



The Zorilla Peer-to-Peer Middleware system

Niels Drost

niels@cs.vu.nl



vrije Universiteit



vl·e

Current Middleware

- Hard to install
- Hard to maintain
- Centralized (not very fault tolerant)
- Usually no Co-Allocation

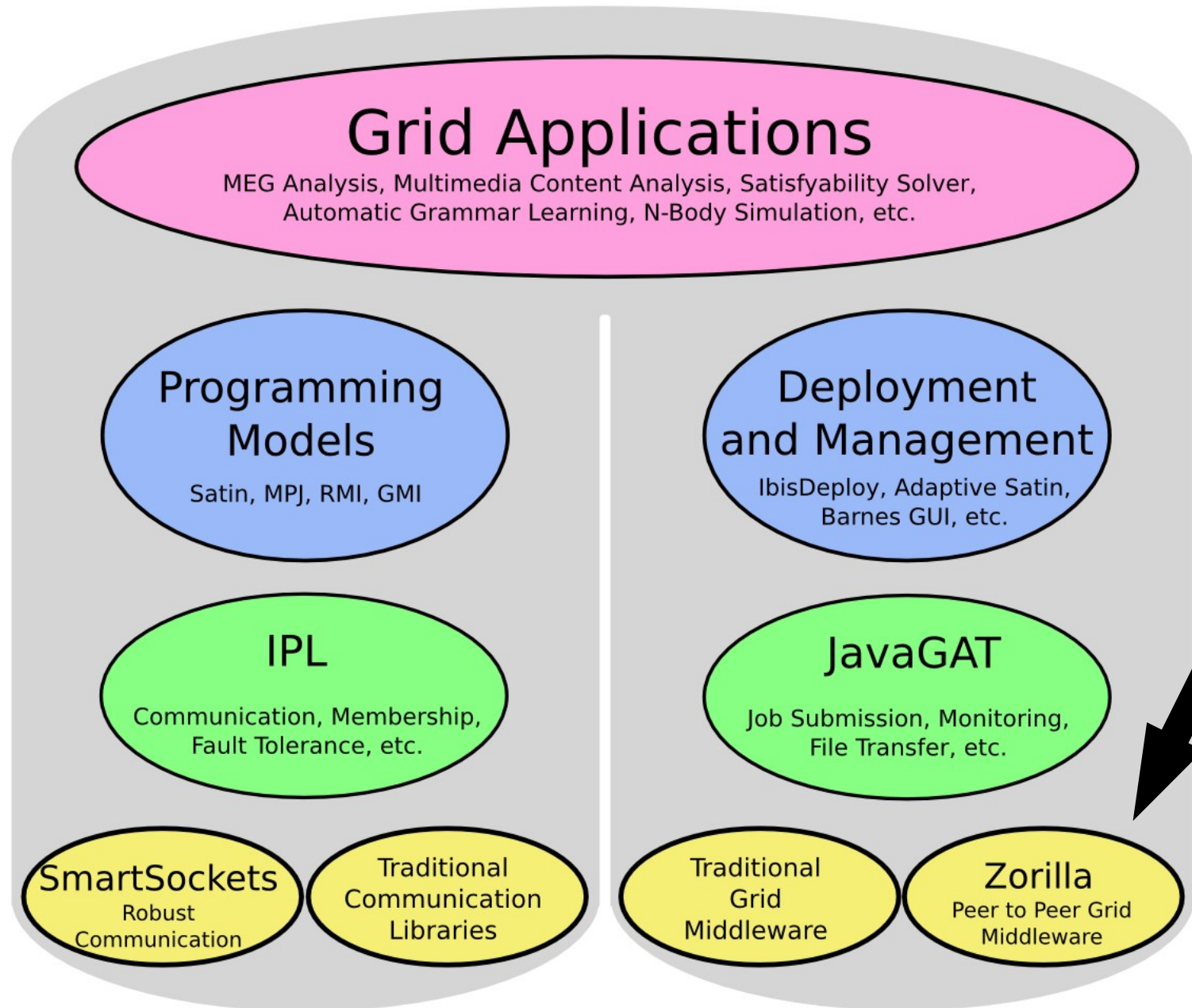


