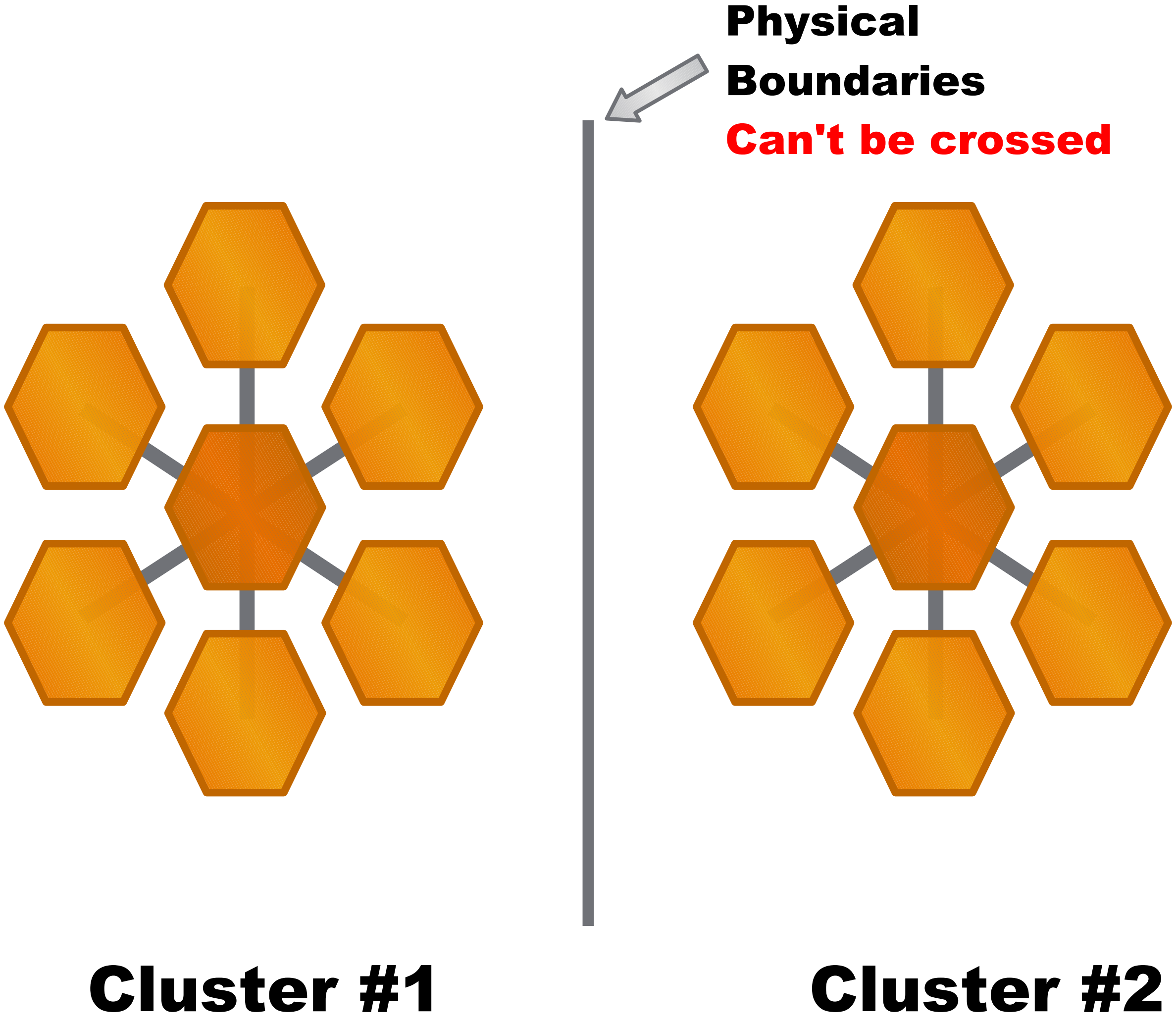
PROJECT HEDEBY – SERVICE DOMAIN MANAGEMENT

Richard Hierlmeier

Andreas Dörr

Sun Microsystems, Inc.

In the Beginning...

