



---

# Sharing Models/Simulations/Analysis: A Unifying Approach

**Norm Lamarra & Julia Dunphy**

Engineering & Communications Infrastructure, JPL Section 366

June 2001

with contributions from Dan Crichton et al. (Section 389)



# Introduction/Background



- JPL need: improved engineering lifecycle support
  - ⇒ Team access to distributed models/tools/simulations
  - ⇒ Unifying data architecture & execution framework
- Approach:
  - Attempt to automate *process* flow
    - e.g., creation/investigation of an engineering design
      - multidisciplinary analysis of performance, cost, risk
      - exploration of design alternatives
  - Capture, reuse, iteration, optimization of process
    - Graphical "*methogram*"
    - Distributed heterogeneous components
  - Management of data products
- Prior development supported by:
  - NASA Code S (CETDP)
  - DARPA (RaDEO)



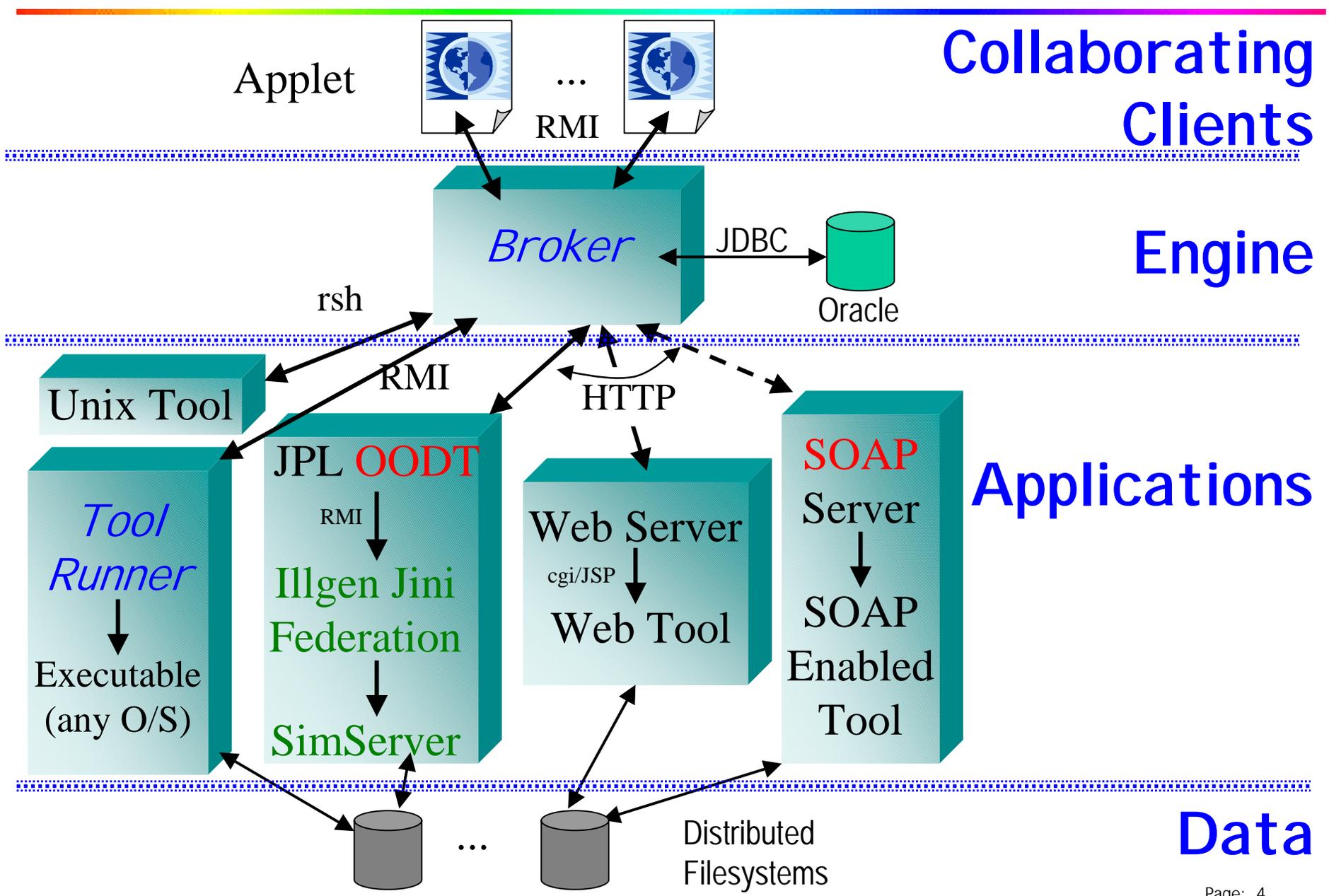
# Evolution of Software Architecture



- Original: C++/X single application using PVM
- Then: 3-tier single-user distributed application:
  - Java GUI , CORBA connection to C++ Engine, PVM back end
- Last year: All-Java 3-tier web-based application
  - *Broker* is the collaboration engine (server):
    - handles multiple clients concurrently
      - each client interacts with Engine and views its state
    - mediates access to "tools"
      - single user (private) or collaborative teams (shared)
  - *Client* is an applet
    - downloadable from the server via HTTP
    - connects back to the broker using Java/RMI
  - *Tool Runner* can encapsulate remote tool
    - also connects to Engine via RMI
    - handles all I/O between Engine and tool



# Software Architecture 2000



Collaborating Clients

Engine

Applications

Data



# Evolution to EJB



- Now: EJB/JSP 3-tier web-based application
- *Broker* is the EJB collaboration server:
  - handles multiple clients concurrently
    - each client interacts with Engine and exchanges states
- *Client* is an applet
  - downloadable from the server via HTTP
  - connects back to the broker using Java/RMI
- *RunnerBeans* are managed by the Broker:
  - Are instantiated by the Engine via RMI
  - Handle all I/O between Engine and “tool” (i.e., proxy)
  - Use JSP to activate “tool”