Alternative: Zorilla

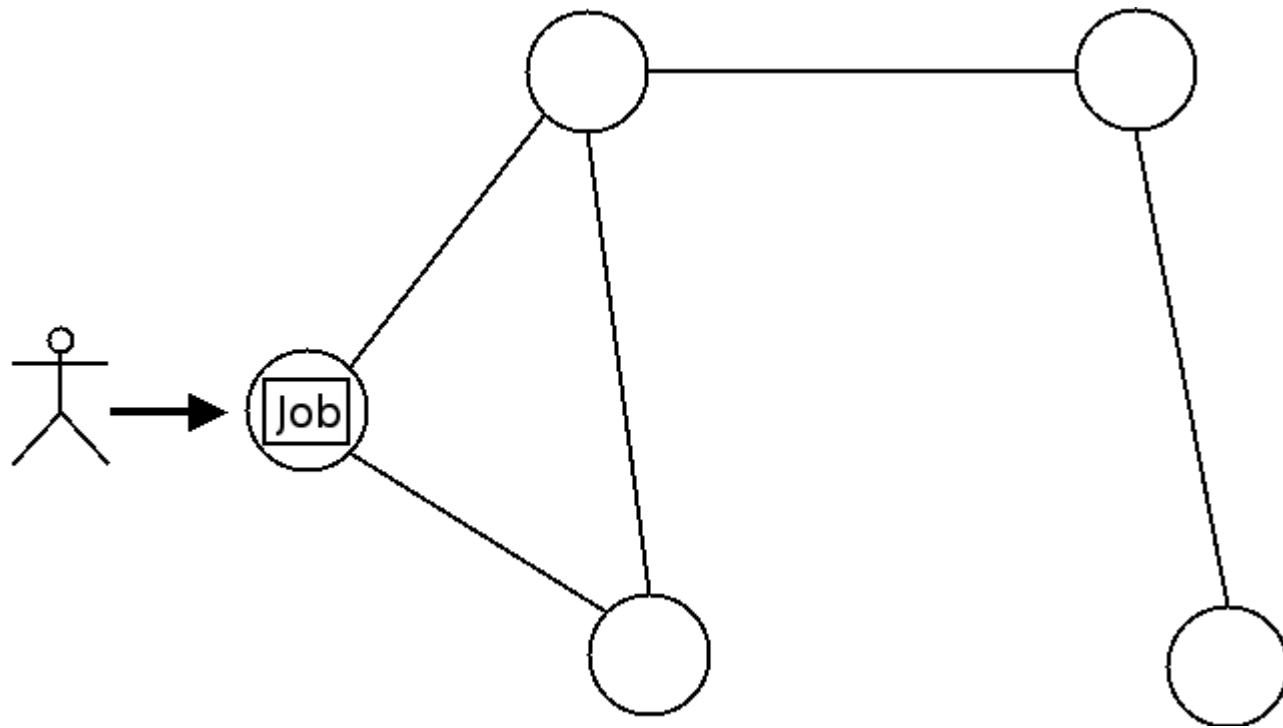
- Prototype Java Peer-to-Peer middleware
- Fully distributed
- Easy to setup (just add Java)
- Fault tolerant (no single point of failure)
- P2P: Limited trust/security
- Interface: JavaGAT



