



# ***The Ibis Project or Grids As Promised***

***(an Introduction)***

***Jason Maassen      Rob van Nieuwpoort***  
***Niels Drost      Roelof Kemp      Frank Seinstra***



***ibis@cs.vu.nl***  
***<http://www.cs.vu.nl/ibis>***



vl·e

# ***Grid Computing***

## ***(our definition)***

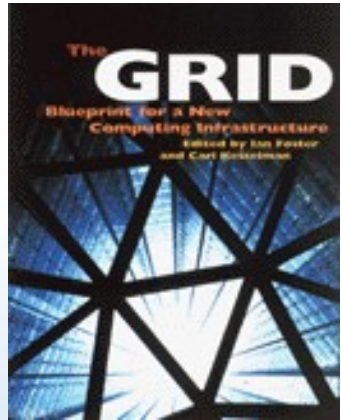
- A collection of *clusters*
- Often a '*social collection*'
  - Access often provided by (former) colleagues, project partners, etc
- Several administrative domains
  - Difference in configurations, security settings and level of maintenance
- May include desktop systems
  - Used for visualization, monitoring, steering



# ***Parallel computing on Grids***

- Mostly limited to
  - trivially parallel applications (parameter sweeps, master/worker)
  - applications that run on one cluster at a time
  - use grid to schedule application on a suitable cluster
- Our goal: run efficient, high-performance parallel applications on a large-scale grid, using co-allocated resources





# ***The 'Promise of the Grid'***

Efficient and transparent (i.e. easy-to-use) wall-socket computing over a distributed set of resources [Sunderam ICCS'2004, based on Foster/Kesselman]

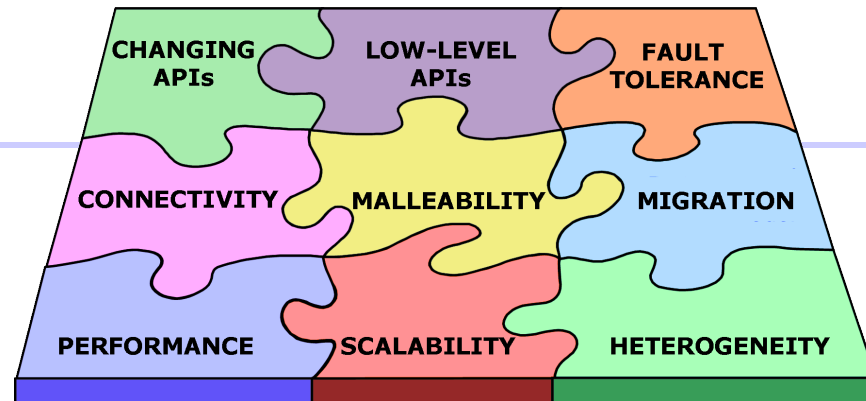


# Reality: 'Problems of the Grid'

- Performance & scalability
- Heterogeneous
- Low-level & changing programming interfaces
- Connectivity issues
- Fault tolerance
- Malleability



User



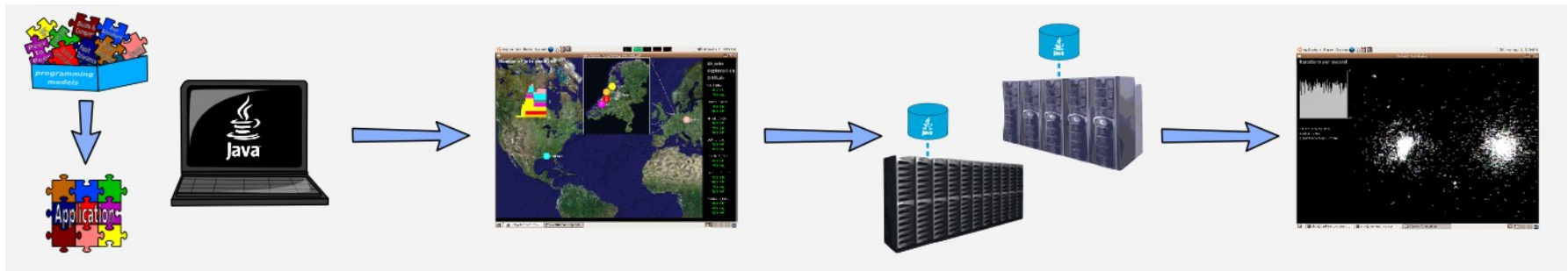
Wide-Area Grid Systems

- *writing & deploying grid applications is hard*



# ***Ibis: Grids As Promised***

- Goal: offer all the functionality needed to create and run grid applications.
- Java based, so highly portable
- Designed for dynamic/hostile grid environments
- Modular and flexible: can replace Ibis components by external ones, including native code





