Grid Computing

The IBM ZetaGRID Solution

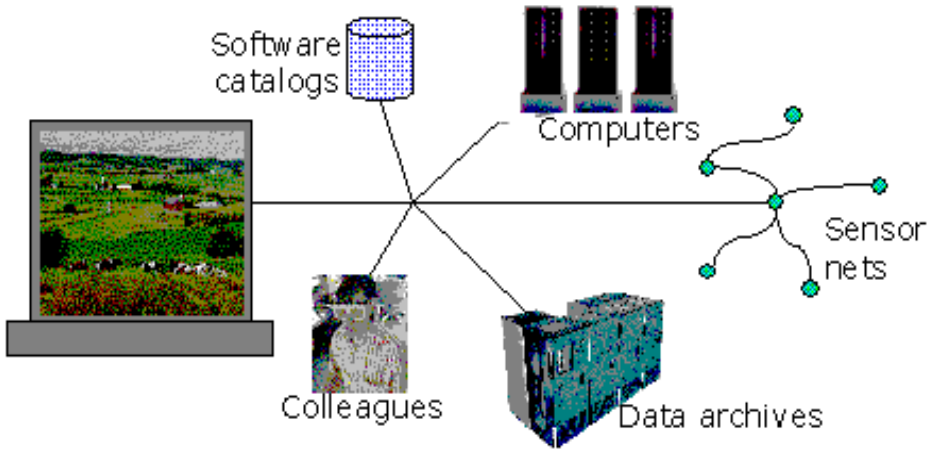
Ralf Grohmann
Dr. Sebastian Wedeniwski

IBM Global Services
IBM Lab Boeblingen, Germany

June 17, 2002

What is Grid Computing?

- **Uniform access resources**
- **Resource can be:**
  - **CPU**
  - **Storage**
  - **Data**
  - **Programs**
  - **Sensors**
  - **...**



On-demand creation of powerful virtual computing systems

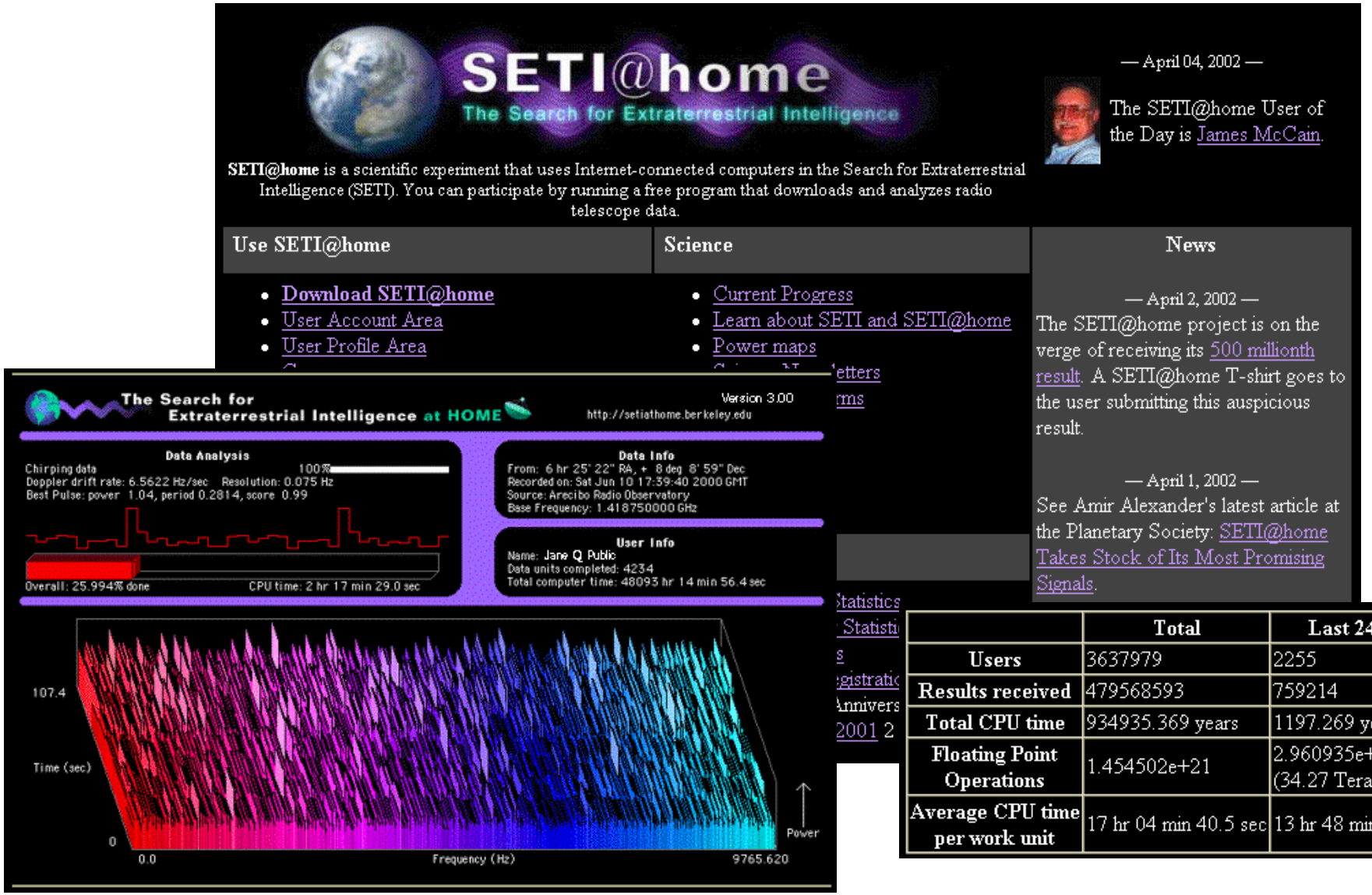
Idea: use idle resources

About 90% of the CPU capability of an office computer is not used.

