

A composite image of NASA aircraft in flight against a clear blue sky. In the foreground, a large white Boeing 747-400 is shown from a low angle, flying towards the right. The tail of the aircraft features the NASA logo and the number 905. Above it, two smaller white aircraft are flying in a similar direction. The top aircraft is a NASA 701, and the one below it is a NASA 8001. The text "Using CORBASec to Secure Distributed Aerospace Propulsion Simulations" is overlaid in large, bold, yellow letters across the center of the image.

Using CORBASec to Secure Distributed Aerospace Propulsion Simulations

**NPSS CORBASec Test Bed
DOCsec 2001**

March 28, 2001

NASA Glenn Research Center

Tammy M. Blaser

Tammy.M.Blaser@grc.nasa.gov

NPSS CORBASec Test Bed

Presentation Overview

- Motivation and Goal of the NPSS CORBASec Test Bed
- NPSS CORBASec Architecture *
- NPSS CORBA Wrapped Architecture *
- Schedule Phased Buildup
- Tools Current, Planned and for Future Study
- Preliminary Phase 1 of 4 Test Results and Environment
- Issues for OMG Attention
- Summary

* NPSS Dev Kit supports wrapping NPSS simulations with CORBA/CORBASec



Computing and Interdisciplinary Systems Office
Glenn Research Center

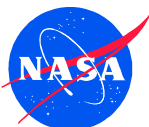
Page 2



NPSS CORBASec Test Bed

Motivation and Goal of the NPSS CORBASec Test Bed

- Develop a test bed using CORBASec software and other security tools (firewalls, etc.) to secure NASA Glenn and cooperative industry partners distributed numerical propulsion simulations.
- Test bed results will drive the Numerical Propulsion System Simulation (NPSS) CORBASec production development and deployment
 - NPSS is part of the High-Performance Computing and Communications (HPCC) Program
- The NPSS allows various aerospace companies and NASA Glenn and NASA Ames to simulate a full-scale system engine at various levels of fidelity (0D, 1D, 3D and back)
 - NPSS is built following the Object Oriented Paradigm using C++ and CORBA
 - Java developments growing rapidly (EJB based Web Servers, GUIs for testing, ...)



