



Khimaira

A unified user-interface for
AFS and Kerberos
credentials management for
Microsoft Windows

Overview

- Introduction
- afscreds
- Leash32
- Khimaira
- Architecture
 - Overview
 - User interface & Configuration
- Conclusion

Introduction

- Credentials management on Windows
 - OAFSfW provides afscreds
 - MIT KfW provides Leash32

afscreds

- Provided by OpenAFS for Windows
- Allows user to
 - Obtain new tokens
 - View/delete existing tokens
 - Receive notifications of token expiration
 - Manage drive mappings
 - Invoke the AFS client configuration applet

Leash32

- Provided by MIT Kerberos for Windows
- Allows user to:
 - Obtain, view and manage Kerberos tickets and AFS tokens
 - Receive notifications of ticket/token expiration
 - Manage Kerberos and AFS configuration
 - AFS configuration through invocation of AFS client configuration applet

Issues

- Two applications with overlapping functionality
 - Confuses users
 - Additional work to maintain two applications
- Extensibility
 - Not feasible to add extensive Kerberos support to afscreds
 - Leash32 not extensible for additional credential types

Khimaira



Khimaira

- Origin
 - “Khimaira” is Greek spelling of “Chimera”
 - WordNet says of Chimera; “fire-breathing she-monster with a lion’s head and a goat’s body and a serpent’s tail; daughter of Typhon”
 - Developed at
**Massachusetts Institute of Technology,
Information Services & Technology**