- **Room for additional computations**
- **This CPU power is available for free**
- **Central control, simple administration, good scalability**
- **Some existing examples are**
  - **SETI@home**
  - **EON**
  - **ZetaGrid**

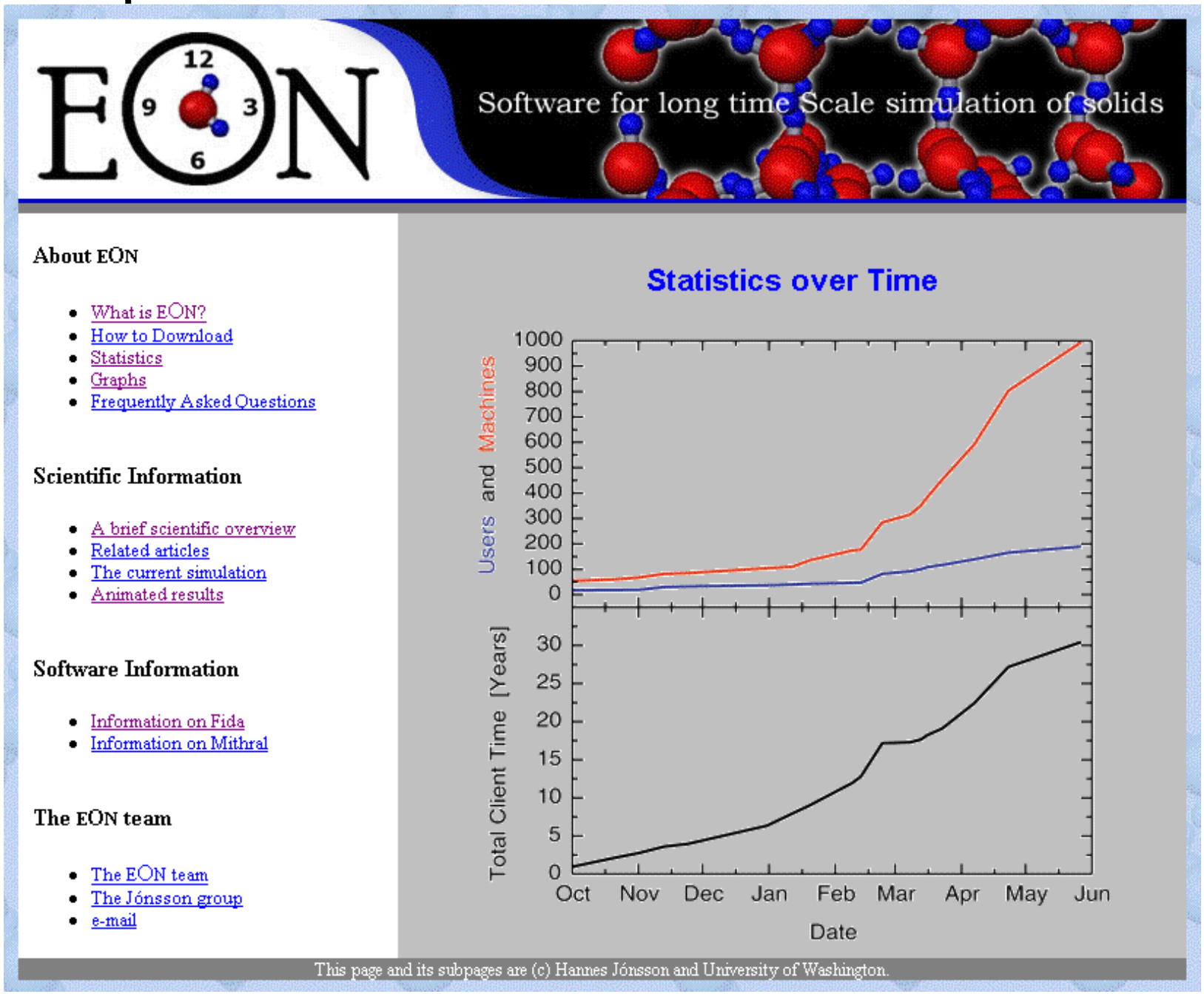
Example: SETI@home

'Search for extraterrestrial intelligence - at home'



