

Session I: TINA and the Internet

Chair: Larry Gabuzda, *Lucent Technologies*

Integration of Commercial Internet Applications in a TINA Environment

Gianni Canal

Intelligent Network Dept., CSELT, Torino, Italy,

Patricia Lago

Control & Computer Engineering, Politecnico di Torino, Torino, Italy



Outline

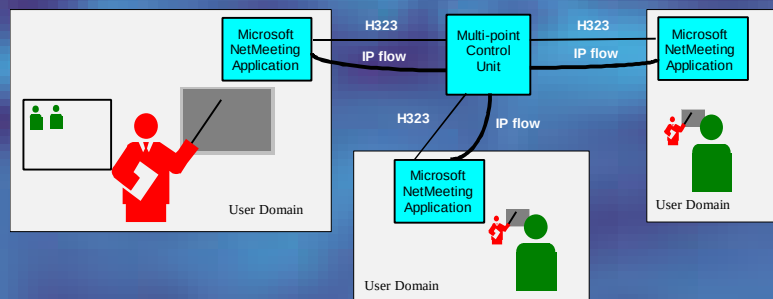
- Context and Objectives of the Work
- Commercial Internet Applications
- Integrated TINA-Internet Service
- TINA Added Values
- Conclusions

Introduction

- Context
 -  Project
 -  Internet
- Objectives
 - Integrability
 - TINA Added Values

Commercial Internet Applications

- Microsoft NetMeeting
- WhitePine MeetingPoint
- Architecture



Pure Internet Service: Setup

- Off-line MCU Configuration

-  1) Conference Creation
- 2) Conference Characteristics
-  3) List of Enabled IP Addresses
- 4) Set Video Access Rights to Conference Owner and Participants

Pure Internet Service: Usage

- Joining the conference

- 1) Choice of the PC Configured with the already Enabled IP Addresses
-  2) Connection to the MCU by Knowing its IP Address or its Full Name
-  3) Choice of the Conference by ID and Name
-  4) Enabling/Disabling Audio or Video
- 5) Switch Received Audio/Video (According to Video Access Rights)

Pure Internet Service: Features

- Joining the conference

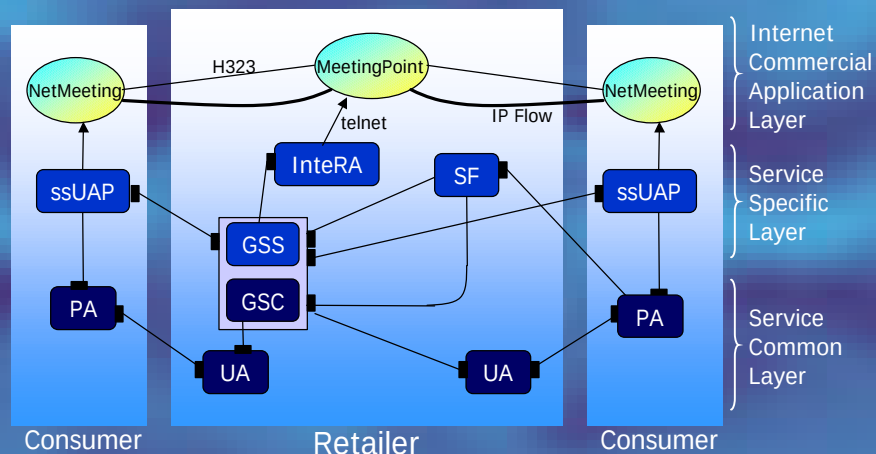
- 1) Audio/Video Conference Support
- 2) Fixed Conference Type
- 3) No Role Based Service Features
- 4) Bearer Service Session Representation
- 5) Bearer Service Session Control



