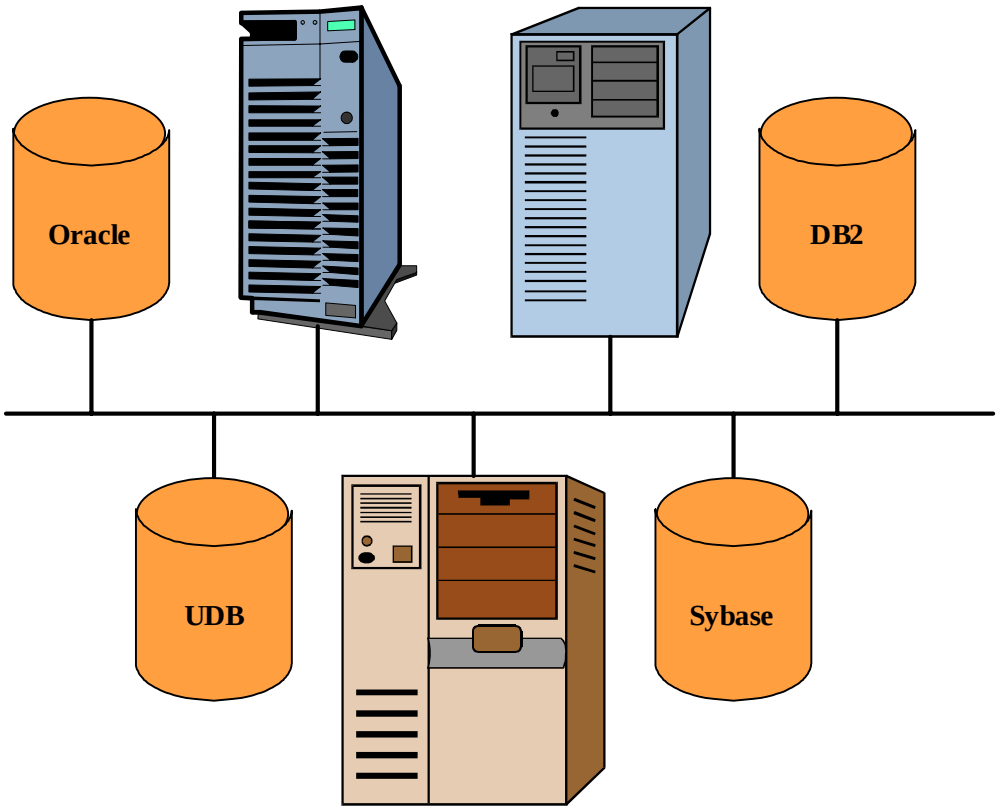
QoS for Enterprise Applications

Paul Tunney

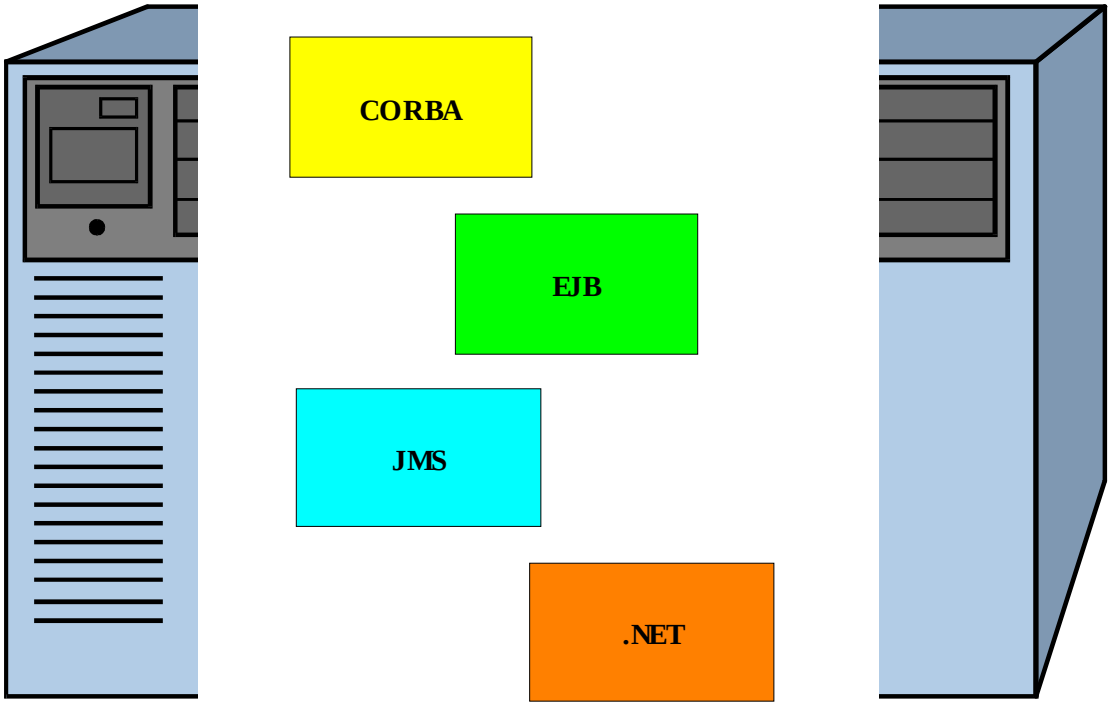
WhiteView

QoS Taskforce: Paris April 2002