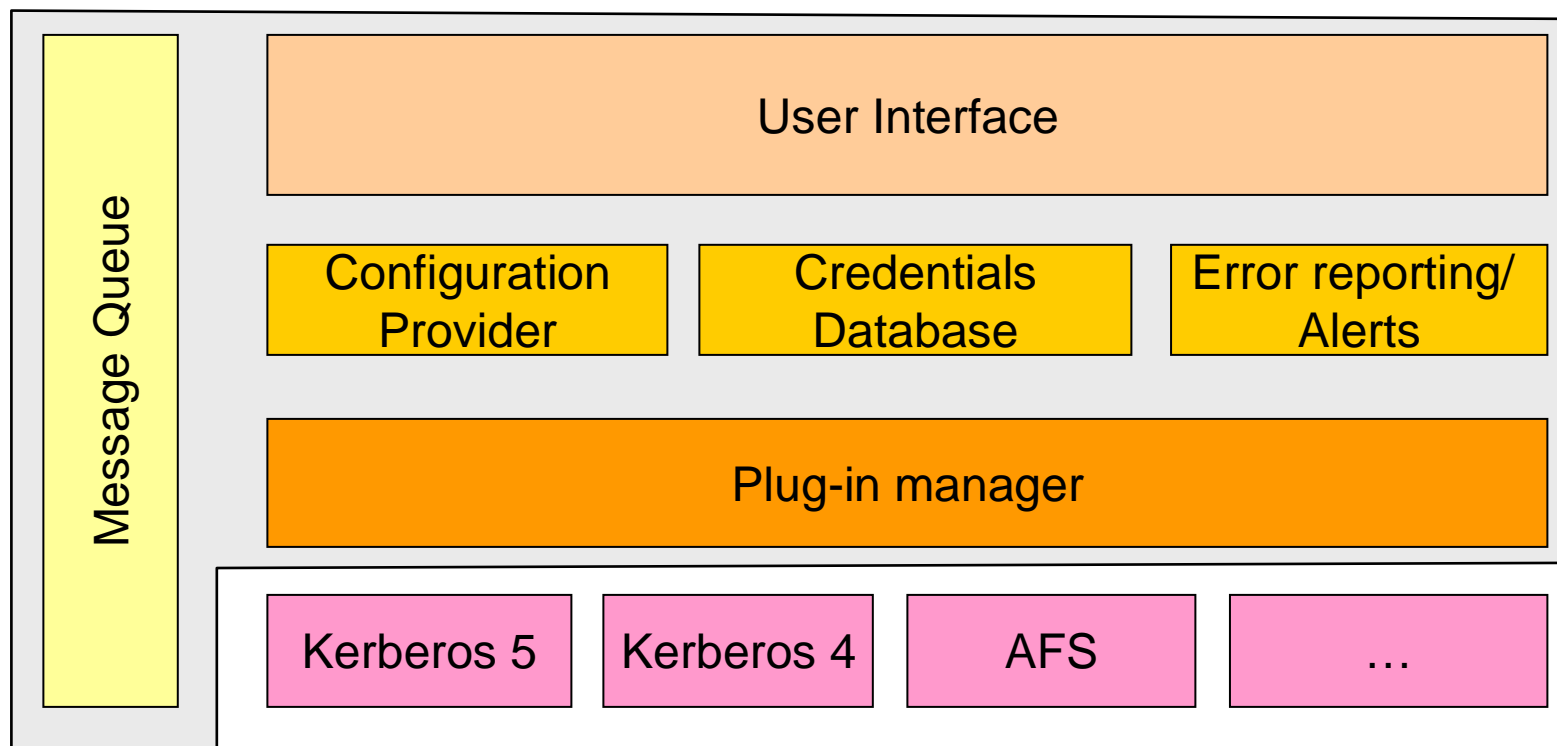
Khimaira

- Goals
 - Provide functionality of afscreds and Leash32
 - User friendly
 - Minimize required knowledge (“Shield” user)
 - Intuitive
 - Extensible
 - Additional credential types
 - Additional functionality
 - Easy to maintain
 - Independent maintenance of Kerberos/AFS/other application areas
 - Robust

Khimaira

- Design
 - “Clean” user interface
 - Identity centric
 - Customizable views
 - Plug-in based architecture
 - Main application has no knowledge about Kerberos or AFS
 - Kerberos and AFS support provided through different plug-ins
 - Comprehensive configuration (...)