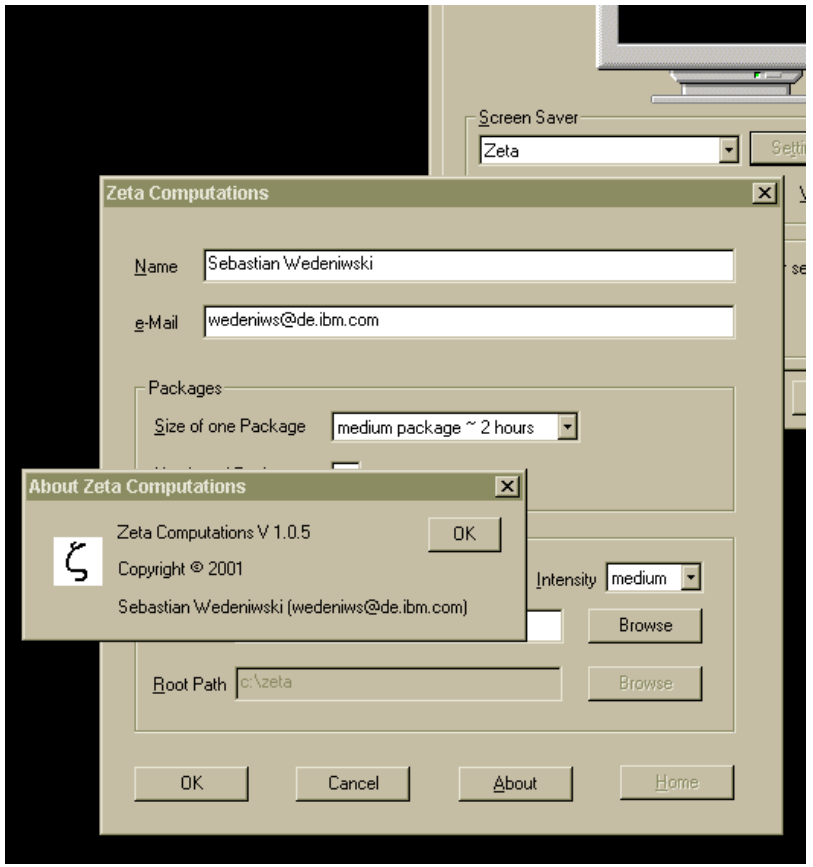
	Total	Last 24 Hours
Users	3637979	2255
Results received	479568593	759214
Total CPU time	934935.369 years	1197.269 years
Floating Point Operations	1.454502e+21	2.960935e+18 (34.27 TeraFLOPs/sec)
Average CPU time per work unit	17 hr 04 min 40.5 sec	13 hr 48 min 51.7 sec

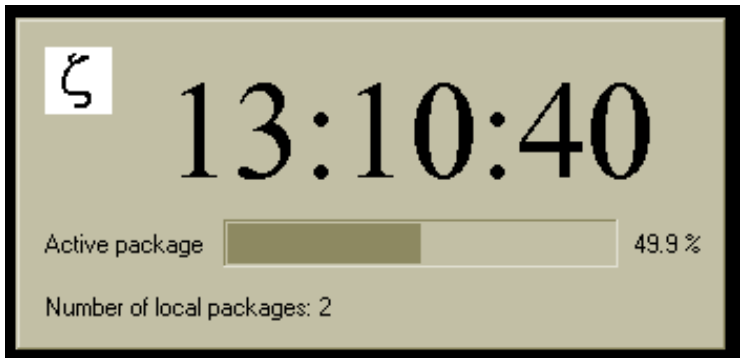
Example: EON


EON
Software for long time Scale simulation of solids
About EON
What is EON?
How to Download
Statistics
Graphs
Frequently Asked Questions
Scientific Information
A brief scientific overview
Related articles
The current simulation
Animated results
Software Information
Information on Fida
Information on Mithral
The EON team
The EON team
The Jónsson group
e-mail
Statistics over Time
Users and Machines
Total Client Time [Years]
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Jun
Date
This page and its subpages are (c) Hannes Jónsson and University of Washington.

Our Offering: ZetaGrid

- **Program runs as screen saver or low-priority process on Windows / AIX / Linux (also z-Series) which can stop at every time.**
- **It downloads 'tasks' using just the HTTP protocol and computes these on the local computer. Works also offline for temporary local tasks.**
- **Flexible tasks via API.**
- **Secured server, back-end database DB2, monitoring, statistics, and full audit trail are available.**
- **Security protocols and methods for authorization (tasks, results)**
- **Proven and stable, runs at IBM Lab with about 550 participating computers without any problems.**





ZetaGrid - math. work package

The first application: computing the zeros of the Riemann zeta function

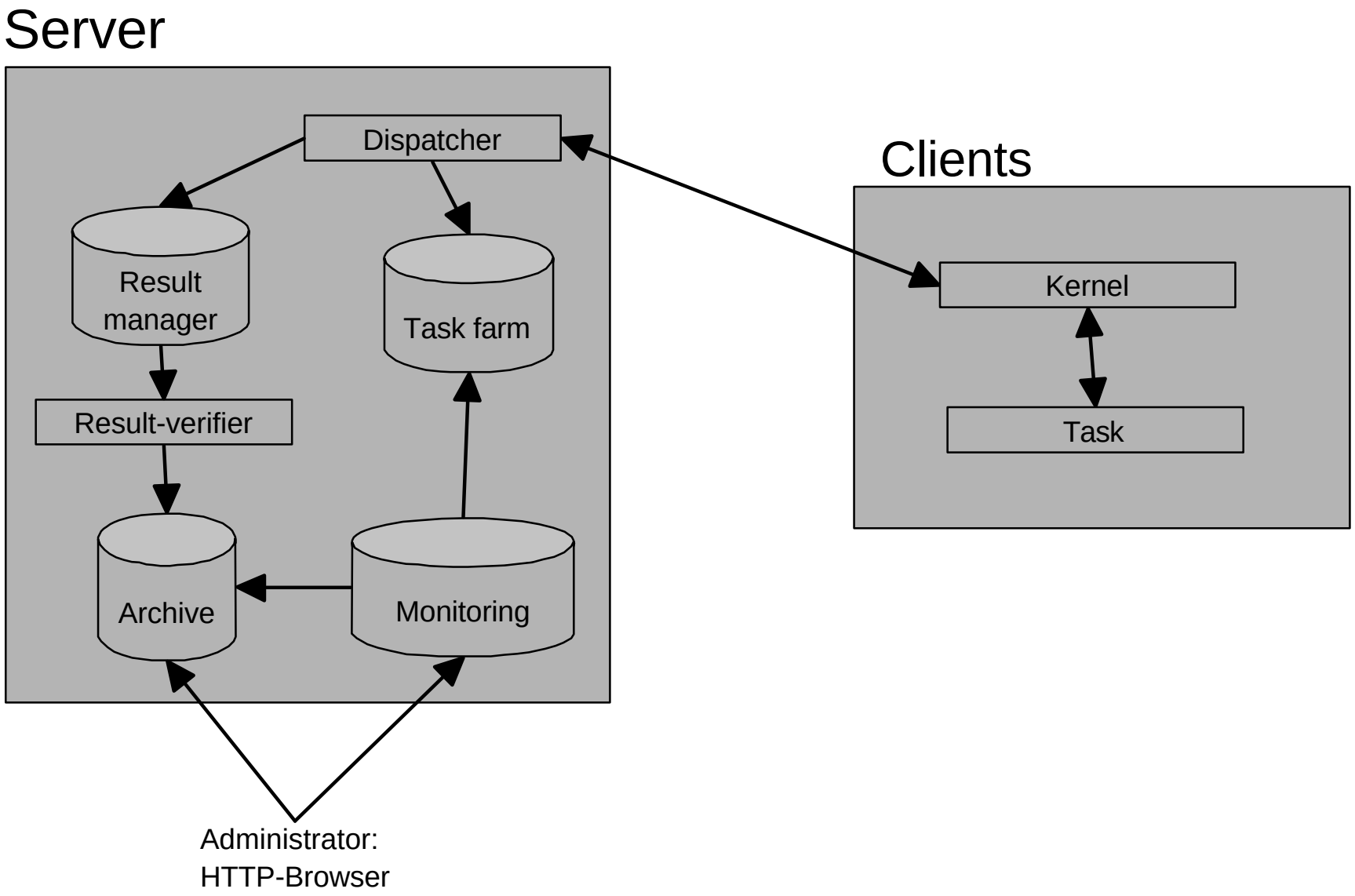
$$\zeta(\mathbf{s}) := \sum_{\mathbf{k}=1}^{\infty} \frac{1}{\mathbf{k}^{\mathbf{s}}}, \ \mathbf{s} \in \mathbb{C}, \mathrm{Re}(\mathbf{s}) > 1$$

