A Secure Next Generation Service Platform: Parlay

Gerald Lorang
T-Systems, ITC Security
Gerald.Lorang@t-systems.com

OMG DOC sec Workshop
2003.03.07
Overview of the Presentation

- Introduction
- Prerequisites
- PKI
- Architecture
- CCM mapping
- Component Architecture
- Security
Introduction 1

- **Service-Platforms**
  - Provide standardized means for
    - single sign-on
    - administration of services
    - Load balancing
    - Reliability of the systems
  
- Security is enforced by the platform,
  - abstraction of security
  - Security is transparent for the services
  - Accountability is handled by the platform
Introduction 2 – Parlay basics

Parlay offers

- Service Management
- Service Trading
  - Services are selected based on required properties
  - Access to services can be handled based on the properties of a person, e.g. age
- User Management
- Groups
- Service Contract Management
Prerequisites for the Secure NGSP

- **CORBA Security Services**
  - Message protection
  - Access Control / Auditing
  - Authorisation Tokens (ATLAS) for delegation purposes (Single sign-on etc.)

- **Threading support**

- **Public Key Infrastructure**
  - Checking validity of certificates

- **Components**
  - Platform Objects have multiple interfaces

- **Optional: Transactions and NR**
  - Will be handled by the CCM container implementation
PKI 1

- Public Key Infrastructure
  - Administration of certificates
  - Generation of Certificates
  - Checking validity of certificates

- DTSC- Group - University of Queensland, Australia
  - Definition of a CORBA PKI binding
  - Adopted OMG Standard
PKI 2

- CORBA binding:
  - Provides an easy to use API for accessing any PKI supporting it from a CORBA application
  - Two possible solutions
    - PKI- Wrapper, works with basically any PKI
    - The PKI provides the CORBA interface directly
PKI-Wrapper working with any PKI
Architecture of the Platform

■ Framework
  ■ Administration of Services
  ■ Session Handling

■ Services
  ■ Basic Services
    ■ Authentication
    ■ Heartbeat (checks if a system is active, cf. ping)
    ■ Load Balancing (limited)

■ Customer Services

■ The overall architecture leads to a modular design
Architecture of the Platform

- Client
  - Session handling
  - User Administration

- Services
  - Service Administration
  - Contract Management

Parlay Framework

CCM

CORBA & CORBASec/ CCMSec
Design and Implementation of the Platform

- **Framework**
  - Component model of the framework
  - Detailed threading model
  - Interfaces and interaction diagrams are specified in the Parlay standard (current Version 3.1)
  - First version without any security other than authentication, but with security in mind

- **Services**
  - Basic services - required to run the platform itself
  - Customer services, sample application
CCM- Mapping

- Parts of the framework will be components, eg.
  - Framework core
    - Service management
    - User Management
  - „Initial“ Component
    - Authentication
    - Opens Access Session
  - Access Session Component(s)
    - Session control interface
    - External Framework interfaces
  - Security will be integrated in all relevant components

- Services will be components
CCM – Mapping
Architectural considerations (1)

- Management Core - internal component without direct connections to the outside world
  - Service Management
  - Service Contract Management
  - User Management (subscription etc.)
  - Lots of data are required across these functions making separate components infeasible
CCM – Mapping
Architectural considerations (2)

- Interfaces - components providing access to external systems
  - Access- session: User-ID, setting User rights for all subsequent session upon availability of CORBASec 2.x
  - All interfaces requested will extend the session (server side only)
    - User ID can be obtained without using CORBASec repeatedly
    - Access Rights are set accordingly
    - session component
    - sessions will be silently closed when parent session is closed
Component Arch. - Overview

- **Client Component**
  - Session control

- **Framework**
  - Operating interface
  - Session control
  - Access Session Component

- **Framework Core**
  - Operating interface
  - Session & Interface Management
    - Sessions
    - Creation and deletion of external Interface components
  - Service Management
    - Service Discovery
    - Service Offering
    - Service Subscription
    - Service Contracts
  - Security Enforcement
    - Authentication
    - Access Control
    - Auditing

- **Session & Interface Management**
- **Service Management**
- **Security Enforcement**
Component Architecture Framework core (1)

- Provides the full set of interfaces described in the specification, but modified for the communication with the access session components (internal interfaces)
- Interacts only with the access session components which provide additional information
- Complete user, service, and subscription management
Component Architecture
Framework core (2)

FWServiceInstanceLifecycleManager
FWFwServiceRegistration
FWServiceProfileManagement
FWServiceProfileInfoQuery
FWServiceContractManagement
FWServiceContractInfoQuery
FWEntOpAccountManagement
FWEntOpAccountInfoQuery
FWClientAppManagement
FWClientAppInfoQuery
FWServiceDiscovery
FWServiceAgreementManagement

Framework Core

Service Data
Users

Service Profiles
EntOps

SAGs

Contracts
Service Providers
Component Architecture
Initial Component

- User Authentication
  - Using the underlying security mechanism
  - Using a special security mechanism implemented for the Framework
Component Architecture
Access Session (1)

- All access to the framework’s functionality and administration functions via the access session (the creation of an extra service session proves unnecessary)
- Several different access session components
- Set of interfaces depending on the client’s Role
  - User
  - Service Provider
  - Enterprise Operator
Component Architecture
Access Session (2)

- The access session object belongs to the user
- The framework receives user information and filters the information sent back to the client
- To be solved: reasonably easy changes of the role (currently a user would need to close the access session and re-authenticate with a different user ID)
Component Architecture
Access Session (3)