Zorilla in the Ibis project

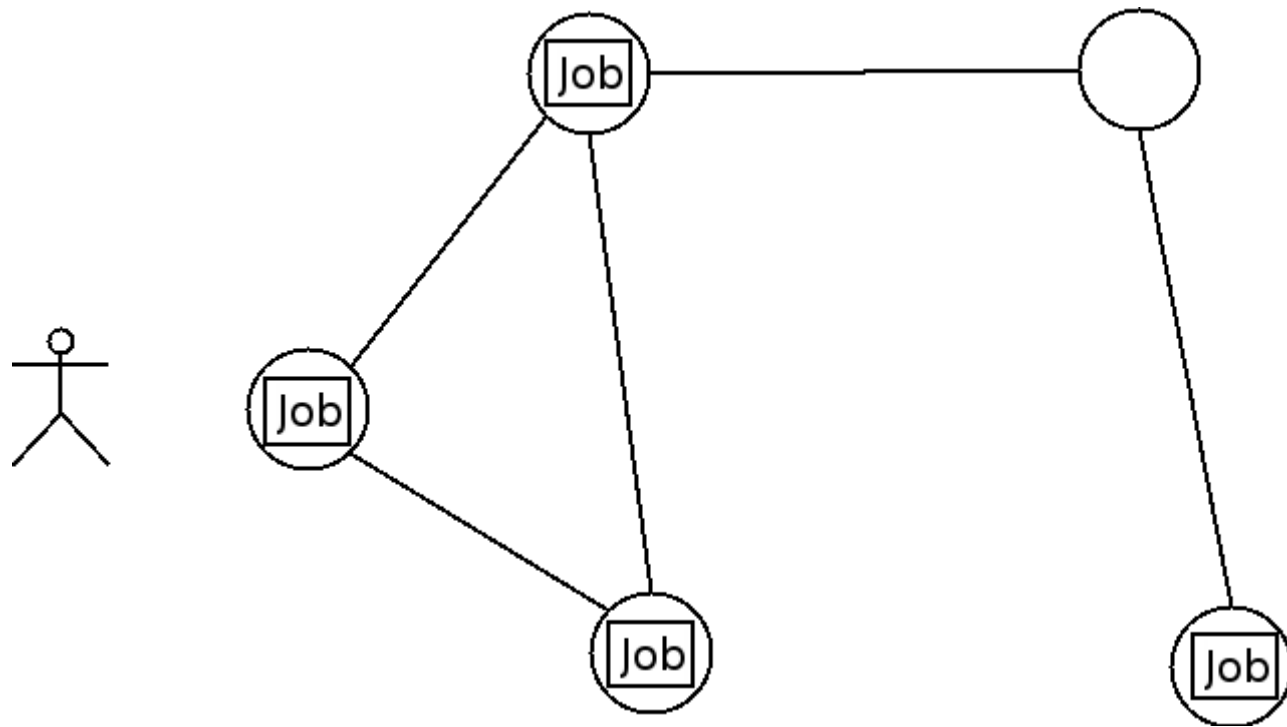


Life of a job (1/4)