# Software Architecture 2001

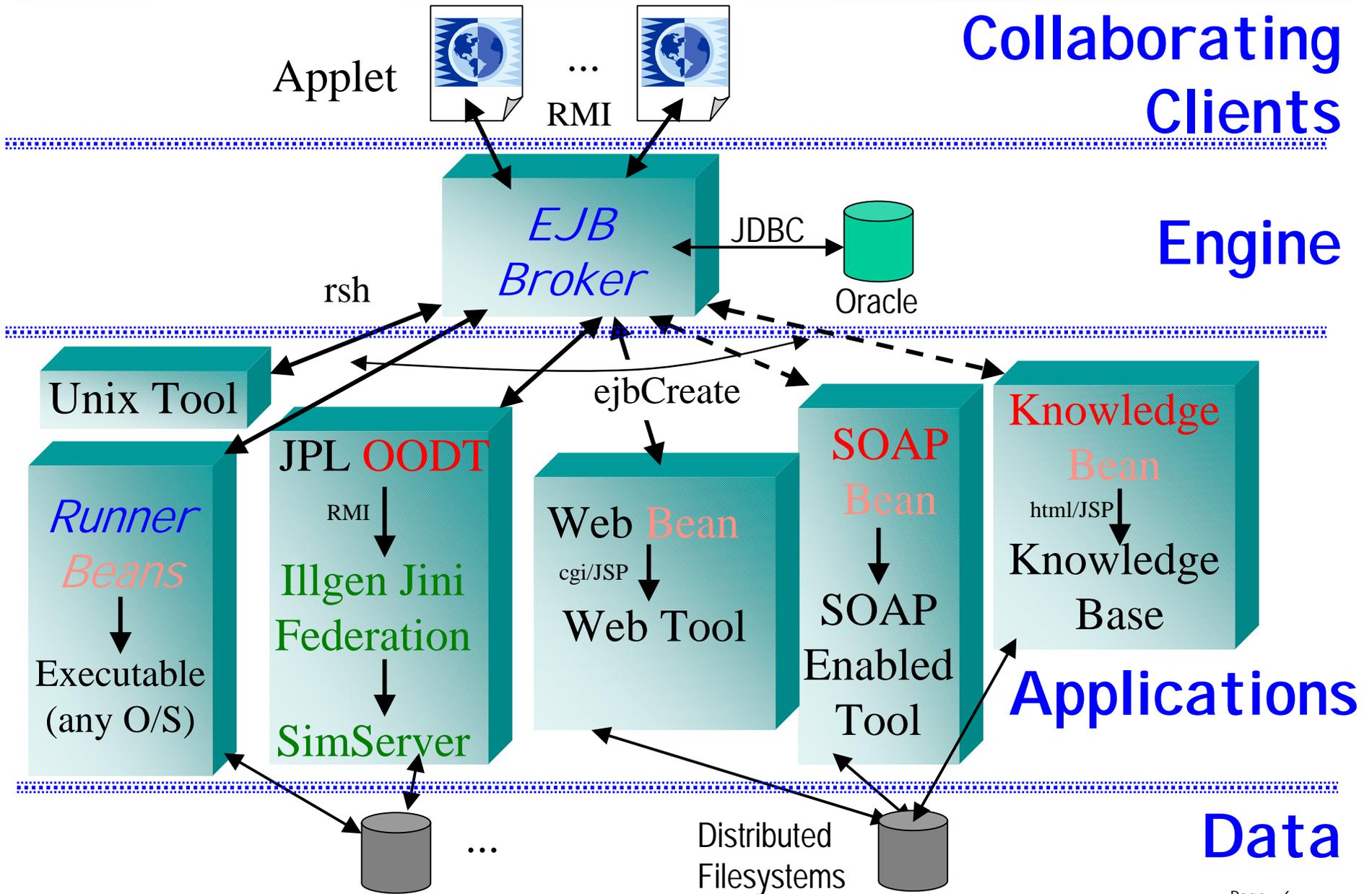


## Collaborating Clients

## Engine

## Applications

## Data





# "Methogram" Creation Wizard



The screenshot shows the Methedit application window with a Project Wizard dialog box open. The dialog box contains the following text and fields:

Project Wizard will guide you through creating a project.  
You may quit the wizard at any time by pressing "Finish"

Project name: Undefined

Project integrator: julia

Multiple users:  Yes

Multiple hosts:  Yes

Uses web tools:  Yes

Allow other users to browse this project:  Yes

Buttons: Next, Finish

The background interface includes a menu bar (Projects, View, Flows, Help), a toolbar, a Palette of icons, and several panels: Tools (Calculator, Web, DB, Analysis, SOAP), Network (table with columns: Archit., Node, User, Pass, AFS, RPC, Status), Executer, and Roles. A 'Save' button is visible in the Roles panel.

Archit.	Node	User	Pass	AFS	RPC	Status
INT	ldun	julia	****	true	true	...



# "Methogram" Editing Session



The screenshot displays the Methedit application window. At the top, the menu bar includes 'Projects', 'View', 'Flows', and 'Help'. Below the menu, the user 'julia' is logged in, and the project is 'Aeroshell'. The main workspace shows a workflow diagram with nodes such as 'Logon', 'SetGeometry', 'ExecOptions', 'TableExtractor', 'SubOptions', 'Maximizer', 'Managers', 'AddressFetcher', and 'E-Mail'. A text overlay on the right side of the diagram reads 'Nodes can be Hierarchical'. The interface also features a 'Methogram Tree' on the left, a 'Palette' of icons, and several panels at the bottom: 'Tools' (with a 'SOAP Tool0' list), 'Network' (a table of network nodes), 'Executer' (with flow control options like 'Reset', 'Single Step', 'Slow', 'Fast'), and 'Roles' (a table of role information).

Archit	Node	User	Pass	AFS	RPC	Status
NT	jdun...	julia	*****	true	true	resp...
Unix	rock...	julia	*****	true	true	resp...
NT	dstla...	julia	*****	true	true	resp...
Unix	nutri...	julia	*****	true	false	resp...
Web	spac...	julia	*****	false	false	resp...
Web	vab0...	julia	*****	false	false	resp...