- 6) No Participants Invitation

Integrated TINA-Internet Service

- Architecture



Integrated TINA-Internet Service

- Service Access and Launch

- 1) Authentication Phase

- 2) Access Session

- 3) Service Selection

-  4) Conference Name and Type Selection

Integrated TINA-Internet Service

- Service Features

-  1) Participant Invitation/Removal

- 2) Conference Scope

- 3) Conference Characteristics

- Conference Types

-  - Audio/Video Switching between Participants

-  4) Conference Representation

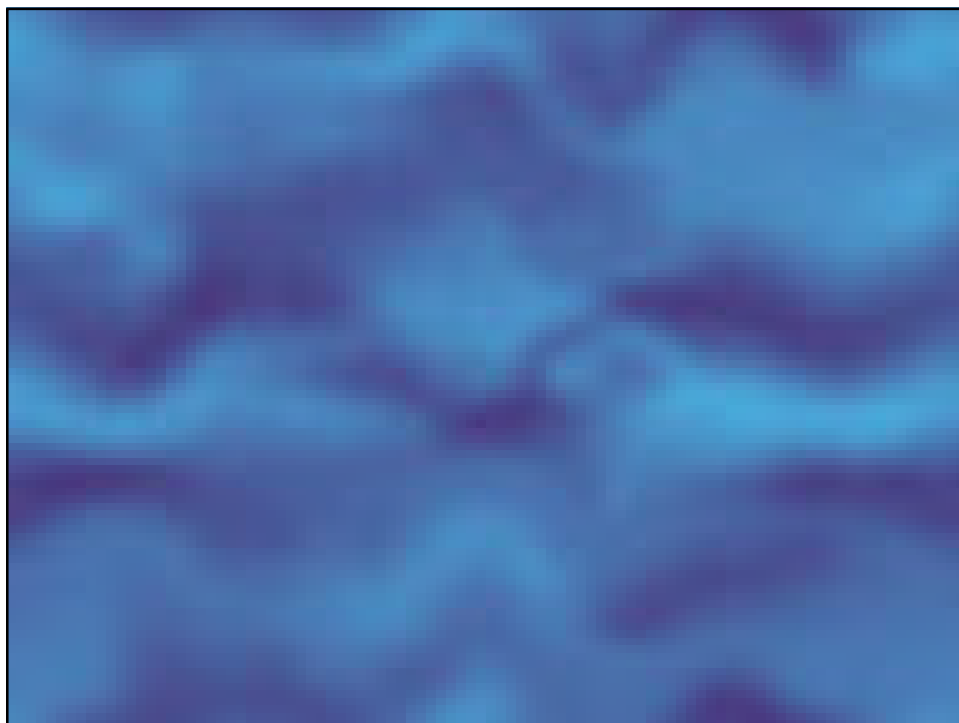
Integrated Solution Added Values

- 1) Transparent Location of Resources
- 2) Dynamic Configuration of Resources
- 3) Multi-resource Support
- 4) Higher Availability
- 5) User Identification by Logical Name
- 6) New Participants Invitation
- 7) Role-Based Service Features
- 8) Pre-Defined Conference Types
- 9) User-Defined Conference Types
- 10) Service Session Representation

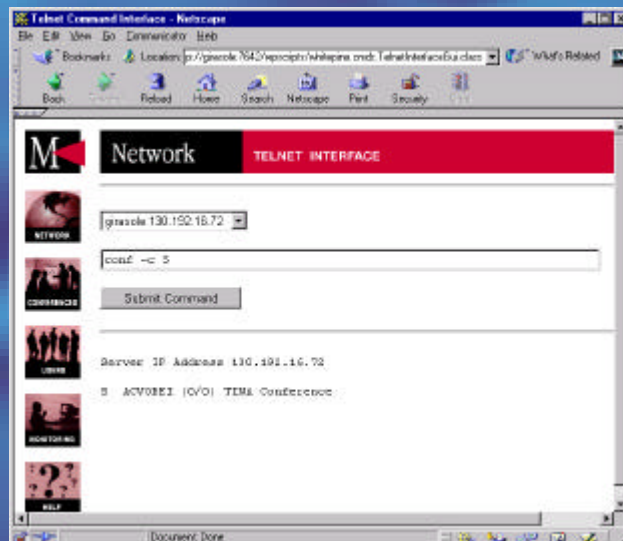
Conclusions

- 1) Successful Integration
- 2) Carrier Grade ORB
- 3) Acknowledgments

Questions?