# ***Grid HOWTO***

- Step 1: Create a user friendly Grid
  - Resource discovery
  - Access to resources
  - Security (passwords, proxies, etc)
  - Connectivity (firewalls, NATs, etc)
- Step 2: Use the Grid
  - Communication Library
  - Programming models



# ***Creating A User Friendly Grid: JavaGAT***

- Java Grid Application Toolkit
- Make applications independent of underlying grid
- Used for file copying, resource discovery, job submission & monitoring, user authentication
- API is currently standardized (SAGA)





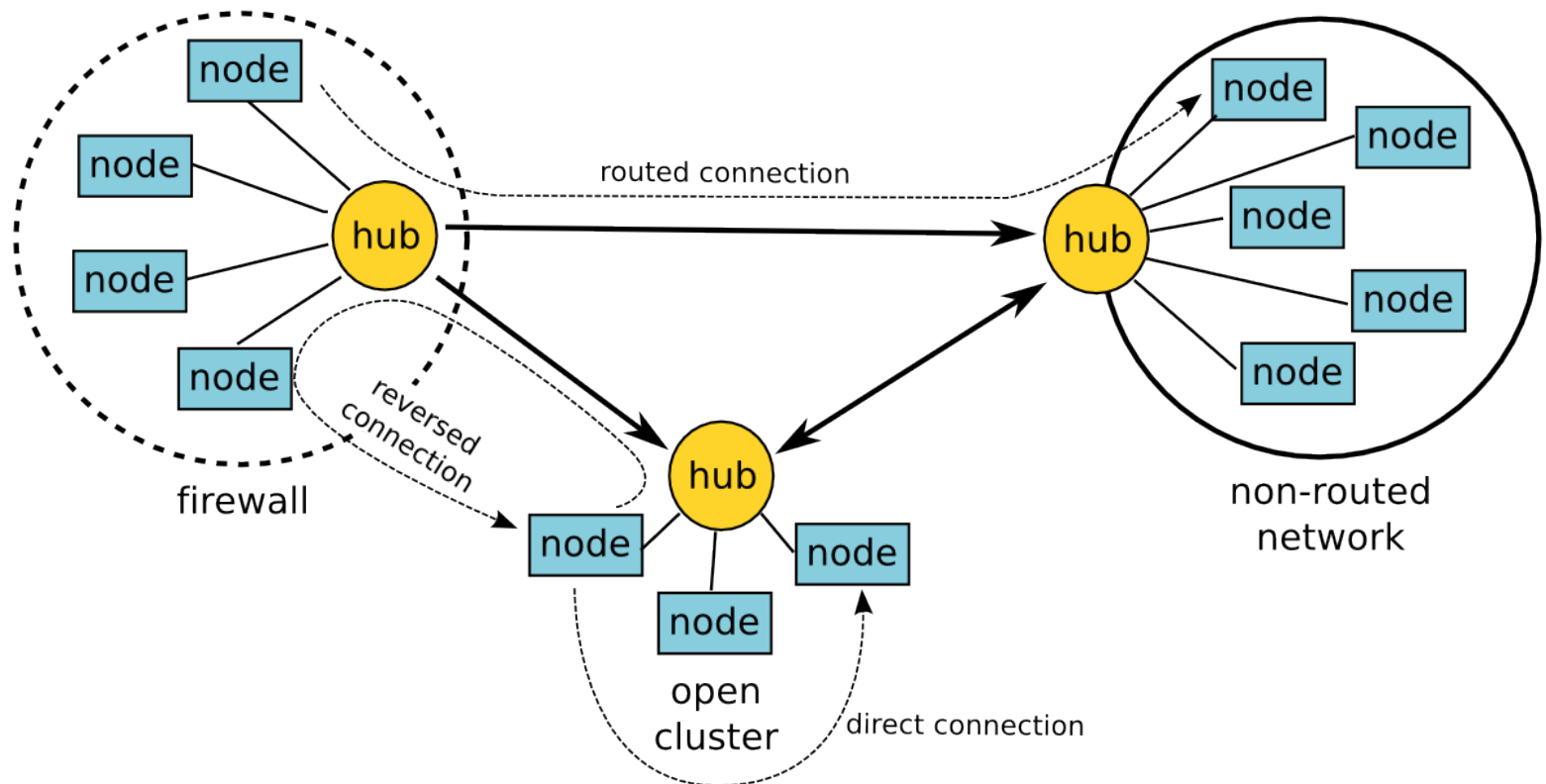
# ***Creating A User Friendly Grid: Zorilla***