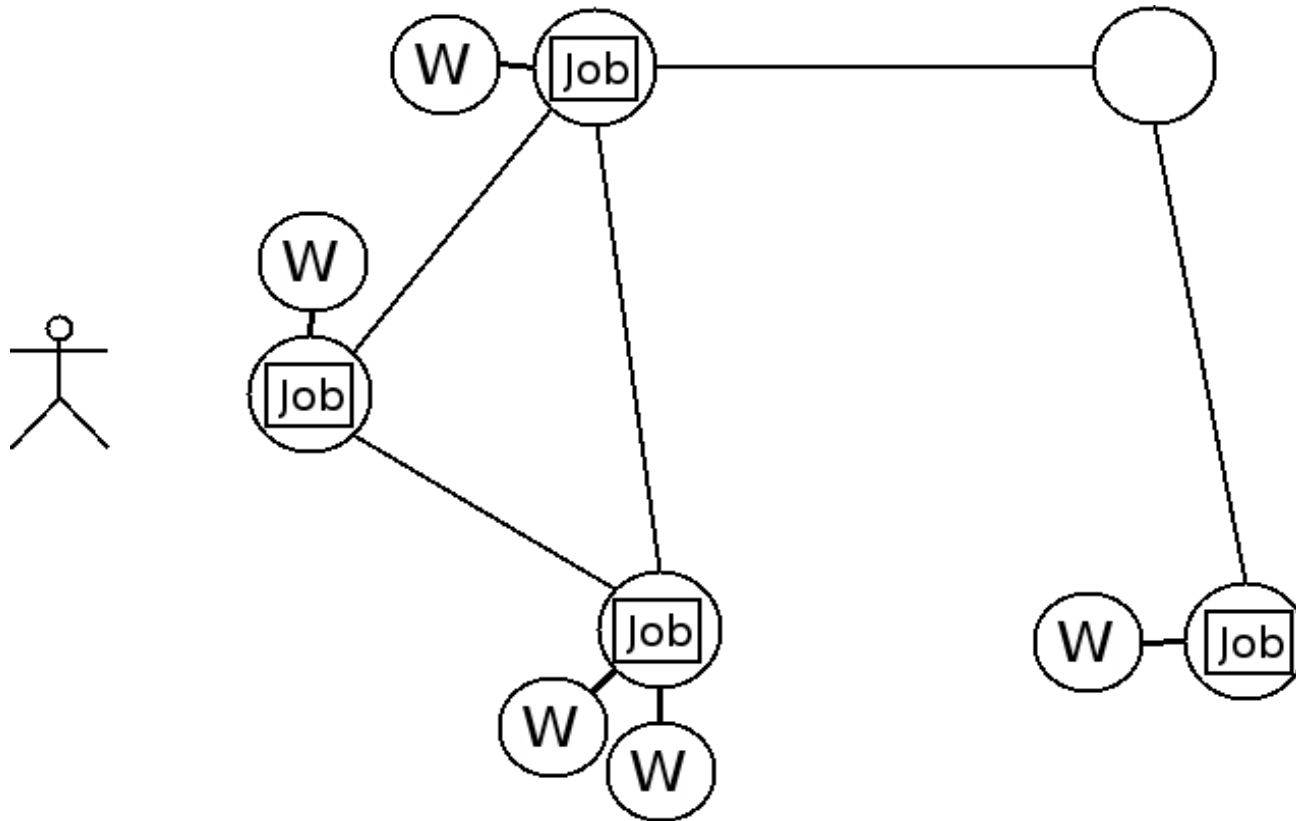
- Submit

Life of a job (2/4)



- Resource Location

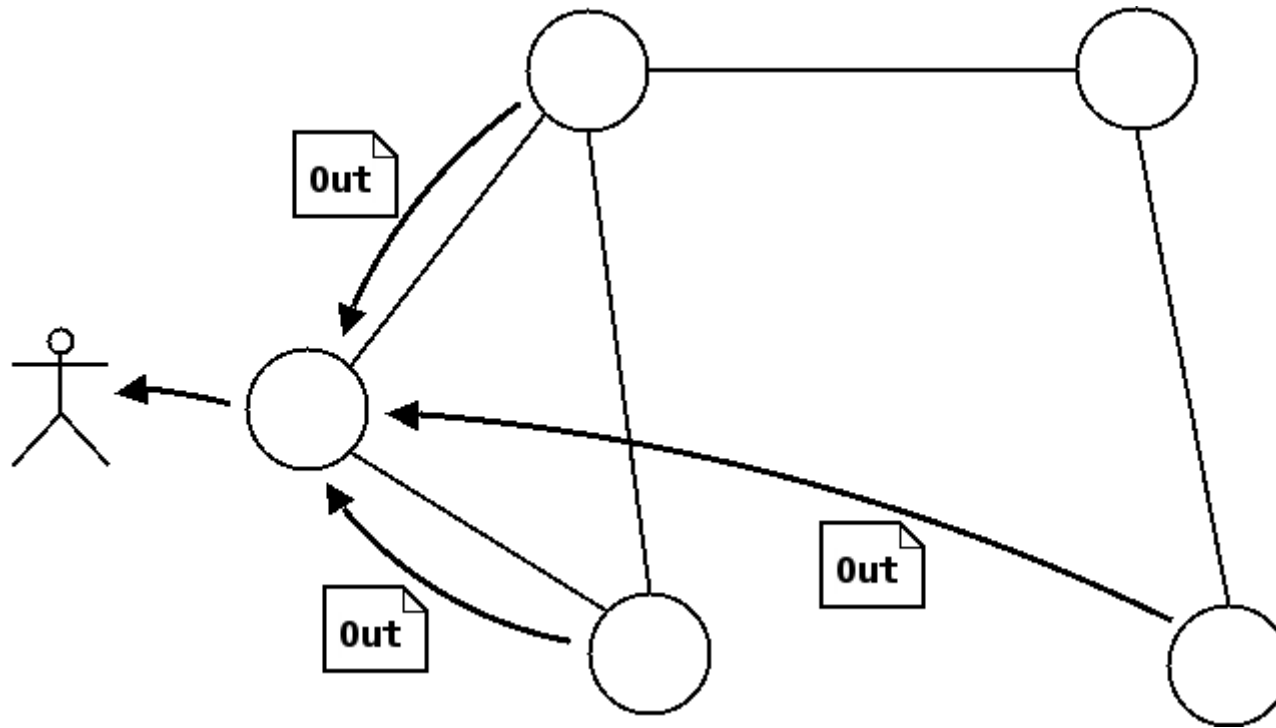
Life of a job (3/4)



