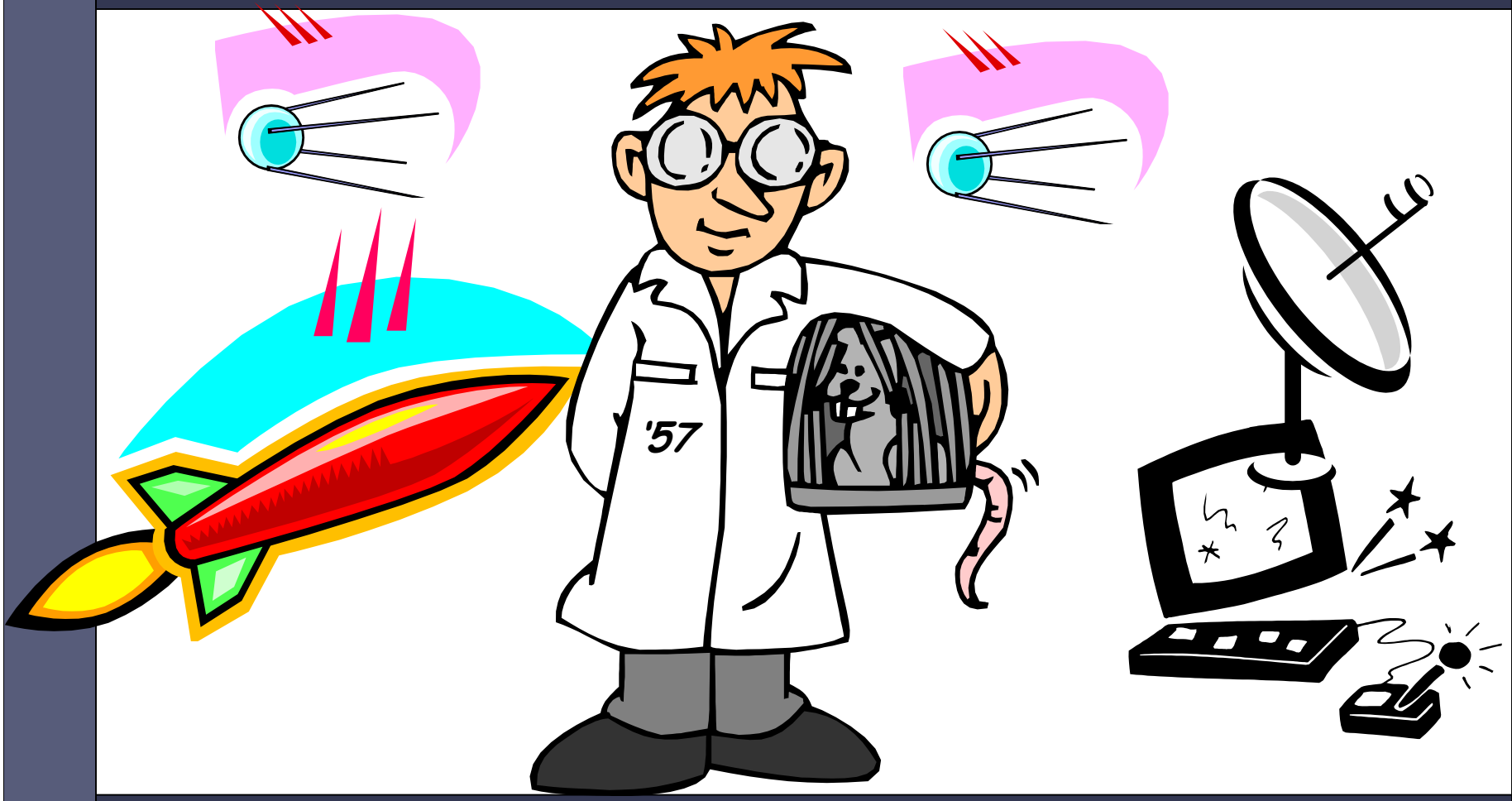
Open Sourcery and Web Services in Ground Segment Architectures

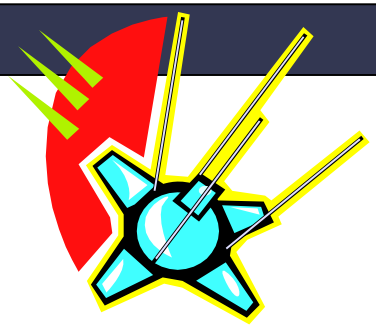
Alan W. Ptak

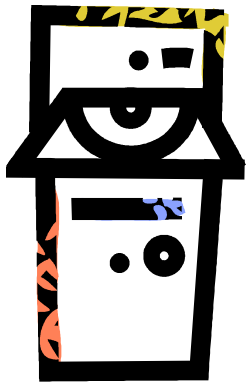
Kronosoft, Inc.

