



# Volunteer Computing with Web-Services as the communication mechanism

*Babu Thomas, SJSU ID - 007669116*

*Nandish Benchalli, SJSU ID – 008036041*

*Veera Venkata Kartik Gurram, SJSU ID – 008025576*

# What is Volunteer Computing

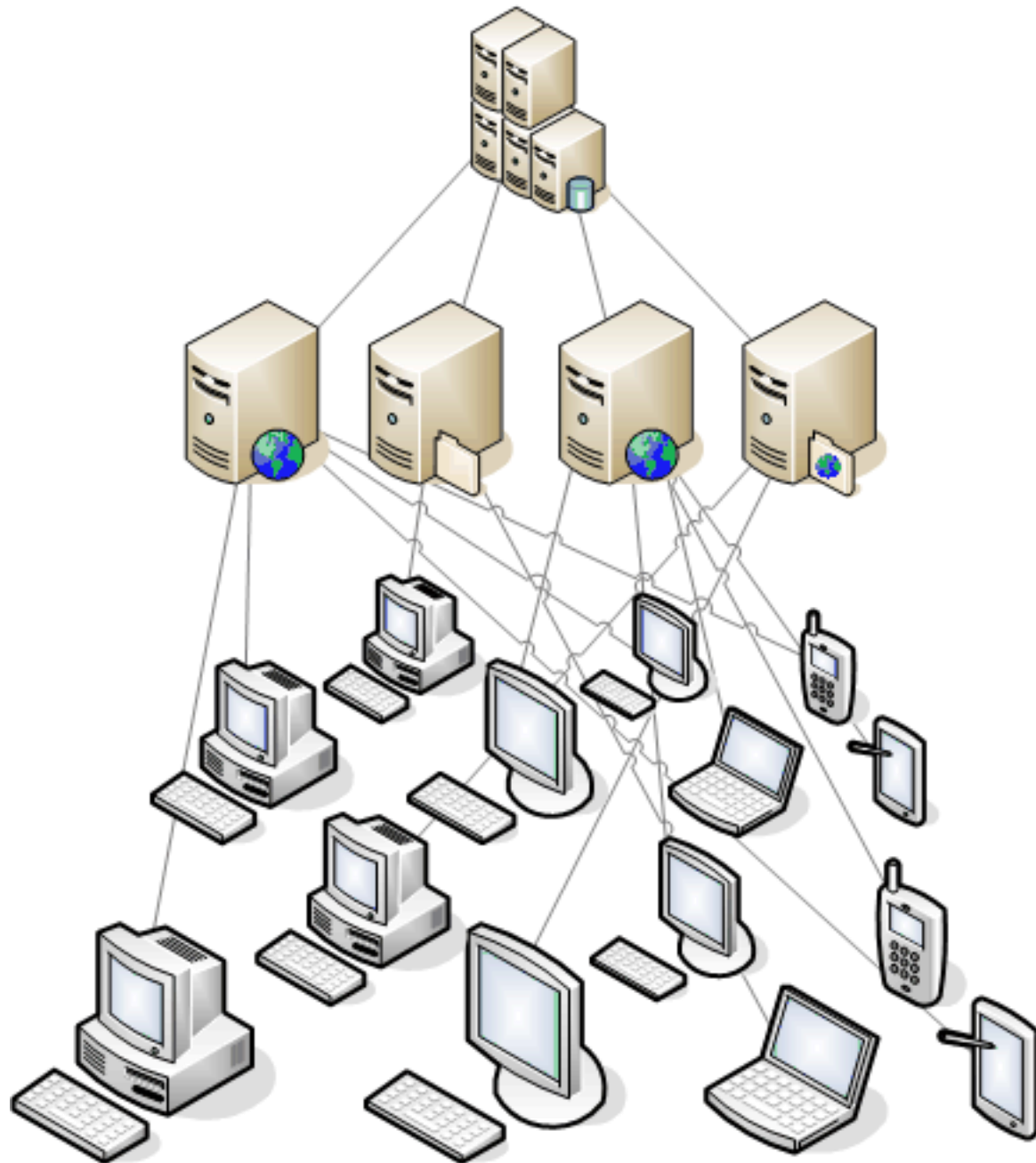
Large Distributed computation capability using idle compute time of machines during inactivity

Can be used for projects which require massive computational power

Can use modern technologies which are not vendor or platform specific



# What is Volunteer Computing





# Trying to achieve

Proposing a Web services model for implementing volunteer computing.