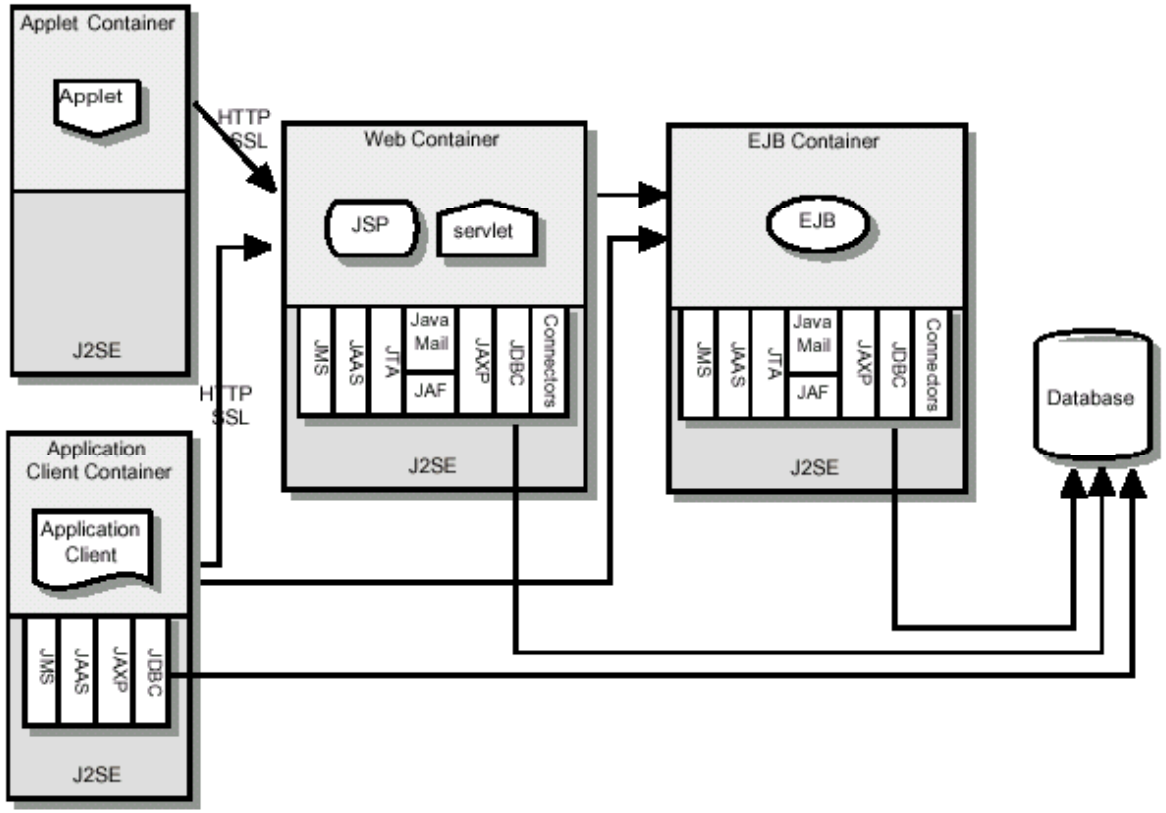
Current Topology



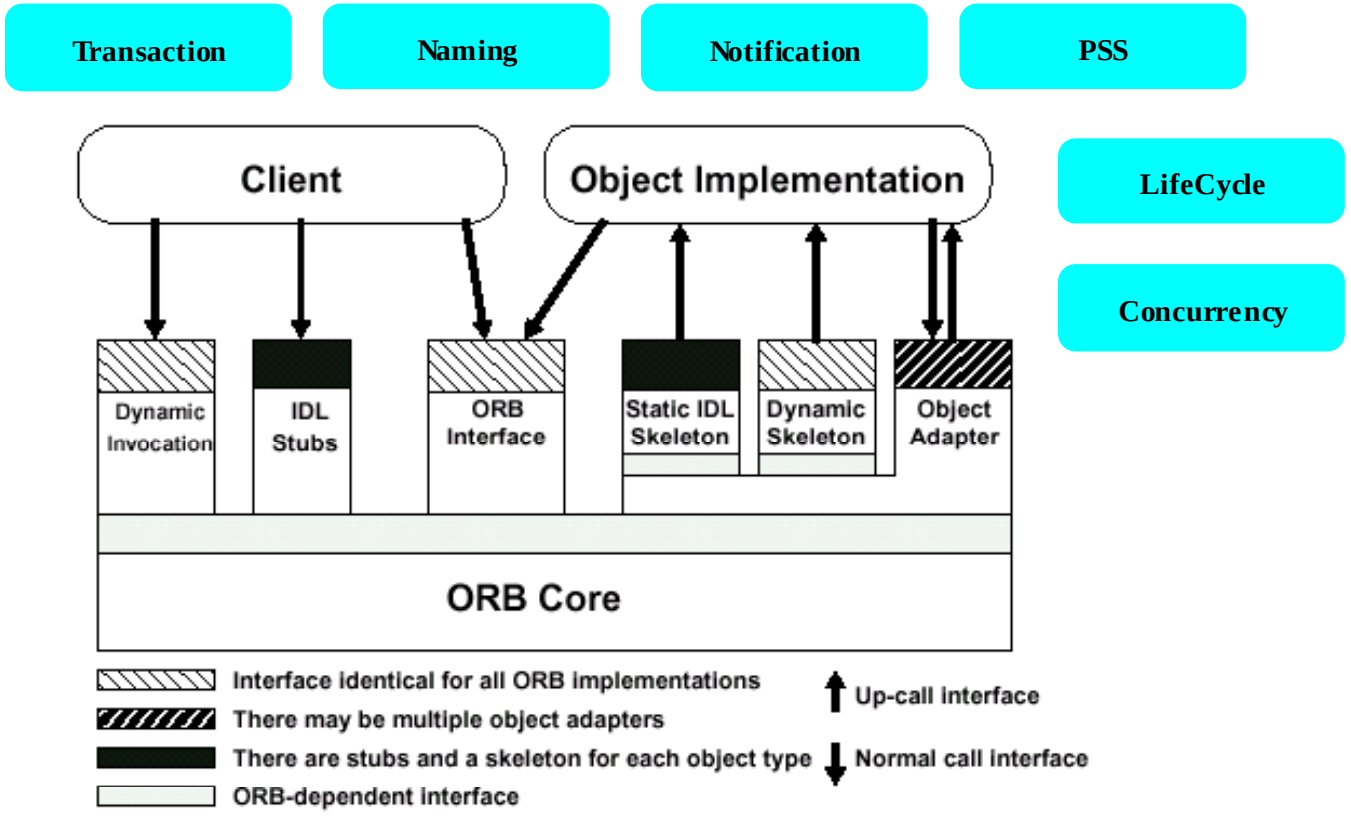
What's in the box



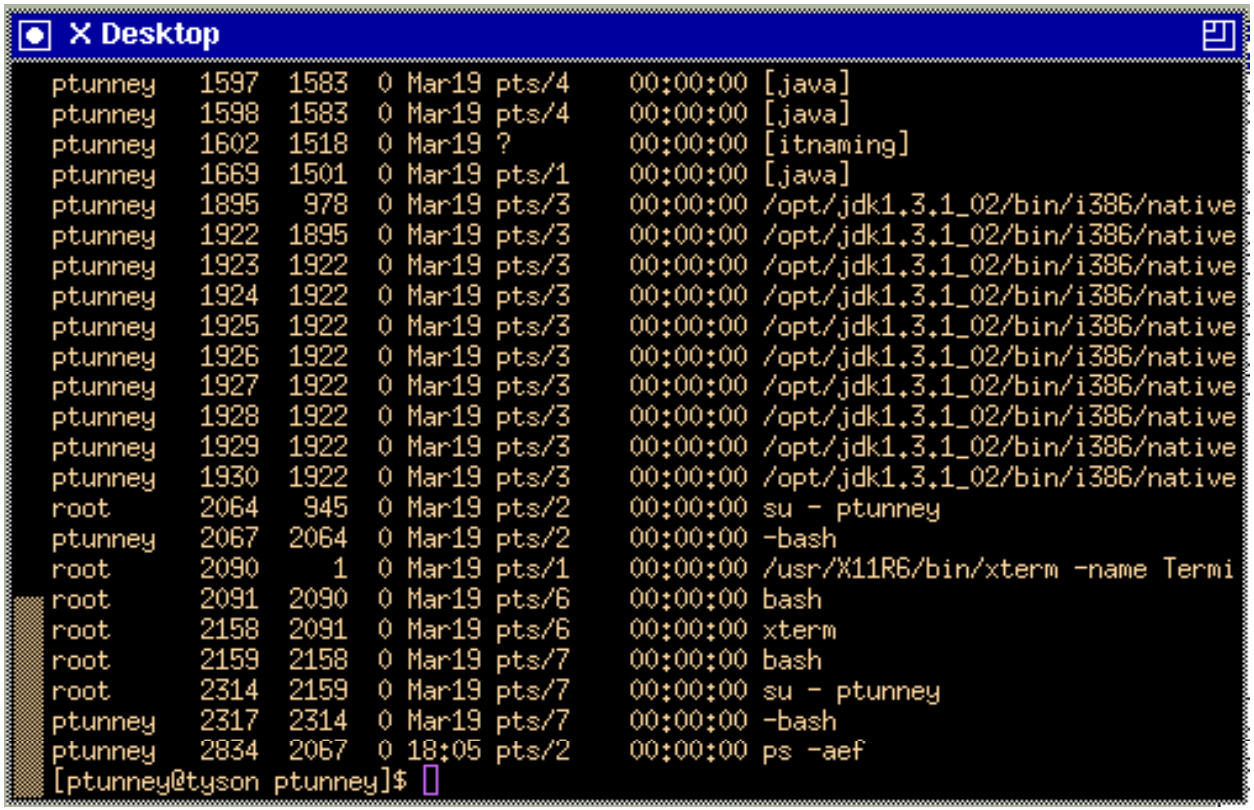