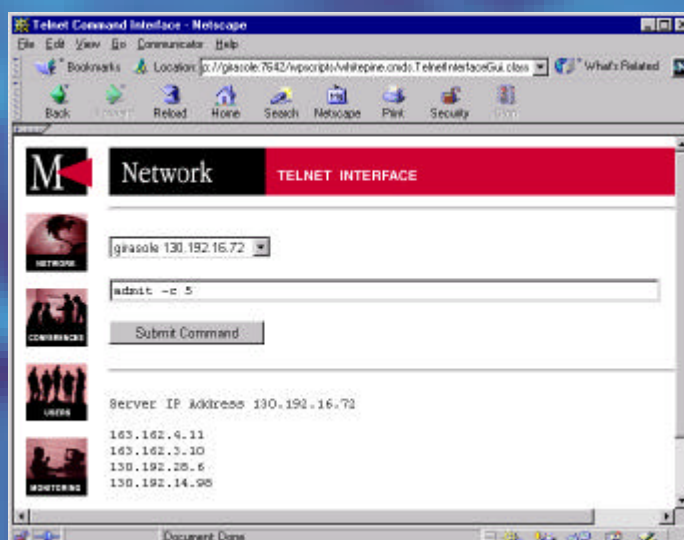
MCU Configuration



TINA Conference '99



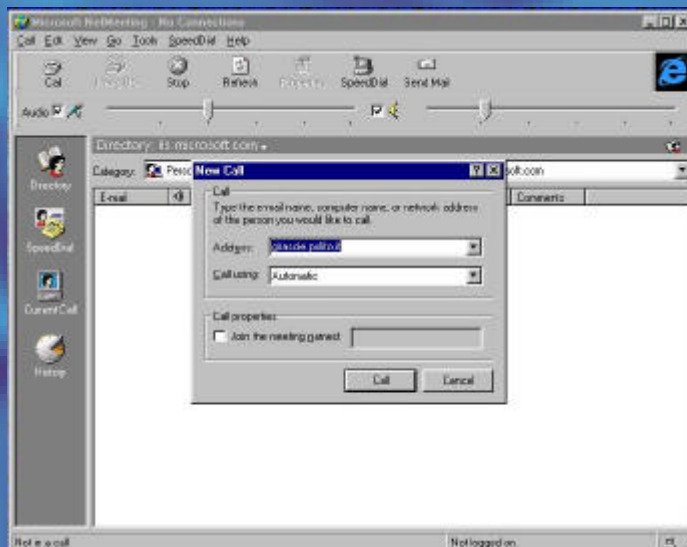
MCU Configuration



TINA Conference '99



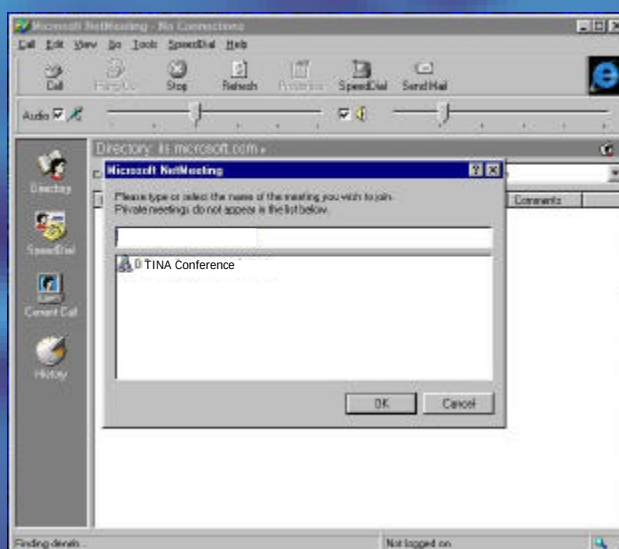
Netmeeting: call dialog box



TINA Conference '99



Netmeeting: conference choice



TINA Conference '99

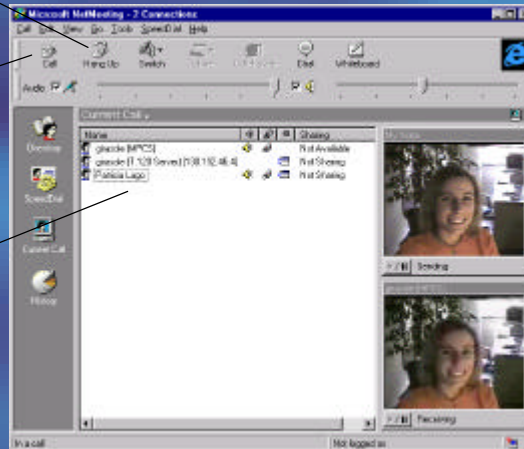


NetMeeting: main GUI

Delete own Participation

Point-to-Point Invitation

Conference participant List

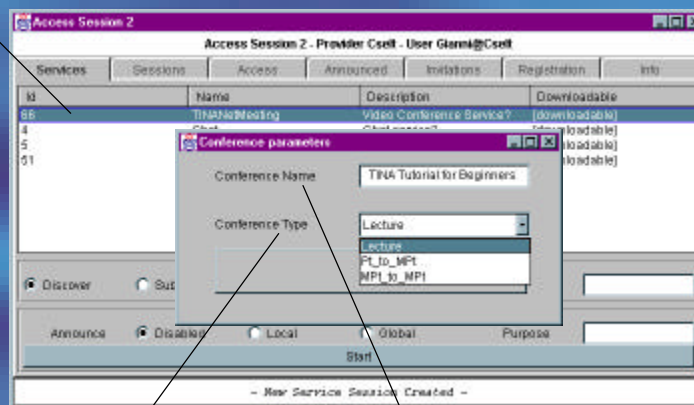


TINA Conference '99



TINANetMeeting Service: start

Service Selection



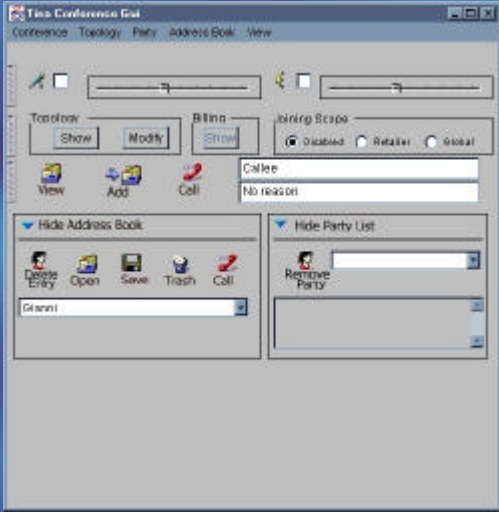
Conference Type

Conference Name

TINA Conference '99



TINANetMeeting: main GUI

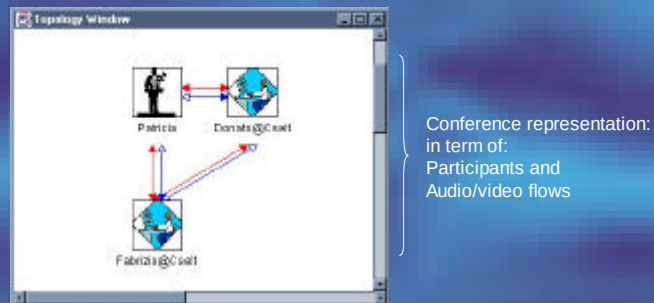


The screenshot shows the 'Tina Conference GUI' window. It features a menu bar (Conference, Topology, Party, Address Book, View) and a toolbar with icons for microphone, camera, and other functions. The main area is divided into several sections:

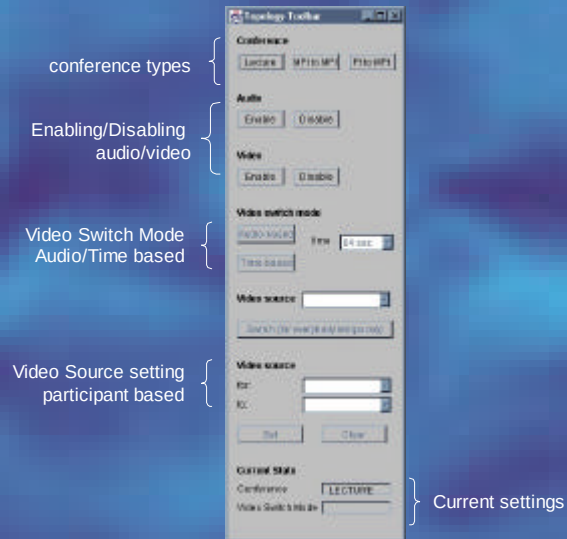
- local A/V H323 setup**: Includes sliders for audio and video settings.
- conference control**: Includes buttons for 'Show', 'Modify', 'Show', and 'Joining Scope' (Disabled, Retain, Global).
- local address book management and usage**: Includes buttons for 'View', 'Add', 'Call', 'Delete Entry', 'Open', 'Save', 'Trash', and 'Call'. It also has a 'Hide Address Book' checkbox and a list of entries (e.g., 'Glenm').
- indication/ information dynamic area**: Includes a 'Hide Party List' checkbox and a list of participants (e.g., 'Remove Party').
- Conference scope**: Includes a 'Joining Scope' dropdown and a 'Call to' field with a 'No reason' button.
- Conference participants' list**: Includes a 'Delete participation (party name selection)' button.
- Video window**: A small window in the bottom right corner showing a video feed of a person.