Computing and Interdisciplinary Systems Office
Glenn Research Center

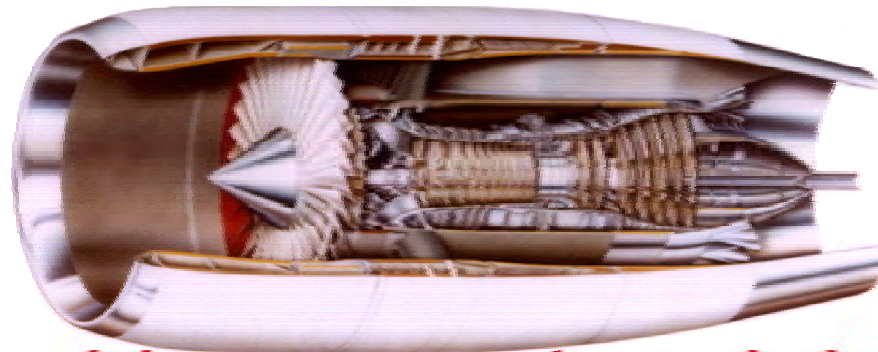
Page 3



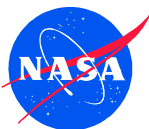
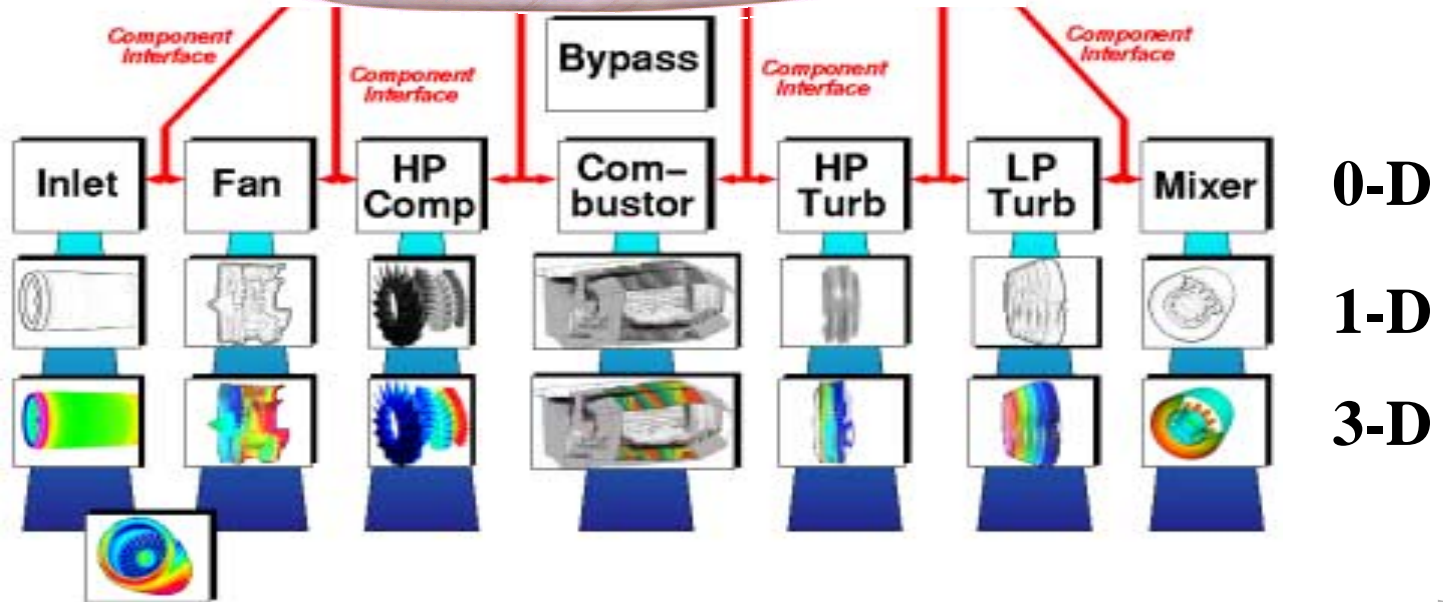
NPSS CORBASec Test Bed

NPSS Production and Simulation Architecture

NPSS Production System Model



NPSS Dev. Kit supplies tools for integrating codes, accessing geometry, zooming, coupling, security.



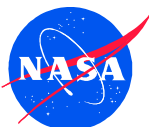
Computing and Interdisciplinary Systems Office
Glenn Research Center



NPSS CORBASec Test Bed

Collaborate on Multiple Component Architecture

- Collaborative/Multiple Domain First Phase Users:
 - NASA Glenn Research Center
 - Company 1
 - Company 2
- Multiple commercial ORBs, supporting CORBASec ORB Interoperability, will be implemented in the final phases (scheduling details in later slides)
- Exercising use of the SSL for on wire encryption (3DES) with CORBASec.
- Multiple Authentication techniques implemented and planned.
- NPSS Simulation developers use a CORBASec enhanced NPSS API Development Kit to enable and deploy CORBASec.
- Configured with multiple Firewalls.



Computing and Interdisciplinary Systems Office
Glenn Research Center

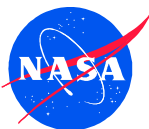
Page 5



NPSS CORBASec Test Bed

Primarily CORBASec Security-Unaware Architecture

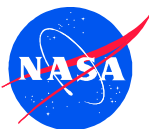
- Primarily uses CORBASec Security-Unaware (Enabled) *interceptor invoked* services
 - Simulation and Interpreter Interfaces with required privileges (rights) for public access
 - Security Policy Administered at Interface level with required rights for public access.
 - No methods configured for Security Policy Administration.
 - NPSS Access Control (AC) Administered based on the following combined attributes:
 - Aerospace Company or NASA Agency
 - Domain
 - Citizenship
 - Project
 - Role
 - General Users, Developers, and Restricted Users assigned privileges (rights):
 - » Private access limited to General Users and Developers (General Users have read-only private access)
 - » Public access granted to all User Roles (per Domain Access Policy)
 - » Restricted Users limited to Public access