J2EE Application Server Architecture



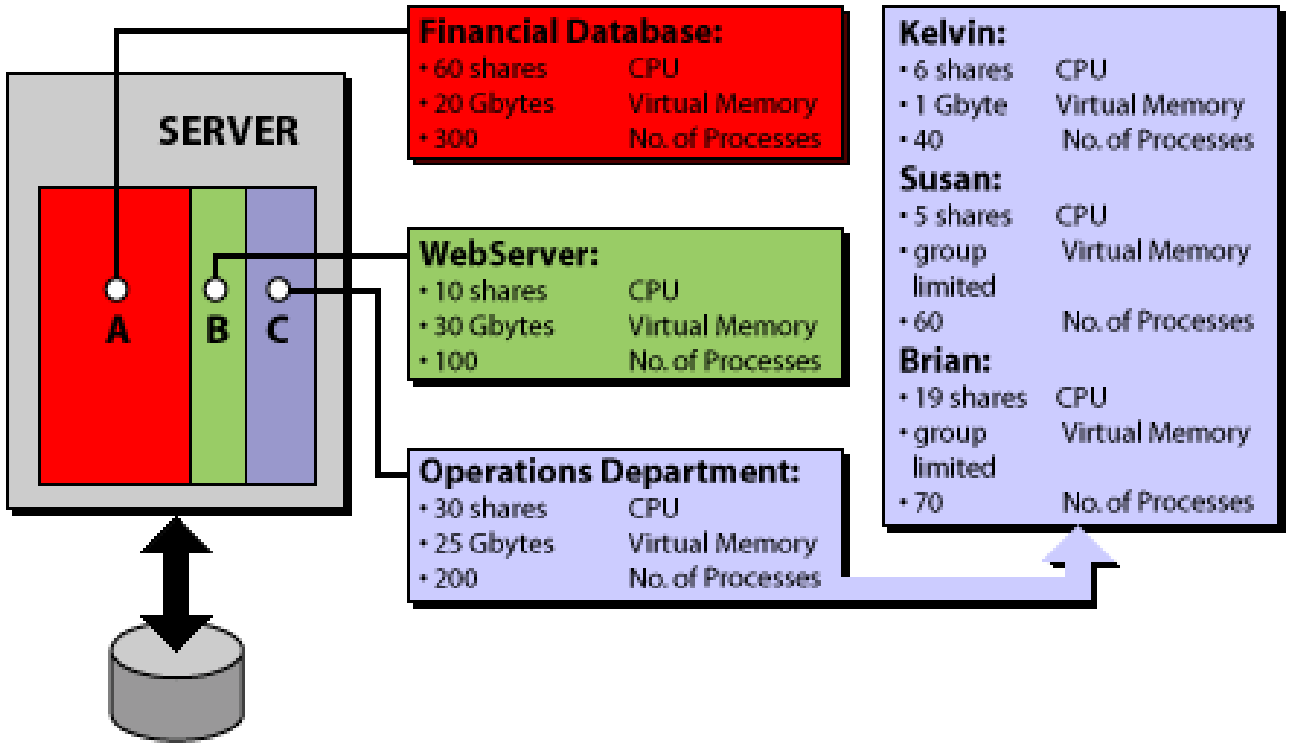
CORBA Architecture



Where’s the applications process



Solaris Resource Manager



QoS Characteristics

- Timeliness (e.g. request/response time)
- Capacity (e.g. throughput)
- Accuracy (e.g. error probability)
- Security (e.g. access control, integrity, auth)
- Precedence (e.g. priority)