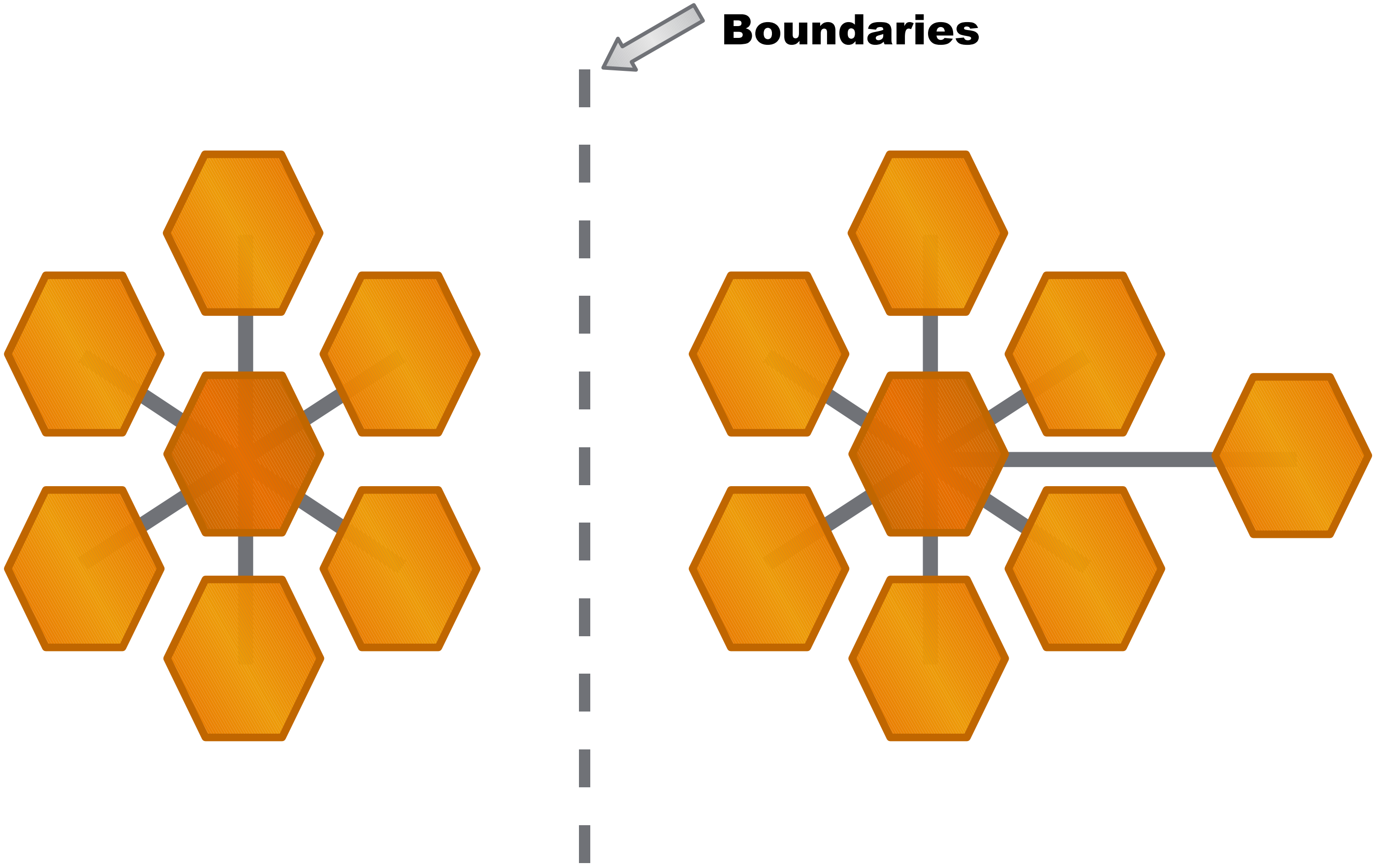


Physical
Boundaries
Can't be crossed

Cluster #1

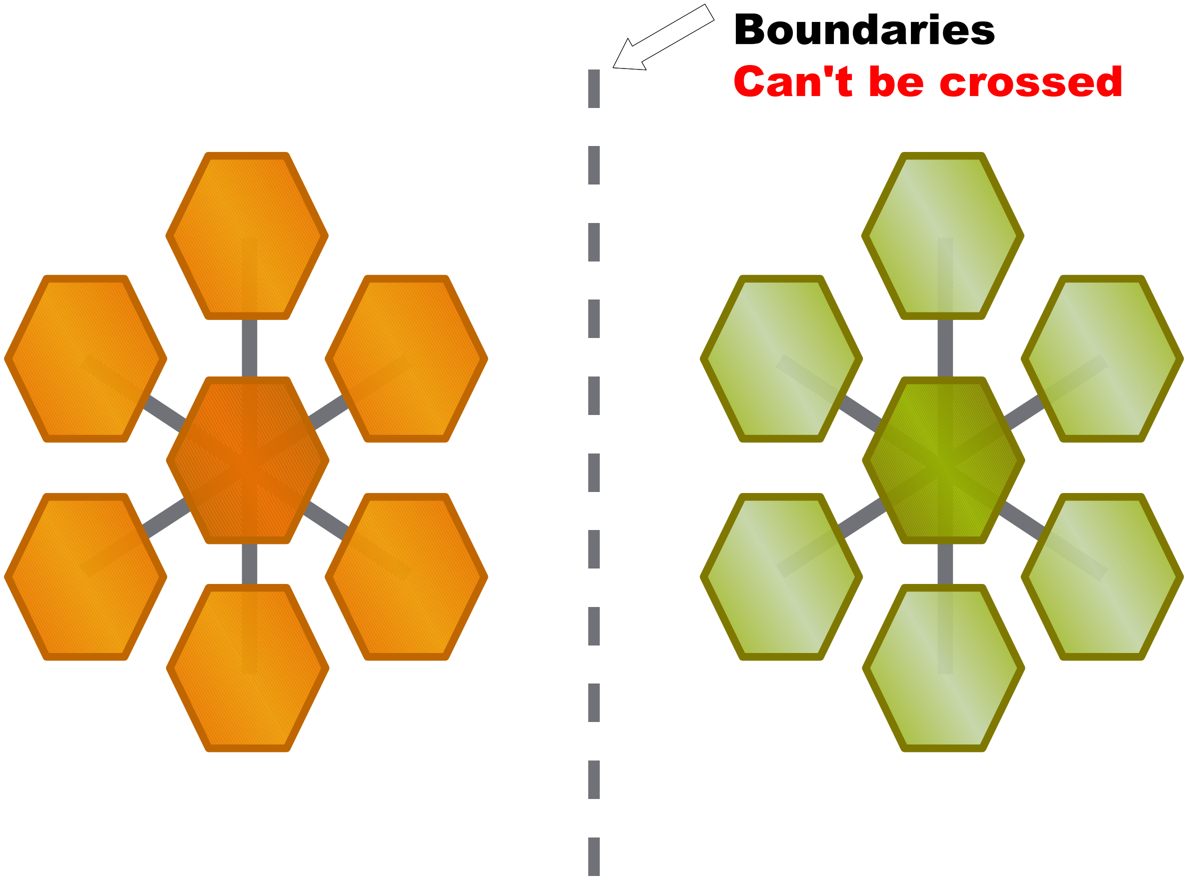
Cluster #2

Along Came N1 Grid Engine



N1 Grid Engine Grid

Not Alone in the Grid