Architecture - Overview

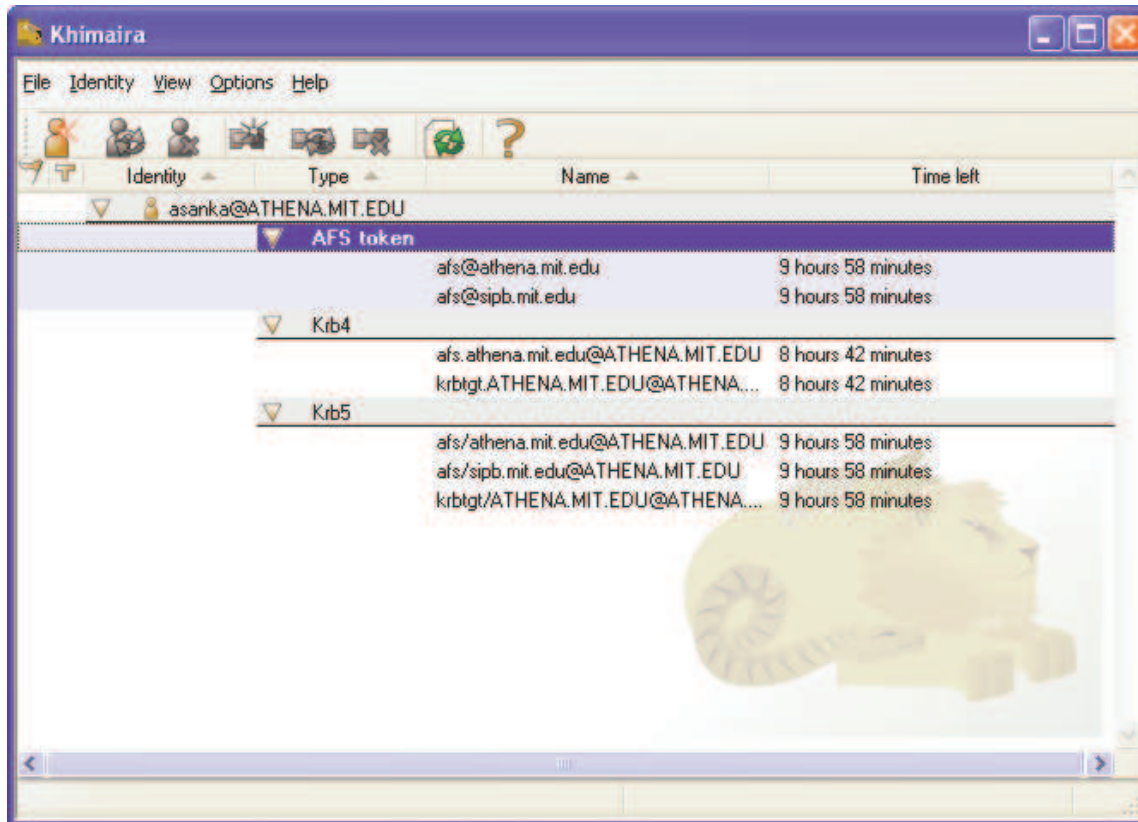


User Interface

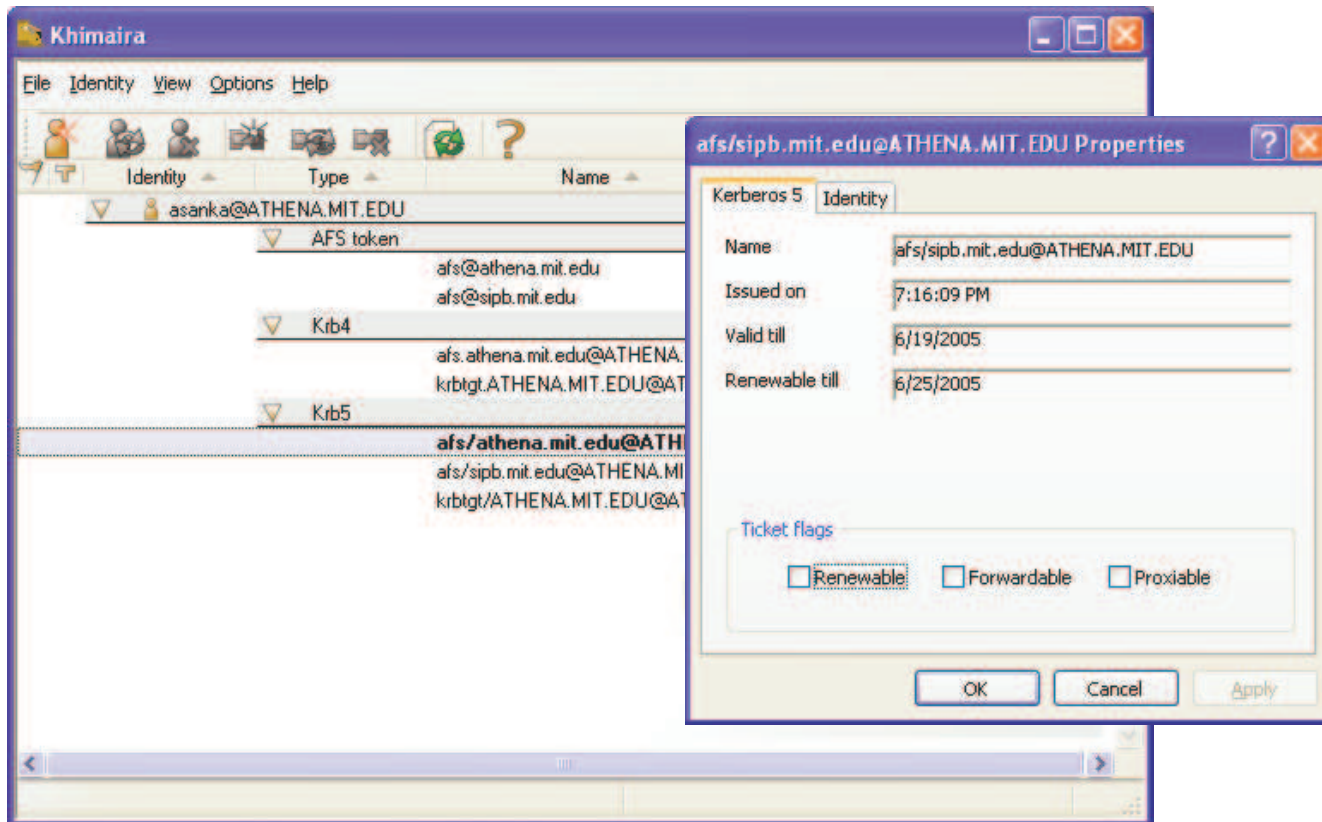
- Identity centric
 - Identities correspond to Kerberos principals
 - Defined by an “Identity provider” plug-in (currently Kerberos 5)
- Hierarchical views
 - Uncluttered view of useful information
 - Configurable
- Modular design
 - Extensible