Deployment Descriptors could be the way

```
<enterprise-beans>
  <entity>
   <description>Inventory CMP EJB</description>
   <display-name>InventoryEB</display-name>
   <ejb-name>InventoryEJB</ejb-name>
   <local-home>com.sun.j2ee.blueprints.supplier.inventory.ejb.InventoryLocalHome</local-home>
   <local>com.sun.j2ee.blueprints.supplier.inventory.ejb.InventoryLocal</local>
   <ejb-class>com.sun.j2ee.blueprints.supplier.inventory.ejb.InventoryEJB</ejb-class>
   <persistence-type>Container</persistence-type>
   <prim-key-class>java.lang.String</prim-key-class>
   <reentrant>False</reentrant>
   <cmp-version>2.x</cmp-version>
   <abstract-schema-name>Inventory</abstract-schema-name>
   <cmp-field>
    <field-name>itemId</field-name>
   </cmp-field>
   <cmp-field>
    <field-name>quantity</field-name>
   </cmp-field>
   <primkey-field>itemId</primkey-field>
   <security-identity>
    <description></description>
    <use-caller-identity></use-caller-identity>
   </security-identity>
   <query>
    <description>EJBQL to find all Inventory Items</description>
    <query-method>
     <method-name>findAllInventoryItems</method-name>
     <!--method-intf>LocalHome</method-intf-->
     <method-params />
    </query-method>
    <ejb-ql>Select OBJECT(a) From Inventory a</ejb-ql>
   </query>
  </entity>
</enterprise>
```

QML - Quality Meta Language from Hewlett Parkard Labs

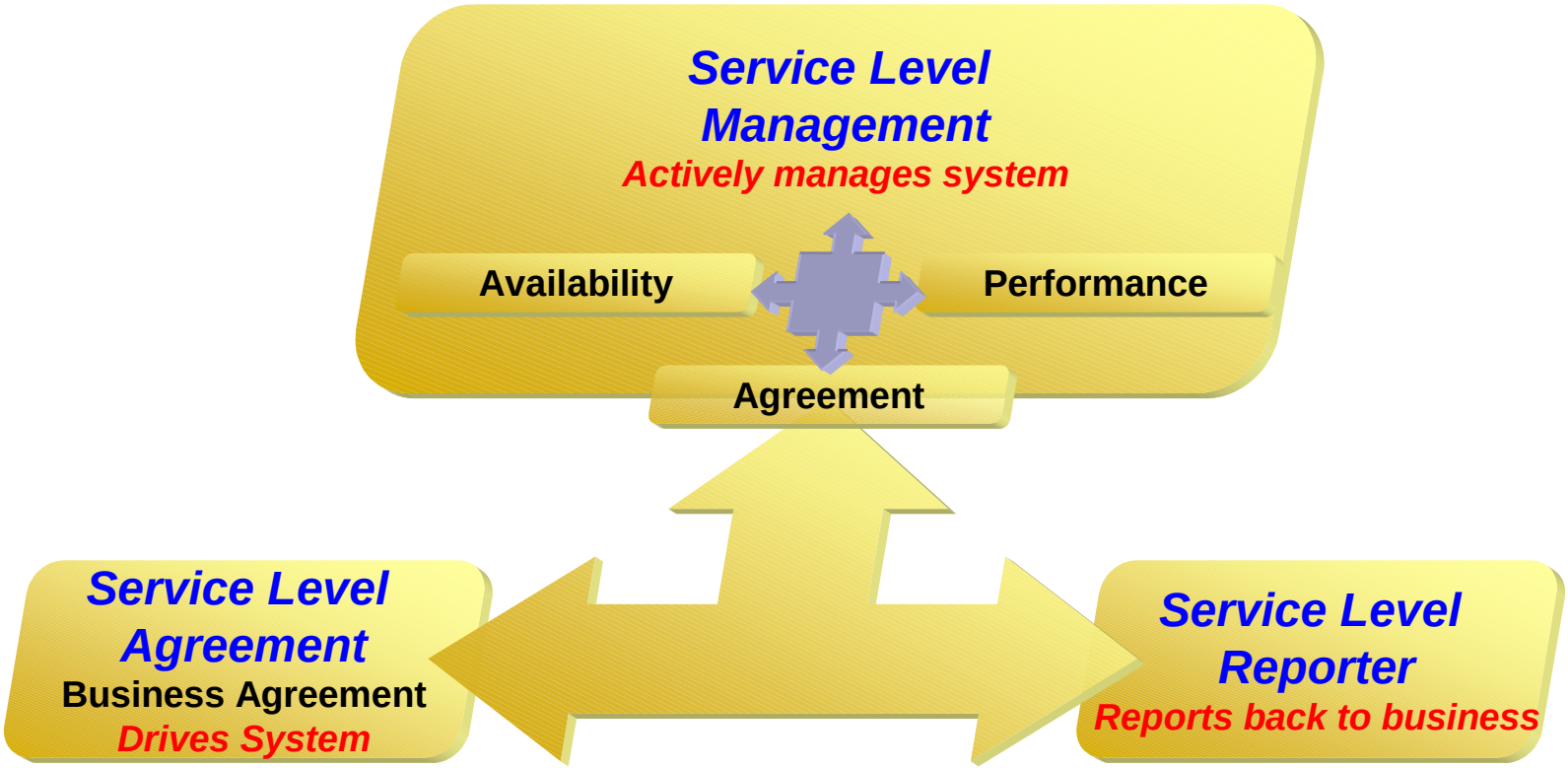
```
type Reliability = contract {
        numberOfFailures : decreasing numeric no / year;
        TTR : decreasing numeric sec;
        availability : increasing numeric;
};

type Performance  = contract {
        delay  : decreasing numeric msec;
        throughput : increasing numeric mb / sec;
};

systemReliability = Reliability contract {
        numberOfFailures < 10 no / year;
        TTR {
                percentile 100 < 2000;
                mean < 500;
                variance < 0.3;
           };
        availability > 0.8;
};
```

