

Dynamic Management of Internet Telephony Servers: A Case Study based on JavaBeans and JDMK

Alexander Keller



IBM Thomas J. Watson Research Center

alexk@us.ibm.com

Helmut Reiser



Munich Network Management Team

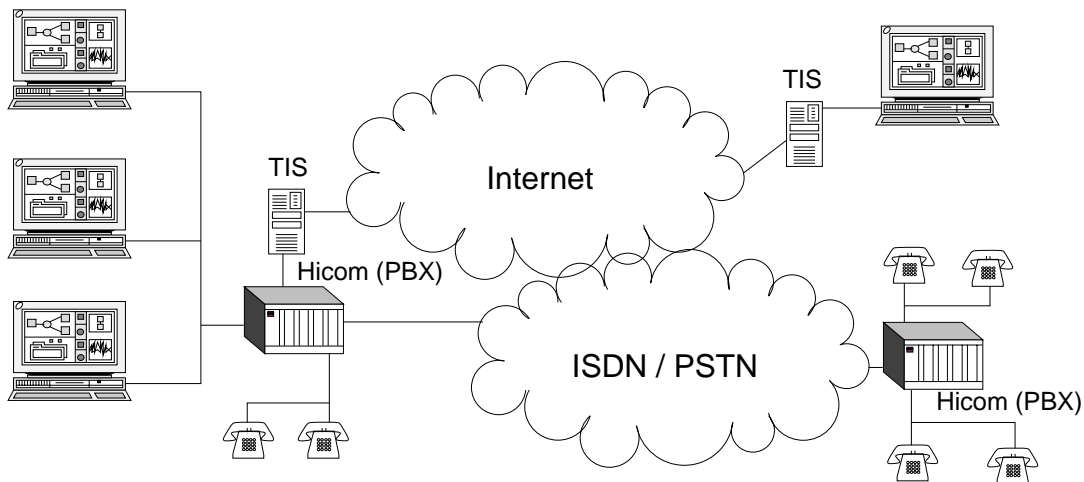
reiser@informatik.uni-muenchen.de

Mannheim, 09/29/1999

EDOC'99



Deployment of Internet Telephony Servers



- ▶ Gateway between PBXes and the Internet
- ▶ Provides ISDN and Videoconferencing Facilities
- ▶ Yields potentially high Cost Savings, deployed in large Quantities



Internet Telephony Servers: Management Requirements

Status Quo:

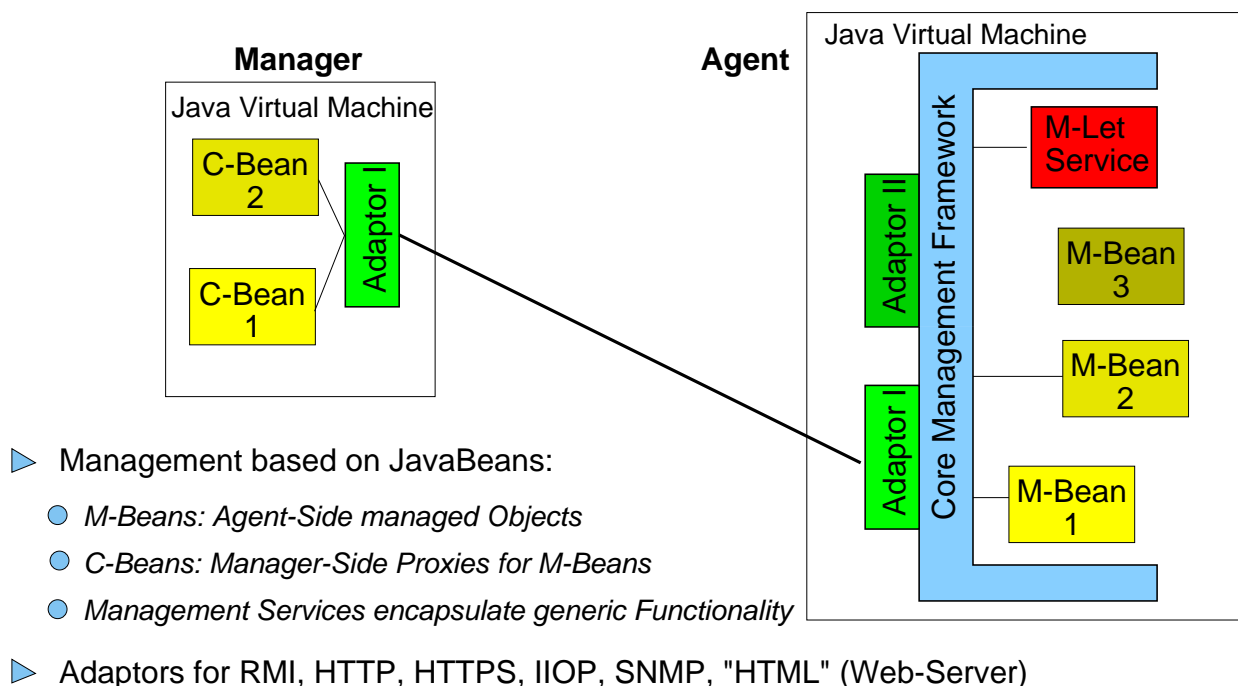
- ▶ Management via SNMP(v2); Enterprise-specific Management Information Base
 - Agent implemented as WindowsNT Service
 - Web-based Administration Interface for individual Servers
- ▶ Adding Extensions to an existing Agent cannot be done at Runtime but requires Restart
- ▶ Isolated Management of individual (dumb) Agents