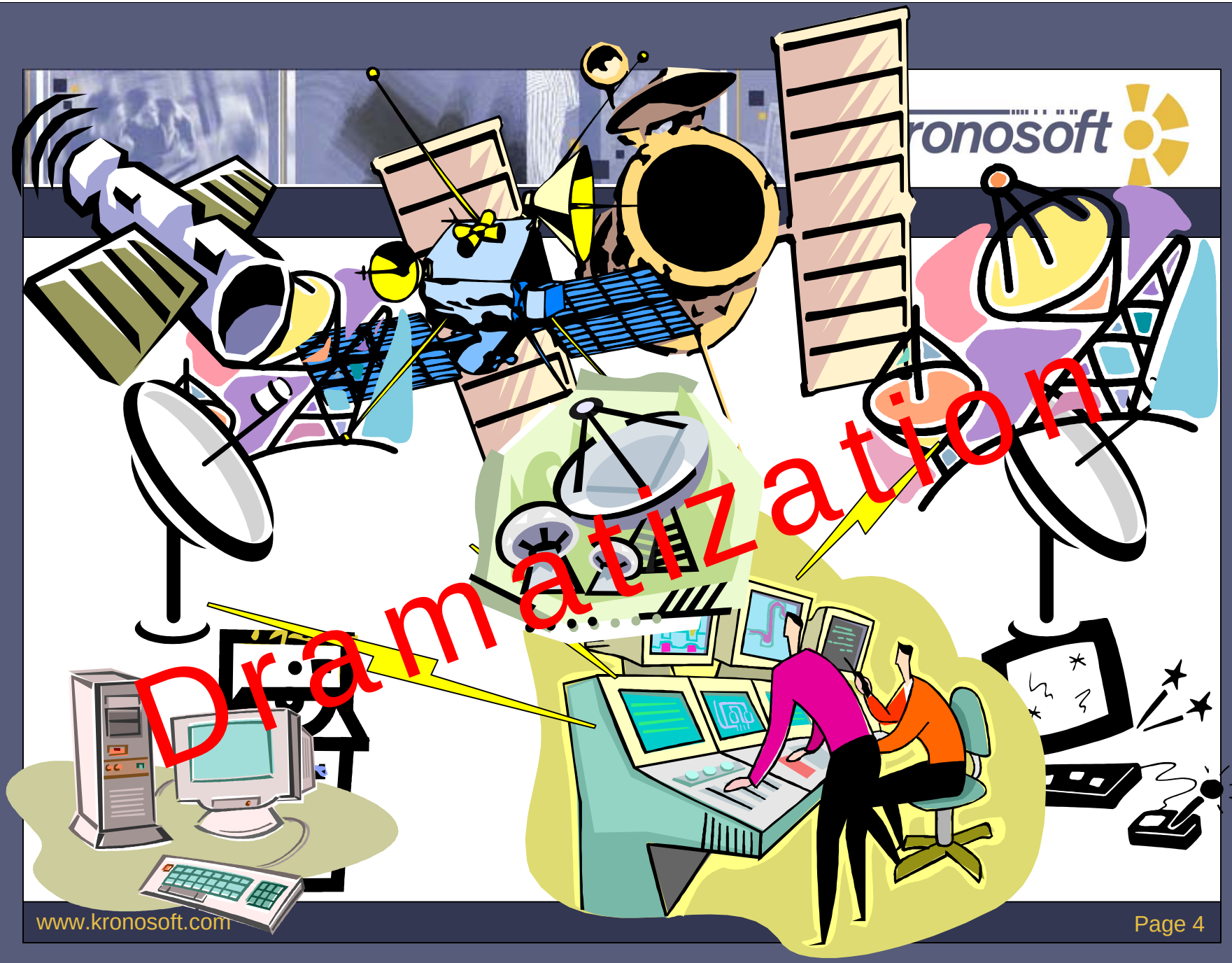
GSAW 2004

March 30, 2004


'57






ronosoft
Dramatization

Presentation Outline

- Motivation and background
- Multi-mission multi-user GS architecture
- Open source software & Internet technologies
- Web service technologies in ground systems
- Future directions