- **The Riemann Hypothesis is one of modern mathematic's most important problems.**
- **The Riemann hypothesis asserts that all non-trivial zeros of the zeta function are on the critical line (1/2+it where t is a real number).**
- **The last 150 years did not bring its proof or disproof. In 2000, Clay Mathematics Institute offered a $1 million prize for proof of the Riemann hypothesis.**
- **The computation of the zeros helps to formulate correct statements in practical applications, e.g. to empirically prove that the fastest known primality test is correct within specific ranges (= important element of cryptography).**

History of the computational records

Year	Author	Number of zeros
1903	J. P. Gram	15
1914	R. J. Backlund	79
1925	J. I. Hutchinson	138
1935	E. C. Titchmarsh	1,041
1953	A. M. Turing	1,104
1955	D. H. Lehmer	10,000
1956	D. H. Lehmer	25,000
1958	N. A. Meller	35,337
1966	R. S. Lehman	250,000
1968	J. B. Rosser, J. M. Yohe, L. Schoenfeld	3,500,000
1977	R. P. Brent	40,000,000
1979	R. P. Brent	81,000,001
1982	R. P. Brent, J. van de Lune, H. J. J. te Riele, D. T. Winter	200,000,001
1983	J. van de Lune, H. J. J. te Riele	300,000,001
1986	J. van de Lune, H. J. J. te Riele, D. T. Winter	1,500,000,001
2001	S. Wedeniwski	10,118,665,300
2002	S. Wedeniwski	50,631,912,399

Architecture of the components



Properties of the framework

- **Flexible tasks (API available).**
- **Kernel is written in Java (independent of platform); a task can also be implemented for a specific platform.**
- **Characteristics (Windows):**
  - **Program runs only during the screen saver time (pull not push mechanism)**
  - **The screen saver can be stopped without any delay at every time**
  - **Not completed work units are stored temporary on the local hard disk (sophisticated cross-platform check-pointing mechanism)**
- **A low-priority background process runs on non-Windows operating systems which uses just idle time.**
- **Integration of the resource in the network is only necessary to get a new task and to deliver a result**
  - **The client works also off-line.**

The HTTP Server

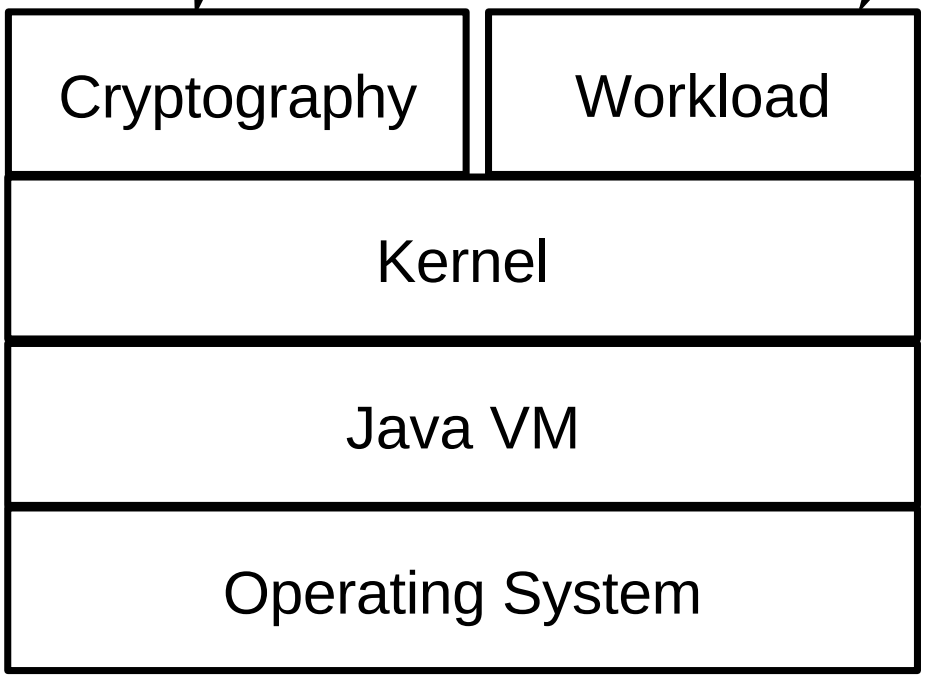
provides four general interfaces to the client