TINA Conference '99

TINANetMeeting: topology GUI



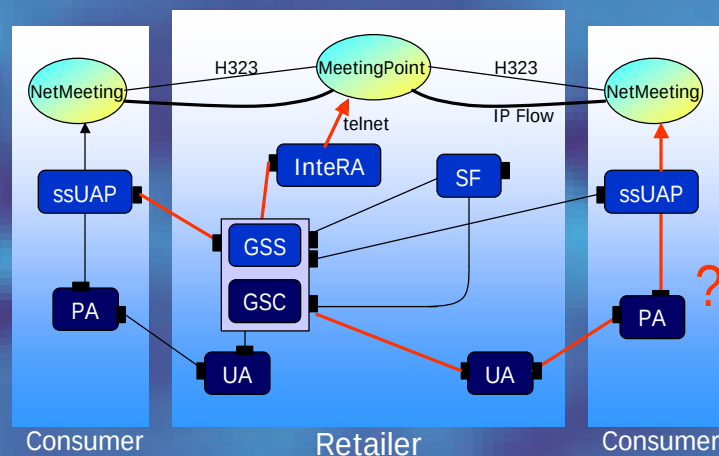
TINANetMeeting: conference toolbar



TINA Conference '99



TINANetMeeting: Invitation Scenario

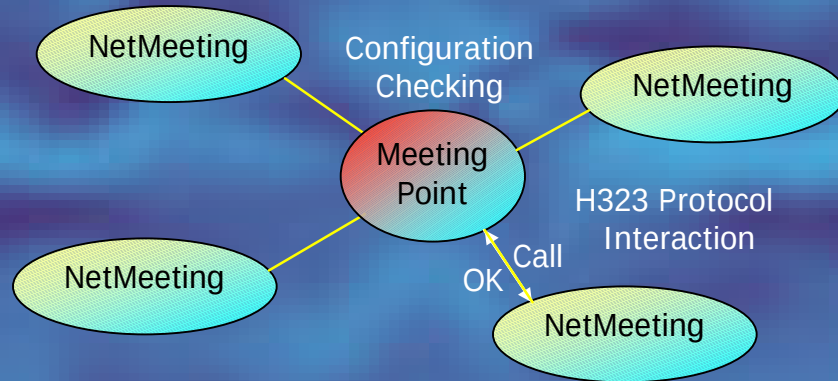


TINA Conference '99



Pure Internet Service

1) Joining Based Participation



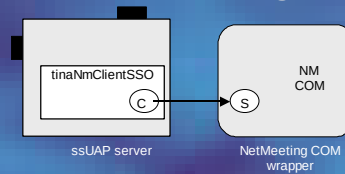
TINA Conference '99



Integrated TINA-Internet Service

- Technological Issues

1) Integration of NetMeeting COM Object



2) Integration of MeetingPoint

MCUTelnetIO

MEETING
POINT

InteRA server

Telnet commands

TINA Conference '99

26 of 14

Value Added Web

1

Aart van Halteren, 13 April 1999



Imagine...

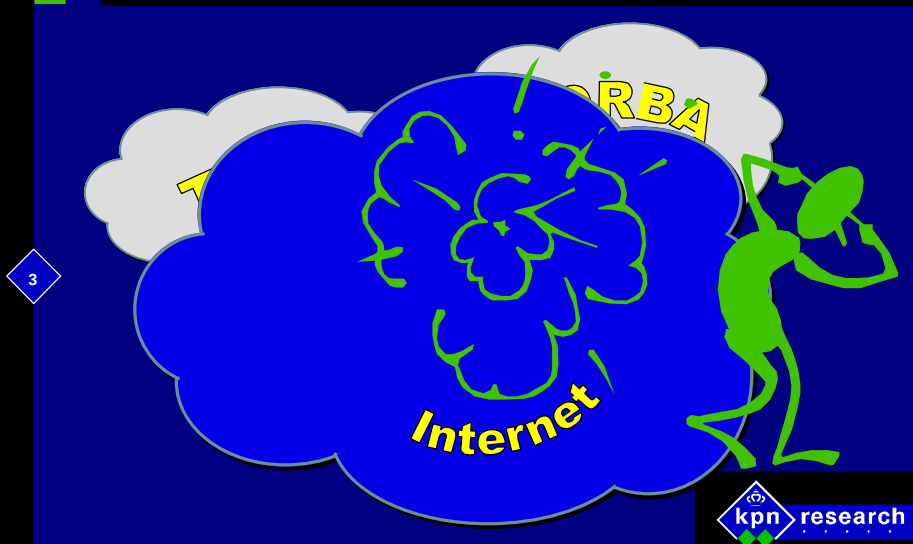
TINA

CORBA

2



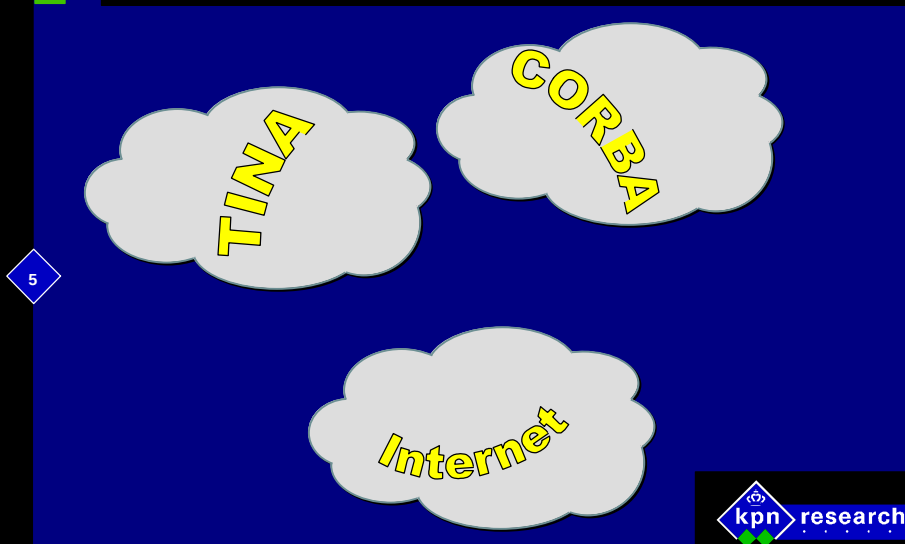
Imagine...