NPSS CORBASec Test Bed

Plus CORBASec Security-Aware

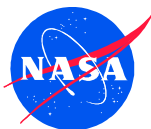
- 100% CORBASec Security-Unaware was our design goal, but ...
- NPSS AC proved runtime bound, therefore our design added a CORBASec Security-Aware *application invoked* “hook” and the Authorization (SecBuddy) Server was born.
- SecBuddy uses simple delegation and supports NPSS Client private access for dynamically invoked runtime bound operations.
 - via one hasPrivateAccess method call
 - Easy to update per application or security policy changes vs. changing multiple CORBASec Security-Unaware (Enabled) redundant methods
 - Old design had redundant methods get_public, get_private, set_public, set_private ...
 - Low coupling with CORBASec Administration
 - Supports growth and scalability



NPSS CORBASec Test Bed

Application Invoked AC

- Requires an Application System (NPSS Simulation, SecBuddy and Interpreter Servers) to be trusted (exercising formal application code inspections) to enforce NPSS AC decisions.
- SecBuddy Server, in a final configuration, will deploy a fault-tolerance design to make up for its dependence on Application invocation.



Computing and Interdisciplinary Systems Office
Glenn Research Center

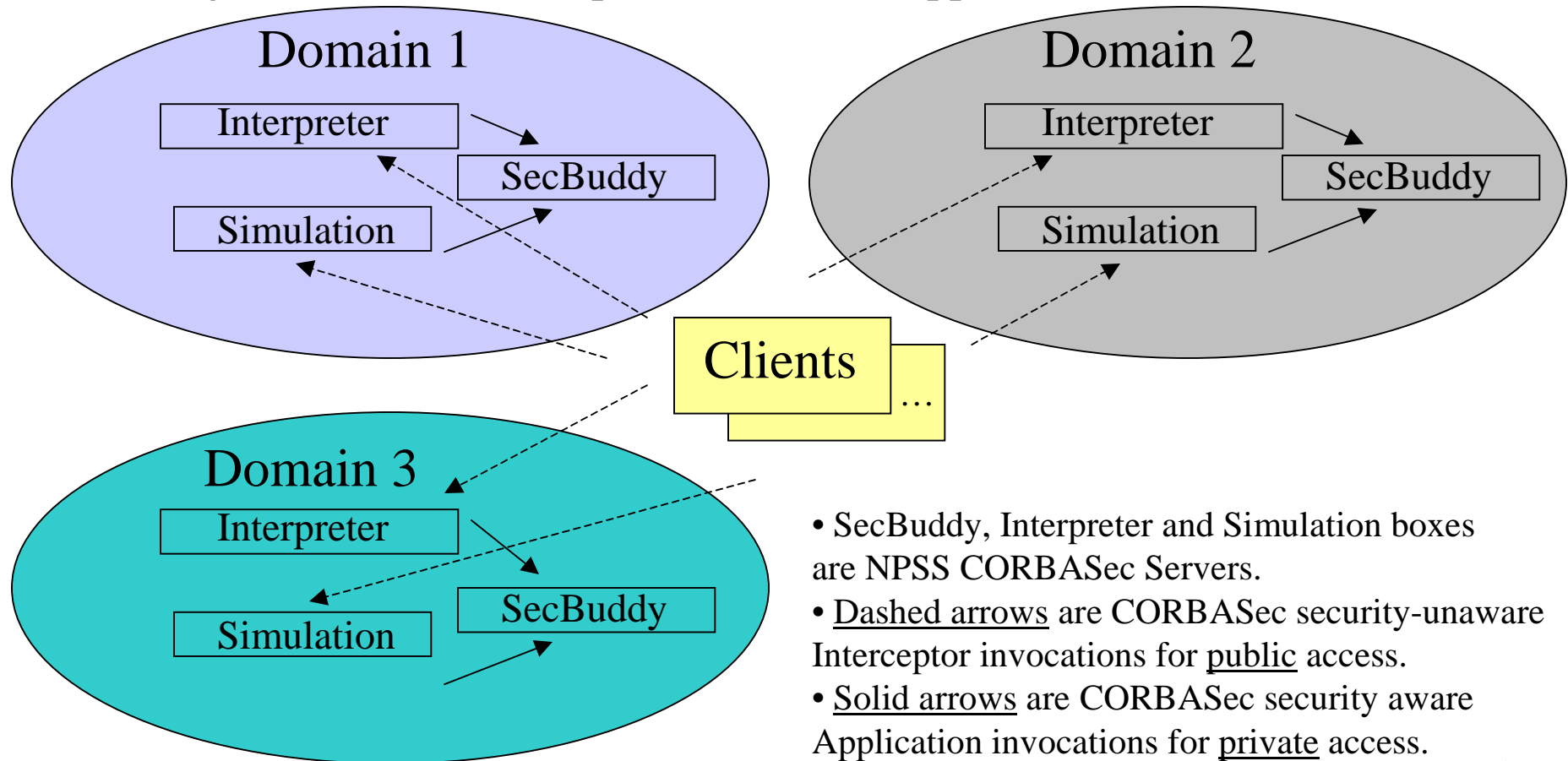
Page 8



NPSS CORBASec Test Bed

NPSS Dynamic AC