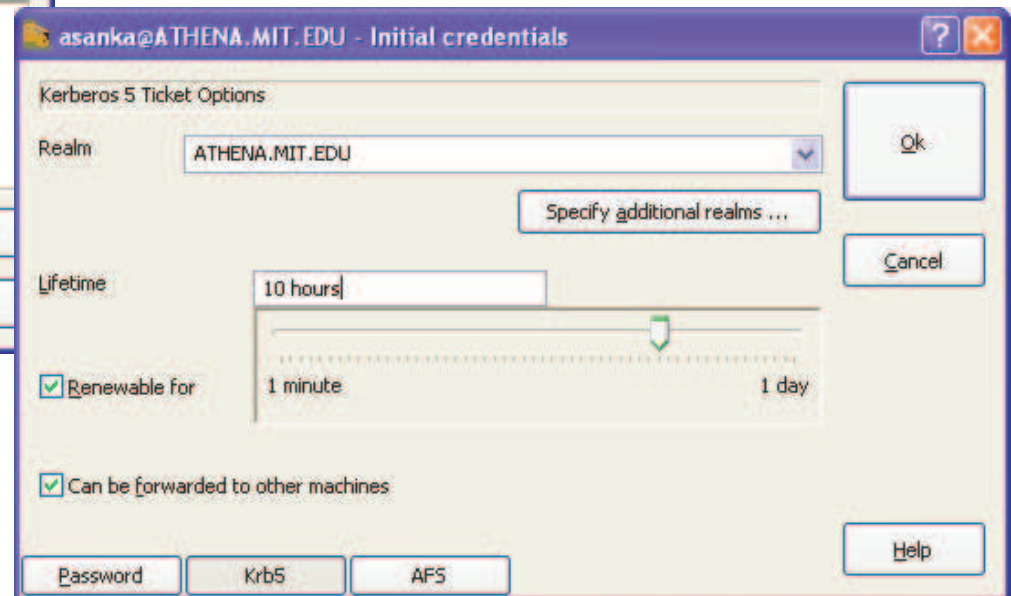
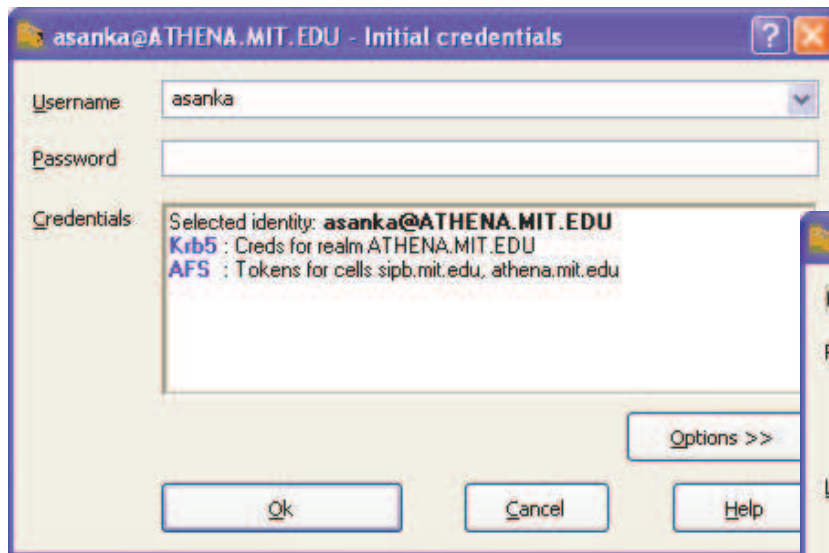
User Interface