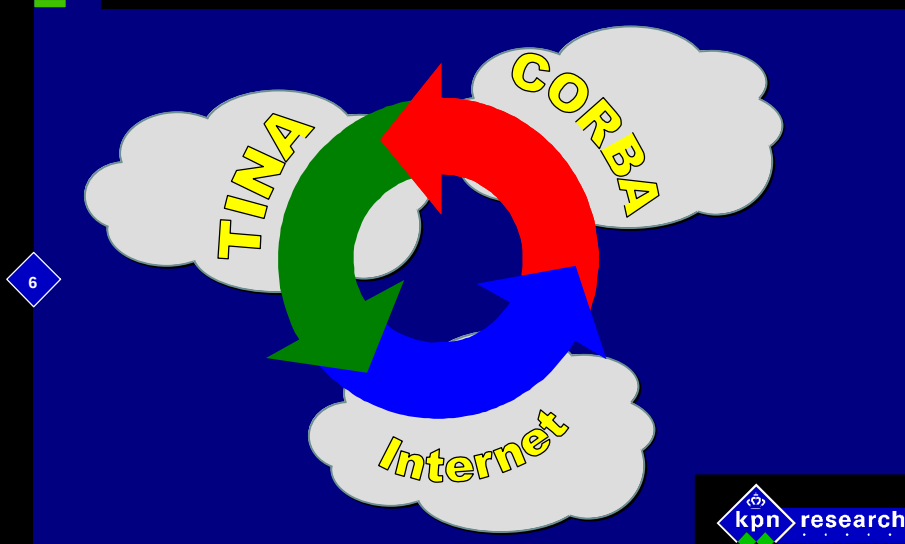
The force field...



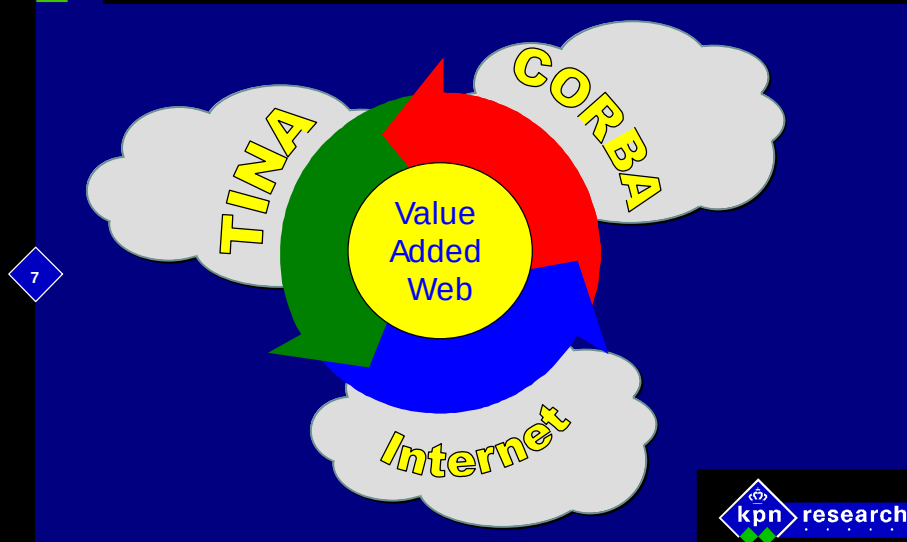
The force field...



The force field...



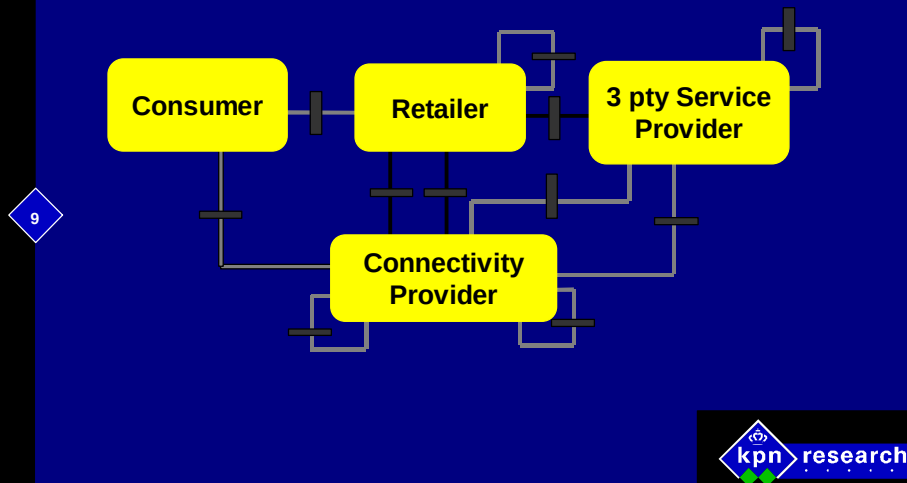
The force field...