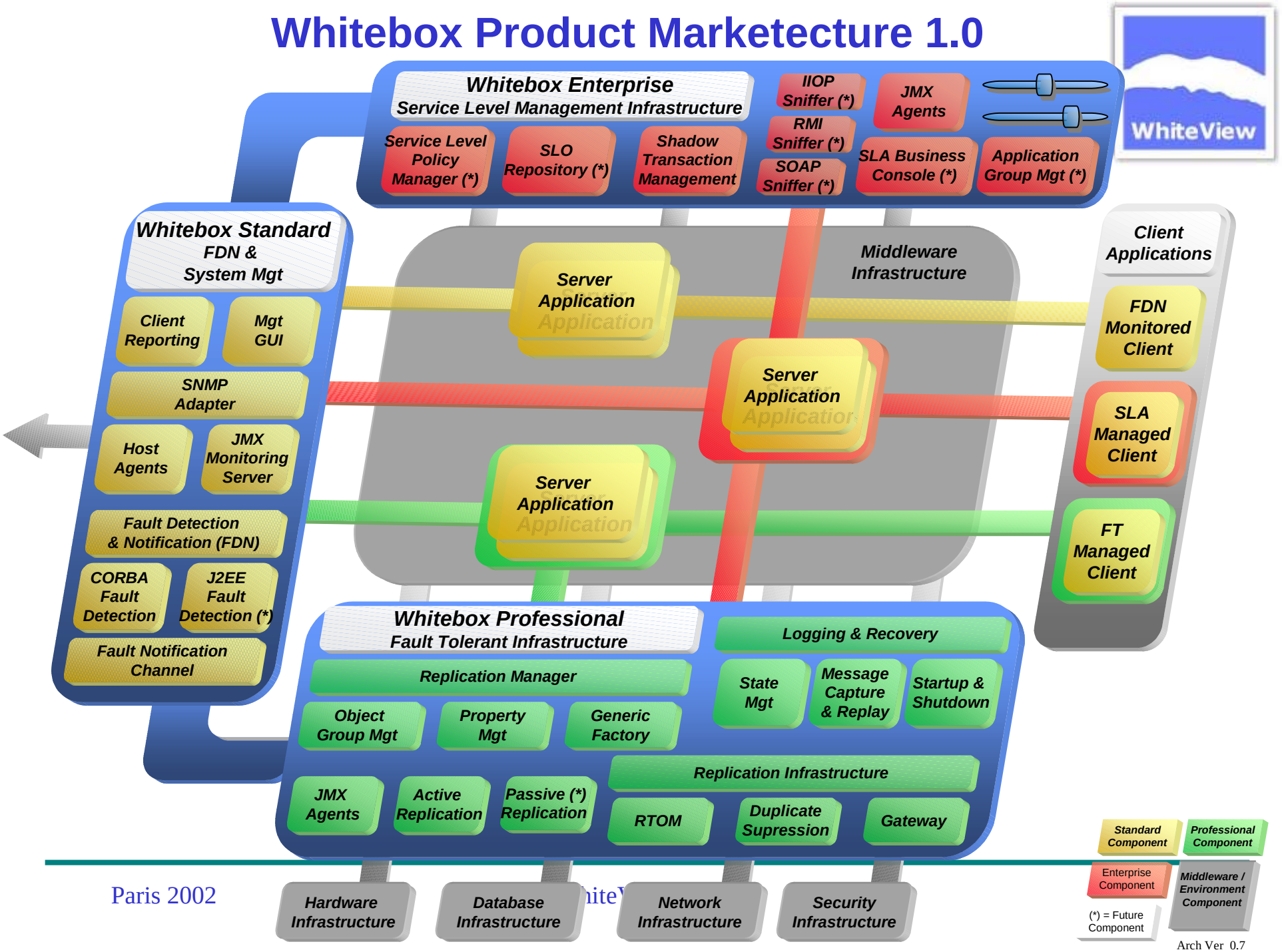
Reference: Svend Frølund, Jari Koistinen, Software Technology Laboratory, HPL-98-10, February 1998

