Global CubeSat Operations

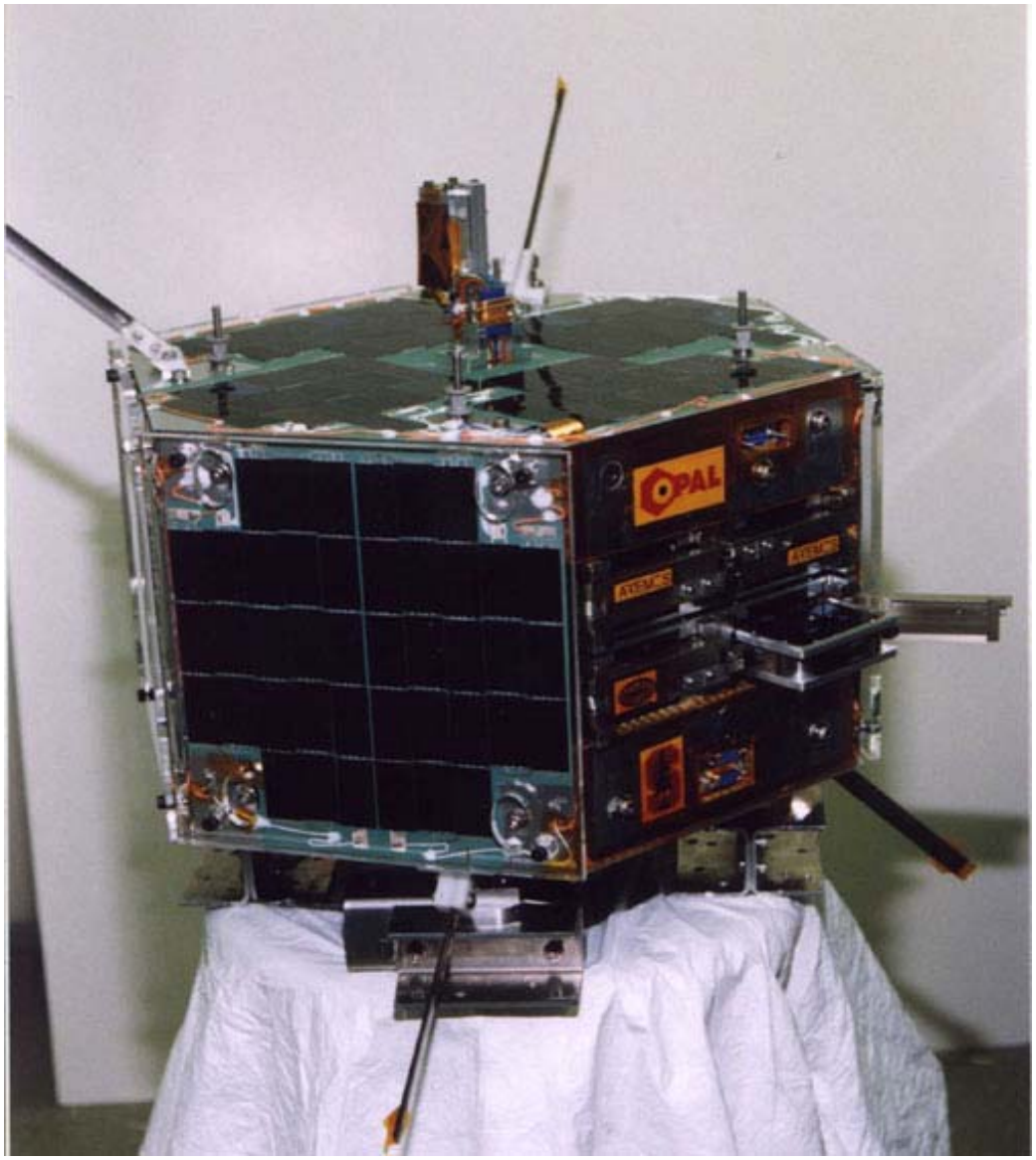
James Cutler

Space Systems Development Laboratory
Software Infrastructures Group
Stanford University