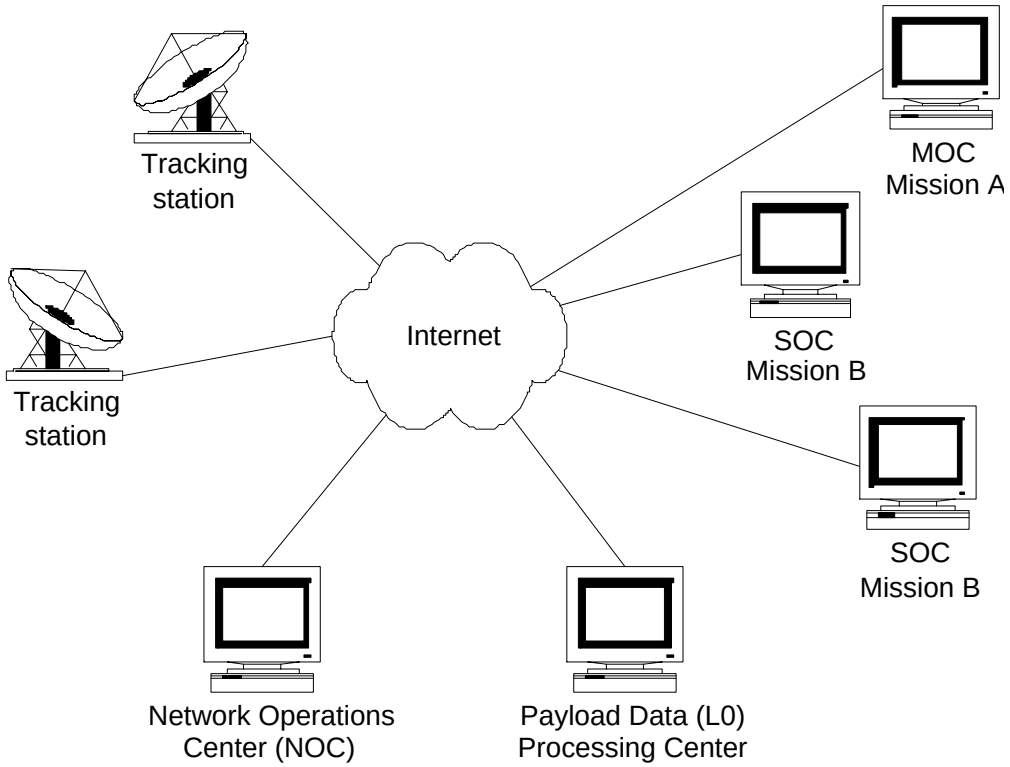
Background

- R&D experiment-in-progress
- Continuation of earlier work in autonomous systems and dependable computing (1993+)
- Kronosoft, Inc.
  - Khalil Foundy, Alan Ptak
- Collaboration with Canadian Space Agency
  - Dr. Leo Hartman
  - Software and Ground Segment Group
  - Space Technology Branch

