

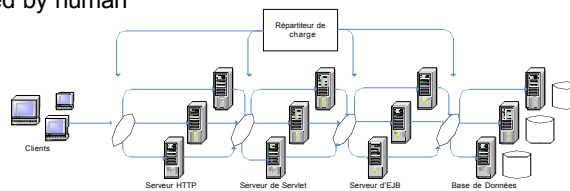
Jade : an autonomic system INRIA Rhone-Alpes

N. De Palma, J. B. Stefani



Problem statement

- Complex distributed system (e.g. clustered J2EE server)
- Management
 - Complex task
 - Achieved by human



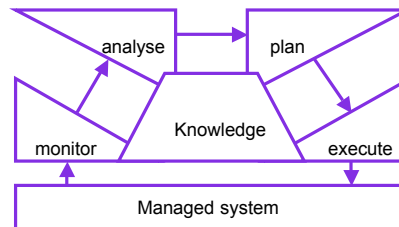
- Consequence
 - Error (mainly configuration)
 - Low reactivity
 - Consume a lot of resources
 - human resources
 - hardware resources (overbooking, estimation in the worst case)

2

Autonomic System



- Jade: Middleware for autonomic management of legacy system
 - Self-configuration
 - Self-healing
 - Self-optimizing
 - Self-protection
- The system embeds control loops
 - Monitor the system, react and reconfigure the system automatically



Requirements for autonomic system

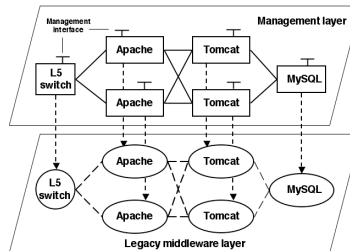


- Deployment
 - Installation
 - Instanciation
 - Configuration
- Monitoring
- Introspection
 - Configuration & structure
- Reconfiguration



Design Principles

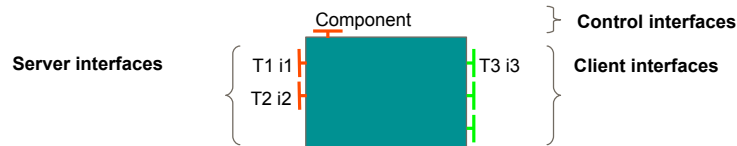
- Architecture-Based management approach
 - High level abstraction to model and control the managed system (legacy) and the management system
- Component model (Fractal - Julia - Think)
 - Wrap legacy managed resource (software/hardware)
 - *An architectural view of a legacy system*
 - *Deployment, Introspection and Reconfiguration mechanisms*
 - *A naming system, A type system, Ensure distributed configuration consistency for legacy complex system*



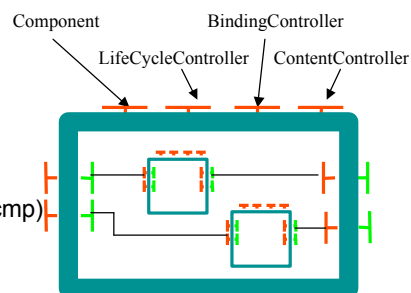
5



Fractal component



- Component
 - Introspection (interfaces)
- LifeCycleController
 - Lifecycle management (start, stop)
- ContentController
 - Content management (add/remove subcmp)
- BindingController
 - Binding management for client interface
- Attributecontroller

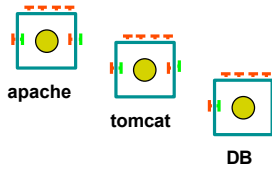


6

Fractal wrapping and ADL description



- Wrapping step (java programming)



(only control aspect is wrapped)

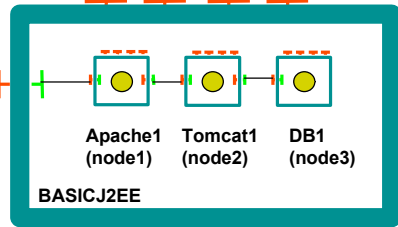
- Complex infrastructure description (ADL)

```
<definition name="BasicJ2EE">
  <interface name="http" role="server" signature="pkg.http" >
    <component name="apache1" definition="apache"/>
    <component name="tomcat1" definition="tomcat"/>
    <component name="DB1" definition="DB"/>
    <binding client="this.http" server="apache1.http"/>
    <binding client="apache1.ajp13" server="tomcat1.ajp13"/>
    <binding client="tomcat1.jdbc" server="DB1.jdbc"/>
  </definition>
```

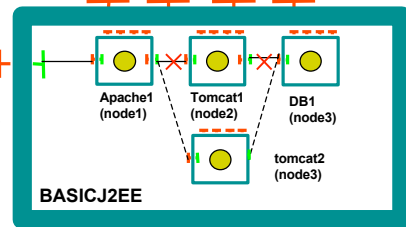
1

2

- Automatic Deployment step (ADL factory)