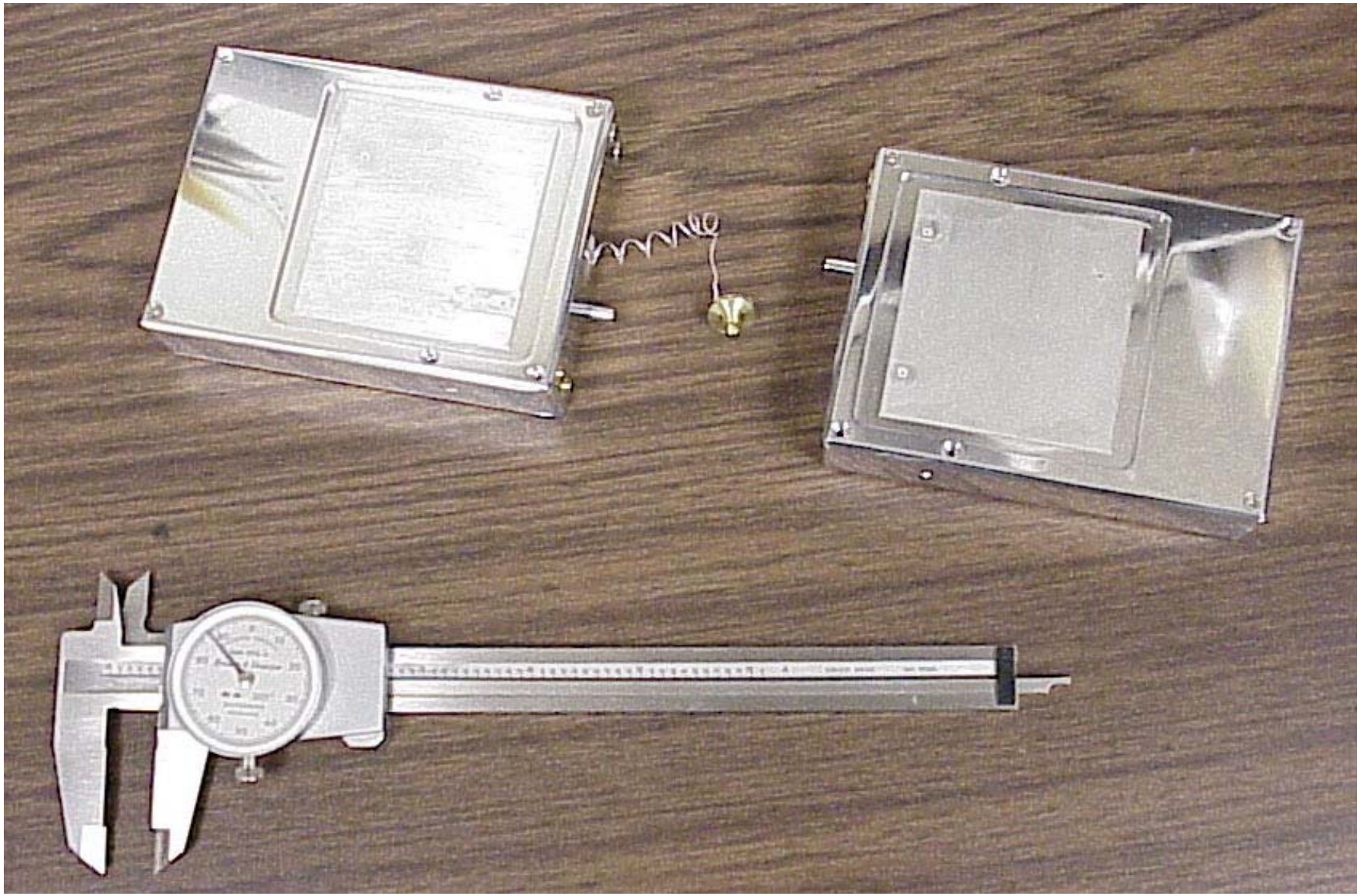
2004 CubeSat Workshop

A Little History...

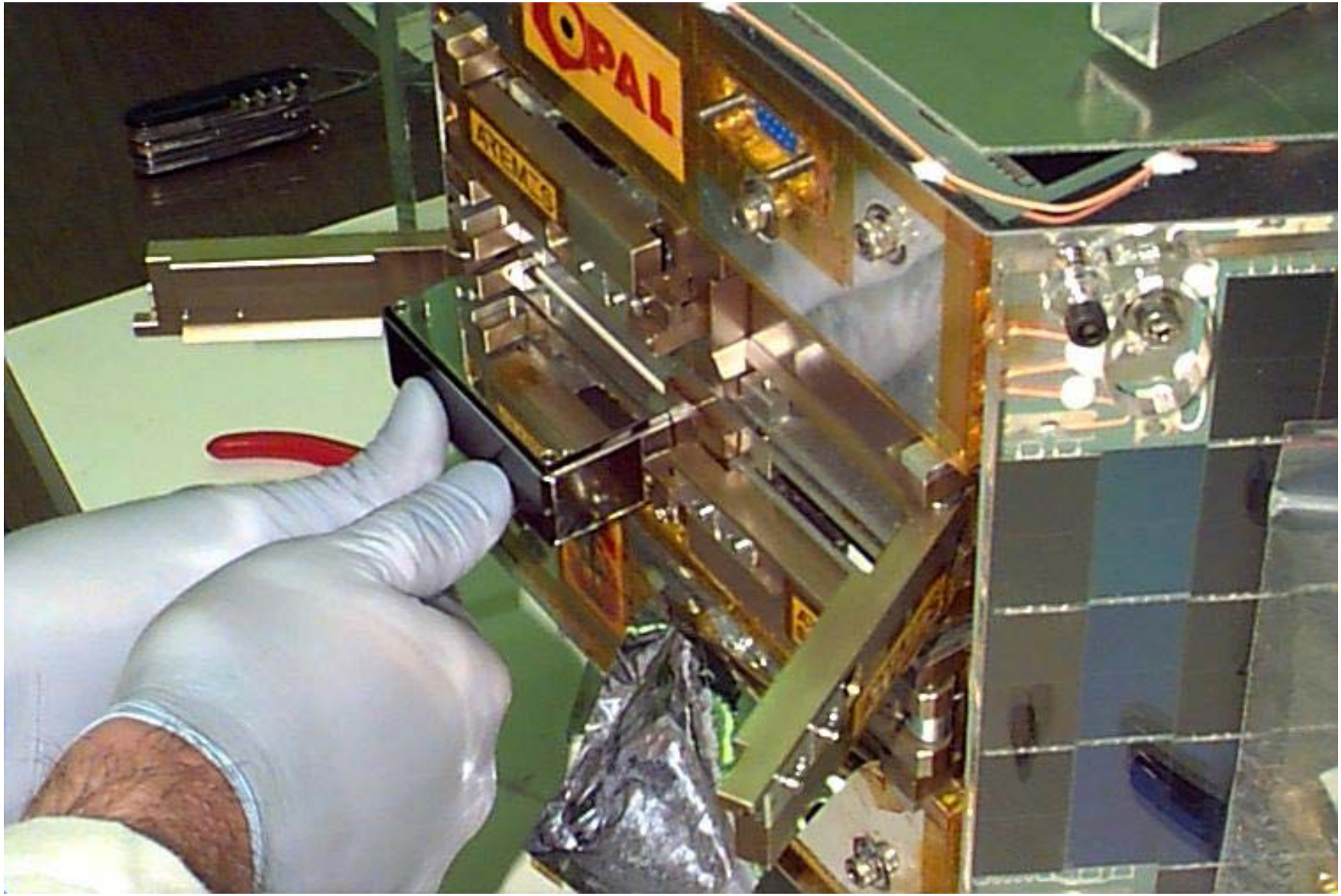
- Stanford's OPAL satellite
  - Our 2nd sat, 1995-1999
  - Mothership/daughtership technologies.
  - Ops: 2000-2002.5
- Missions
  - Magnetometer
  - Accelerometers
  - Mothership/daughtership testbed
- Picosatellites
  - 2 VLF receivers, SCU
  - 1 Beacon, SCU
  - 1 HAM repeater, STENSAT
  - 2 Experimental Comm sats.