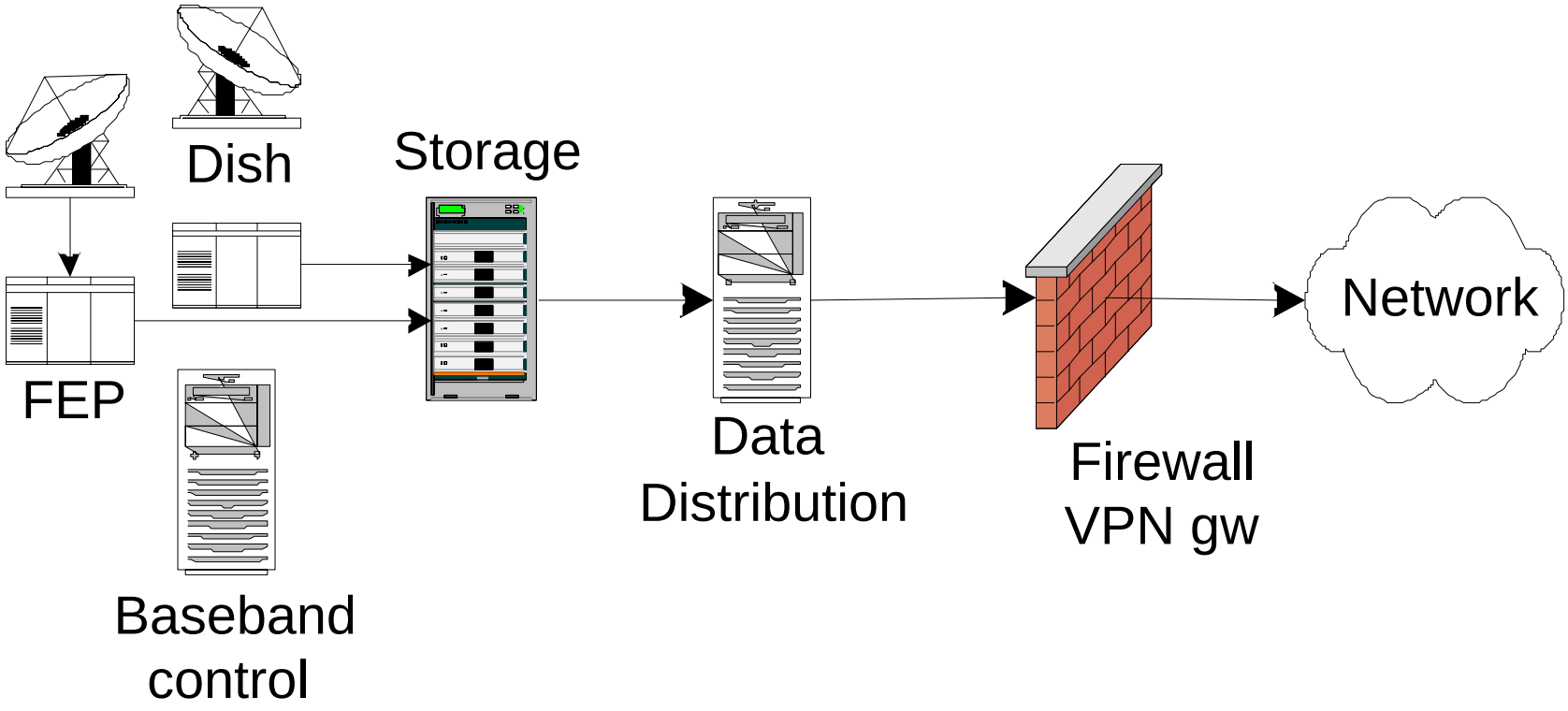
Motivation

- Improve capabilities for ground systems
  - Provide scaleable HA infrastructure
  - Provide feature-rich user applications
  - Reduce manual intervention for routine ops
- Reduce costs to build, operate & evolve
- Evaluate open source software (OSS) in mission-critical applications
- Explore the limits of possibilities with shoestring budget