- Peer-to-Peer Grid middleware
  - Supports running jobs on “random” collection of machines
- Gossip based overlay network
- Flood-scheduling mechanism
- Job management



# Creating A User Friendly Grid: SmartSockets

- Solve connectivity problems automatically (firewalls, NAT, addressing problems)



# ***Using The Grid: IPL (Ibis Portability Layer)***

- Java-centric “run-anywhere” communication library
- Easy to use
  - Simple yet powerful model
  - Flexible
- Efficient communication
  - Highly optimized Object serialization
- Resource tracking
  - Keep track of resources in “pool”



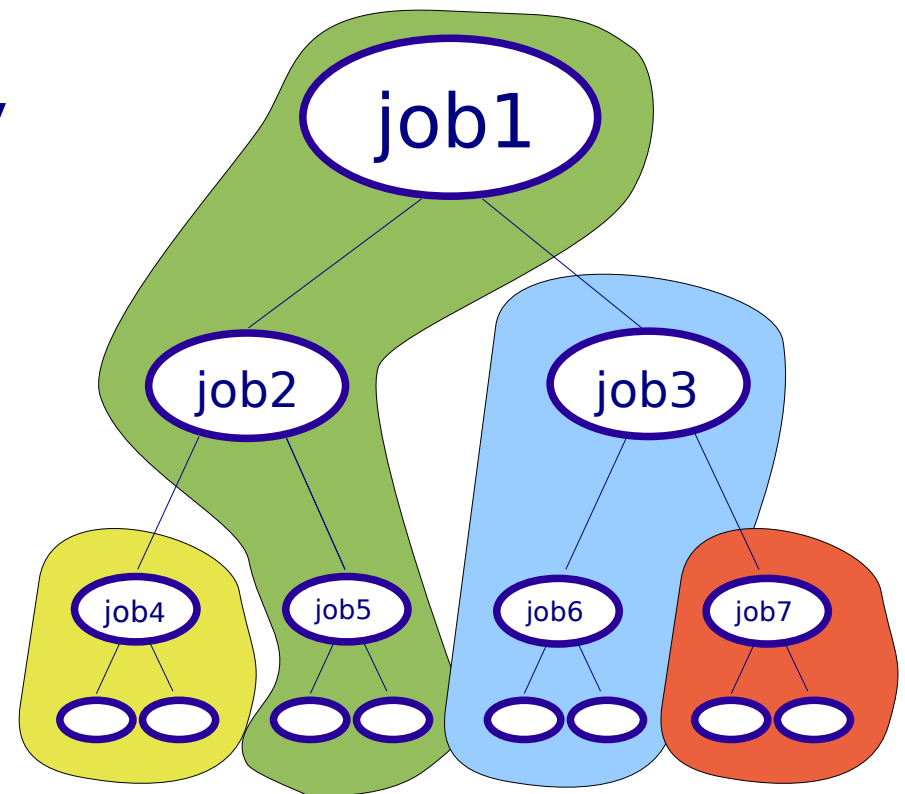
# ***Using The Grid: Programming Models***

- Remote Method Invocation (RMI)
- Group Method Invocation (GMI)
- MPJ (MPI Java 'standard')
- Horus (User Transparent Parallel Multimedia Processing)
- Satin (Divide & Conquer)