Role Name	Locator	Authorized Users
ProjectIntegrator	Aeroshell	julia



# Features of Approach



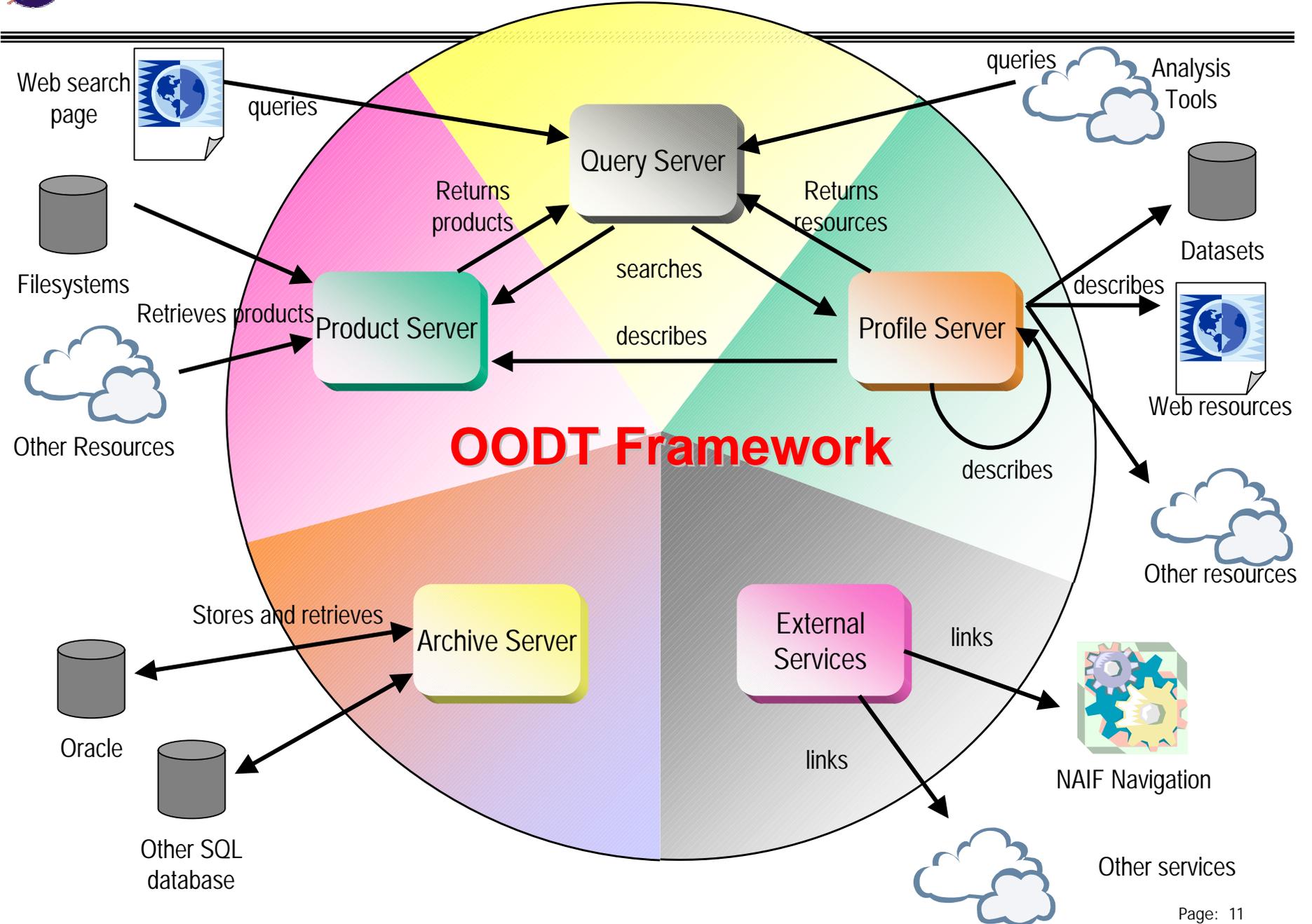
- “Tools” may be located on any network platform
  - Executable Legacy codes using file input/output
    - if source code available, it can be wrapped semi-automatically
  - Codes reachable by a URI or a SOAP interface
  - Web-based via EJB/JSP or cgi/servlet
  - Jini model federation (Army collaboration)
  - COTS products with vendor-supplied API
- Utilizes JPL’s **OODT** for data management
  - Methograms & Metadata stored in Oracle database
- Results can be stored in AFS/NFS
  - Data/notifications can be sent automatically by e-mail
- Graphical Control flow “*methogram*”
- All-Java implementation
  - Currently addressing security (RBAC, privacy, PKI, file)