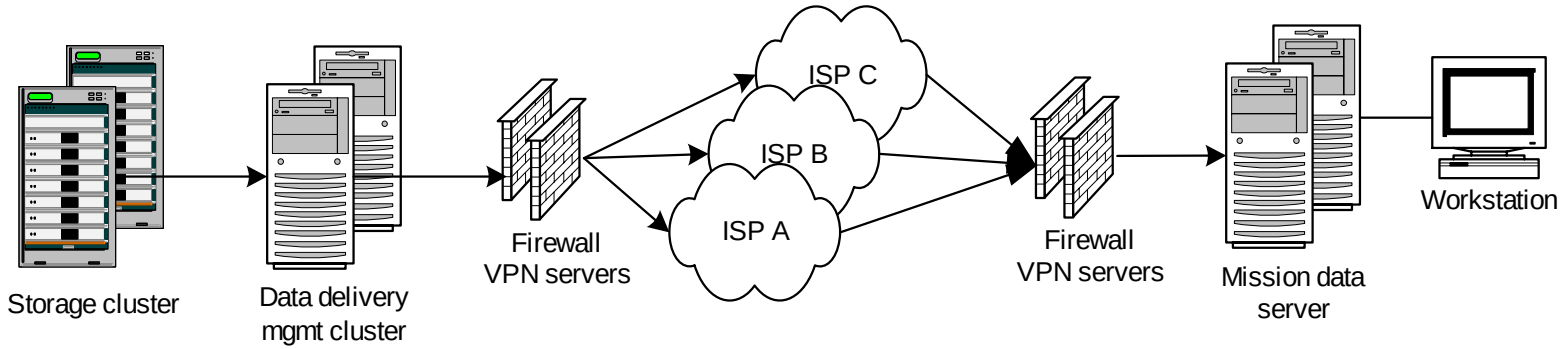
Generalized GS Architecture



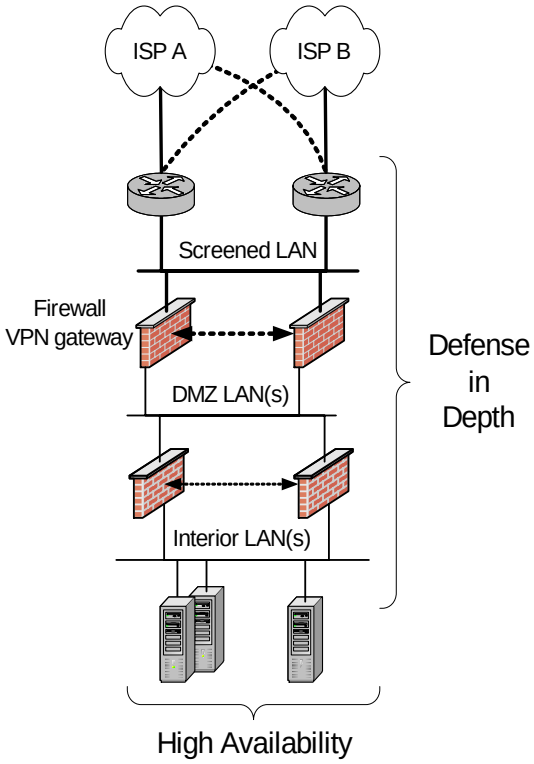
Tracking Station Architecture



Data Delivery Architecture



GS Node Common Architecture



- Redundant communications elements and services
- Automated stateful failover
- IPsec VPN tunnels
- Stateful inspection firewalls
- Application proxies
- Diverse security technologies

Implications of OSS Projects

- Distribute source code, not (only) binaries
- Allows code inspection, in-house fixes, modifications
- Mitigates vendor lock-in issues for users
- Creates new opportunities for developers and vendors

Open Source Challenge for SGS

- Divergence from standard aerospace custom engineering approach
- Explore a novel tradeoff in TCO
  - Components less specialized
    - easier to configure, target, reuse
  - Deployment no longer an issue
    - Focus on functionality, not license costs
  - Tailored/customized in standard ways
    - Fosters and simplifies experimentation

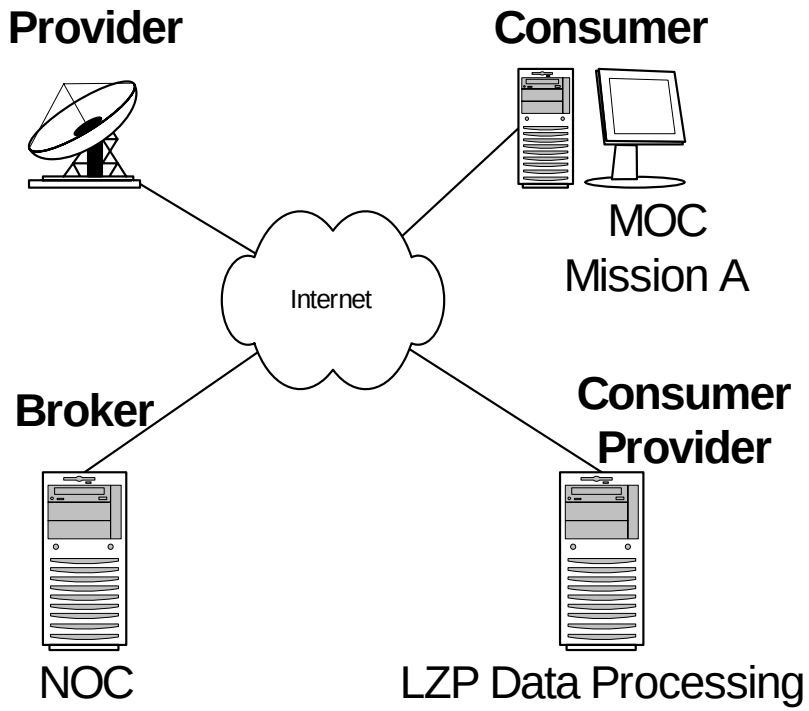
Future of Distributed Apps

- Autonomic computing will play a strong role in future networks
- Depends heavily on standarized components and interfaces
- Exploit off the shelf components and advances in terrestrial networking
- Modular GS network nodes (MOC, SOC, NOC …)