Aerospace Picos



Post Delivery Pico Loads



Launch- Jan 26, 2000

- Successful launch from VAFB.
  - OSP Minotaur 1
- We quickly realized…
  - Opal works! Now what?
  - Very early AM passes.
  - Our ground station (GS) didn't work well.
- New research agenda born:
  - Can we access our satellite resources like we access Google, Yahoo, etc.?
  - Can we build a reliable GS network from unreliable stations?

Trends in Space Ops

- The trends in space operations include:
  - 24x7 connectivity between ground and space.
  - End-to-end access between users and satellites.
  - Intersatellite coordination and automation.
  - Some foresee/hope for the end of communication as a constraint.
- With this in mind, the goal of our work is two-fold:
  - Develop infrastructure to make space-based information more accessible.
  - Make this infrastructure robust and reliable while built from unreliable components.
- Primary approach:
  - A network of composable ground stations—a university ground station network.
  - High availability through recovery oriented computing.
  - OMNI—Operating Missions as Nodes on the Internet (NASA-GSFC)

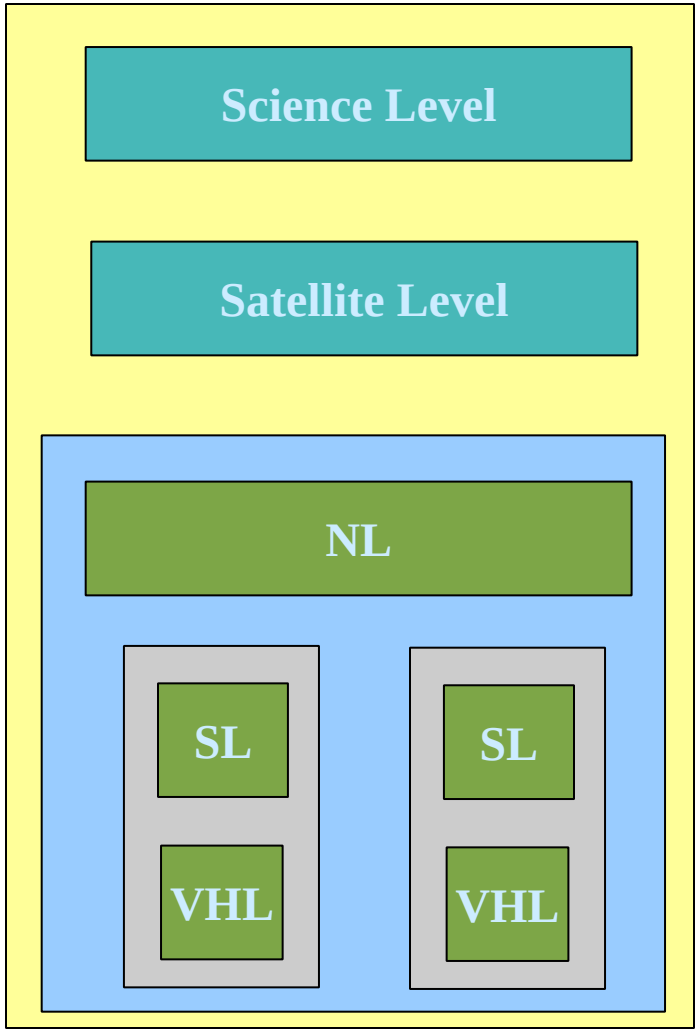
Status of Ground Stations

- Now, consider the state of ground stations.
  - They are tackling layers from low level RF to high level application issues.
  - Standardization battles => Complexity and high costs.
- What has enabled grad students in their spare time to create the likes of Yahoo, Google, and Paypal?
  - Fertile ground for innovation. Why?
  - Some say the *simplicity of the core Internet* that provides basic building blocks for users to develop applications.
  - At the heart of this is the *end-to-end principle:* "A lower layer of a system should support the widest possible variety of services and functions, so as to permit unanticipated applications" [Saltzer, Reed, Clark].
- This leads to a couple questions:
  - Can we apply the E2E argument to ground stations and provide the same fertile ground for innovation?
  - Can we standardize fundamental ground station services and provide a standard mechanism for *flexible application level support*?