Access Session (client side)

Access Session (server side)

Session Control Interface

Client Role specific Interface

Client Data

Usage Data (later)

Connection to the corresponding Framework Core interfaces
Component Architecture
Non user-specific Services

- Session control
- Operating interface
- Interface component
- Service component

(for eg. a directory service, where a single component is shared between all users)
Component Architecture
User-specific Services

Service Component
(for eg. H323 communication where a component is created for every user/session component)

Session control

Operating interface

Client Component

Platform
Security 1 – General Requirements

- Secure transport
- Access control on any object carrying or transporting sensitive data
  - Session objects
  - Framework Core
  - Service Objects
- Access control for fault and integrity management
- Shutdown of abandoned sessions by the framework
Security 2 - Security functionality

- Low level, provided by the CCM security services
  - SSL
  - CSIv2
  - Access control/ Auditing / NR etc.

- High Level, inside the platform
  - Service usage or visibility can be restricted based on several properties of the person accessing the system, e.g. age
  - Service usage only permitted if a contract exists

- We focus here on the low level security features
Security 3 – General

- Based on a secure CCM implementation
- Certificate based authentication
- Access control
  - Discrete Access Control for sessions
  - Role Based Access Control to Management functions, service subscription functionality
  - Special considerations for certain services, e.g. multiparty services like conferences
- Accountability/ Auditing
  - Handled by the session components
Security 4 – System Architecture

- **Interacting parts**
  - **Platform**
    - Framework Core
    - Initial session components
    - Access session components
    - Service Components
  - **Outside**
    - ATLAS
    - PKI
Security 5 – Access Session

PKI

Verify (2) → Verify (2)

Initial Session (customer) → Authenticate (1) ← Initial Session (Platform, provider)

Acquire access session token and transfer to opposite side

ATLAS

Access Session (customer) ← Communicate ← Access Session (Platform, provider)

Communicate

Framework Core
Security 6 – Service Session

Access Session (customer) ➔ Find service ➔ Access Session (Platform, provider) ➔ Find service ➔ Framework Core

- Acquire service session token and transfer to opposite side
- ATLAS
- Communicate
- Service Session (customer) ➔ Service Session (Platform, provider)
- Sec. enforcement
- Sec. enforcement

Gerald Lorang
ITC Security
2003.03.07
Seite 29
Security 7 – Trusted Parties and Trust Relationships

- The PKI always acts as TTP (Trusted Third Party)
- Possible Arrangements of ATLAS
  - ATLAS acting as TTP (as shown)
  - ATLAS may also be hosted by the platform provider, if the customer trusts the provider
    - The platform will create all access tokens
  - An ATLAS may reside on both sides, requiring a big client
- Trust relationships
  - The customer trusts the platform provider (to some extent)
  - The service provider trusts the platform provider
  - Everybody trusts the TTP(s)
Security 8 – Trust Domains

- Customer side
  - Initial, Access, and Service Sessions
  - ATLAS (if located here)
- Framework
  - Initial and Access Session
  - Framework Core
  - ATLAS (if located here)
- ATLAS if acting as TTP
- PKI
- Goal: Create and assign access tokens only within the same trust domain
Security 9 – Distribution Scheme

- Currently, the framework core and the service session are on the same machine, connected via local interfaces.
- Sessions may run on a different machine than the framework core, requiring access control between elements of the framework.
- For load balancing and fault tolerance purposes, it may be considered to have more than one framework core running.
- Any remote access between framework core and access session objects requires:
  - Access control between framework core and access session components.
  - Solution: Delegation using CSIv2 features.
Security 10 – Distribution Scheme

1. Authenticate (1)
2. Open (2)
3. Find service (3)
4. Open (4)
5. Communicate (5)
Load Balancing and Fault Tolerance (1) – Fed. Frameworks

- **Two Basic Layouts**
  - **Tree**
    - Relies on a single Master Framework
    - Only the Customers can use the Framework if the Master goes down
    - No Administration possible if the master is down
    - Simple update procedure
  - **Star**
    - The Platform is supposed to remain fully operational if on or more Frameworks go down
    - Difficult update procedure
Load Balancing and Fault Tolerance (2) – Tree

- Master Framework
  - Framework
  - Framework
  - Framework
  - Service Provider

Updates:
- Update (1)
- Update (2)
- Update (3a)
- Update (3b)
- Update (3c)
Load Balancing and Fault Tolerance (2) – Star

Framework

Update (2a)

Update (2b)

Update (4b)

Update (4c)

Framework

Framework

Framework

Framework

Service Provider 1

Update (1)

Service Provider 2

Update (3)

Update (4a)

Update (4b)

Update (2c)
Conclusion

■ The current architecture
  ■ solves security problems
  ■ can fulfil security requirements
  ■ maintains full compatibility with the Parlay standard

■ Implementation plan
  ■ 2002 - done
    □ Service- and User management
    □ (Parlay-) Session components
    □ Documentation
  ■ 2003
    □ Services
    □ Demo Application
Contacts and Partnerships

- Gerald.Lorang@t-systems.com
- This implementation is funded by the IST project COACH
  
  ![COACH Logo]

- http://WWW.IST-COACH.ORG
Contacts and Partnerships 2

Contact Person:
Uwe G. Wilhelm
T-Systems
D-64307 Darmstadt
Germany
Uwe.Wilhelm@t-systems.com