Web Services

- A new web application that provides programmatic access to your business
- Applications share data and invoke capabilities from other applications through vendor-agnostic interfaces
- Service providers
  - “write once, serve everyone”
- Service consumer
  - data consumer
- Service brokers
  - match requestors with appropriate providers

Web services GS example



A Software Paradigm Shift

- Web services applications are more flexible, quicker to build, and quicker to adapt than their monolithic counterparts
- Web services signify a shift in software development paradigms:
  - Segment large applications
  - Individual components exist as Web services.
- Diverge from standard aerospace custom engineering approach

Future directions

- An experiment in progress
- Data management applications
- Mission/network operations planning tools
- Security
- Prototypes to demonstrate capabilities

Conclusions

- Exploit advances in terrestrial computing and COTS
- Web services reduce costs for development, deployment and operations
- Modular GS architecture
  - Individual components evolve independently (wrappers)
- Standardization is key
- Novel tradeoffs in TCO:
  - Use of “generic” components
  - Deployment and development become non-issues

Open Source Software

- OSS model: open, sharing, collaborating, distributed
- Sourceforge.net: 76,550 projects, 800,735 users
- Freshmeat.net: 31,950 projects, 254,330 users

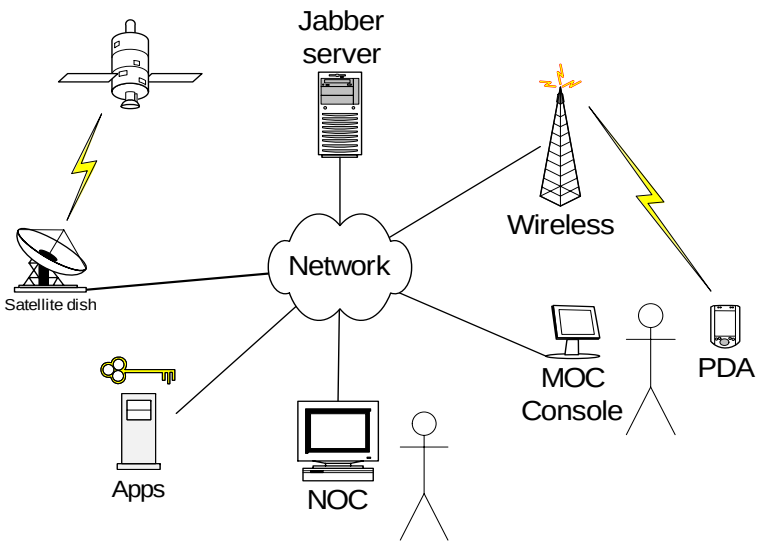
> "*It may well turn out that one of the most important effects of open source's success will be to teach us that play is the most economically efficient mode of creative work.*"

Eric Steven Raymond, *The Cathedral and the Bazaar*

Is Open Source for real?

- Successes:
  - LAMP: Linux, Apache, MySQL, PHP
  - GCC, CVS
  - Nmap, Nessus, openSSH, TightVNC
  - Samba, Spamassassin, Squirrelmail, Webmin
  - Mozilla, Gaim
- Linux 2.6 kernel ready for prime time

Jabber: Intelligent Messaging



- IETF draft protocols
- XMPP-core: transport
- XMPP-IM: application
- Based on std protocols
  - TCP, HTTP, XML, URI …
- Structured messages:
  - peer-peer, app-app …
- Simple clients, smart server
- Real-time intelligent messaging
- Extensible protocols
  - e.g., Publish/subscribe model

Web Services Foundations

- Standard Internet protocols and data formats:
  - TCP/IP and HTML – the ubiquitous foundation
  - XML – Data representation
  - SOAP – Intercommunication between systems
  - WSDL – Description of services
  - UDDI – Dynamic discovery of Web services

eXtensible Markup Language

- Self-describing *document*
- Separates content from presentation
- Represents any form of data: Web pages, database tables, telemetry, imagery …

```
<?xml version='1.0'?>
<!-- A very simple XML document -->
<bookmark>
  <article>
    <pageurl>http://science.slashdot.org/05207.shtml</pageurl>
    <pagetitle>Debugging the Spirit Rover</pagetitle>
  </article>
</bookmark>
```