Requirements:

- ▶ Persistent Storage of Agent State allowing Restart after Failure
- ▶ Giving the Agents a certain Degree of Autonomy
- ▶ Extend or change Functionality during Runtime
- ▶ Access to standardized Naming and Directory Services
- ▶ Cooperation among Agents enabling the cooperative Determination of Problems



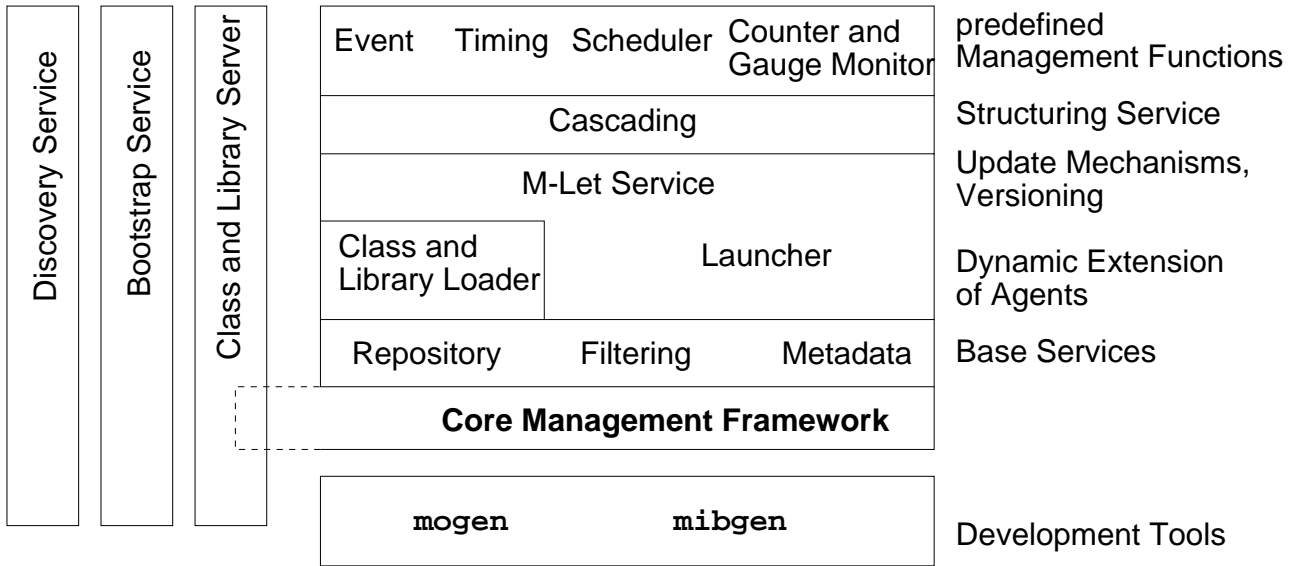
Java Dynamic Management Kit (JDMK)