1. **To provide the environment for the work unit**
   - ***Input:*** **Operating system, processor architecture, TCP/IP address**
   - ***Output:*** **Libraries**
2. **To request a work unit**
   - ***Input:*** **Name of the resource provider, Size of the work unit, TCP/IP address**
   - ***Output:*** **Work unit**
3. **To finalize a work unit**
   - ***Input:*** **Name of the resource provider, TCP/IP address, work unit**
4. **To generate statistics**
   - ***Input:*** **Name of the statistic**
   - ***Output:*** **HTML pages**

The client layers

Secured communication, authorisation and authentication

The workload may be OS specific and should ideally be a closed unit of code.



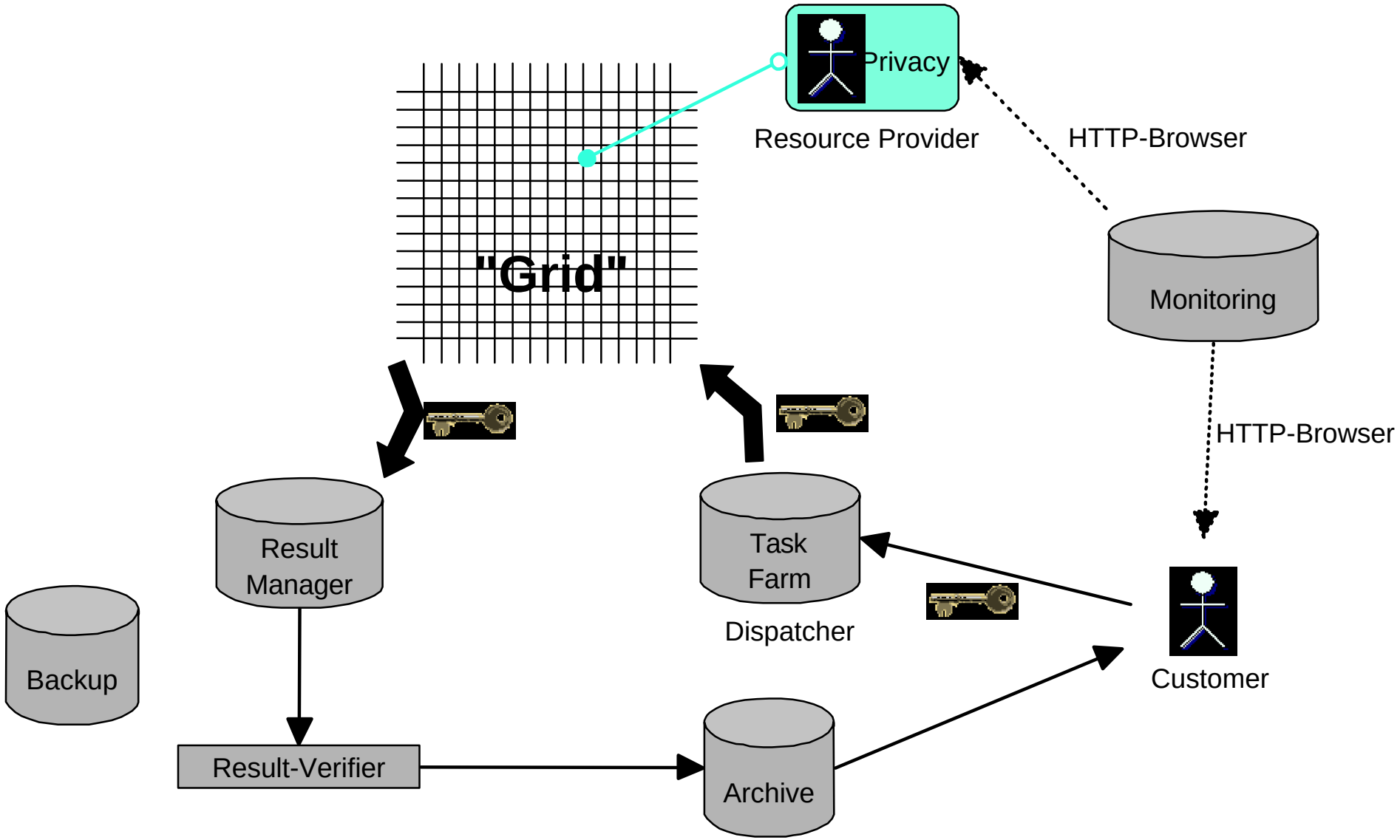
Starting and stopping the workload, server comms and management functions.