Cryptographic algorithm breaking challenge.





# Existing System

- Implemented in C, C++.
- Java implementations uses RMI for Communication.
- Require client user to have knowledge in installation of software components
- Not interoperable across multiple platforms

# Cryptographic algorithm breaking challenge

**Brute-force attack**, strategy that can, in theory, be used against any encrypted data

The **Key Length** used in the encryption determines the practical feasibility of performing a brute-force attack, with longer keys exponentially more difficult to crack than shorter ones

Source - [http://en.wikipedia.org/wiki/Brute-force\\_attack](http://en.wikipedia.org/wiki/Brute-force_attack)

Symmetric key length vs brute-force combinations

Key size in bits <sup>[2]</sup>	Permutations	Brute-force time for a device checking $2^{56}$ permutations per second
8	$2^8$	0 milliseconds
40	$2^{40}$	0.015 milliseconds
56	$2^{56}$	1 second
64	$2^{64}$	4 minutes 16 seconds
128	$2^{128}$	149,745,258,842,898 years
256	$2^{256}$	50,955,671,114,250,072,156,962,268,275,658,377,807,020,642,877,435,085 years

# Volunteer Computing Task Decomposition

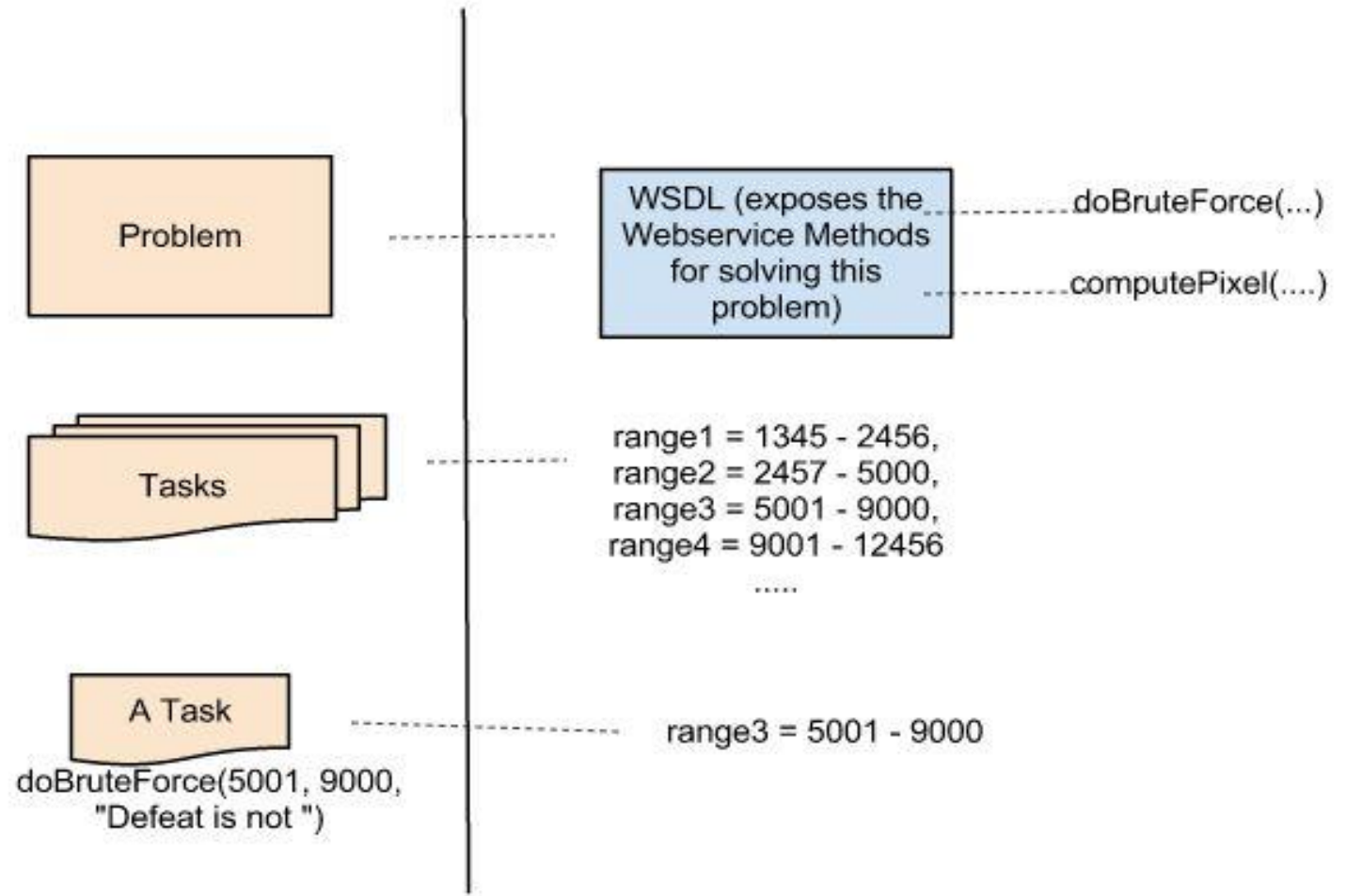


Fig 3: Problem to Web-service mapping



# Volunteer Computing Architecture

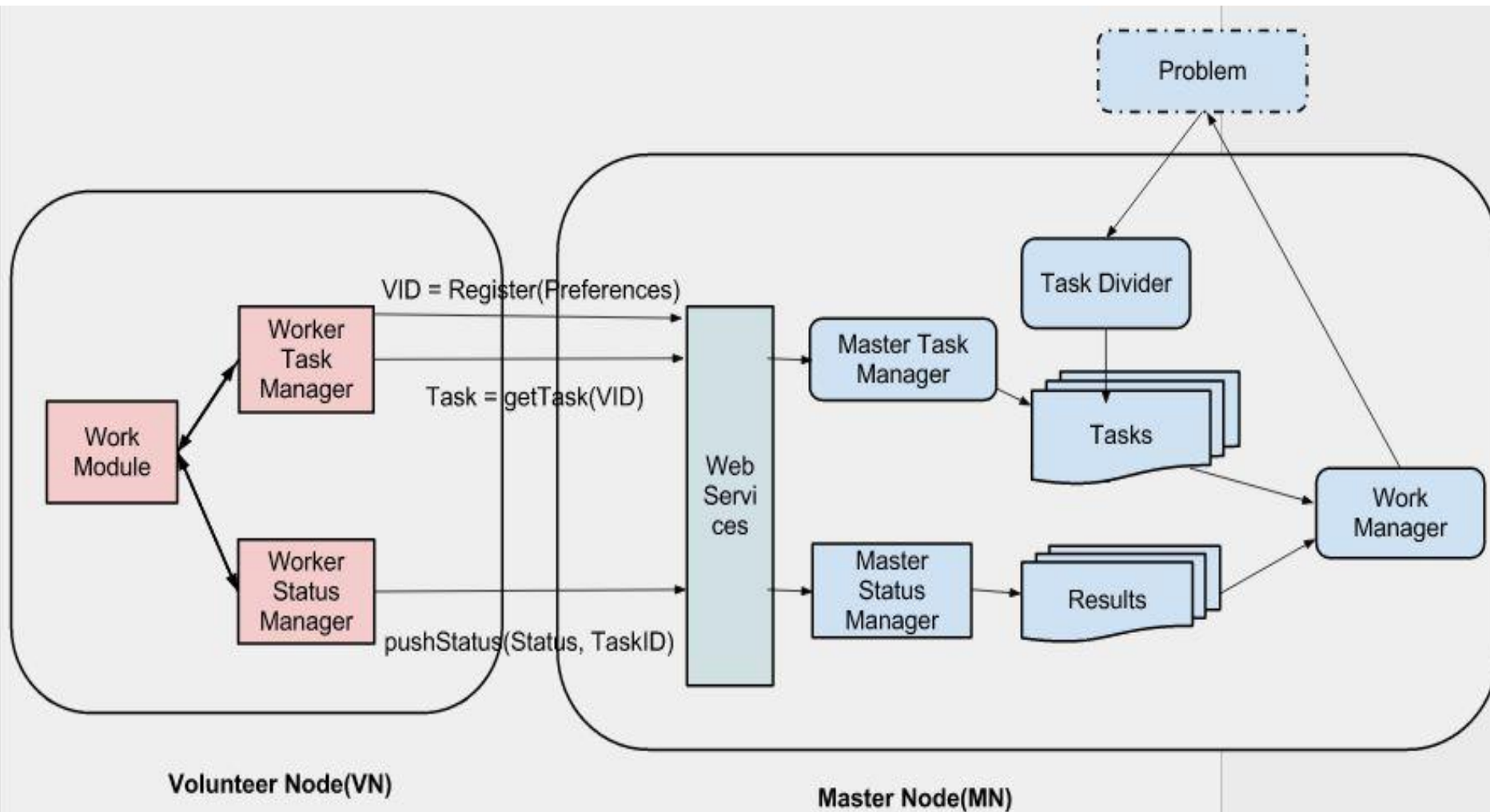


Fig:1- Volunteer-Computing Overall Architecture



# Volunteer Computing - Deployment

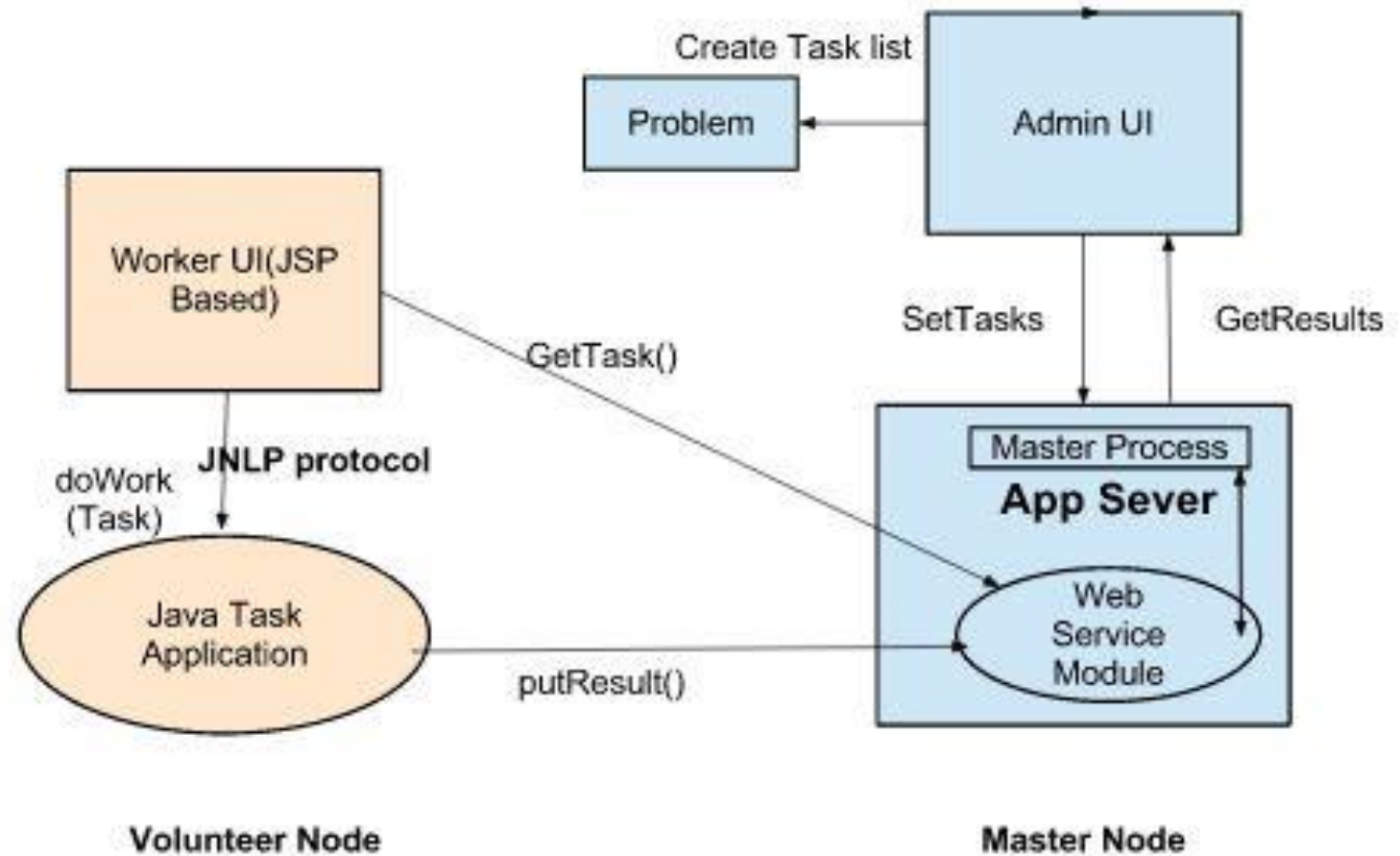


Fig :2 - Deployment Diagram



# Advantages of using Web Services

Application will be loosely coupled

Not blocked by firewalls, which is not true for other technologies say using C++ or RMI