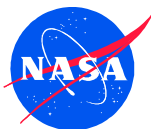
Using CORBASec interceptor services and application invoked Architecture



NPSS CORBASec Test Bed

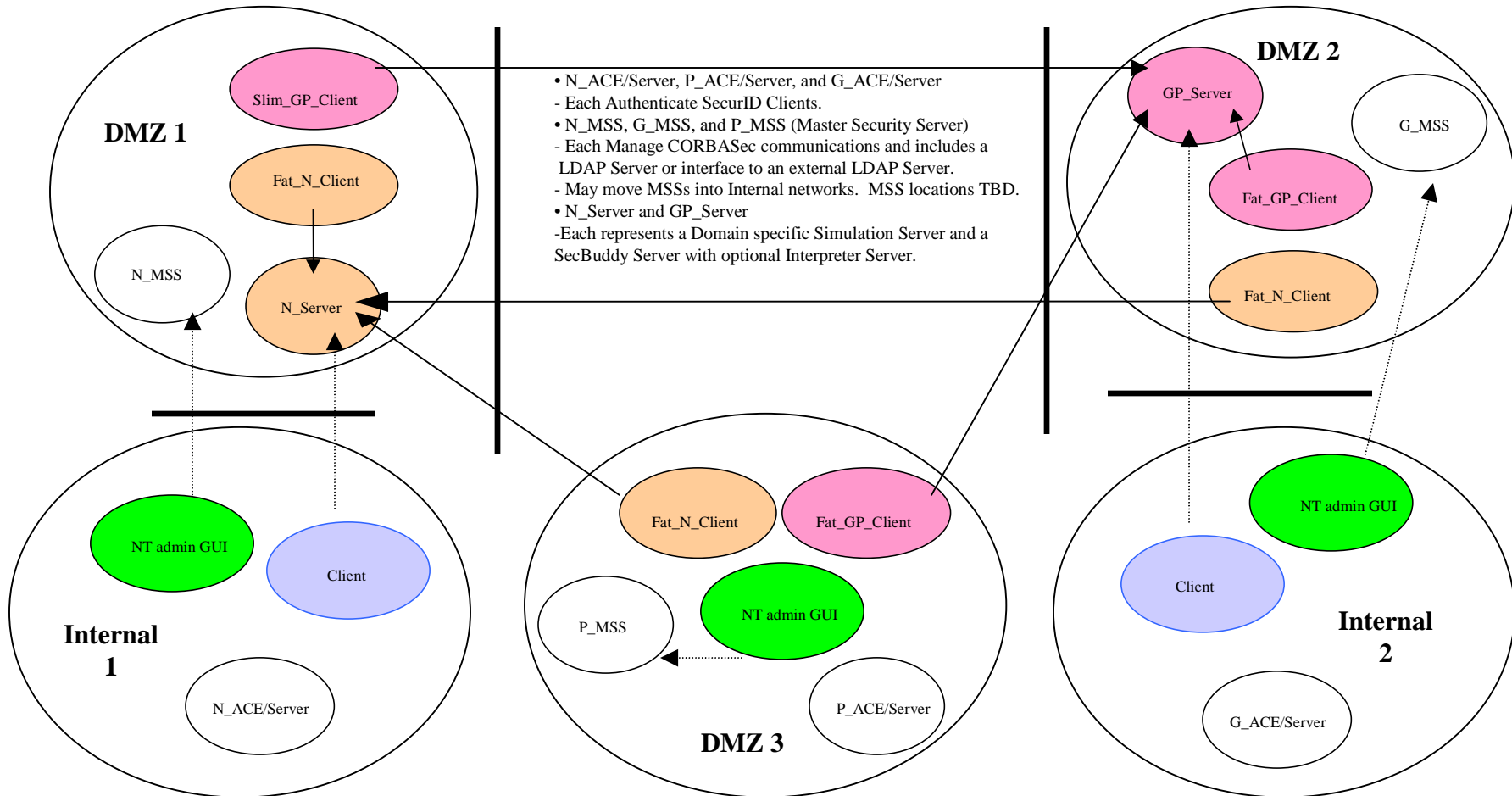
NPSS CORBASec Test Bed Example IDL

```
module npssCORBA {  
    interface npssObject {  
        // Every npss object can hold a table of protected variables. Security Policy is not directly Administered for Interface npssObject.  
        string get(in string varName);  
        boolean set(in string varName,  
                    in boolean isPrivate); // Many other npssObject methods exist in the production IDL  
    };  
    interface Interpreter: npssObject {  
        // Security Policy Administered for the Interpreter Interface at the Interface level (not at parseString method) with required rights for public access.  
        boolean parseString(in string cmdLine);  
    };  
    interface Simulation: npssObject {  
        // Security Policy Administered for the Simulation Interface at the Interface level with required rights for public access.  
        boolean runSim();  
    };  
    interface SecBuddy {  
        // Security Policy Administered at Interface level with required rights. Includes one method hasPrivateAccess with required rights for private access.  
        void hasPrivateAccess();  
    };  
};
```