Computer architectures used

... the example of the zeta computation task

Name	Processor	Number of computers	Part of the computation
AIX	ppc	153	3,1%
Linux	x86	52	13,7%
Linux	s390	3	0,4%
Windows 2000	x86	189	31,1%
Windows NT	x86	90	51,6%
Windows 98	x86	4	0,1%
Windows 95	x86	2	0,06%

Security Architecture

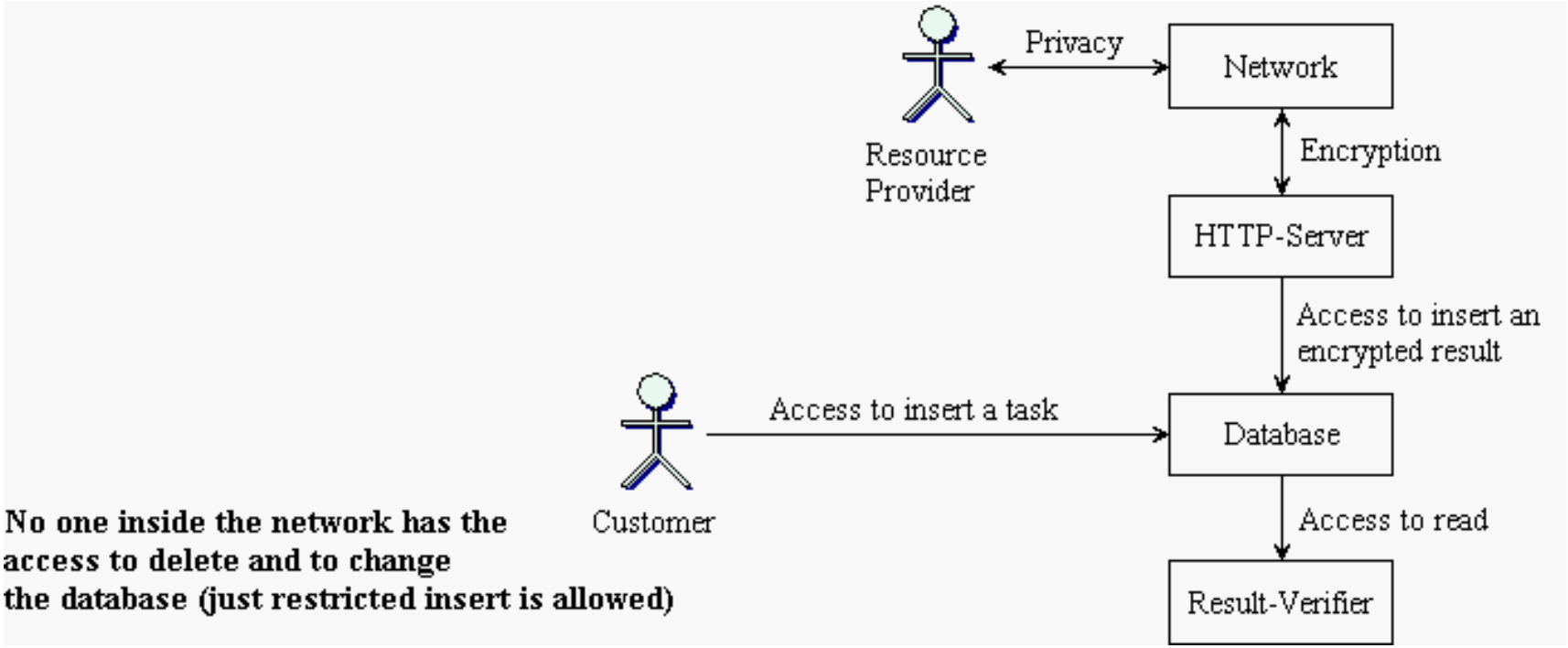


Security aspects (part I)

- **The total system is secure against all known passive and active attacks, e.g. viruses.**
- **The encryption used is secure against:**
  - **ciphertext-only, known-plaintext, chosen-plaintext, adaptive chosen-plaintext, chosen-ciphertext, key-only, known-message, chosen-message, adaptive chosen-message**
- **The protocols used are secure against:**
  - **known-key, replay, inpersonation, dictionary, forward search, interleaving, reflection, forced delay, intruder-in-the-middle, misplaced trust in server**
- **The digital signatures used are secure against:**
  - **index-calculus, Pohlig-Hellman, weak generators**

Security aspects (part II)

- **Every task has a different digital signature based on ElGamal signature scheme with keys which have a length of 1024 Bit.**
  Only the customer knows the private key of the digital signature.
- **Every transfer of a result use a key establishment protocol (half-certified Diffie-Hellman) with keys which have a length of 1024 Bit.**
  Only the Result-Verifier knows the private key of the protocol (ElGamal key agreement)
- **Restricted Layer Access:**