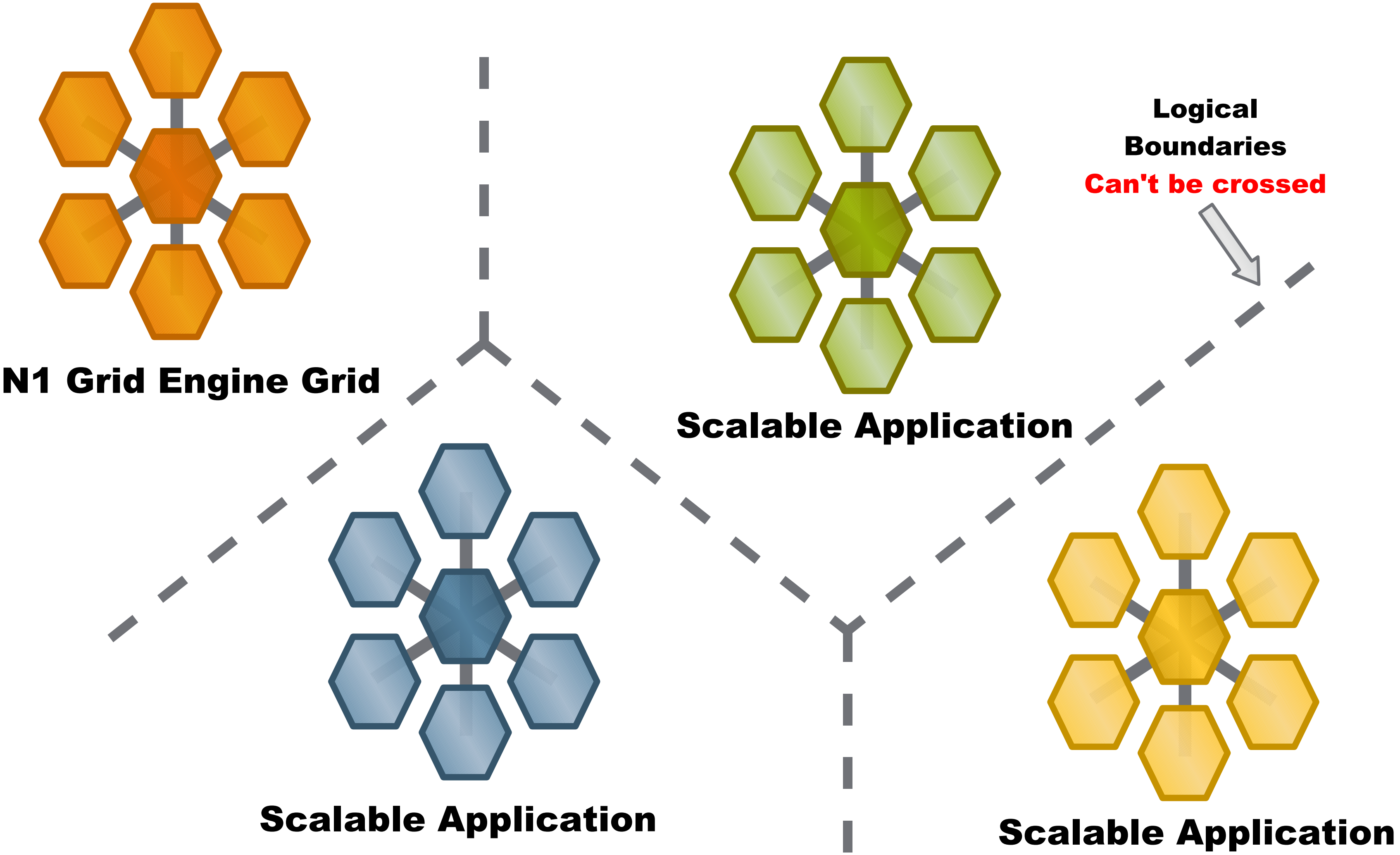


Logical Boundaries Can't be crossed

N1 Grid Engine Grid

Scalable Application

Really Not Alone…



N1 Grid Engine Grid

Scalable Application