# Data Management: OODT Framework



- **Object-Oriented Data Technology (JPL):**
  - Distributed system for disparate data management
  - Cross-discipline data search, analysis, and automatic correlation
  - <http://oodt.jpl.nasa.gov>
- **Components:**
  - Profile service - describes resources
  - Product service - retrieves data products
  - Query service - crawls through distributed profile and product servers, discovering results
  - Other services:
    - Archive service for storing products
    - Bridges to external services for extra features

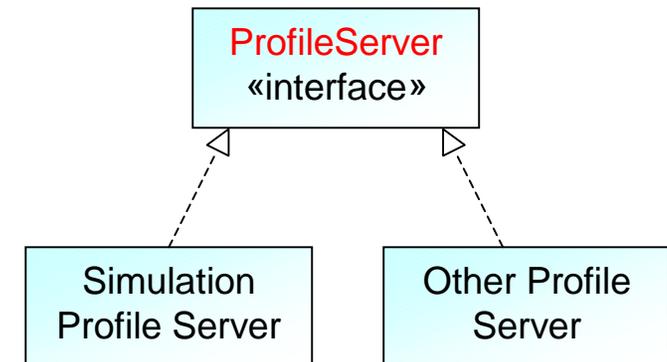




# Using Profile Server for Simulation



- Profile Server:
  - Serve profiles based on query criteria
  - CORBA-accessible interface
- Simulation Profile Server:
  - A profile server that communicates with a Jini simulation federation
  - Accepts a profile query
  - Delivers profiles that describe:
    - What simulations, models, and other objects are available
    - Or, the configuration parameters of a single model



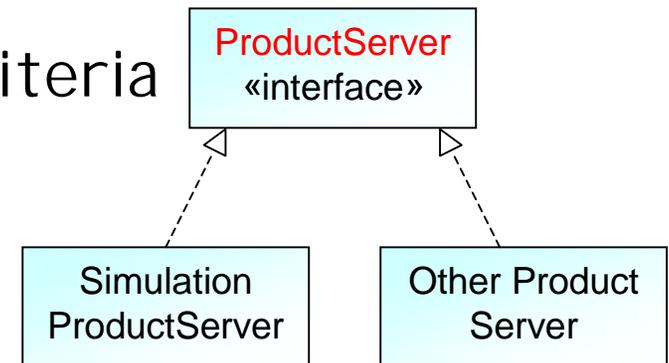


# Using Product Servers for Simulation



- Product Server:

- Serve products based on query criteria
- CORBA-accessible interface
- Results:
  - translated into standard format
  - Based on Internet MIME types
  - Encoded into XML
  - May be text, images, documents, Java objects, ...



- Simulation Product Server:

- A product server that communicates with a **Jini** simulation federation
- Accepts a product query
  - Query contains model configuration parameters
  - Runs the model
  - Gathers its result
- Returns the result in standard format