- Reconfiguration step (java programming)

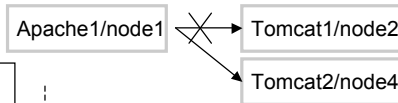


3

Component model abstraction



- Exemple of abstraction
 - AJP13 connection APACHE/TOMCAT



```
1/ log on node1
2/ stop the apache server (script shutdown)
3/ update config file
4/ restart the apache server (script httpd)
```

```
workers.tomcat_home=/tmp/depalma_tomcat_local
workers.java_home=/cluster/java/j2sdk1.4.2_01
ps=/
worker.worker.port=8098
worker.worker.host=node4
worker.worker.type=ajp13
worker.worker.lbfactor=100
worker.list=worker, loadbalancer
worker.loadbalancer.type=lb
worker.loadbalancer.balanced_workers=worker
```

Worker.properties

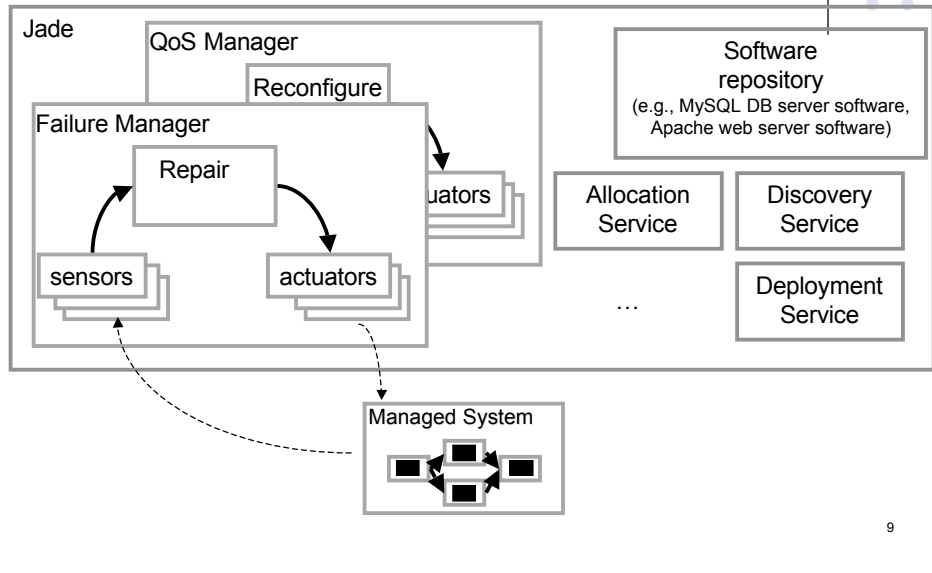
```
1/
Apache1.stop()
Apache1.unbind("AJP13")
Apache1.bind("AJP13",tomcat2.AJP13)
Apache1.start()
```

1'/ change the distributed ADL config
1''/ use a GUI
i.g: fractal explorer

Jade

8

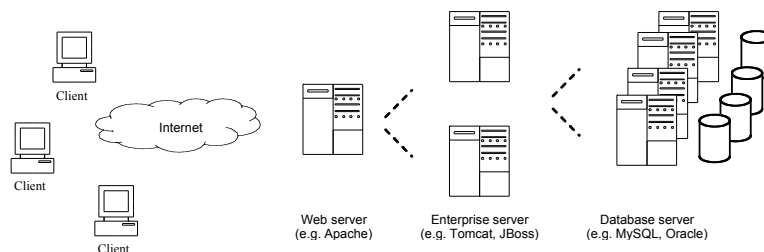
Jade Architecture



9

Self-configuration

- Specify configuration rules for the system to automate its deployment
- Automatically deploy a (distributed) application in the cluster