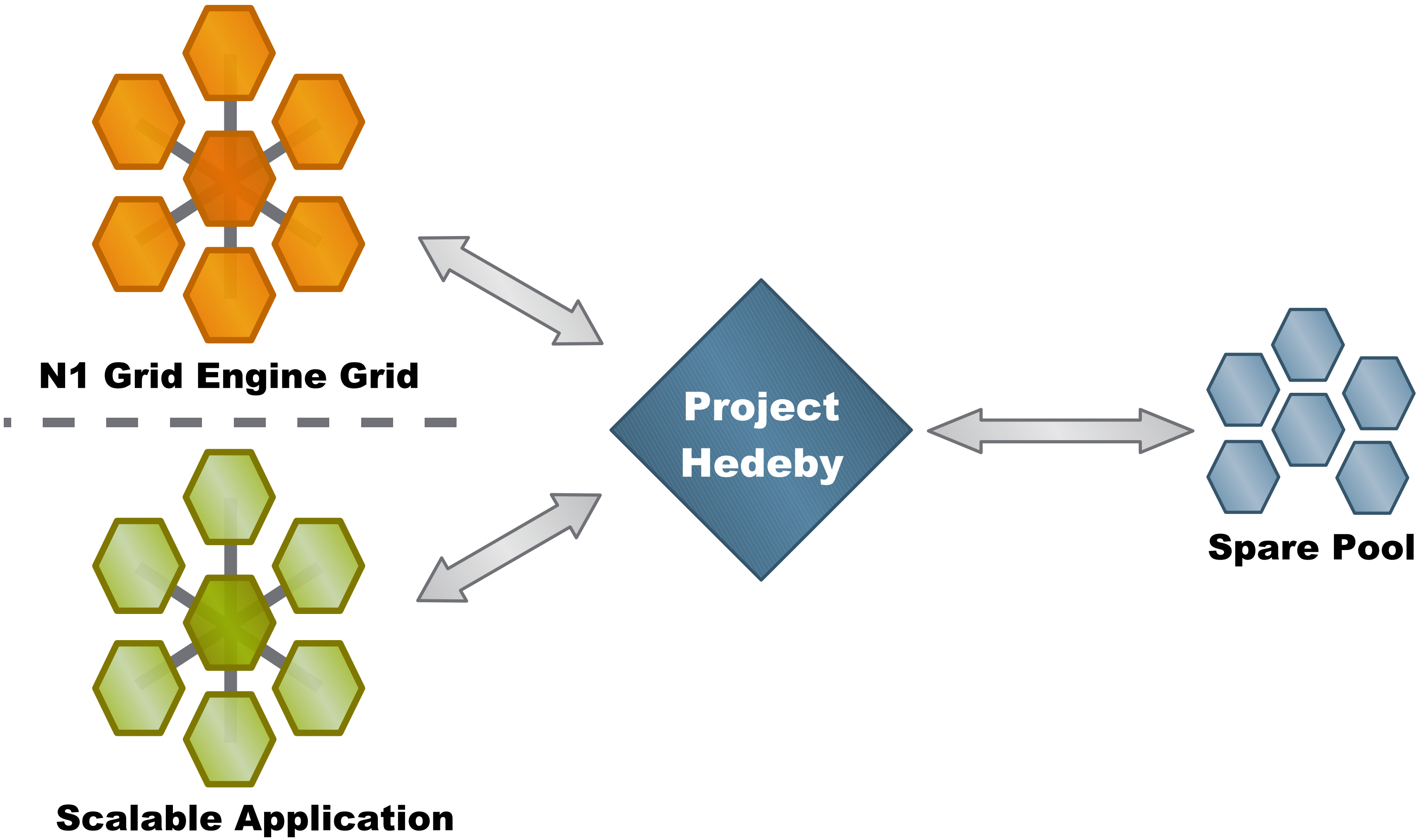
Logical Boundaries

Can't be crossed

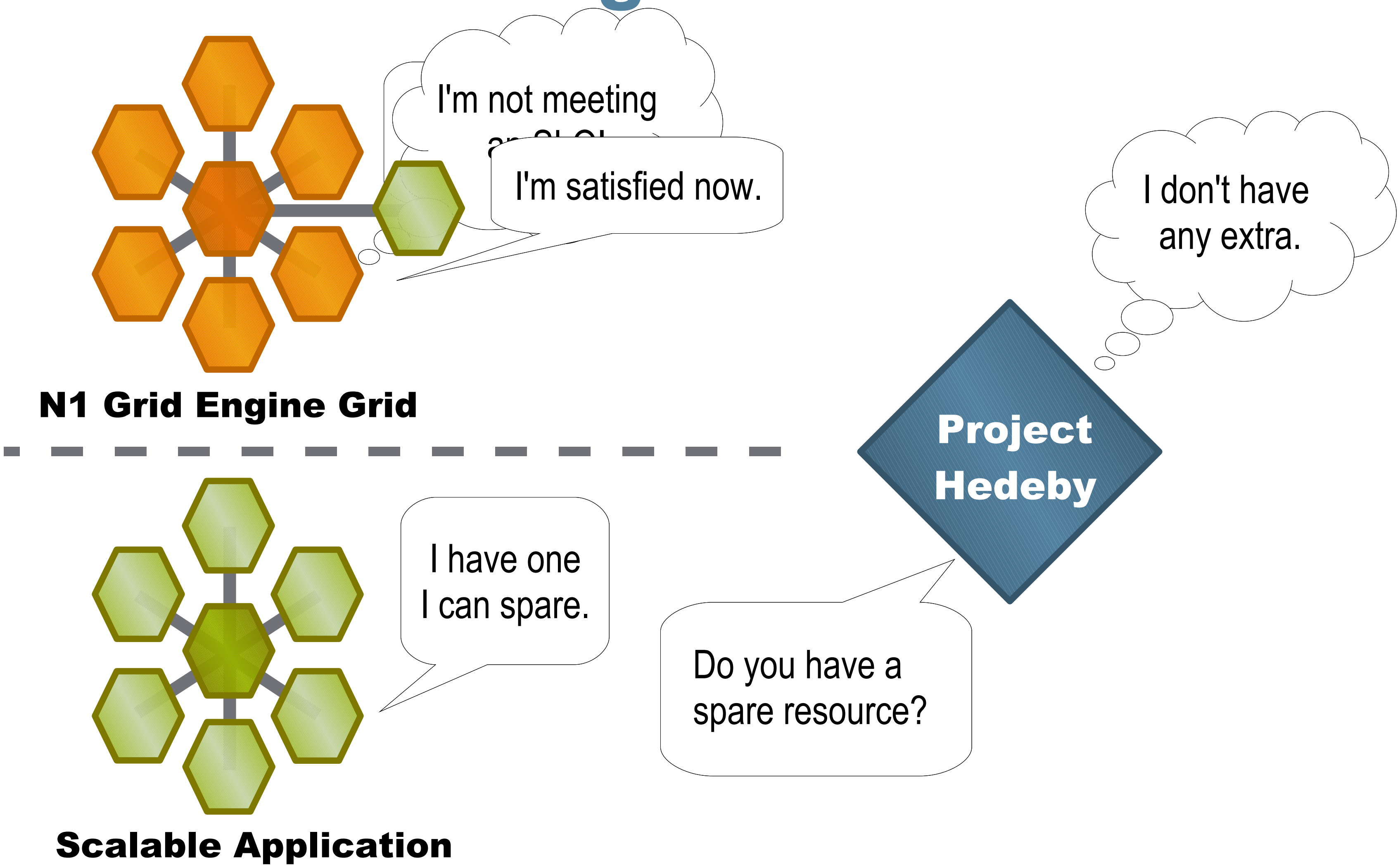
Scalable Application

Scalable Application

Project Haithabu



Resource Sharing



I'm not meeting an SLO!