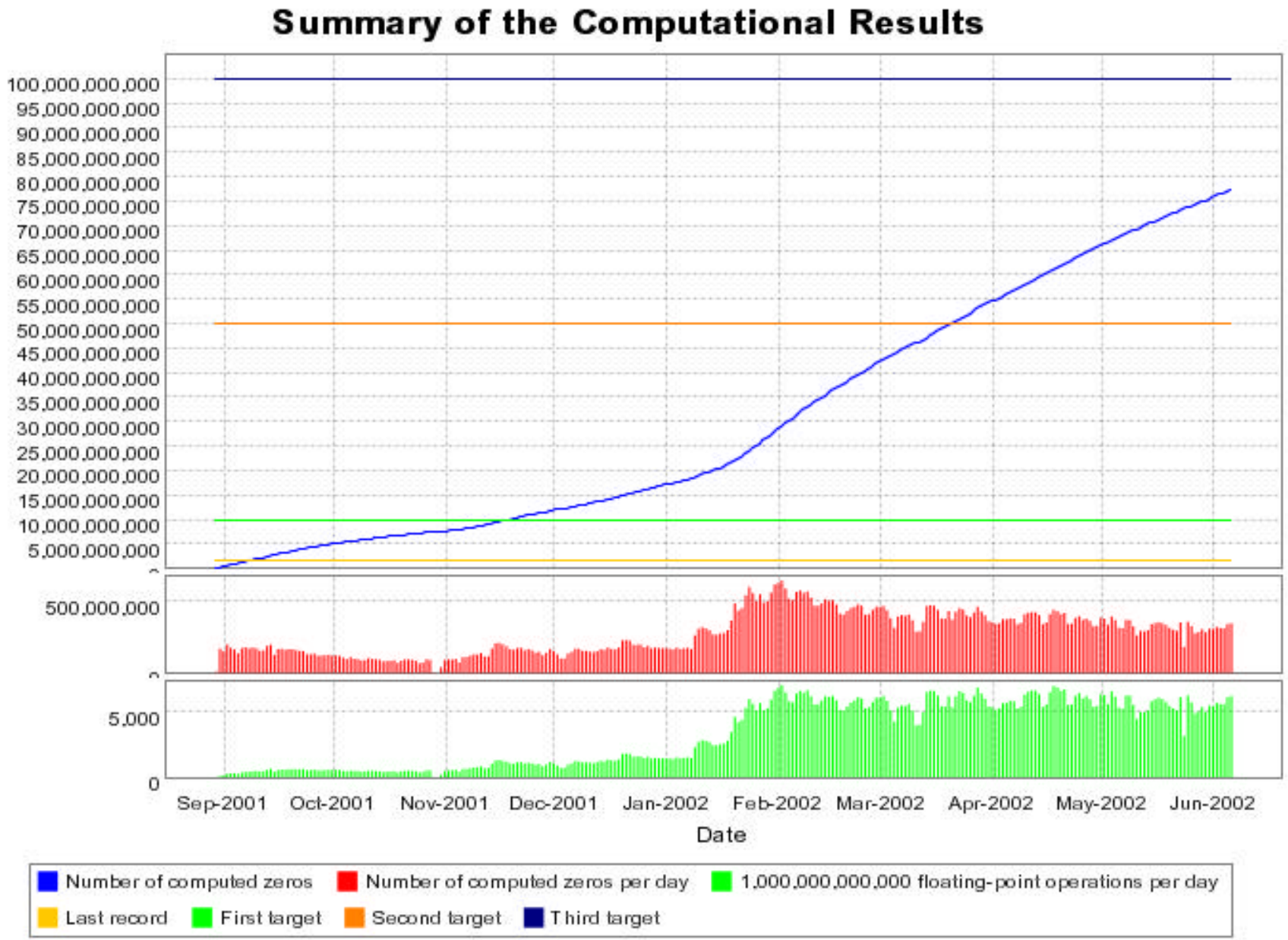


No one inside the network has the access to delete and to change the database (just restricted insert is allowed)