NPSS CORBASec Test Bed

NPSS CORBASec Test Bed Phase 1 Firewall Architecture



NPSS CORBASec Test Bed

Integrated SecurID NPSS Client Authentication

- SecurID is two-factor authentication that is based on something you know (a password or PIN), and something you have (an authenticator; we use key token fobs known for their light-weight and compact size). The SecurID token generates a new, unpredictable access code every 60 seconds.
- Currently testing a SecurID workaround prototype that uses the RSA ACE/Server authentication API to integrate SecurID-based authentication with Hitachi's TPBroker Security Service 3.4 Login API .
- The ACE/Server authentication API was provided as C libraries.
 - NASA has wrapped the SecurID authentication API methods for use with NPSS & CORBASec using C++.
- The code looks something similar to the following:

```
NPSS prototype client {  
    Do SecurID login           // Call ACE/Server authentication APIs  
    If failed, exit  
    Do Security Service login // Call Security Service login API (necessary to create Credentials)  
    If failed, exit  
    Continue with processing  // Processing can continue only if both authentications succeed  
}
```

- After Hitachi Security Service 4.0 has implemented SecurID authentication

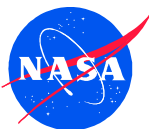
```
NPSS 4.0 client {  
    Do Security Service login // Call Security Service login API now integrated with SecurID  
    If failed, exit  
    Continue with processing  // Processing can continue if authentication succeeds  
}
```



NPSS CORBASec Test Bed

NPSS CORBA Wrapped Architecture

- NPSS is a component-based object oriented engine simulator.
- NPSS supports the use of external codes to extend a simulation to a higher fidelity and/or multidisciplinary analysis.
- Codes are CORBA wrapped to allow communication between NPSS and external codes.
- External codes use a direct CORBA wrapping scheme.
- Direct CORBA wrapping is accomplished via the easy to use NPSS Dev Kit (NPSS API for CORBA developers)
 - Transparent to the NPSS CORBA developer the API modifies the external code to become a CORBA Server conforming to the npssCORBA IDL
- The test bed results will provide the development roadmap required to add production grade CORBASec to the NPSS Dev Kit.