I'm satisfied now.

I don't have any extra.

N1 Grid Engine Grid

Project Hedeby

I have one I can spare.

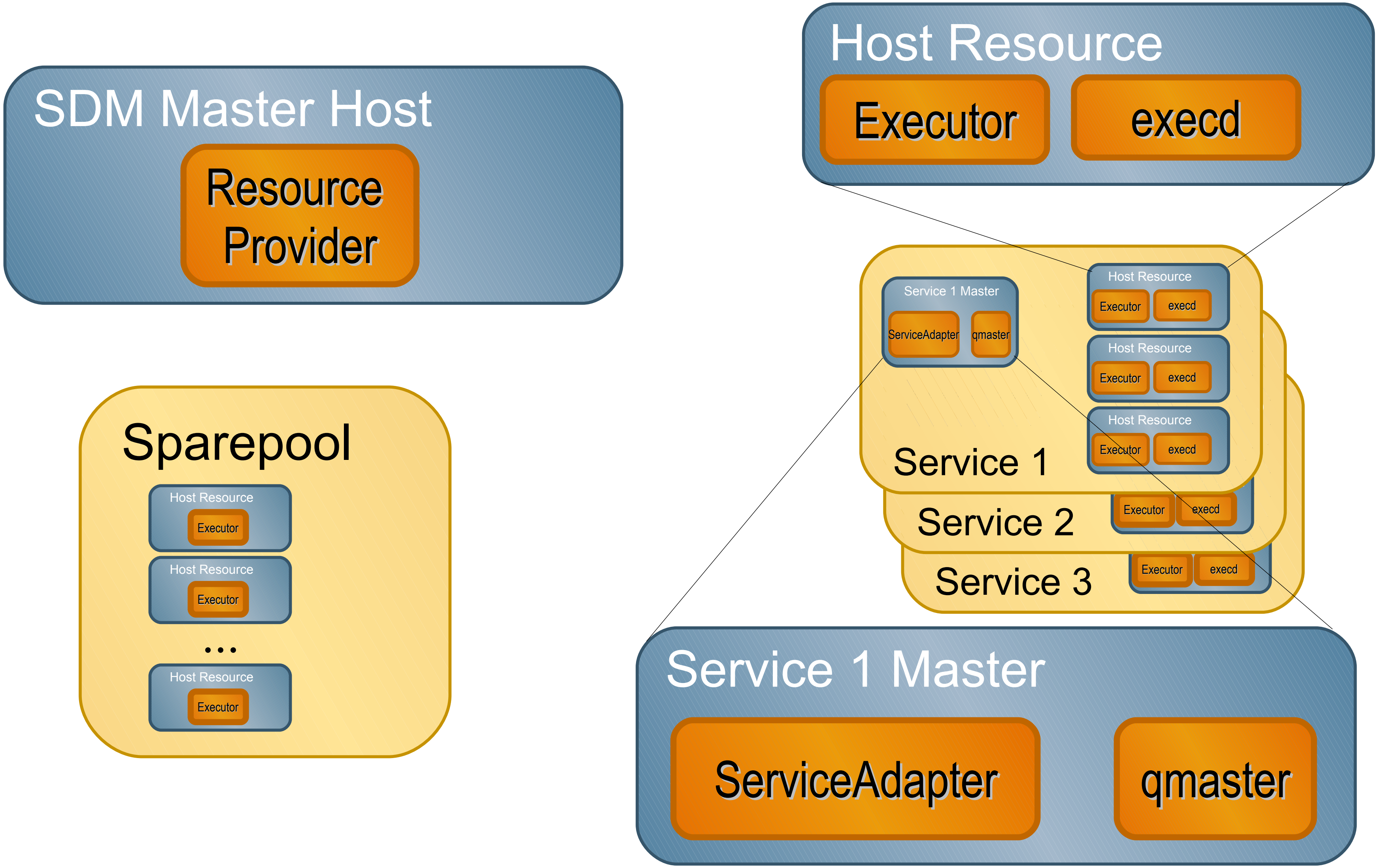
Do you have a spare resource?

Scalable Application

Service Level Management

- Service Level Agreement (SLA)
  - An agreement between the service provider and customer specifying service performance
  - Translates into a series of SLO's
- Service Level Objective (SLO)
  - Specify customer requirements for performance, availability, etc. which have to be met by a service
  - Represented by acceptable KPI value ranges
- Key Performance Indicator (KPI)
  - Performance statistic for a service

Typical Deployment



SDM Master Host

Resource Provider

Sparepool

Host Resource
Executor

Host Resource
Executor

...