GS Reference Model

- First, let's capture core ground station services:
  - Goal is to develop an architecture with core, simple services and a standardized mechanism for flexible application-level services.
  - We divide along lines of autonomy.

- *Virtual Hardware Level (VHL)* — fundamental capabilities of low-level hardware.
  - Master/slave control paradigm.
- *Session Level (SL)* — typical automation tasks of a single station.
  - Automated tracking, scheduling, health monitoring,etc.
- *Network Level (NL)* — services of a network of ground stations.
  - Scheduling and GS registry services.
  - Teamed ground stations cooperating on a pass.

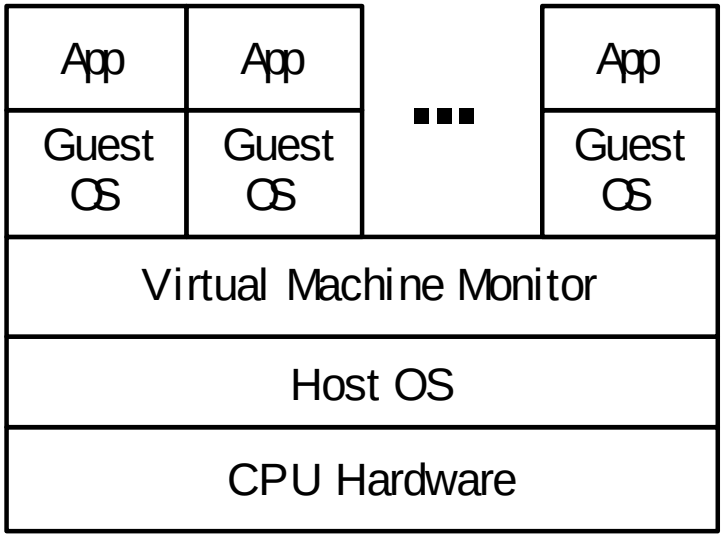


GSML

- GSML—Ground Station Markup Language.
  - We have captured this reference model in an API and protocol.
- GSML is an XML derivative.
  - Provides API and protocol for accessing ground stations and a network of ground stations.
  - Formally defined in XML Schema.
  - Currently used in SSDL ground station software, Mercury.
- Goal: explore the issues of multi-mission GS support.
  - GSML is NOT meant to be THE standard.
  - In fact, with Internet skills, we can easily move back and forth between different systems.
  - But we have laid some ground work from which to build.

Virtual Machines—A quick digression

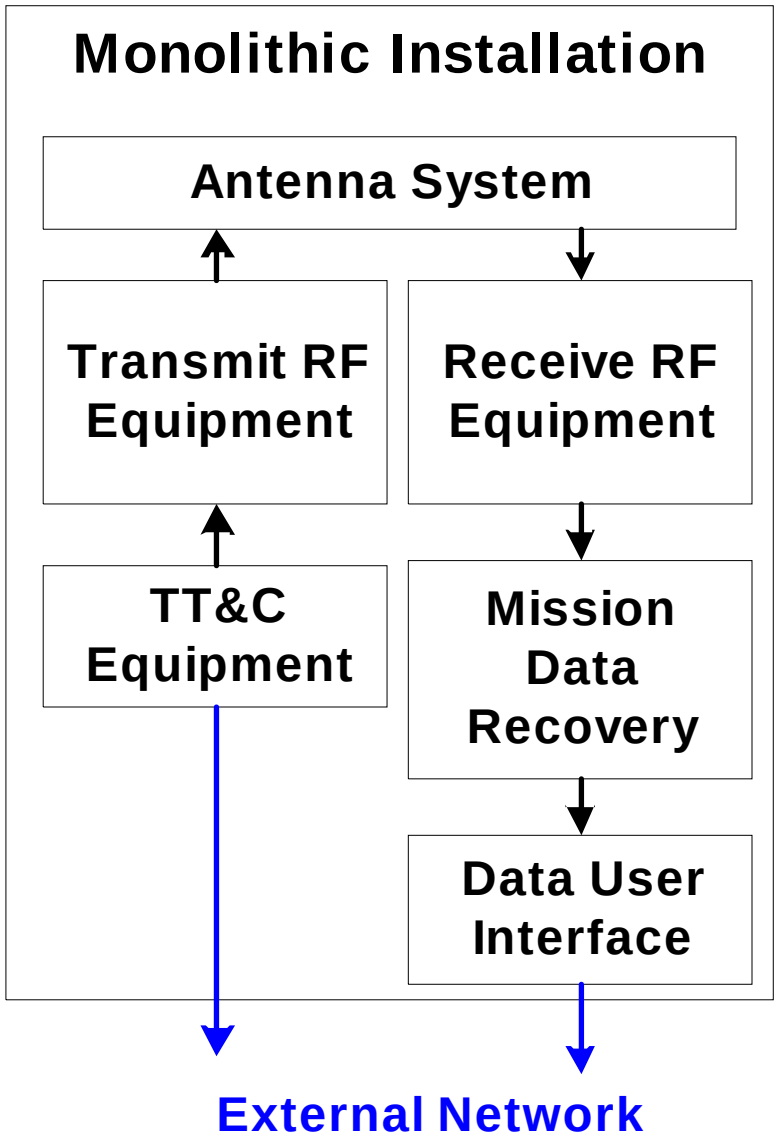
- A host OS and hardware running multiple guest OS, the virtual machines.
  - To the guest, it appears to them as if they are the sole machine.
  - To the host, it just appears as a running. A Virtual Machine monitor (VMM) controls and monitors the VMs. The VM is encapsulated in a file.
  - Common place in IBM main frames for years, but now making their way into mainstream computing (ie Vmware, Xen).
- Uses of VMs
  - Guest OS free from the hardware it is running on. Consider HW upgrades now. Just copy.
  - Facilitates backups and restorations.
  - Higher utilization of CPU resources.
  - Isolation, sandboxing, and security.
- What if a core GS service included the ability to run a VM?