Computing and Interdisciplinary Systems Office
Glenn Research Center

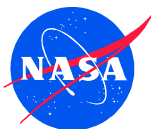
Page 13



NPSS CORBASec Test Bed

Schedule Milestones Phased Buildup

- Developed in four phases.
- Preliminary results of the NPSS CORBASec Test Bed first phase effort, will be available in March 2001, at that time, the NPSS software team will finalize the NPSS Dev Kit detail design using the BOA based CORBASec and using the POA based CORBA Server.
- The second phase will be completed in June 2001 and will support the POA CORBASec architecture and NPSS training events; to include a Dry Run of NPSS Dev Kit Training.
- A release of CORBASec will be provided for the majority of the NPSS platforms with the November NPSS Release.
- The third phase will be completed in December 2001 and will support the POA CORBA architecture and NPSS Dev Kit Training.
- The NPSS Onsite is scheduled in December 2001.
- The fourth and final phase release will support all NPSS platforms and will be provided in the February 2002 NPSS Release.
 - Due to a recent project budget cut back the final NPSS CORBASec release is expected to slip 4-6 months.



Computing and Interdisciplinary Systems Office
Glenn Research Center

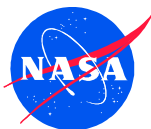
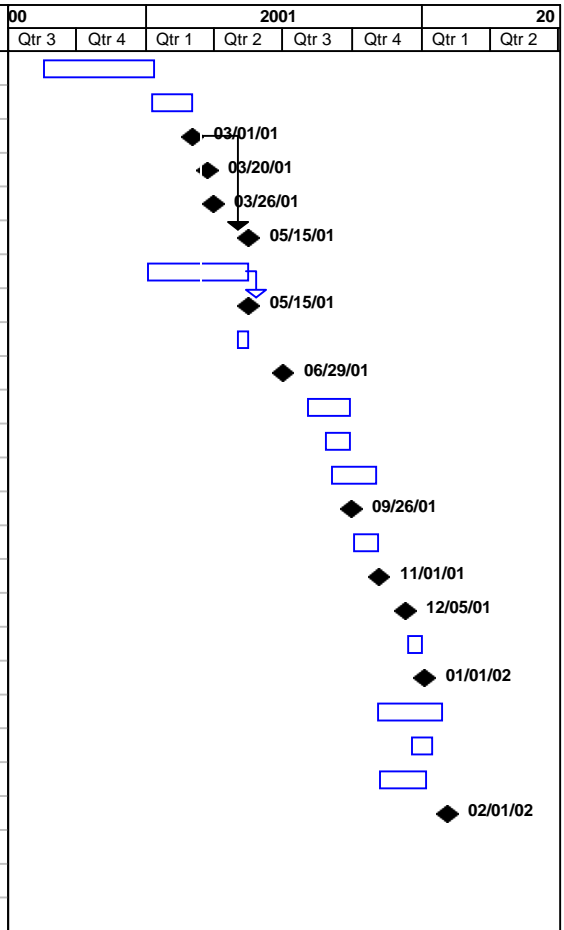
Page 14