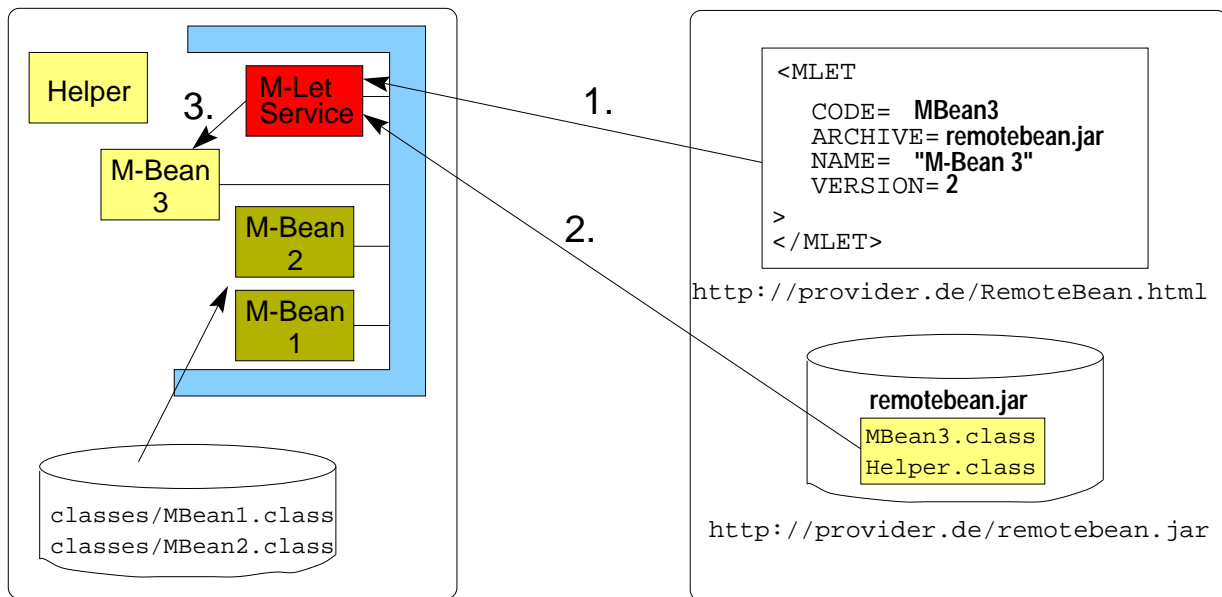
- ▶ Management based on JavaBeans:
 - M-Beans: Agent-Side managed Objects
 - C-Beans: Manager-Side Proxies for M-Beans
 - Management Services encapsulate generic Functionality
- ▶ Adaptors for RMI, HTTP, HTTPS, IIOP, SNMP, "HTML" (Web-Server)



Management Services in JDMK



The JDMK Management Applet (M-Let) Service



► Download and Configure M-Beans at Runtime => Management by Delegation



Managing Internet Telephony Servers with JDMK

1. Transforming existing SNMP MIB Descriptions into JavaBean Interfaces:

- ▶ handled by MIBGEN Compiler (similar to JIDM Approach: SNMP MIB -> CORBA Objects)
 - SNMP Groups, Table Rows become M-Beans
 - Scalar Variables (in Groups or Table Rows) become Attributes of appropriate M-Beans
 - get / set Methods generated for readable / writeable Attributes