- **Workers Started**

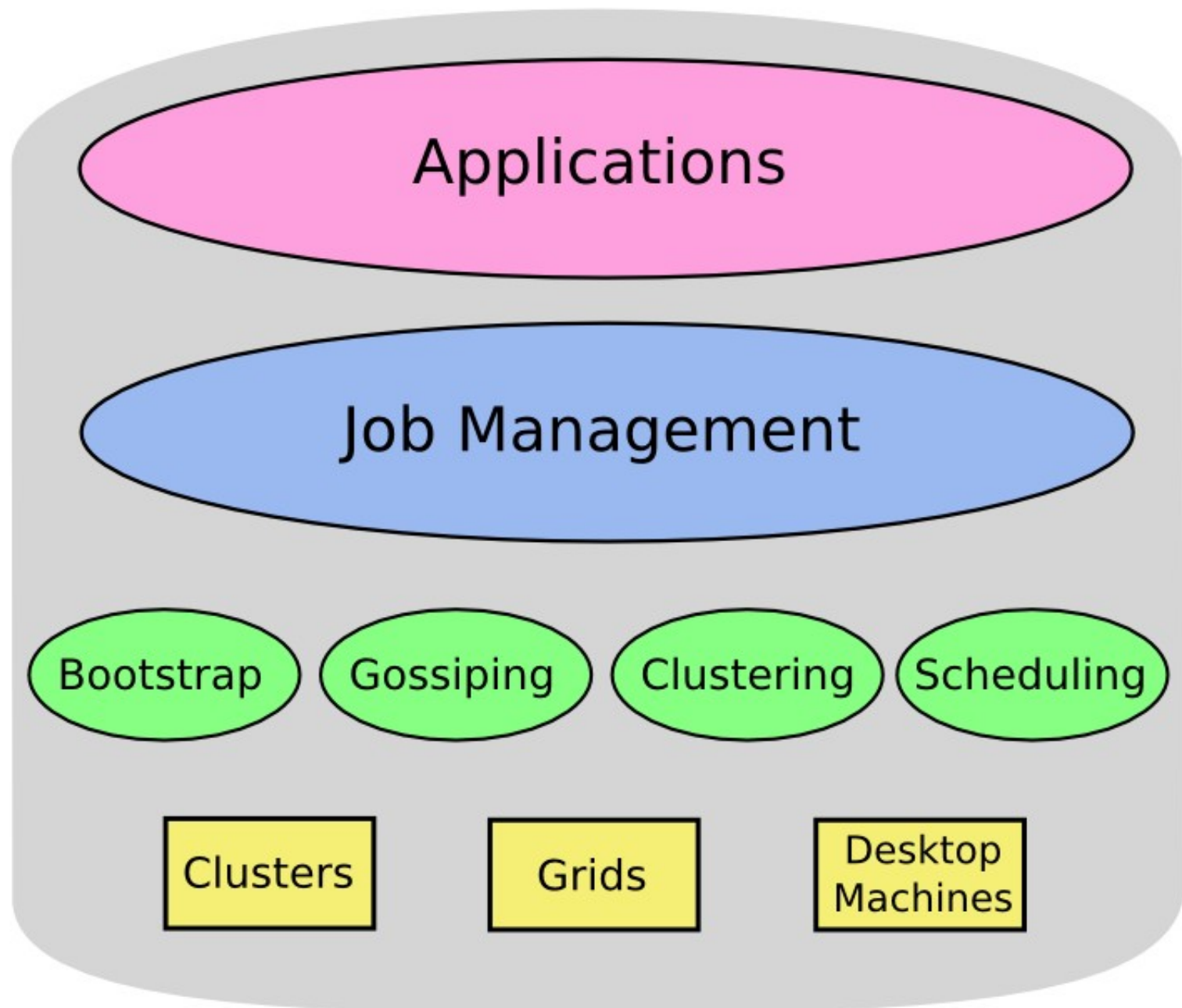
W = Worker running Application

Life of a job (4/4)



- Results returned to user

Zorilla Overview



Zorilla Components

- Bootstrap
 - Initial set of contact points
 - UDP broadcast or provided by user
- Gossip overlay network
 - Actualized Robust Random Gossip (ARRG)
 - Withstands Firewalls et al.
- Clustering
 - Nearest neighbor list



Zorilla Components (2)

- Flood scheduling
 - Incrementally search for resources at more and more distant nodes
- Job Management
 - Status (scheduling, running, done, etc)
 - File transfers
 - Malleability / crashes



Zorilla Usage

- 1) Install recent JVM
- 2) Download Zorilla at
<http://www.cs.vu.nl/ibis>
- 3) Run “zorilla”
(see --help for options)
- 4) Repeat 1-3 for all machines
- 5a) Start application using JavaGAT
- 5b) Start application using “zubmit”