NPSS CORBASec Test Bed

Detailed Task Schedule

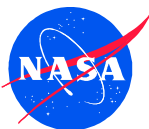
ID	Task Name	ORB Arch.	ORB	ORB Ver.	O.S.	2001									
						Qtr 3	Qtr 4	Qtr 1	Qtr 2	Qtr 3	Qtr 4	Qtr 1	Qtr 2		
1	NPSS CORBASec TestBed Solaris Java	BOA	TPBroker Java	3.3.2	Solaris										
2	NPSS CORBASec TestBed Solaris Reconfigure Java and C++	BOA	TPBroker Java & C++	3.2/3.4	Solaris										
3	NPSS CORBASec TestBed (Phase 1) Solaris Preliminary Testing Java and C++	BOA	TPBroker Java & C++	3.2/3.4	Solaris										
4	NPSS Onsite NPSS Dev Kit Briefing	POA	VisiBroker	4.1	Solaris										
5	DOCSec 2001 NPSS CORBASec Presentation	BOA	TPBroker Java & C++	3.2/3.4	Solaris										
6	NPSS Training Release w/CORBASec (BOA)	BOA	TPBroker C++	3.4	Solaris										
7	NPSS CORBA Server POA Solaris CR	POA	VisiBroker C++	4.1	Solaris										
8	NPSS CORBA Server (POA) Dry Run Training (Phase 2)	POA	VisiBroker C++	4.1	Solaris										
9	NPSS CORBA Server Port HP-UX	POA	VisiBroker C++	4.1	HP-UX										
10	NPSS CORBASec Port Solaris	POA	VisiBroker C++	4.1	Solaris										
11	NPSS CORBA Server Port NT	POA	VisiBroker C++	4.1	NT										
12	NPSS CORBA Server Port Linux	POA	VisiBroker C++	4.1	Linux										
13	NPSS CORBA Server Port Irix	POA	Mico C++	2.3.x	Irix										
14	NPSS Release w/CORBA Server (POA) UnSecured Majority OS	POA	VisiBroker C++	4.1	All w/o Irix										
15	NPSS CORBASec Merge Solaris/HP-UX/NT	POA	VisiBroker C++	4.1	Solaris/ HP-UX/NT										
16	NPSS Release w/CORBASec (POA) Majority OS	POA	VisiBroker C++	4.1	Solaris/ HP-UX/NT										
17	NPSS Onsite Dev Kit Training (Phase 3)	POA	VisiBroker C++	4.1	Solaris/ HP-UX/NT										
18	NPSS CORBASec Merge Linux	POA	VisiBroker C++	4.1	Linux										
19	NPSS Release w/CORBASec (POA) Linux	POA	VisiBroker C++	4.1	Linux										
20	NPSS CORBASec (MicoSec) Irix CR	POA	Mico C++	2.3.x	Irix										
21	NPSS CORBASec (MicoSec) Merge Irix	POA	Mico C++	2.3.x	Irix										
22	NPSS Web Security Merge	POA	VisiBroker Java ...	4.x	All										
23	NPSS Release w/CORBASec & Web Security (Phase 4)	POA	Visi Java/C++ & Mico	various	All										



NPSS CORBASec Test Bed

Tools Current

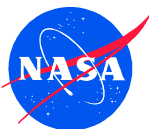
- Solaris 2.6 (2.8 - later phases)
- Hitachi TPBroker Security Service (SS) for Java and C++ 3.4
- Hitachi TPBroker (VisiBroker repackaged/hardened (BOA) ORB) for Java and C++ 3.x
- LDAP iPlanet Directory Server 4.12
- Sun JDK 1.2 (1.3 – later phases)
- Sun Sparcworks 5.0 C++ compiler
- RSA ACE Server and Agent v.4.1 for Solaris
- NAI Gauntlet 5.5 Firewall (6.0 – later phase)
- Checkpoint 4.0 Firewall



NPSS CORBASec Test Bed

Tools Planned

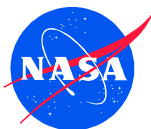
- In June, Phase 2 we will
 - Upgrade to Forte 6.1 C++ compiler.
 - Upgrade to Solaris 2.8.
 - Upgrade CORBASec using Hitachi SS 4 to POA based on VisiBroker 4.x. ORB.
- Phase 3 we will add
 - POA based on Orbix 2000 ORB using Hitachi SS for Orbix 2000.
 - Ports to HP/UX 11, Linux RedHat 6.2 and (NT 4 and/or Windows 2000)
 - Additional C++ compilers (HP aC++, GCC, Microsoft, ...)
- Phase 4 we will add
 - Port to Irix using MICO 2.3.4 ORB
 - Irix C++ compiler
 - MICOsec for NPSS CORBASec Irix implementation
 - EJB/J2EE Web Security (BEA WebLogic)
- Need to add software tools to support:
 - Security Policy Generation
 - Develop detailed Security Policy Plans
 - If an intruder breaks in will our policy be?
 - Shutdown or Track the intruder?



