Hierarchical structure of loosely coupled services which are independent & autonomous entities able to cooperate using a dynamic set of proxies or self-describing protocols.

They need a dynamic registration and discovery & subscription mechanism.

For an effective use of distributed resources, these services should provide adaptability and self-organization (aggregation and hierarchical orchestration).

Reliable on a large scale network distributed environment:
- Avoid single points of failure
- Automatic re-activation of components and services

Scalable & Flexible for adding dynamically new services and automatically replicate existing ones to cope with time dependent load.
“Traditional” Distributed Object Models (CORBA, DCOM)

The Stub is linked to the Client. The Client must know about the service from the beginning and needs the right stub for it.

The Server and the client code must be created together!!
The client can dynamically generate the data structures and the interfaces for using remote objects based on WSDL

**Platform independent**
MonALISA
Design Considerations

Any well suited protocol for the application

Dynamic Code Loading

Services can be used dynamically

- Remote Services   Proxy == RMI Stub
- Mobile Agents     Proxy == Entire Service
- “Smart Proxies”   Proxy adjusts to the client

Act as a true dynamic service and provide the necessary functionally to be used by any other services that require such information (Jini, interface to WSDL / SOAP)

- mechanism to dynamically discover all the “Service Units"
- remote event notification for changes in the any system
- lease mechanism for each registered unit
A Service Registers with at least one Lookup Service using the same ID.

It provides information about its functionality and the URL addressed from where interested clients may get the dynamic code to use it. The Service must ask each Lookup Service for a lease and periodically renew it.

If a Service fails to renew the lease, it is removed from the Lookup Service Directory. When problems are solved, it can re-register.

The lease mechanism allows the Lookup Service to keep an up-to-date directory of services and correctly handle network problems.
Monitoring: Data Collection

- Dynamic Thread Pool
  - SNMP get & walk
  - rsh | ssh remote scripts
  - End-To-End measurements

- Farm Monitor
- Trap Listener
- WEB Server
- Trap Agent (ucd – snmp) perl
- Dynamic loading of modules or agents
- Configuration Control
  - Other tools (Ganglia, MRT…)

- PULL
- PUSH

Tools:
- PULL
- Dynamic loading of modules or agents
- Trap Agent (ucd – snmp) perl
- Configuration Control
- Other tools (Ganglia, MRT…)

- PUSH
Registration / Discovery / Remote Notification

MonALISA Service

Registration

Lookup Service

Discovery

Services Proxy Multiplexer

Client (other service)

Data Filters & Agents

Client (other service)

MonALISA Service

MonALISA Service

MonALISA Service

Services Proxy Multiplexer

Lookup Service

Proxy Multiplexer

Services Proxy Multiplexer

Client (other service)
Service Monitor UNIT & Data Handling

Client (other service) Web client

WEB Service
WSDL SOAP

McKoi DB
MySQL

Other tools

Client (other service) Java

Predicates & Agents

Configuration Control (SSL)

UDP

MySQL

MDS

User defined loadable Modules to write /sent data

Data Cache Service & DB

Monitor Data Stores

Registration

Discovery

Lookup Service

Predicates & Agents

Client

JAVA

Discovery Service Monitor UNIT & Data Handling
Global Client for Farms and Network Connectivity

@ CALTECH
Mobile Agents and Filters

Simple “Global Load” filter agent

Maximum Flow Data Replication
Path Agent Deployed to each RC
and evaluates the best path for
real-time data replication

From FNAL to all

From CERN to all
Pseudo – Clients & Dedicated Repositories

MySQL

IDB

MonaLisa Service

Discovery

Lookup Service

Filter Agents / Data

MySQL

TOMCAT

JSP/servelts

Pseudo Client

Filter Agents / Data

WEB

WAP
Reflectors are hosts that interconnect users by permanent IP tunnels.

The active IP tunnels must be selected so that there is no cycle formed.

The selection is made according to the assumed network links performance.
Barůvka’s Algorithm
Barůvka’s Algorithm
Barůvka‘s Algorithm
Monitoring VRVS Reflectors
Global Client / Dynamic Discovery
SUMMARY

- MonALISA is able to dynamically discover all the "Service Units" used by a community and through the remote event notification mechanism keeps an update state for the entire system.
- Automatic & secure code update (services and clients).
- Dynamic configuration for services. Secure Admin interface.
- Access to aggregate farm values and all the details for each node.
- Selected real time / historical data for any subscribed listeners.
- Active filter agents to process the data and provide dedicated / customized information to other services or clients.
- Mobile Agents for decision support and global optimization.
- Dynamic proxies and WSDL & WAP pages for services.
- Embedded SNMP support and interfaces with other tools (LSF, PBS, Ganglia, Hawkeye, IEPM-BW...)
- Dedicate pseudo-clients for repository, WAP access or decision making units.
- It proved to be a stable and reliable distributed service system. It is currently running at ~100 sites.

http://monalisa.cacr.caltech.edu