
State:

parents //these are parents
services
myId
fallback

Upon Init do:

fallback \leftarrow **getFallback()**
myId \leftarrow **randomString()**

Upon event Receive(QUALITY_NOT_ASSURED, service, location) do:

newChild \leftarrow **getNodeClosestTo(location)**
Trigger Send(DEPLOY_SERVICE, newChild, service, myself)
Call addChild(service, newChild)

Upon event Receive(DEPLOY_SERVICE, service, sender) do:

if $service \notin services$ **then**
 services \leftarrow services \cup {service}
end

Call updateParent(service, sender)

Procedure updateParent(service, parent):

if $service.Parent == parent$ **then**
 return
end

parents \leftarrow parents \cup {parent}
service.parent \leftarrow parent
foreach $child \in service.children$ **do**
 Trigger Send(GRANDPARENT, child, service, parent, myself)
end

Upon event Receive(GRANDPARENT, service, grandparent, sender) do:

service.grandparent \leftarrow grandparent

Procedure updateChildren(service):

orderedChild \leftarrow **order(service.children)**
foreach $child \in service.children$ **do**
 Trigger Send(SIBLINGS, child, service, orderedChild)
end

Procedure addChild(service, child):

if $child \notin children$ **then**
 children \leftarrow children \cup {child}
end

if $child \notin service.children$ **then**
 service.children \leftarrow service.children \cup {child}
 servicesByChild[child] \leftarrow servicesByChild[child] \cup {service}
 Call updateChildren(service)
end

Procedure deleteChild(child):

children \leftarrow children \setminus {child}
childServices \leftarrow servicesByChild[child]
foreach $service \in childServices$ **do**