User Interface



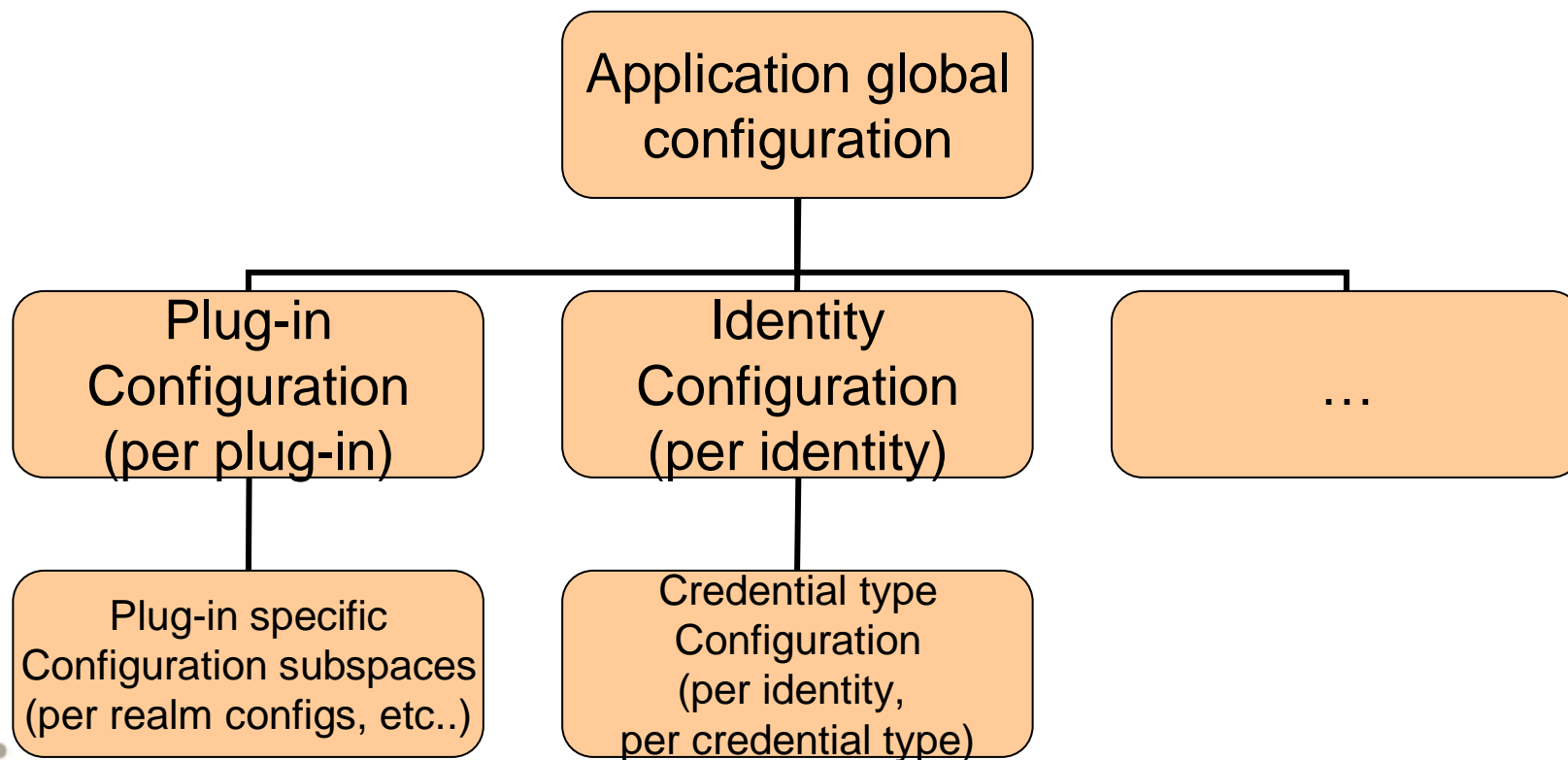
User Interface



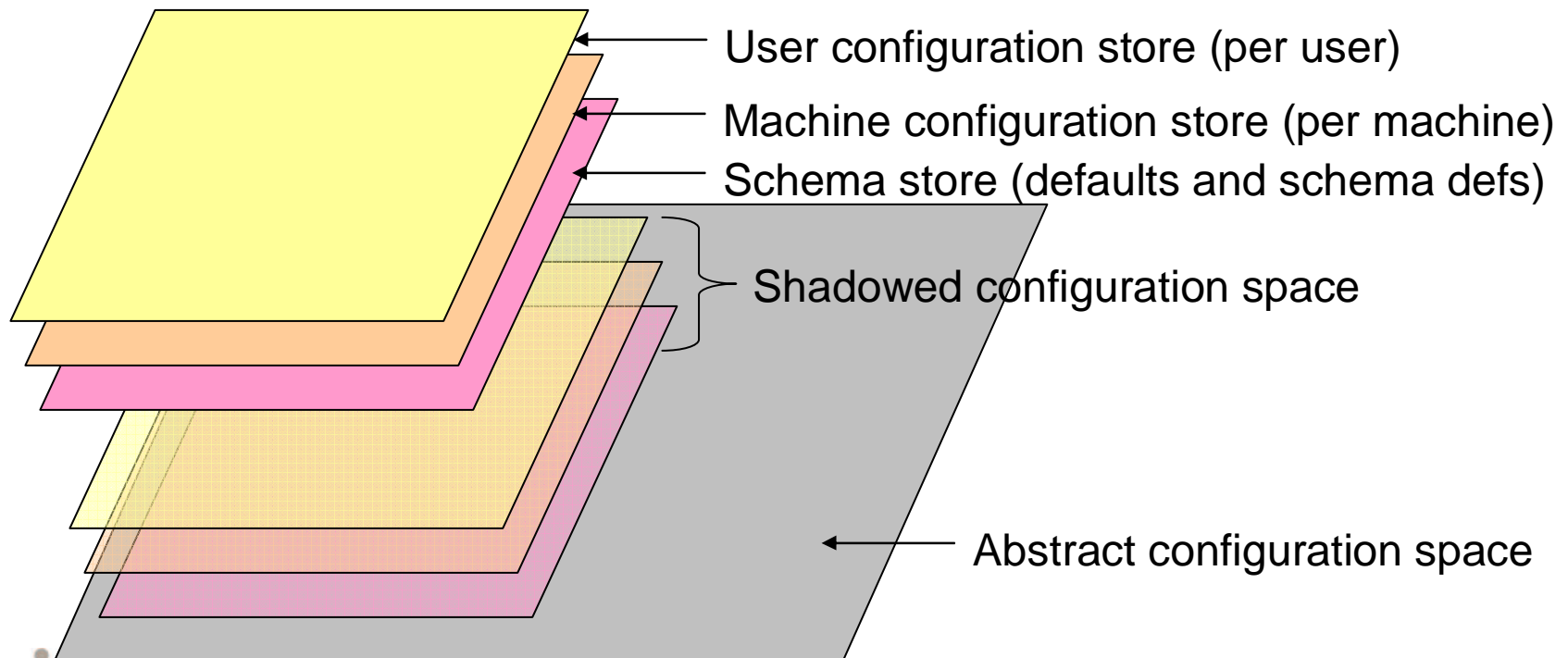
Plug-ins

- Message based communication
- Several “weak” types
 - Credentials Provider / Identity Provider / Configuration Provider / etc...
- Plug-ins for plug-ins
- Run in isolated threads