Flexible Application Support

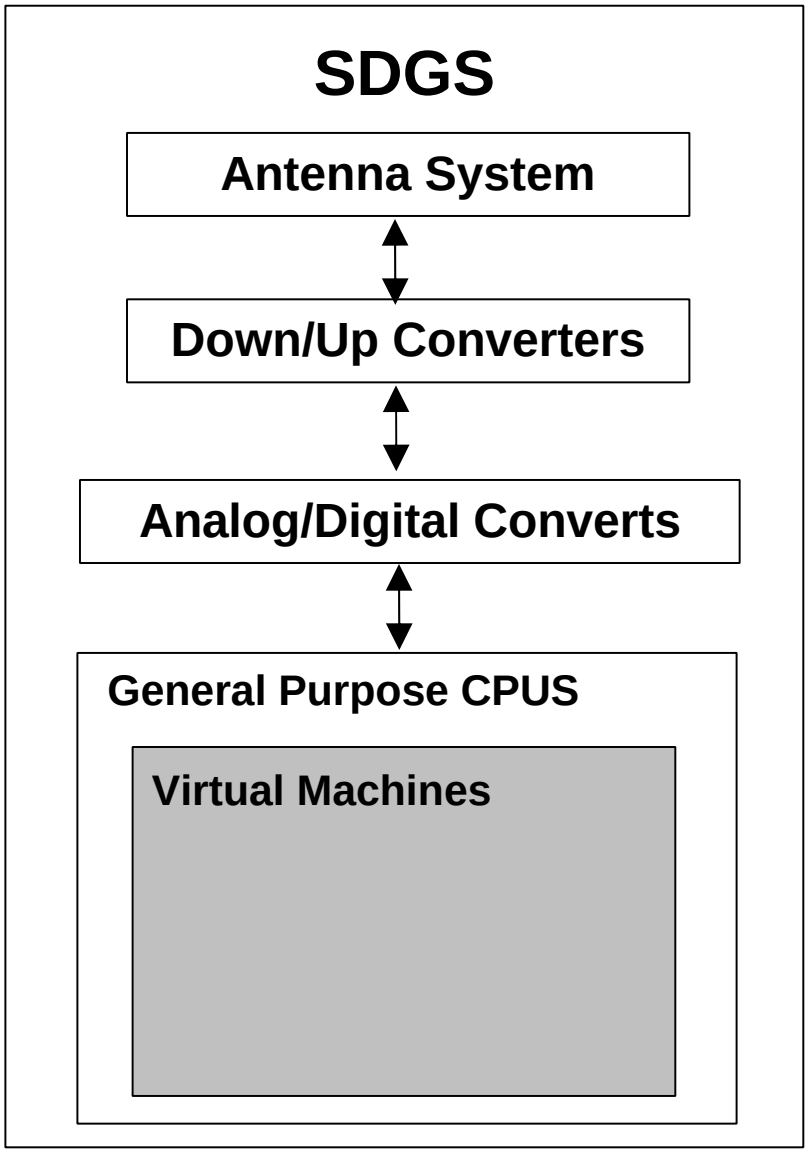
- Due to the nature of ground stations, some application support is likely needed.
  - Custom control of low level hardware devices
  - Large downlinks requiring longer term store and forward.
  - Supporting legacy missions.
- An interesting trend is the move to software intensive systems.
  - Traditionally, distinct hardware devices have been used for bit synchronization, FEC, packetization, security.
  - These services are now being captured in software on general purpose CPUs.
  - A ground station is...an antenna, amplifiers, a ADC, and a CPU.
- Now, combine these software centric ground stations with the concept of a virtual machine.
  - GS users can download their own, custom VM to perform all their bit sync, FEC, packetization, store and forward, data delivery, etc.
  - GS provides a standard mechanism for VM execution. Free from app specific knowledge.

Monolithic Installations



- Traditional, legacy systems.
- Characterized by custom hardware for each of the system components.
- Difficult to upgrade and support multiple missions.
- Fixed components.