Host Resource
Executor

Host Resource

Executor

execd

Service 1 Master
ServiceAdapter
qmaster

Host Resource
Executor
execd

Host Resource
Executor
execd

Host Resource
Executor
execd

Service 1

Executor
execd

Service 2

Executor
execd

Service 3

Service 1 Master

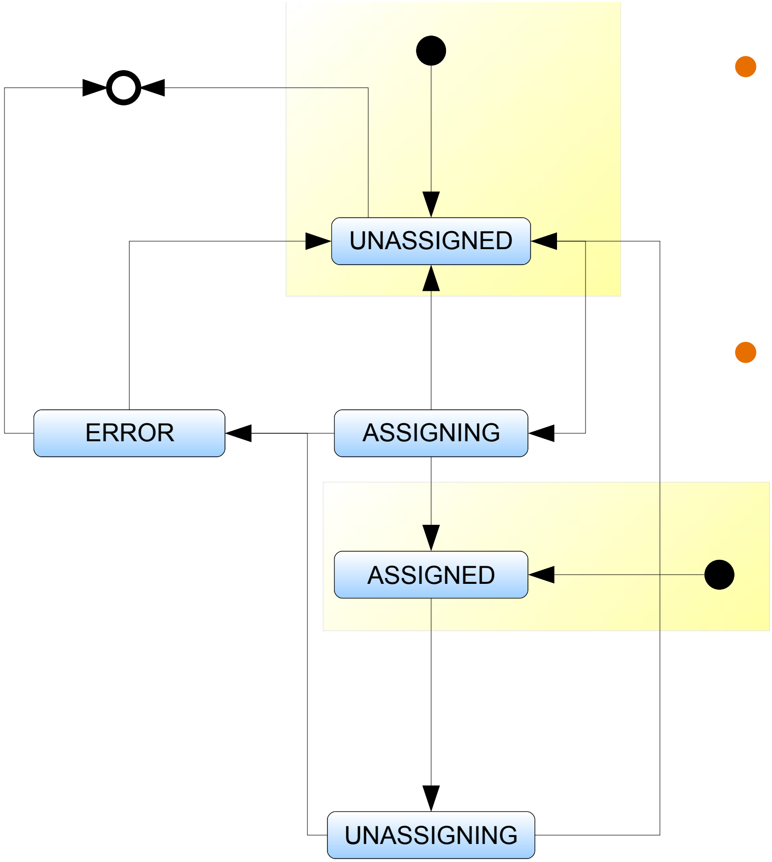
ServiceAdapter

qmaster

Resource

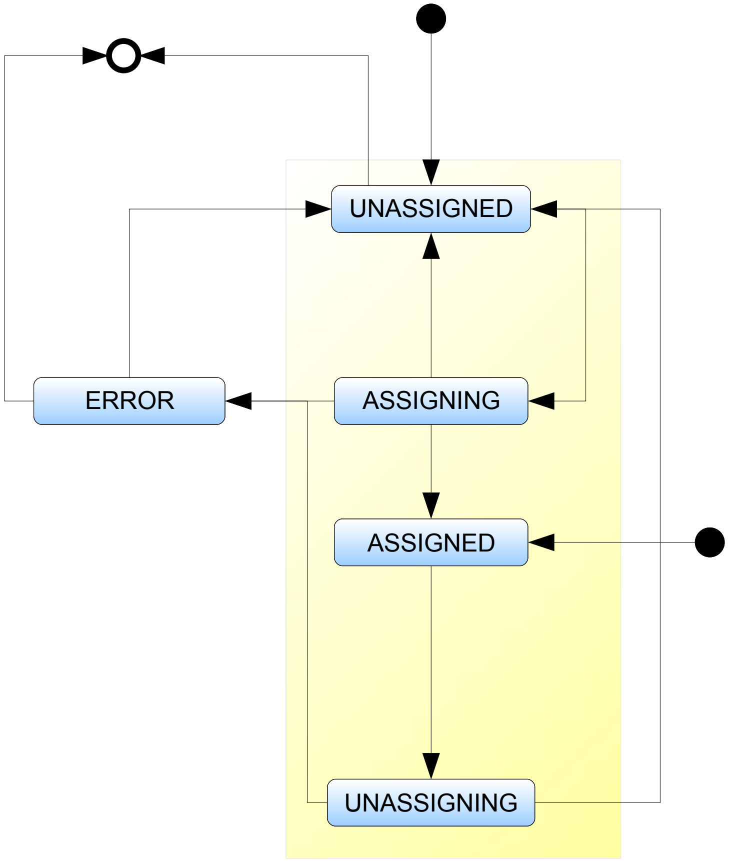
- Can be anything used by a service
- In the first release only host resource are possible
- A Resource has a name, a type and additional properties

States of a Resource



- Initial state UNASSIGNED Administrator has defined a resource
- Initial state ASSIGNED ServiceAdapter has discovered that service has already an unknown resource assigned (static resource)

States of a Resource

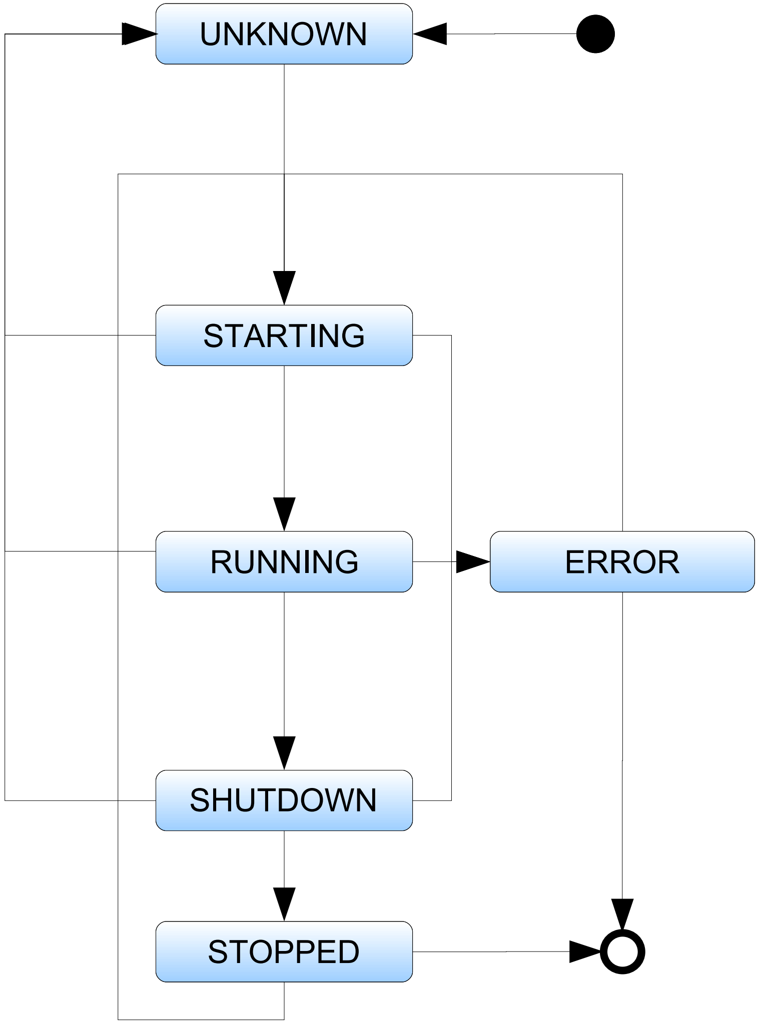


- State transition UNASSIGNED <-> ASSIGNED
- Triggered
  - > Manually over UI
  - > Automatically (SLA of service not fulfilled)
- On unrecoverable errors => ERROR state