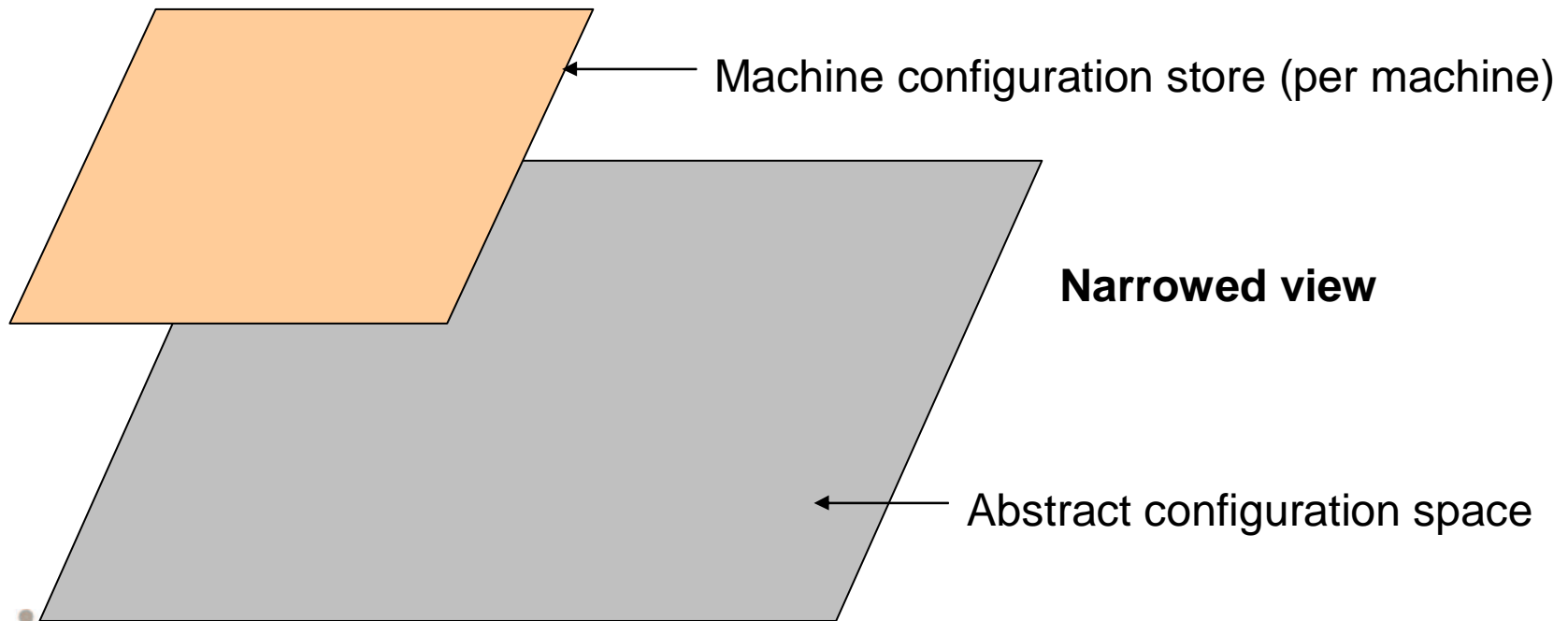
Configuration Provider



Configuration Provider



Configuration Provider



Configuration Provider

- Transparent handling of user specific and per machine settings
 - Easy for plug-in developers to manage configuration
 - Easy to push global settings in a managed environment
- Hierarchical organization with shadow configuration spaces
 - Granular settings (per identity etc...) with useful failover
 - Local settings failover to global settings
- Abstract / extensible
 - By default, each configuration space maps to a registry key
 - “Configuration Provider” plug-ins can provide computed configuration parameters

Documentation

- Inline
 - Using Doxygen

Maintenance Plan

Component	Maintained in
Main application and core-components	MIT Kerberos source tree
Krb5 plug-in	Krb5 source tree
Krb4 plug-in	Krb4 source tree
AFS plug-in	OpenAFS source tree

Summary

- Better for users
- Better for developers
- Better for administrators

Availability

- Platforms
 - Windows 2000/XP/2003
- Distributed with
 - MIT Kerberos for Windows 3.0
 - OpenAFS for Windows 2.0

Questions?

Thank you

A special thank you to

MIT Information Services & Technology

**as well as MIT's Professor Rob Miller and participants of
class 6.831 (User Interface Design & Implementation)**

Contact Information:

Asanka C. Herath
Secure Endpoints Inc.

asanka@secure-endpoints.com