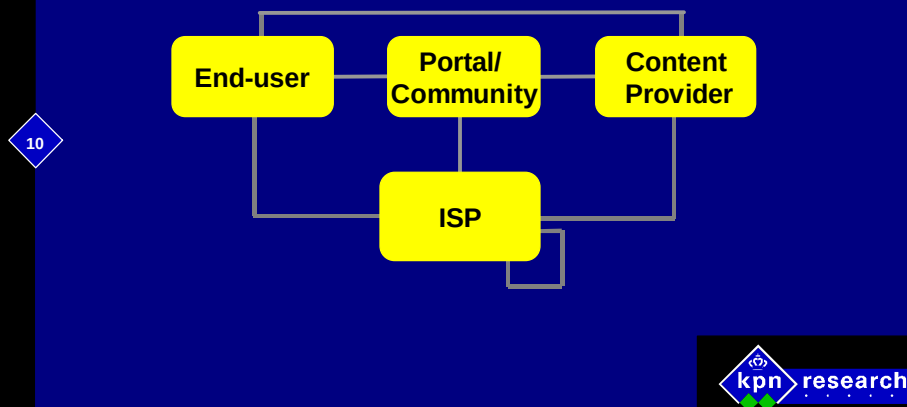
Overview

- 8
- * Business models
 - TINA
 - WWW
 - VAW
 - * Design of Value Added Web
 - available technologies (Web, CORBA)
 - implementation environment (EURESCOM P715)
 - how it works

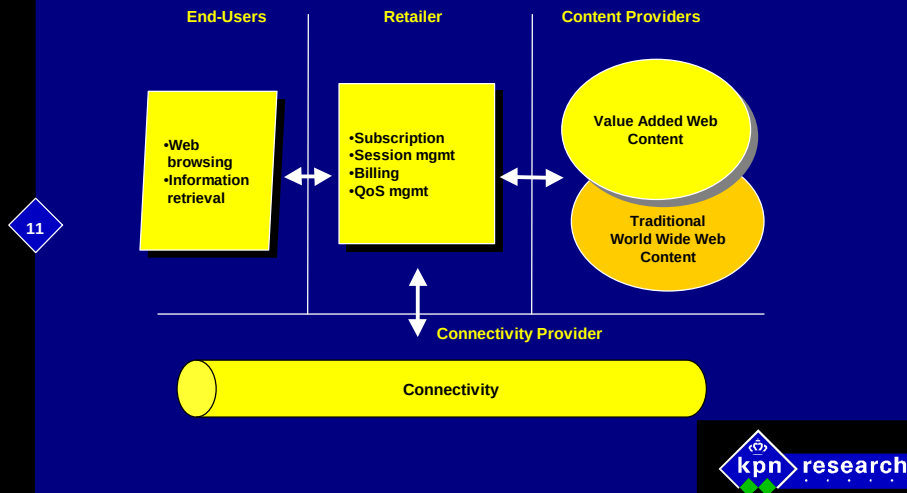
TINA business model



WWW business model



VAW business model

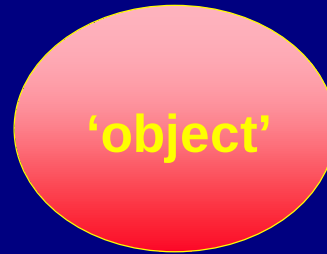


What technologies are available?



Object-Orientation

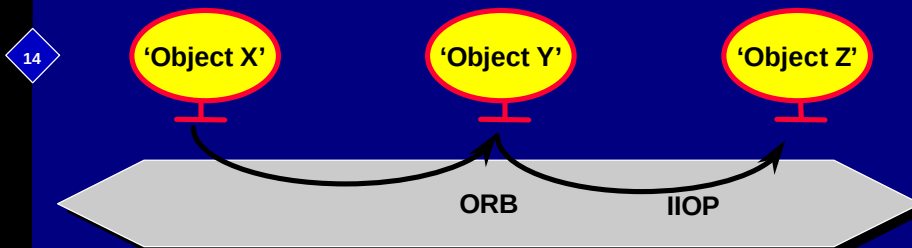
- * objects are modelled entities from the real world
 - an object can be anything
- * objects can do "actions"
 - behavior
- * actions have no influence on other objects
 - encapsulation
- * internal details of an object are hidden from other objects
 - abstraction



13

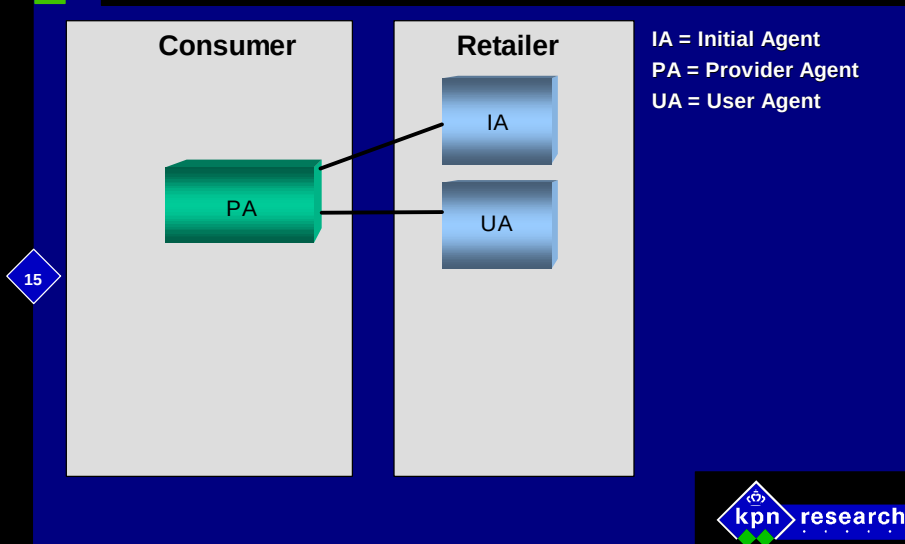
Distributed objects

- * objects distributed over the network
 - location not important
- * objects interact through interfaces
 - invoke an action
 - obtain a result from the object
- * interface hides object implementation