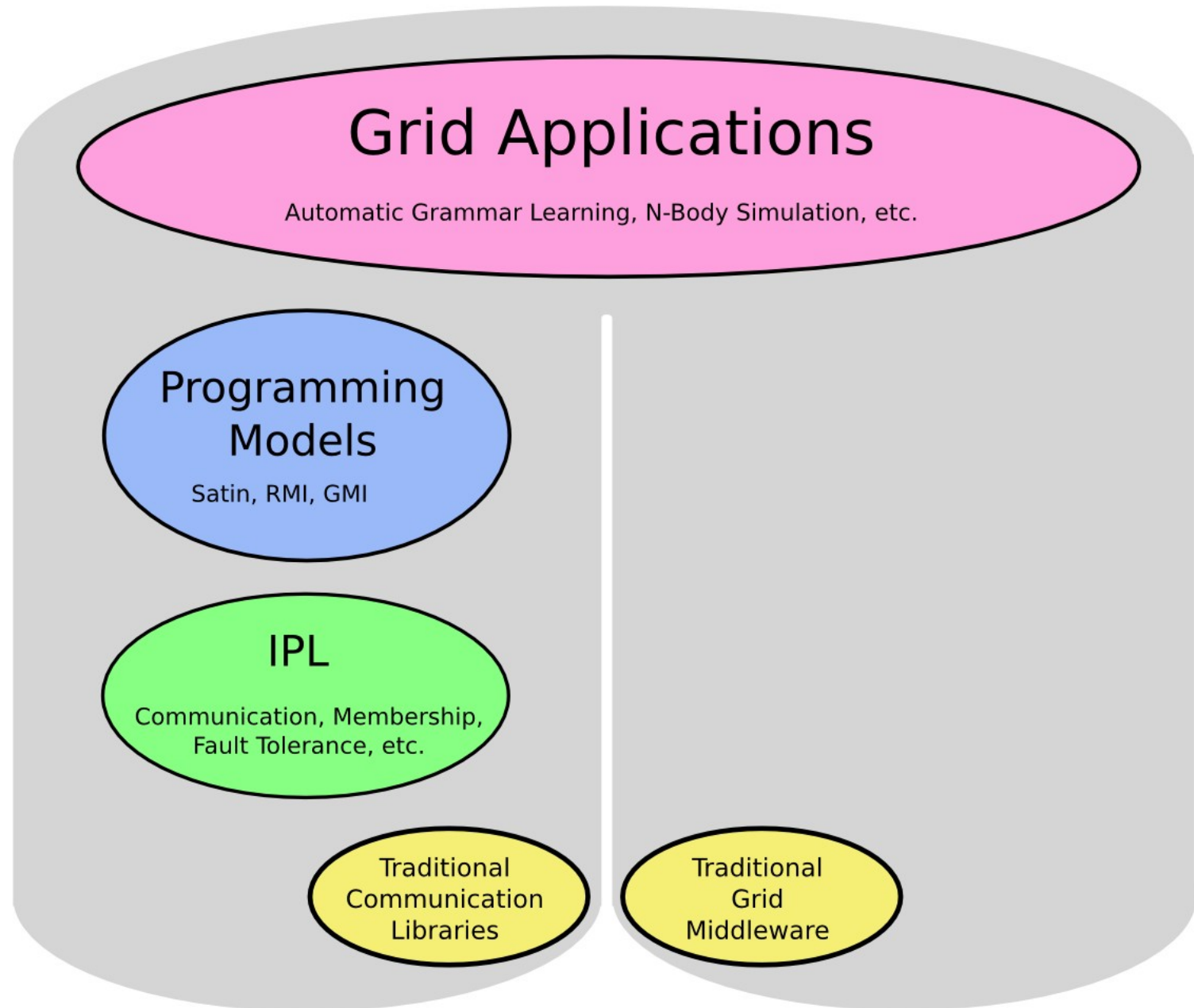


# ***Satin: Divide-and-Conquer***

- Divide-and-conquer programming model
- More general than master/worker
- Cilk-like primitives (spawn/sync) in Java
- Supports malleability and fault-tolerance

