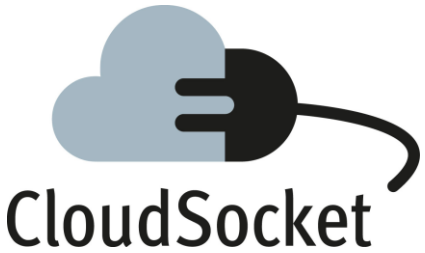


CloudSocket

Summary

- CAMEL:
  - Multi-DSL covering many aspects
  - Covers both IaaS, PaaS & SaaS levels
  - Cross-layer adaptation rule modelling
  - Good support level – many editors
  - Usage in many European projects
    - PaaSage, CloudSocket, Cactos, Melodic
  - Very active community
    - Various extensions performed or in their way
  - Possible standardisation possibility
    - TOSCA – coverage of instance level

Join us!



www.cloudsocket.eu

@CloudSocketEU