High scalability and inter-operability assured

Easy to maintain and can upload new versions of software



## Challenges of Volunteer Computing:

- **Obtaining New Volunteer Users**

"Pay Model" can be tried

- **Retaining Existing Users**

Use less system resources and tasks should be processed in less time

- **Volunteer Computing Costs for Users**



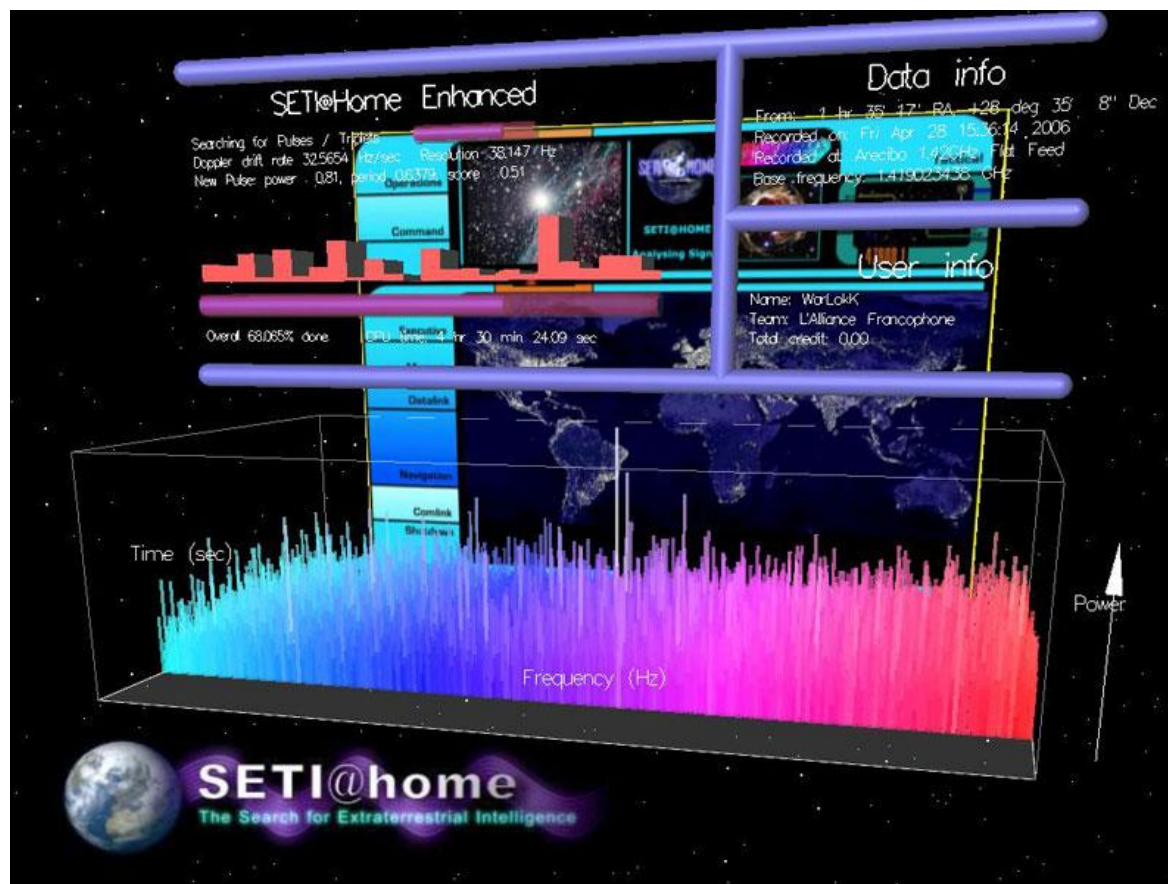
# Applications of Volunteer Computing:

- SETI@home by University of California, Berkeley
- Einstein@home by University of Wisconsin, Milwaukee and Max Planck Institute, Germany
- Clean Energy Project by Harvard University
- PrimeGrid project which searches for different types of longest prime numbers
- Applications in Earth Sciences, Biology and Medicine, Physics & Astronomy and Mathematics



# Future of Volunteer Computing:

- Enhanced version of SETI@home runs as a screen saver
- Started implementing on VMware and Android platforms





# Conclusion

Volunteer Computing fits into the current day technological trend towards massively distributed and scalable architectures

A viable solution in solving problems, when there the immediate business value are not known or explainable

Newer technologies are adaptable to most of present day computing platforms

Enables in mankind's inherent goodness to giveback to the community