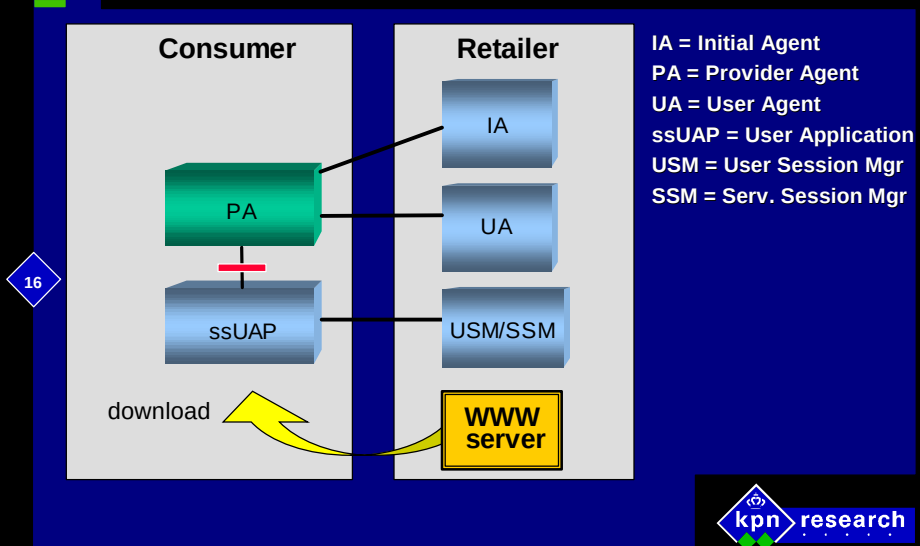


14

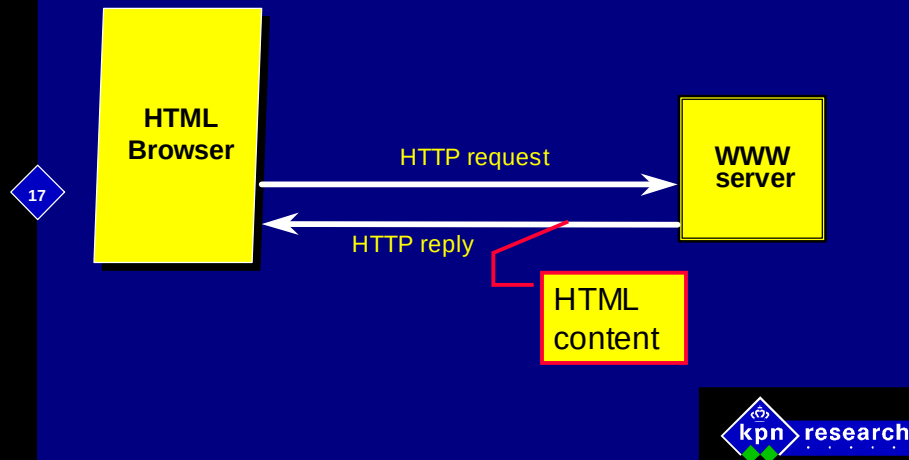
EURESCOM P715: Service Management platform



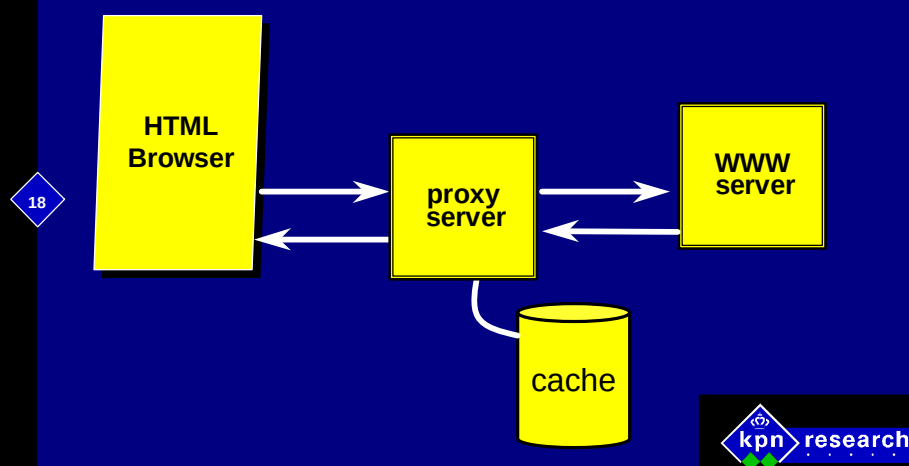
EURESCOM P715: Service Management platform



Web technologies



Web technologies

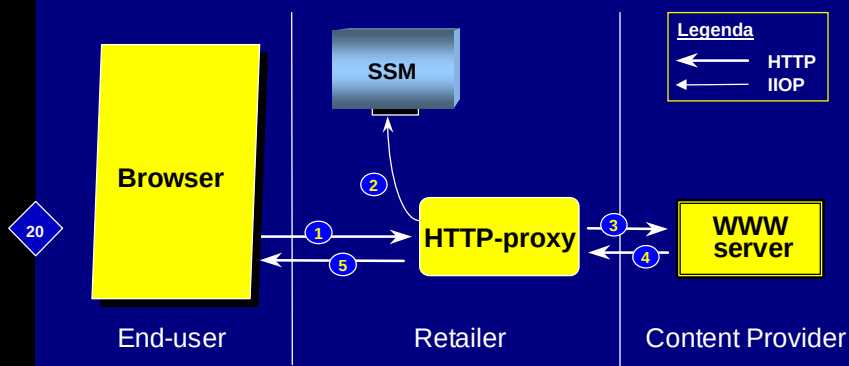


VAW design issues

- ★ Combining Web sessions with TINA sessions
 - intervention in the HTTP request chain?
- ★ Charging for content
 - how to register VAW content?
- ★ Initiating a VAW session
 - how to start a VAW specific ssUAP?