NPSS CORBASec Test Bed

Tools Planned/Recommended

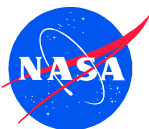
- Intrusion Detection
 - Choose a Intrusion Detection System based on detailed NPSS Security Policy Plans
 - Recommend Security Interfaces between CORBASec Non-Repudiation and Intrusion Detection Systems
- Certification
 - FIPS 140-1
 - FIPS 140-2 - when approved will replace FIPS 140-1
 - References Common Criteria
 - RSA BSAFE Crypto-C FIPS 140-1 (level 1) Certified (Integrated in VisiBroker SSL Pack)
 - Common Criteria
 - Solaris 2.8 (EAL 4) certified, but what about CORBASec?
 - CORBASec not ready, but to meet near term NPSS schedule ...
 - Recommend SSL to be Common Criteria Certified. Why?
 - » ANSI X.9F is in the process of embracing FIPS 140
 - » IV&V good engineering assessment
 - Pursue SSL Common Criteria (EAL 3) Certification for:
 - » RSA BSAFE SSL-C and SSL-J (Integrated in VisiBroker SSL Pack)
 - » Baltimore Technologies KeyTools SSL (Integrated in Orbix 2000; beta now)



NPSS CORBASec Test Bed

Tools for Future Study

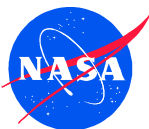
- Need to study potential of using other Authentication Controls:
 - PKI Digital Certificates/Smart Cards
 - Methods for managing multiple CA endorsed digital certificates
 - Biometrics
 - Fingerprint
 - Retinal Scan
 - Iris Scan
 - Voice Recognition
 - Face Recognition



NPSS CORBASec Test Bed

Preliminary Phase 1 of 4 Test Results

- Test Build Up Approach:
 - CORBA only Site Specific
 - CORBASec Site Specific
 - CORBASec Collaborate Network
- Completed IIOP proxy (CORBA only/non-CORBASec) tests between Site 1 DMZ (Firewall) and Internal network:
 - Tests Used Hitachi TPBroker Security Service example Client/Server s/w.
 - And NAI Gaunlet 5.5 Firewall IIOP Proxy
- Completed plug proxy CORBASec tests between Site 1 DMZ (Firewall) and Internal network:
 - Test Used SecurID and CORBASec Login workaround prototype.
 - Test Checked Out our NPSS CORBASec Prototype.
 - With SecBuddy, Interpreter and Simulation NPSS CORBASec Servers with various Clients.
 - Used NAI Gaunlet 5.5 Firewall Plug Proxy
 - SSL Proxy not used until enhancements made and Firewall Traversal Specification Baseline.



NPSS CORBASec Test Bed

Preliminary Phase 1 of 4 Test Environment

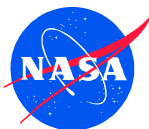
- Test bed effort requires coordination with many groups (networking, developers, system administrators, various companies, etc.)
 - Network engineers
 - Are the “keeper of the keys”
 - Do not know CORBA/CORBASec (but Network engineers are gaining elevation)
 - Understand Firewalls and VPNs
- Hitachi has proved to be a very professional company and their company policy is to actively work with the NPSS CORBASec project.
- Currently we are moving forward with our first set of collaborative (between companies) CORBASec tests.



NPSS CORBASec Test Bed

Issues for OMG Attention

- No Standard SSL API
 - SSL API required for multi ORB vendor CORBASec.
 - SSL Portability interfaces implemented by Hitachi Security Service as they port to both VisiBroker 4.x and Orbix 2000 ORBs.
 - SSL Interoperability required for end-to-end communication between NPSS partners.
- No Standard LDAP API
 - LDAP API will reduce risk of integrating multiple LDAP Servers (Site and CORBASec specific).
- Need Firewall Traversal Specification Baseline
 - Forward Identity (Delegation) of Client/Server Credentials may be required for the NPSS Collaborative Project in later phases.
 - Bi-Directional GIOP



NPSS CORBASec Test Bed

Summary

- We are integrating a production grade CORBASec capability with our component-based object oriented engine simulator (NPSS)
 - Porting to
 - Commercial ORBs (VisiBroker 4.x and Orbix 2000)
 - C++ and Java
 - Multiple operating systems (Solaris, HP/UX, NT, Windows 2000 and Linux)
 - With dedicated MICOSec development for SGI Irix platform
- Our test bed effort is key to the safe use and success of the NPSS project in its deployment phase.



Computing and Interdisciplinary Systems Office
Glenn Research Center

Page 23