2. Accessing the existing C++ Agent Code from the generated JavaBeans:

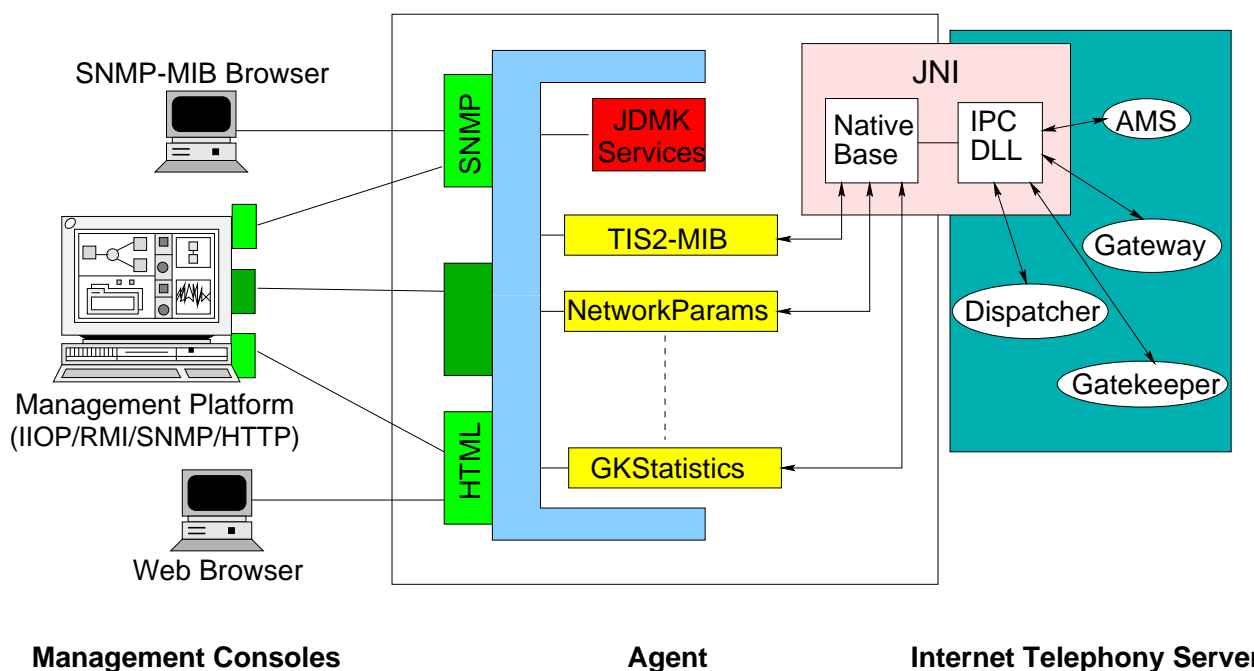
- ▶ Encapsulating the Agent DLL with Java Class "Native Base"
- ▶ Access via Java Native Interface

3. Using Web-Browser as Management Console & Integration w/ Management Platforms:

- ▶ Register HTML (and/or SNMP) Adaptor with Agent Core Management Framework
- ▶ CORBA/IIOP Adaptor insufficient: unaware of Interface Repository & CORBA services



Overview: Components of our Management Solution



Evaluation of JDMK for managing large IT Environments

Strengths:

- ▶ Ease of Use for prototyping flexible, dynamic Management Systems
 - *HTML Adaptor creates simple Web-based User Interfaces for existing M-Beans*
 - *Total Development Time: 4 Person Months*
- ▶ Delegation of Management Tasks from Platforms to Agents
- ▶ Dynamic Code Updates / Extensions (Support of Push- or Pull-Models)
- ▶ Interworking with heterogeneous Environments: Existing or new (own) Adaptors

Weaknesses:

- ▶ Access to Agent Metadata limited to a single CMF (no simultaneous Queries)
- ▶ Insufficient Integration with CORBA: No Access to CORBA Management Services
- ▶ Lack of uniform Security Infrastructure; Access Control Implementation by Developer



Java Management Extensions (JMX)

Since June 1999, JMX is the Successor of JMAPI (Java Management API):

- ▶ Relationship between JDMK and JMX:
 - *JDMK: Product from Sun Microsystems, Inc. (currently at version 3.2)*
 - *JMX: joint Industry Standard with Reference Implementation and Conformance Test Suite*
- ▶ JMX Services are (currently) a Subset of JDMK Services
- ▶ introduces additional kinds of M-Beans: standard, open, dynamic, model
- ▶ defines advanced SNMP Interworking and CIM/WBEM Interoperability

Applicability of JMX to our Solution:

- ▶ Model M-Beans particularly suitable (instantiable by the Agent itself, Persistence)
- ▶ JMX Notification Model provides advanced filtering Capabilities
- ▶ JMX (Infrastructure) and CIM (Information Modeling) complement each other



"Lessons learned" and Outlook

Achievements and Experiences:

- ▶ Migrating an existing SNMP Agent to a JavaBeans/JDMK Environment:
 - *Translation of Management Information done automatically by MIBGEN Compiler*
 - *Making existing Codebase accessible: Java Native Interface*
- ▶ First Set of generic Management Services
- ▶ Ease of use; rapid Prototyping of flexible, dynamic Management Systems
- ▶ Weaknesses of JDMK w.r.t. large Environments (Scalability, Security) also apply to JMX

Further research:

- ▶ Suitability of Directories for storing Meta-Information about M-Beans and Agents?
- ▶ Management of Pervasive Devices, Integration with JINI?
- ▶ Use of recent CORBA services (Property, Query, Managed Sets, Notification, (Topology))