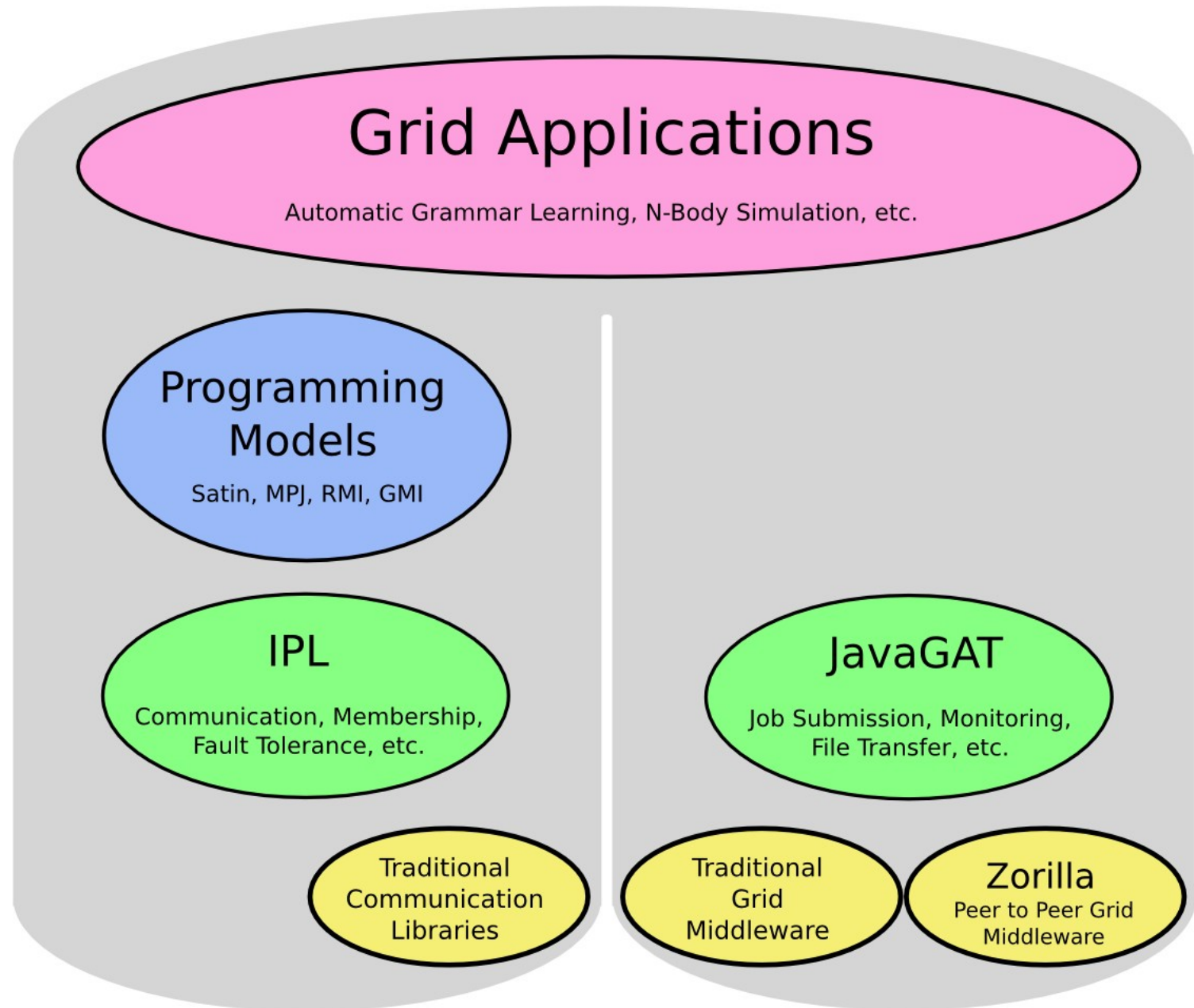


# ***Ibis Overview, C.A. 2004***

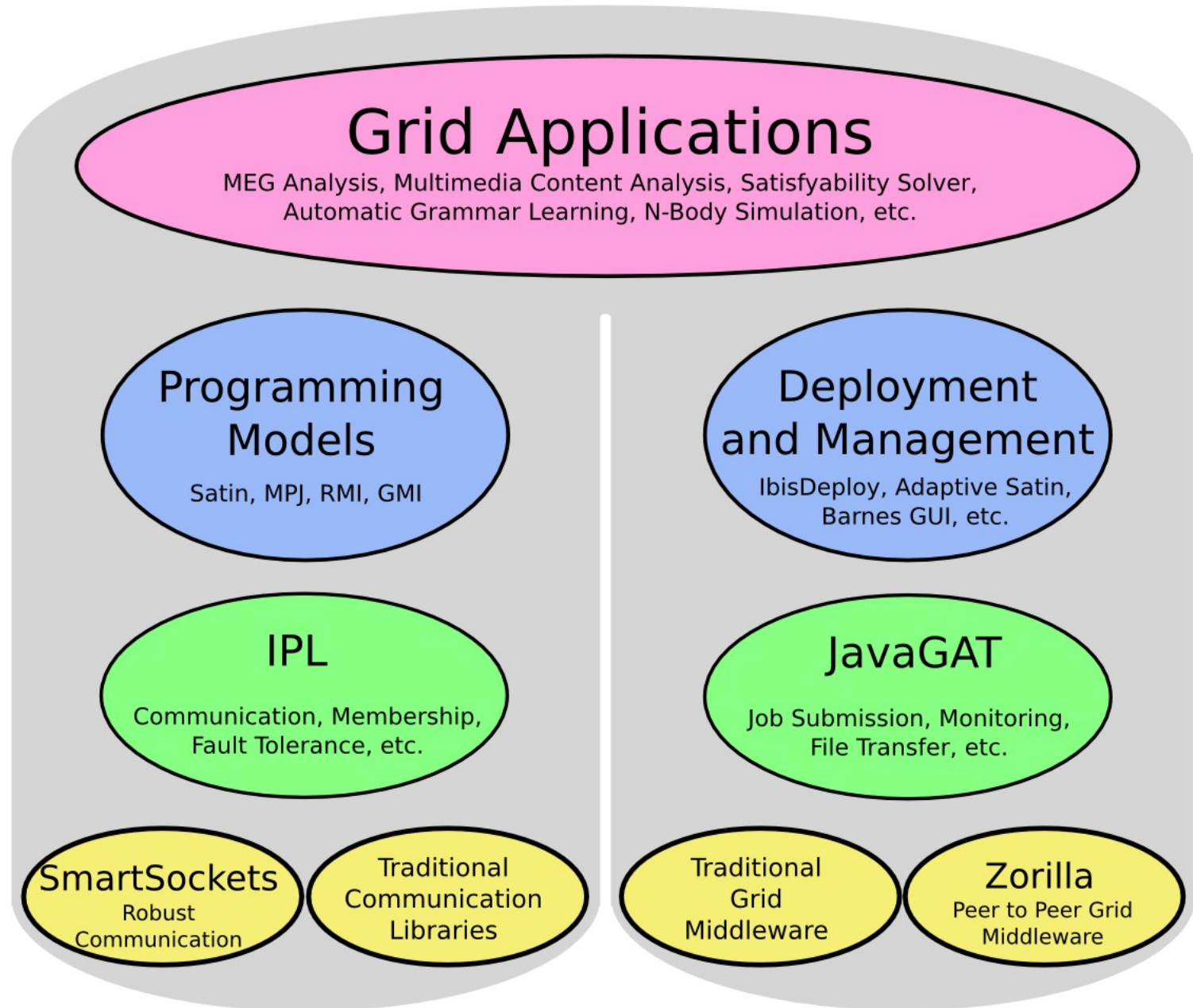




# ***Ibis Overview, C.A. 2006***



# ***Ibis Overview, Current***



# ***Questions ?***

**ibis@cs.vu.nl**

**downloads, more info,  
slides for today:**

**[www.cs.vu.nl/ibis](http://www.cs.vu.nl/ibis)**



# ***Program for Today***

- Introduction
- Creating a user friendly grid
  - JavaGAT
  - Zorilla
  - SmartSockets
- Lunch break
- Using the grid
  - IPL
  - Satin
  - Other Programming Models
- Real world examples