Software Defined Ground Stations



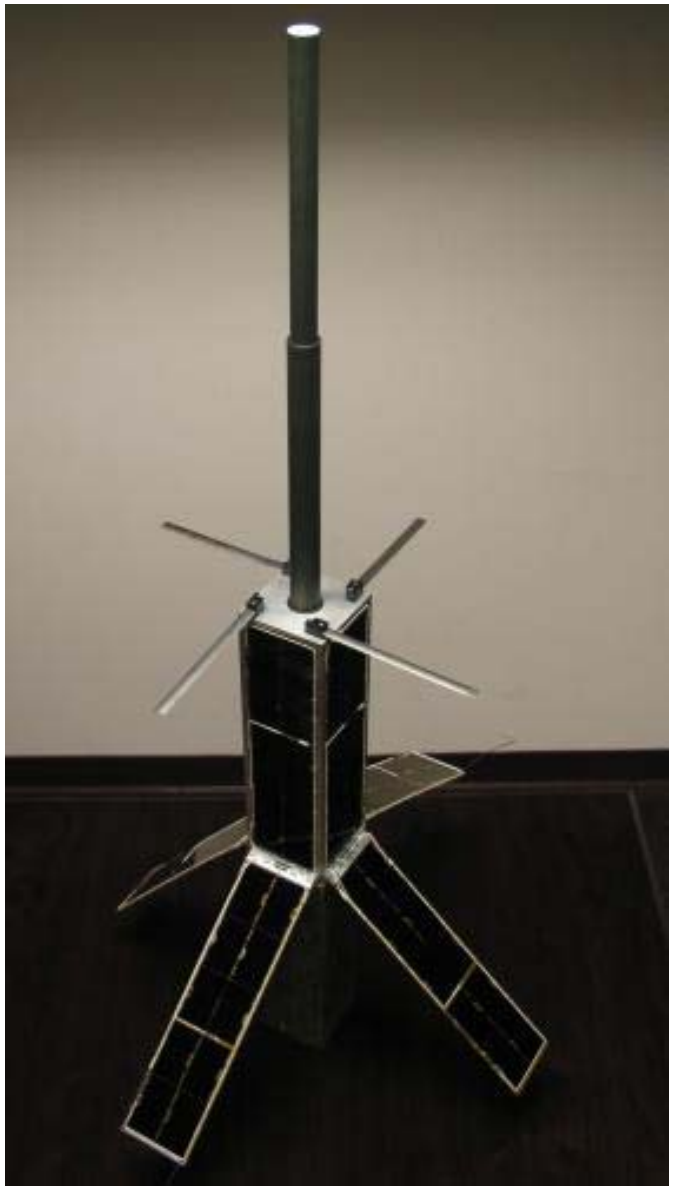
- Reduction in custom hardware
  - Antennas, amplifiers, up/down converters, ADCs, DACs
- Move everything else into a VM
  - Bit sync, FEC, packetization.
  - TTC, misson data, etc.
- VMs are now:
  - Portable
  - Upgradable
  - Customizable
  - Etc.

VM Examples

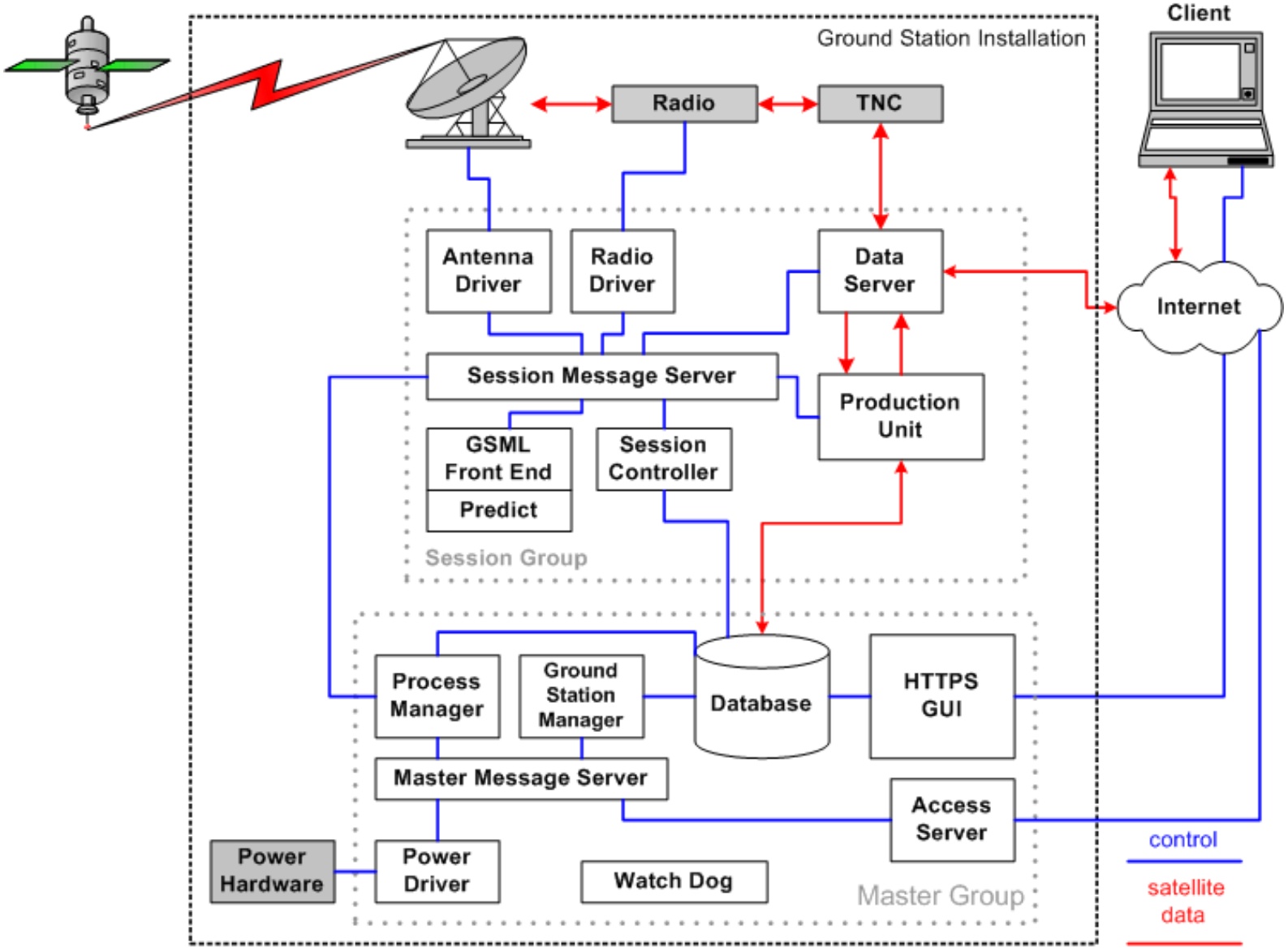
- We've developed VM's for our testbed.
  - QuakeSat VM to run application software in Alaska. Fairbanks station plagued by intermittent connectivity during winter.
  - AX.25 and IP over AX.25 VM for use at Stanford.
  - Pacsat VM. Using legacy windows software not internet enabled.
- Other potential VMS.
  - Linux router VMs.
  - CCSDS VMs.
- Other systems using virtual machines.
  - PlanetLab—an open, globally distributed platform for developing, deploying, and accessing planetary scale network services.
  - Emulab—another wide area platform for testing and development.
  - Industry support from IBM, Intel, Yahoo, etc. Being deployed to support Internet applications.