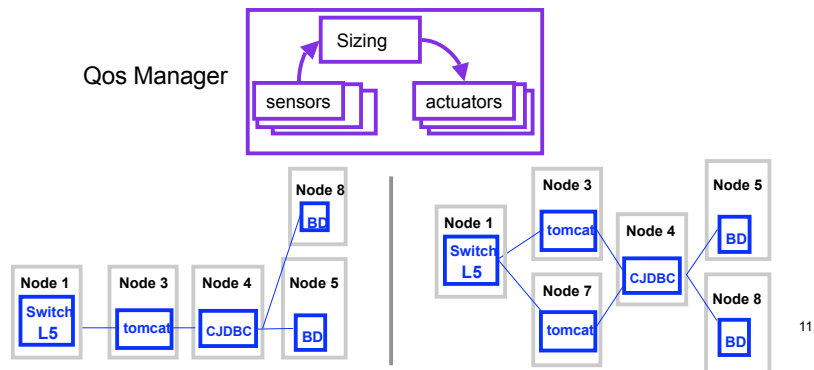


10



Self-optimization

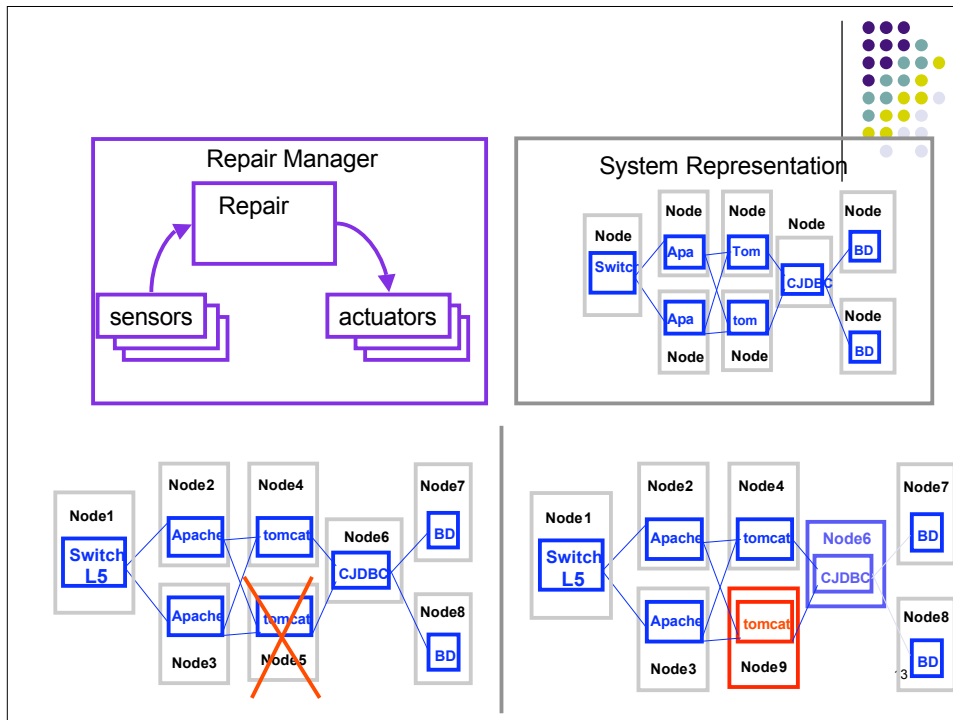
- QoS-based reconfiguration
- Optimize resource usage and performances
 - Dynamically increase/decrease the number of replicas when the server load increases/decreases
 - E.g. online applications with user perceived QoS requirements



Self-healing

- Tackled faults
 - Fail-stop faults (e.g. node failure)
- Structural repair of the distributed application
 - Detection of failure occurrence
 - Which application components are impacted by the failure
 - Redeployment of the failed components on (possibly new) available resources
- Global consistency
 - Hypothesis : Legacy software provides save/restore operations.
 - Consistency is driven by the wrappers
 - (eg : Replicated R/W databases, ensure global consistency upon failure recovery)
- Require a system representation
 - Provides a view of the runtime system architecture, which is isomorphic, introspectable and causally connected to runtime components.

12



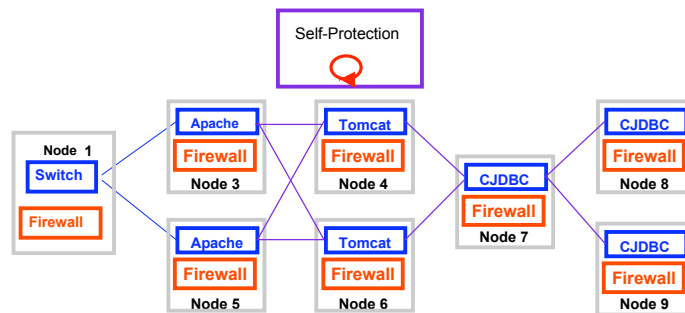
Self-Protection

- Pessimist hypothesis
 - 100% secure is a dream
- Self-protection
 - Not to avoid attacks
 - Not an automated static security configuration
 - But ... to react to attacks while they are still in progress
 - We are looking for the exploitation step



Self-Protection

- Detect and react to anormal behavior
 - The architecture description provides a structural behavior
- Detection is achieved by self-configured firewalls
 - Legacy independent, 0 false-positive, detect well-known and unknown attack
- Different kind of reactions
 - Network isolation, blacklisting, self-destruction and self-healing



15



Conclusion

- Jade is a framework to build autonomic systems
 - Based on fractal
 - Can be used for legacy software
 - First experiment for J2EE cluster
- Current interest
 - Managing P2P system ? Services over p2p system ?
 - A dynamic ADL
 - A wrapping language
 - Management of multiple control loops

16