Service

- Scalable
- Manageable over the network
- Service Adapter (SA) in Hedeby is a proxy for the real service
- Adding a service to a Hedeby system means
  - > Hedeby starts SA component
  - > SA discovers and manages the service (start/stop, add new resources)

States of a Service



- Administrator defines the service over the UI
- Service Adapter manages (start/stop)
- Service Adapter observes service (detects errors, discovery)
- Administrator can removes service

Service - KPI's

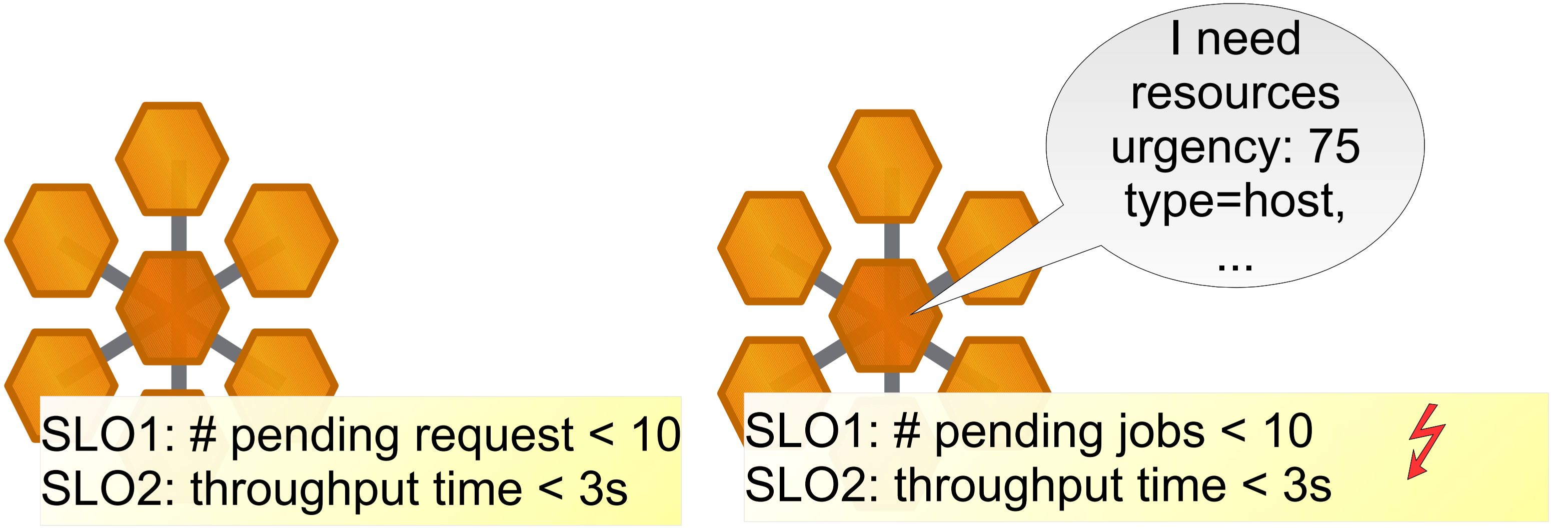
- SA has direct connection to the service
- Service sends to events SA if objects are modified (jobs, hosts, queues)
- SA calculates out of the events the KPI's of the Grid Engine

Service - Typical KPI's

- Number of hosts
- Number of pending jobs per host/queue/queue instance
- Free slots per queue/queue instance
- Public API to added customer specific KPI's (planned)
- For Grid Engine any consumable can be a KPI

Service - SLO

- Based KPI's a set of SLOs is defined for each service
- Set of SLOs is called SLA
- If not fulfilled SLO produces a Resource Need (contains the Urgency and the Resource Properties)