Testbed—Mercury Ground System

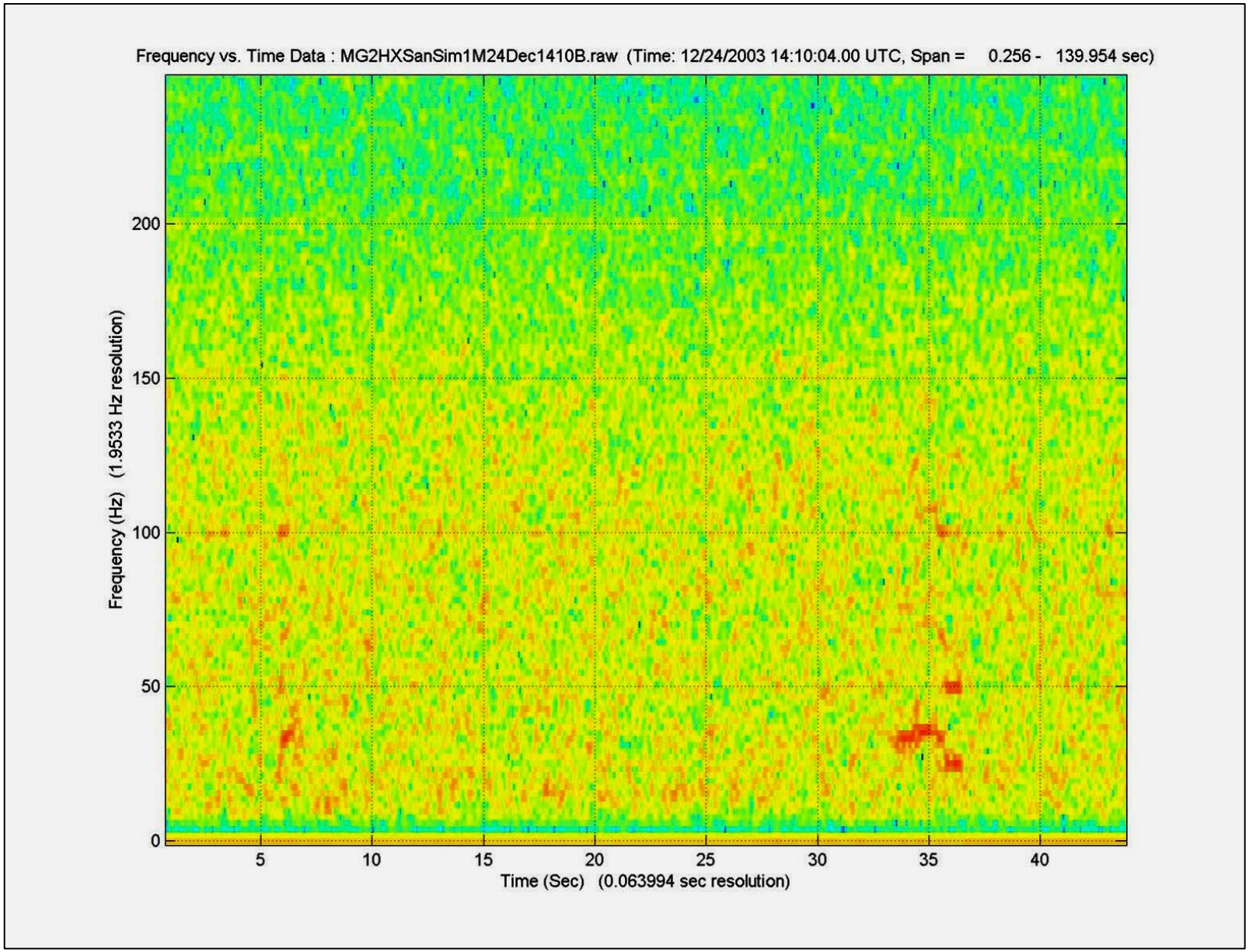
- Technology:
  - Open source: http://mgsn.sourceforge.net/
  - Linux, Apache, MySQL, PHP (but not explicity tied to these resources).
  - Custom code in Java, C, Perl.
  - Single CPU station possible.
  - XML schema of GSML compiled with databinding software. Auto code generation.
- Capabilities:
  - Supports heterogeneous hardware—GSML drivers.
  - Originally AX.25/TNC paradigm.
  - Generic app support with sat-specific Vms.
  - Production use since June on QuakeSat.
- Available to CubeSat community.