'The How' of
Active Service Level Management

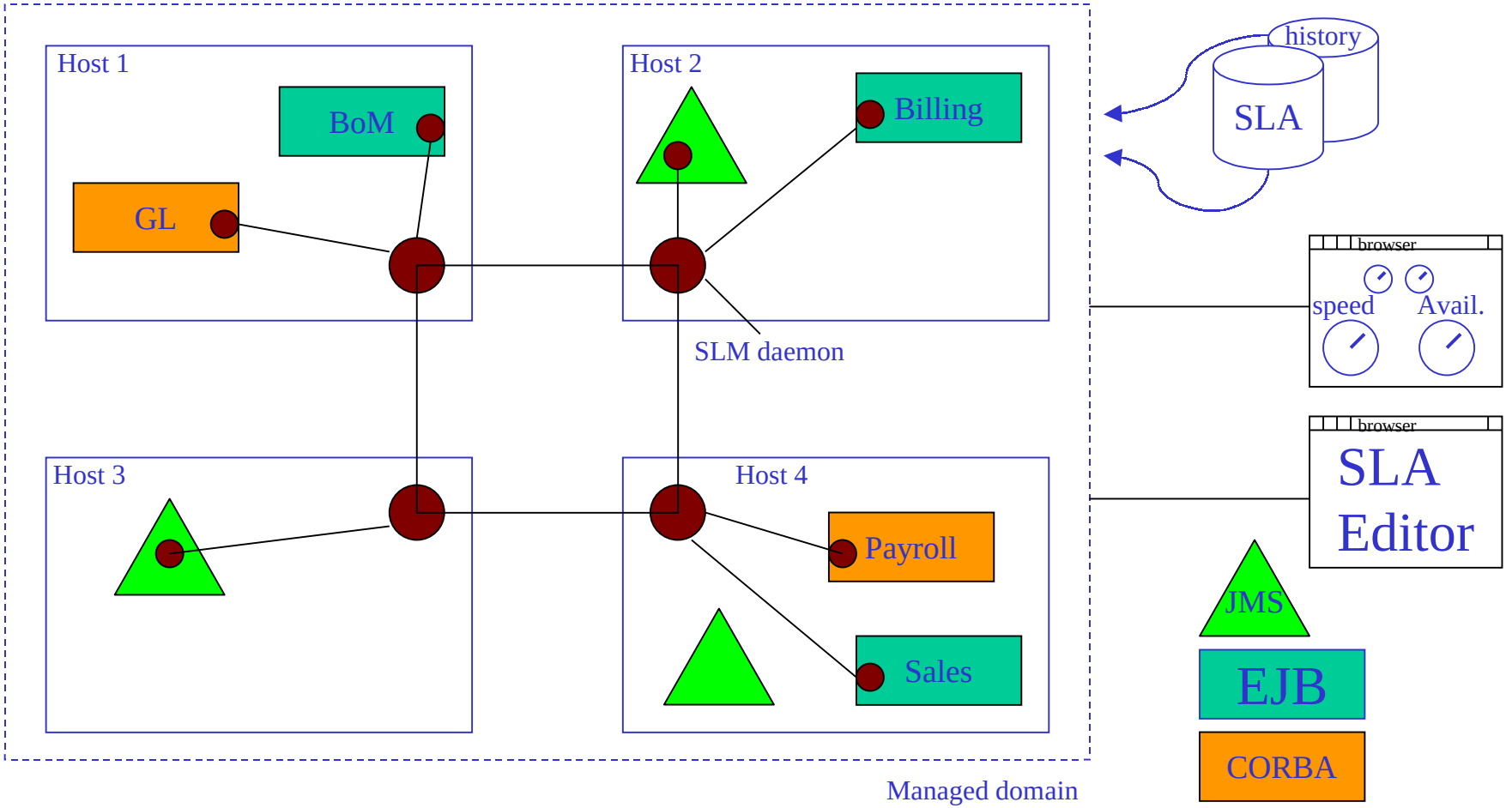


- Standards based Active Service Level Management
- High Availability, Applications focus