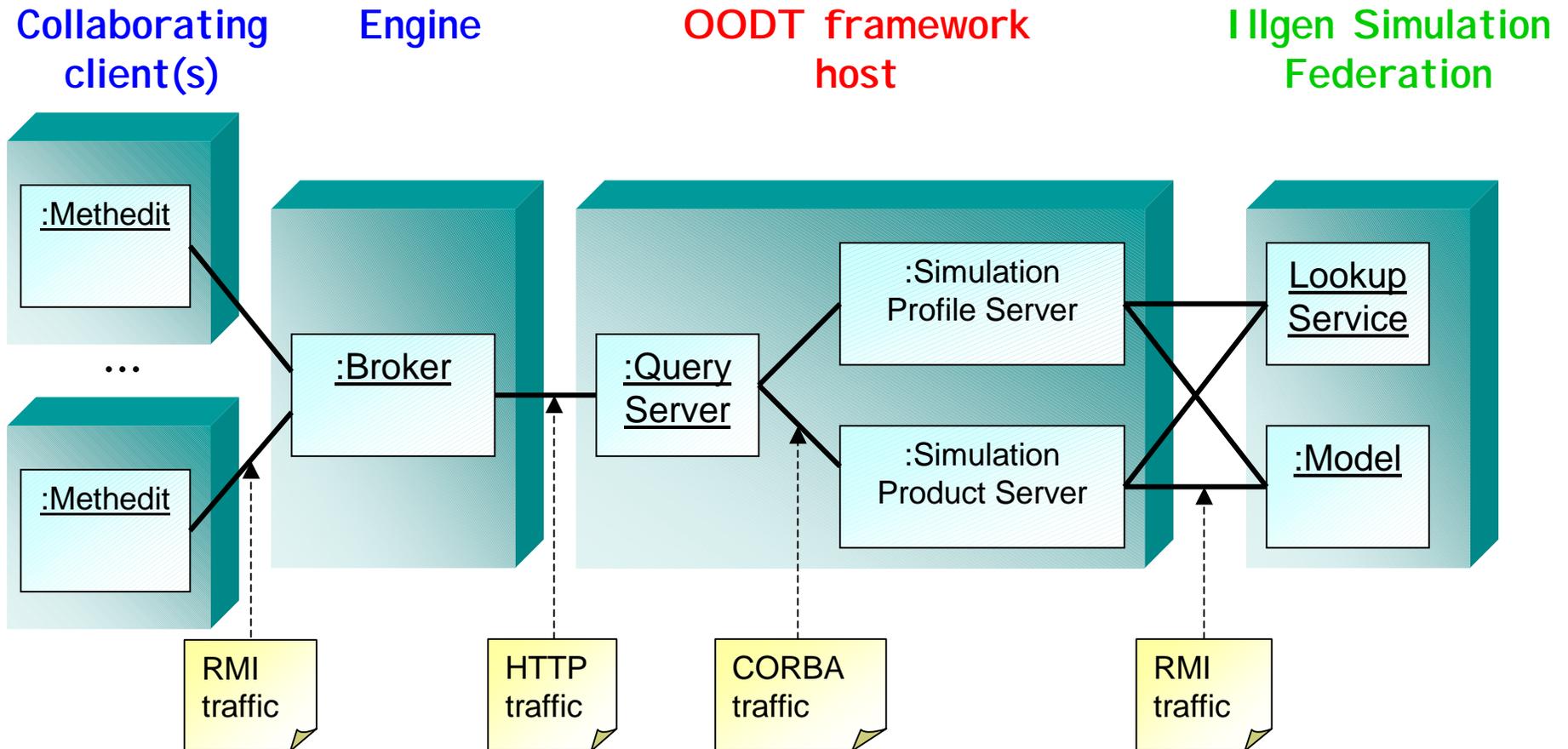
# Integration of Application



- **Engine** can dynamically create and edit a collaborative distributed application “on the fly”:
  - Using direct access to remote tools (JSP, etc.)
  - Using **OODT** Framework servers to configure, select, and run a simulation or model in a **Jini** federation (e.g., **Illgen's**), e.g.:
    - During methogram build:
      - User queries **OODT Profile Server** for list of available models
      - User selects desired model (e.g., **NVTherm**)
      - Interface is retrieved & displayed to user via (generated) form
      - User connects inputs/outputs to methogram
    - For execution: **Engine** uses **OODT Simulation Product Server**
      - Runs model & handles I/O to **Illgen's SimServer**
- **Engine's** only back-end interface is HTTP
  - XML documents passed back and forth over HTTP connection
    - Documents are profile or product queries and their results



# Example Interoperability Prototype



*Heterogeneous components  
connected re-usably on-the-fly...*



# Future Directions



- Security
  - Integration with PKI
    - Role-based access control based on X.509 certificate
  - Distributed resource access management
- Data Management
  - Model Repository
    - Configuration Management
  - Archive products
- Knowledge Management
  - Capturing rationale
  - Locating “similar” work
  - Process Integration/Improvement
- Focus on eNASA



# Background: NASA Initiatives



- eNASA:
  - Enhance key mission, program, and project activities by providing revolutionary tools and services to solve critical problems;
  - Transform the way we work by enabling collaboration within and beyond NASA regardless of time, distance, or technology
  - Enable the Agency to conduct its business more efficiently and effectively
  - Leverage our workforce and other corporate resources by minimizing the duplication of efforts for shared/common services & capabilities;
  - Increase employee productivity by facilitating the availability of customizable interfaces to enhance information delivery, foster person-to-person communication, and increase awareness of NASA significant events; and
  - Help the Agency to demonstrate value to our customers, partners, and members of the public by enhancing external communications and information delivery.
- IT Security:
  - Controls access to shared resources based on authorization & team roles
  - Secures connections between users and distributed resources
  - Ensures that even (apparently) ad-hoc connections are managed via existing security infrastructure