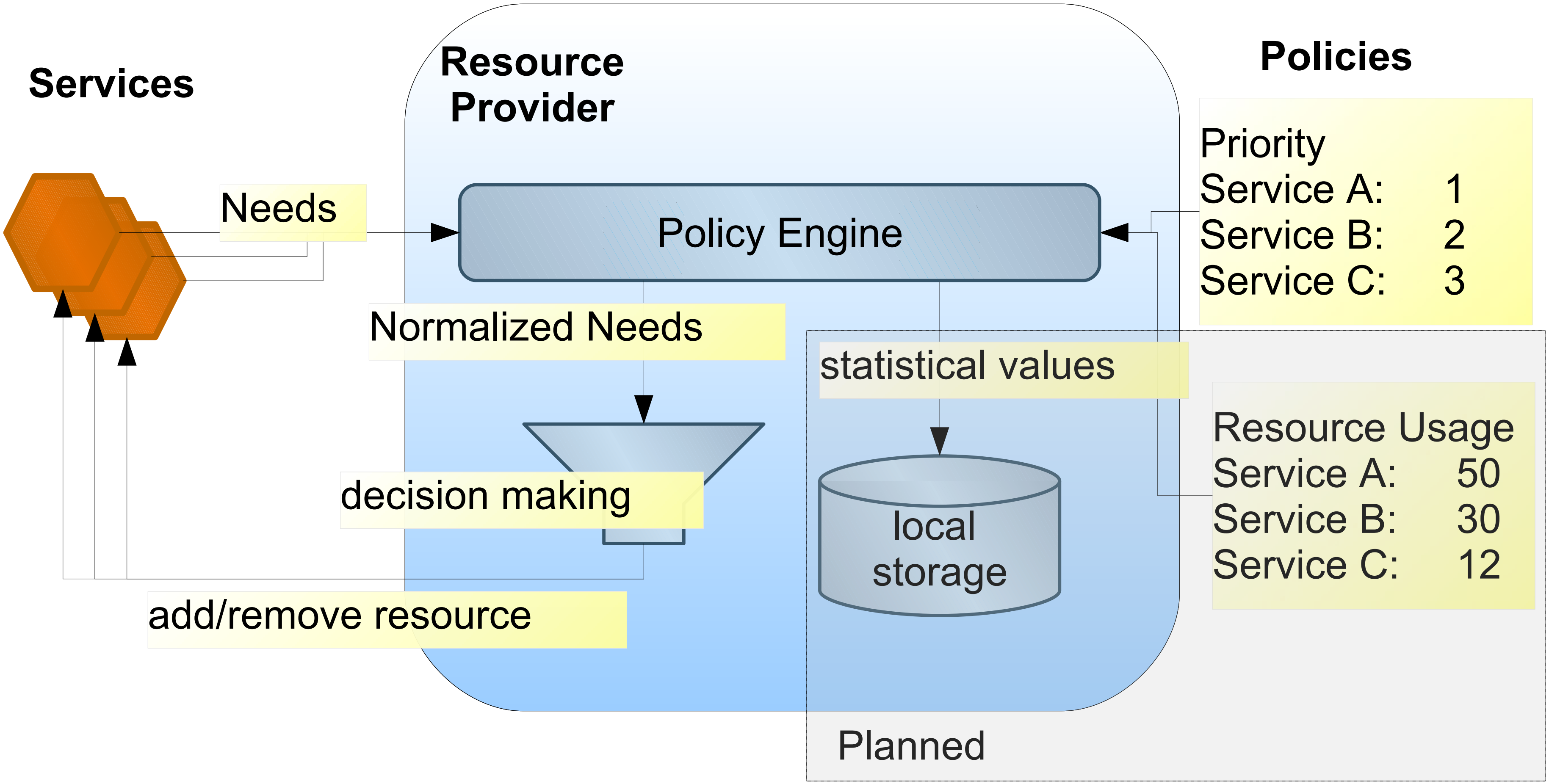


SLO1: # pending request < 10
SLO2: throughput time < 3s

I need resources urgency: 75 type=host, ...

SLO1: # pending jobs < 10 ⚡
SLO2: throughput time < 3s

ResourceProvider



Services

Resource Provider

Policies

Needs

Policy Engine

Priority

Service A:	1
Service B:	2
Service C:	3

Normalized Needs

statistical values

Resource Usage

Service A:	50
Service B:	30
Service C:	12

decision making

local storage

add/remove resource

Planned

User Interface

- Single command line tool for
  - install/deinstall master host/managed host
  - start/stop processes and components
  - managing services (definition, configuration, removal, monitoring)
  - managing resources (definition, assignment, monitoring ring)

UI – Install master

```
% sdmadm -s <system_name> inst_master \
   -ca_admin_mail <admin_email>       \
   -ca_state "Germany"                \
   -ca_country "DE"                   \
   -ca_location "Regensburg"          \
   -ca_org_unit "Development"         \
   -ca_org "Sun Microsystems"         \
   -a <admin_user>                    \
   -cs_port <port>                    \
   -l <local_spool_dir>               \
   -sge_root /opt/sge
```

UI – Install managed host

```
% sdmadm -s <system_name> inst_managed_host \
     -a <admin_user>
     -l <local_spool_dir> \
     -cs_url <master_host>:<cs_port>
```

```
% sdmadm -s <system_name> start
cs_vm started
executor_vm started
```

```
% sdmadm -s <system_name> shutdown
Shutdown has been triggered for following JVMs:
[executor_vm, cs_vm]
```

UI - Examples

```
% sdmadm show_status
Services:
s1_master:  s1 -- status: started
s2_master:  s2 -- status: started
Resource Providers:
master: rp -- status: started

Executors:
exec2: executor -- status: started
...
execn: executor -- status: started
```

UI – Configuration

- Similar to qconf
- sdmadm opens configuration in an editor
- User makes it's modifications and exits the editor
- sdmadm sends modified configuration to the component
- Possibility to store/load configuration from file
- Configuration objects are defined as XML schema
- sdmadm allows to add/modify/delete configuration objects

UI – Configuration - Example

```
% sdmadm mconf
<global name="test_system">
    <jvm port="31016" user="admin" name="cs_vm"/>
    <jvm port="0" user="root" name="executor_vm">
        <component classname="com.sun.grid....ExecutorImpl"
                   name="executor">
            <hosts><include>.*</include></hosts>
            <config>executor</config>
        </component>
    </jvm>
    <jvm port="0" user="admin" name="service_vm">
        <component classname="com.sun....GridEngineServiceImpl"
                   name="executor">
            <hosts><include>.*</include></hosts>
            <config>s1</config>
        </component>
    </jvm>
    ...
</global>
```

UI - Monitoring

- ResourceProvider stores each service/resource state transitions for a limited time (including the reason why the state transition has been triggered)
- sdmadm prints out
  - current state of services/resources
  - history of services/resources
- In future integration into ARCo possible

UI – Monitoring - Example

```
% sdmadm show_resources
resource      service
ex1           s1
ex2           s2
ex3           s2
...
ex<n>      sparepool

% sdmadm show_resource_history ex1
time      service     urg   reason
12:44:00  s2          41    SLO1 not fulfilled
18:01:43  sparepool   54    RP requested resource
18:01:45  s1          54    SLO2 not fulfilled
```

PROJECT HEDEBY – SERVICE DOMAIN MANAGER

Richard Hierlmeier
richard.hierlmeier@sun.com
Andreas Dörr
andreas.doerr@sun.com