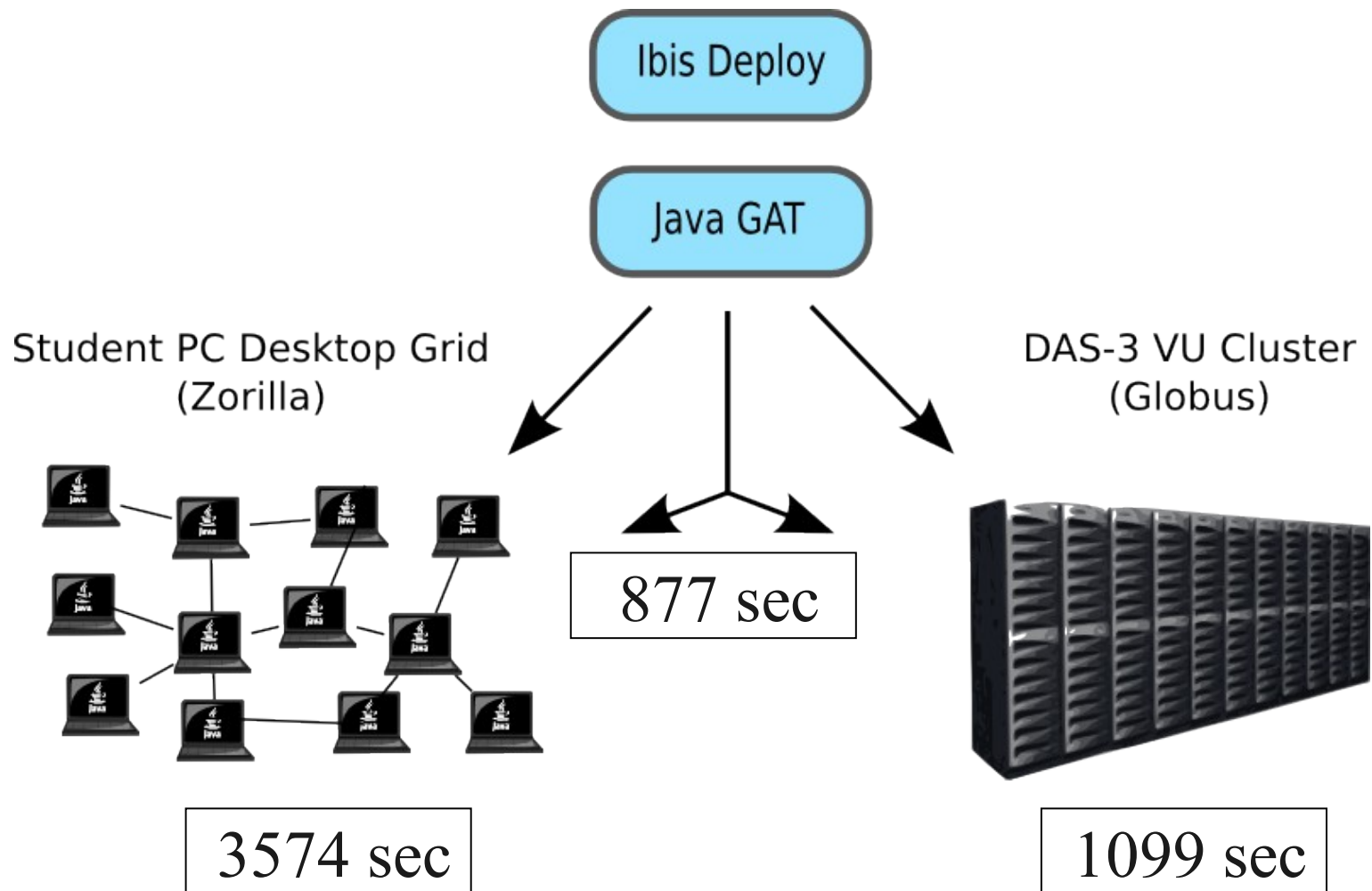


Desktop Grid Experiment

- Small experimental desktop grid setup
 - Student PCs running Zorilla overnight
 - PCs with 1 CPU, 1GB memory, 1Gb/s Ethernet
- Experiment: gene sequence application
 - 16 cores of DAS-3 with Globus
 - 16 core desktop grid with Zorilla
 - Combination, using Ibis-Deploy



Desktop Grid Experiment



Questions ?

ibis@cs.vu.nl

downloads, more info:

www.cs.vu.nl/ibis