Mercury Architecture

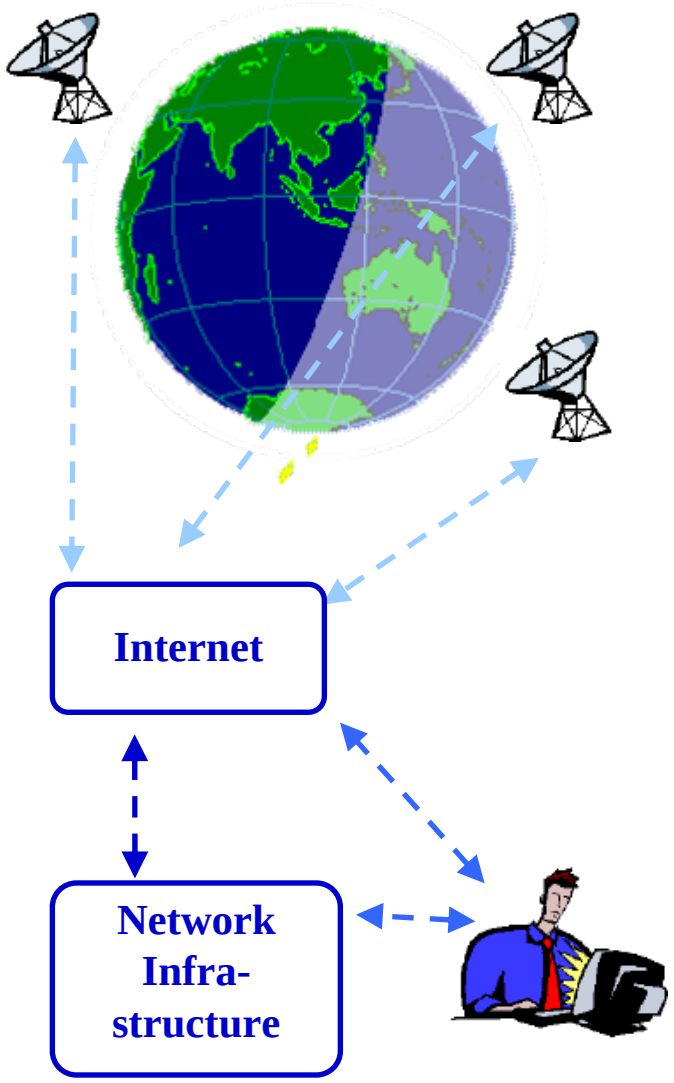


Quake Data


Frequency vs. Time Data : MG2HXSanSim1M24Dec1410B.raw (Time: 12/24/2003 14:10:04.00 UTC, Span = 0.256 - 139.954 sec)
Frequency (Hz) (1.9533 Hz resolution)
200
150
100
50
0
5
10
15
20
25
30
35
40
Time (Sec) (0.063994 sec resolution)

Next Step – MGSN

- A federated ground station network (FGN)
  - 100's of stations under different administrative domains—universities.
  - Globally distributed facilities that can dynamically join and leave the federation.
  - Heterogeneous and networked via Internet.
- Ability to designate teams of stations
  - Teams collaborate on high level task (e.g. "track this spacecraft").
  - Global teams to increase access windows.
  - Local clustering to optimize ground stations and provide path and node redundancy.
- Mercury Ground Station Network (MGSN)
  - Supporting university satellite missions and networking global ground stations.
  - Testbed for operations and Internet accessible.
- Deployed at Stanford, Alaska. Work underway in Norway, Germany. Who else?



Open Questions

- How do we handle clustered communication?
  - Swarms of cubes coming over at the same time?
  - TDMA, CDMA, multiple stations?
- How do we handle the mobility when performing end-to-end IP access?
- How do we leverage this new infrastructure to enhance mission automation?
  - Lights out operations.
  - Failure detection and recovery.
  - Opportunistic science execution.
- “Fair” scheduling of GS resources?
- Web services to support missions ops?
  - Scheduling.
  - Telemetry archival and analysis.

Conclusions

- Trends in space operations:
  - 24x7 Internet-like access by end users to space assets.
  - Software defined ground stations—flexible application support for multimission, legacy missions, and new technologies.
- SSDL has developed an open source GS network system
  - Mercury—a single station control system.
  - MGSN—beginnings of a global university network.
- CubeSat community efforts:
  - Tackle open questions.
  - Let's harness our GS capabilities and build a global network.
  - Cooperative development effort on standards and infrastructure development.
  - How are we going to handle the 12+ cubes launching this summer?
- More information:
  - http://swig.stanford.edu/, http://ssdl.stanford.edu/, http://www.mgsn.net/