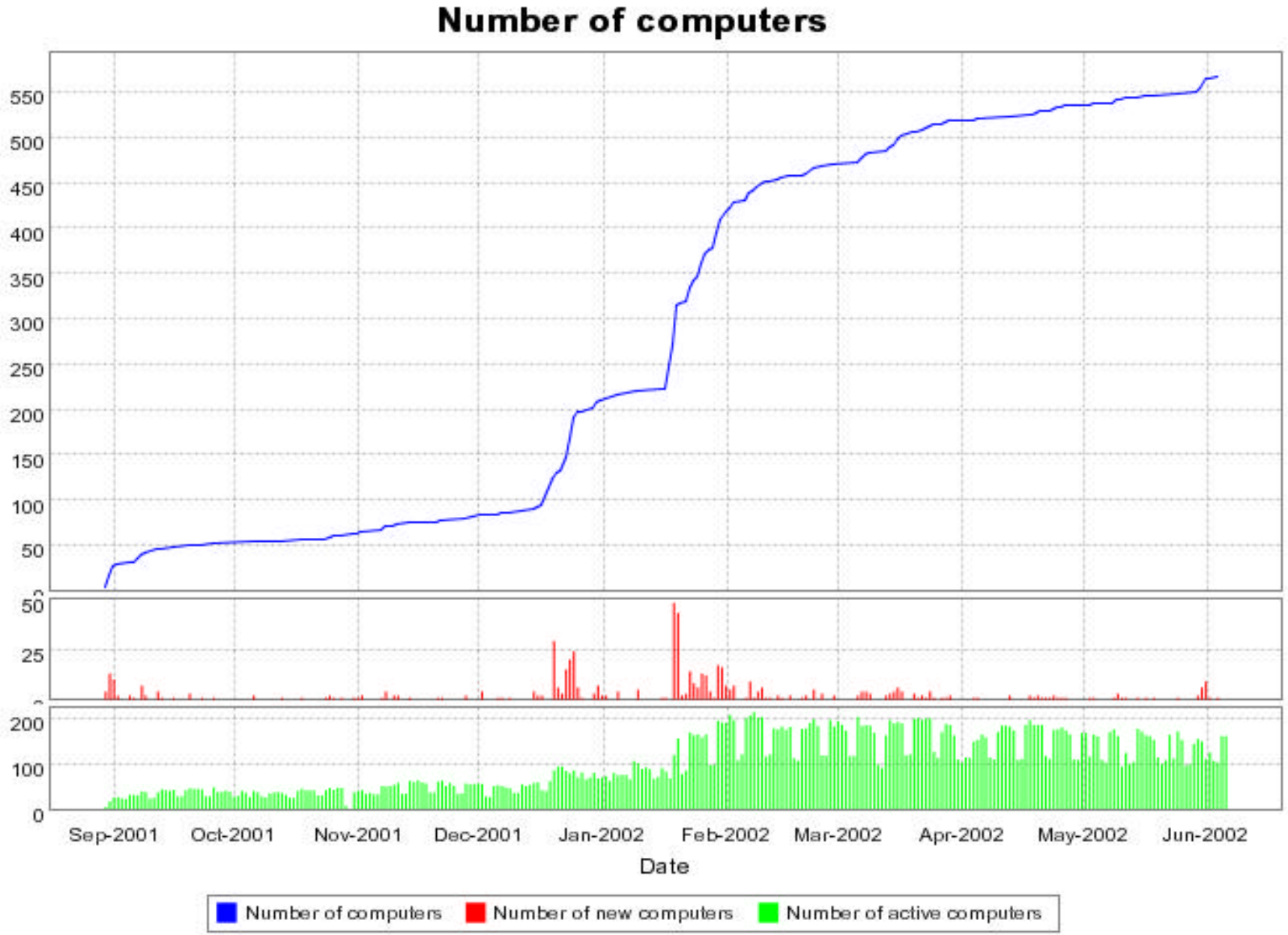
Performance Characteristics

- **Participating (5/19/2002):**
  - **230 users and 543 computers**
- **8.41 × 10^{17} floating-point operations (5/19/2002) for calculating more than 72 billion zeros of the Riemann zeta function in 264 days**
  - **~37 GFLOPS**
  - **~32 hours maximal performance of IBM ASCI White, 8192 Power3 375 MHz processors (place 1, 11/2001, www.top500.org)**
- **day with best performance (1/31/2002):**
  - **7 * 10^{15} floating-point operations for calculating more than 642 million zeros**
  - **~81 GFLOPS**
- **hour with best performance (2/12/2002, 10:00-10:59 a.m.):**
  - **5,9 * 10^{14} floating-point operations**
  - **~164 GFLOPS (place 204, 11/2001, www.top500.org)**

Progress of the computation



Number of computers



Daily activities (part I)

Daily activities (part II)

Characteristics of a fitting workload

- **'Musts':**
  - **CPU intense application which is to hard to handle for a single Computer.**
  - **Task must be of a parallel nature, i.e. not a single sequential calculation but can be split into chunks.**
  - **No user interaction necessary.**
  - **There should be a possibility to do 'checkpointing', i.e. saving of intermediate results if the screensaver metaphor is used.**
  - **The exact throughput cannot be calculated. Therefore you either need enough buffer capacity or the task is not time-critical.**
- **Maybe's:**
  - **Only few data accesses needed or only small data chunks are accessed.**
  - **The calculation should be a codesnippet or a batch. If it itself is a big, complex program with complex configurations the administrative overhead may get tricky.**

Some samples

of good tasks to solve using ZetaGrid.

Mathematical Calculations

- Need to calculate more zeros of specific functions? Factorize numbers? Compute finite elements?
- Characteristics of these calculations are usually a small algorithm, mostly well parallelizable, which needs a lot of CPU power. ZetaGrid can give you the additional resources you need to solve your problem faster.

Banking: Risk assessment

- Banks during the night calculate their 'risk' by extensive Monte Carlo simulations. The asses the value of their assets by taking current exchange rates, interbank rates etc. into account and want to know what happens if e.g. the Dollar goes up or down or the rates change or ....
- There is always a wish to do just one more what-if calculation. ZetaGrid can make use of all the desktop PC's idling at night and give you that probably critical additional information.

Industry: Chip design

- Complexity of the chips in the Semiconductor industry is rising exponentially. Elaborate logic testing routines are run for every component of a chip. These nicely parallelizable calculations are usually done on compute clusters, and the capacity of the available computers limits the amount of tests done. ZetaGrid may be an answer to doing some more test to ensure error-free designs.

Life Sciences

- In drug tests a lot of simulations are run, e.g. to match a new vaccine against known virus DNA strings to find out possible areas of effectivity.

Others:

- Car crash simulations
- oil field detection
- weather forecast
- ....

Pick ZetaGrid if ...

a list of key advantages to compare with

- **Simple-to-use solution - easy to configure, administer, rollout**
- **Workloads are easily integrated. Java or Non-Java programms supported, OS-specific execution possible**
- **It downloads 'tasks' using only the HTTP protocol (firewall proof)**
- **Works also offline for temporary local tasks.**
- **Program runs as screen saver or low-priority process on Windows / AIX / Linux (also z-Series).**
- **SECURITY was a main topic in the design. Any communication is secured and the workload insertion and delivery of results from the clients are fully authenticated. No separate PublicKey server or other security infrastructure is required.**
- **Serverside: NT or Linux server, back-end database DB2, in the entry version Tomcat application server is used.**
- **Monitoring tools, statistics, and full audit trail are available.**
- **Proven and stable, runs at IBM Lab with about 550 participating computers.**
- **Full Java support.**
- **Complete Code control: Almost any special customer wishes can be implemented by IGS Team. Thus a perfectly fitting solution can be reached.**

Project approach

- **Solution Outline Phase**
  - **Collecting the requirements**
  - **Workload consulting (what and how)**
  - **Creating an implementation proposal**
  - **Creating architecture, system diagram, project plan**
  - **Duration 1-3 weeks**
- **Implementation phase**
  - **Adopting the client part**
  - **Creating or adopting the workload**
  - **Custimizing the server, esp. workload partitioning and reconsolidation.**
  - **Duration strongly depends on effort, avg. 1-6 months.**
- **Rollout phase**
  - **stepwise client activation.**

Questions / Discussion

Ralf Grohmann
+49 7031 16 4420
grohmann@de.ibm.com

Dr. Sebastian Wedeniwski
+49 7031 16 3243
wedeniws@de.ibm.com

IBM Global Services
IBM Germany Development, Boeblingen