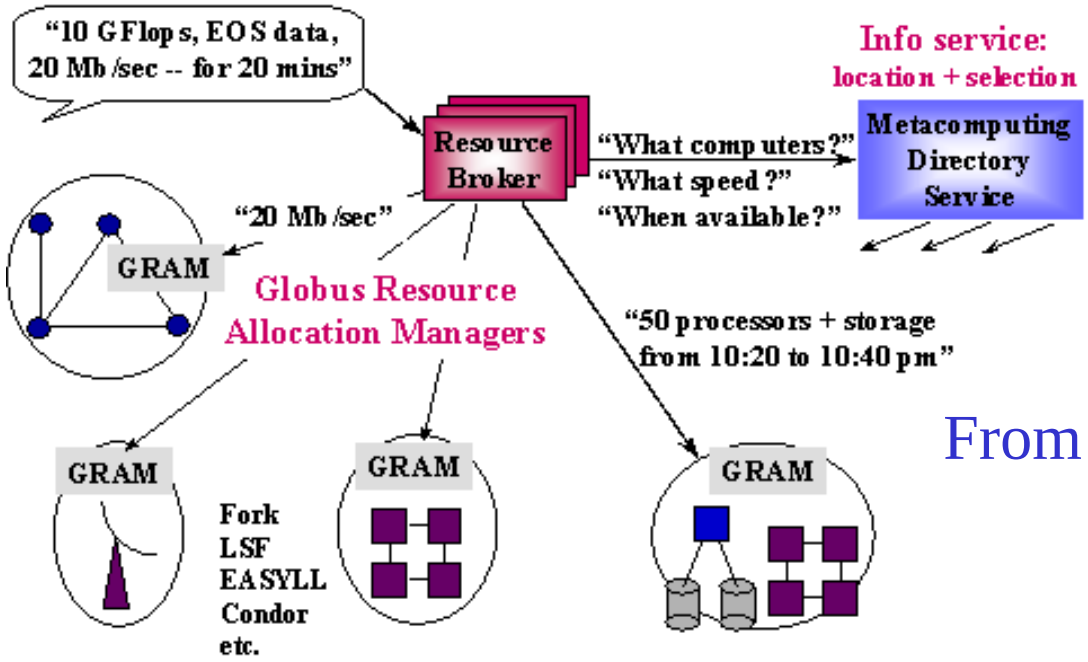
Whitebox Product Marketecture 1.0



My little daemon

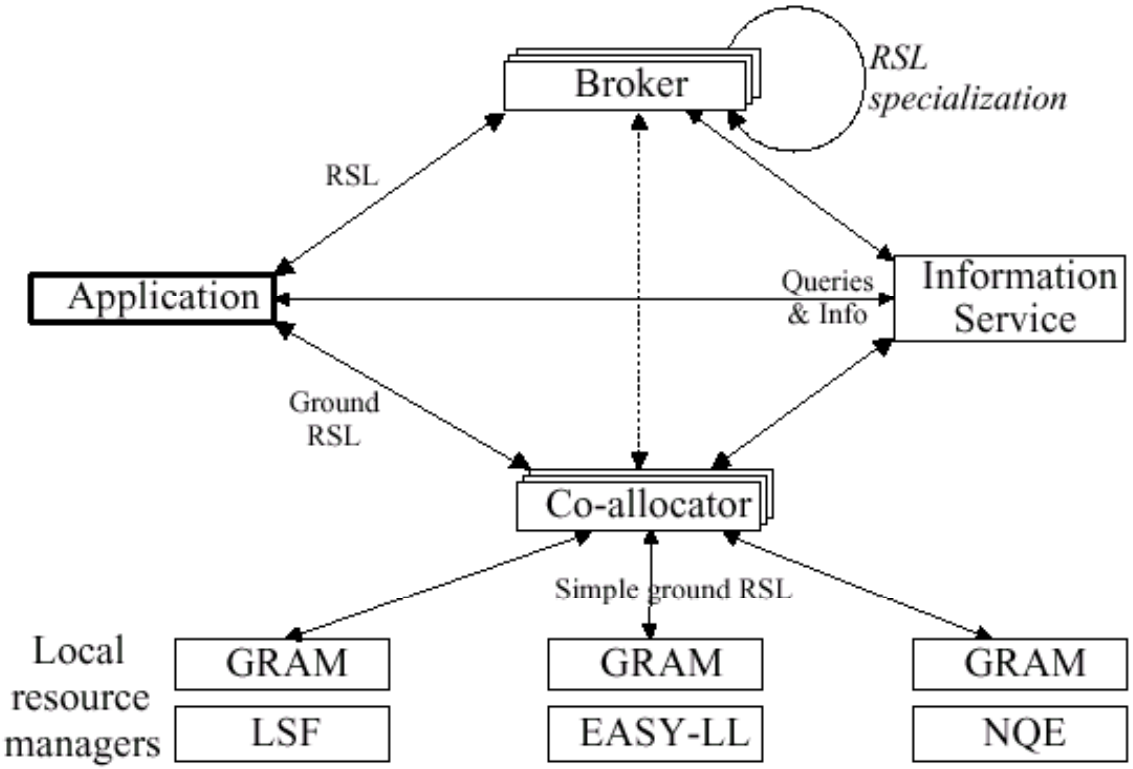


Globus Resource Allocation Manager



From globus.org

Globus Resource Broker



Where we need standards

- Portable SLAs
  - a single SLA for a service should be understood by all components in the chain
- Portable Metrics
  - need to compare apples with apples
  - this may be by component type
- Agreed interfaces
  - each component should be able to request SLA performance information from another