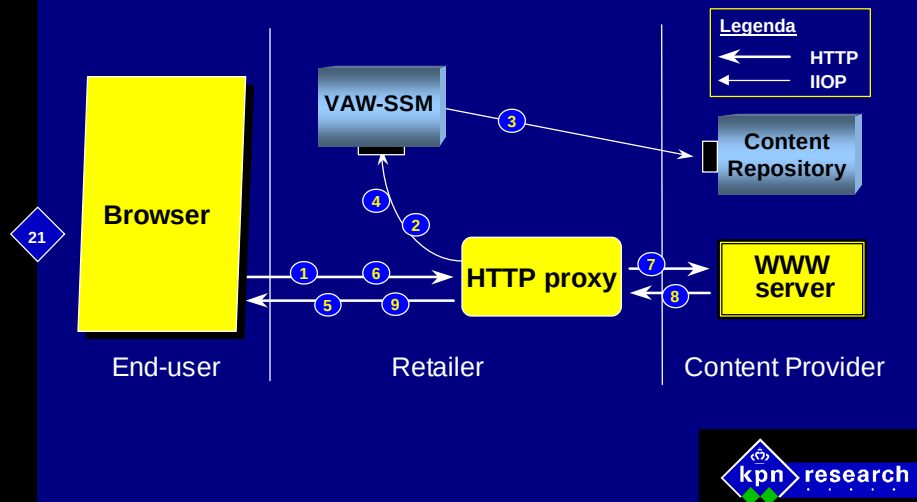
19

Adding an SSM to the HTTP request chain

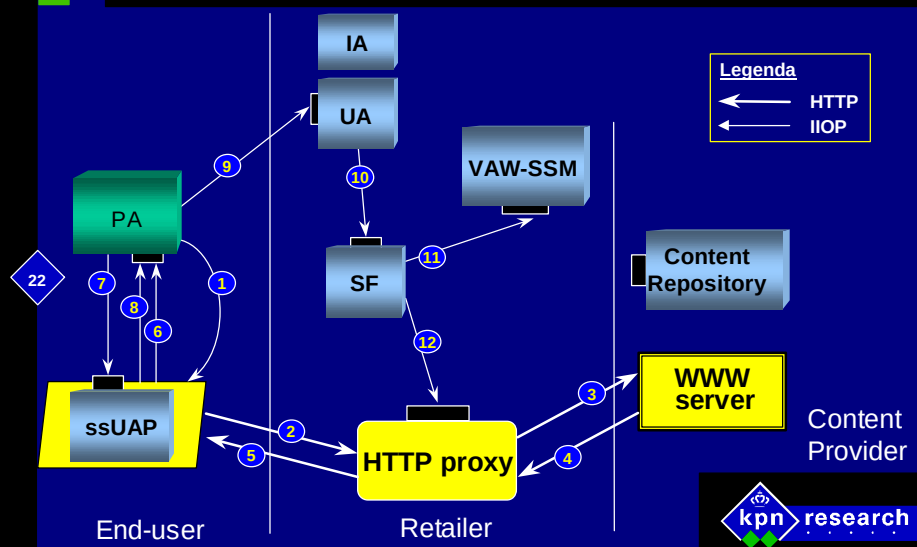


20

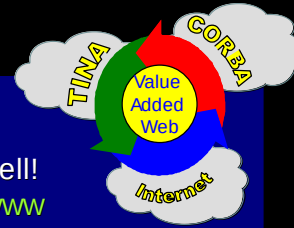
Charging for content



Initiating a VAW session



Conclusions



- * WWW and TINA go together well!
 - VAW business model simplifies WWW
 - WWW technologies enhanced with TINA
- * Benefits for several parties
 - end-users
 - retailers
 - content providers
- * VAW prototype demonstrates feasibility

A "TINA Light" Service Architecture for the Internet-Telecom scenario

Roberto Manione roberto.manione@cse.lt.it
Paola Renditore paola.renditore@cse.lt.it



Via Reiss Romoli, 274
TORINO
ITALY

Overview

A Lightweight Service Architecture has been developed and implemented for the Internet-Telecom Service Scenario.

Lite is inspired to the principles of the TINA SA but introduces a number of simplifications and few improvements to better match the peculiarities of the reference scenario.

Presently Lite it is being exploited within the WebCentric® Web Call Center, presently under field trial within Telecom Italia.

Contents:

- Peculiarities of the Internet-Telecom Service Scenario
- the Lite Service Architecture
- The WebCentric® Web Call Center, implemented within Lite



Peculiarities of the Internet-Telecom Scenario

- 1 Highly standardized User Domain (a PC with a Browser): virtually all the software running on the User Domain to support the service must be loaded from the network
- 2 The Network setting of the reference scenario is relatively simple: telephones can be reached via their numbers, while IP server/clients can be reached via their IP address
- 3 Services have their own authentication (when needed)
- 4 Interaction of Clients and Servers (Service Session) is “connectionless” on the Internet side
- 5 Services are sometimes built on top of other services, possibly provided by other providers, sometimes without a business relation among them: actually providers struggle for being the ultimate Retailer (the “me-Retailer” game)



Implications upon the Service Architecture

- No specific persistent Daemon running within the User Domain; users already have one, very generic, always up and running when services are used: the Browser, with all its persistence facilities (e.g. Bookmarks, Cache, Cookies)
- The business model which regulates the provision of services within the Internet makes very little difference among Retailer and Provider: since most of the time services are free of charge and subsidized by the advertising, every provider wants to be “the” Retailer
- Simplification of the Provider/Retailer Domain: authentication and service state keeping collapsed within the Service Session



Implications at the Engineering VP.

- Need to widen as much as possible the population of potential users of the service (i.e. the population of Browsers).
- Java is the language of choice for the implementation of the Client Side of Internet applications (UAP).
- Choice was made to use a Java native ORB, in order to be able to exploit all the advanced features of Java and to reduce the implementation effort.
- Java-RMI was considered and successfully tried but, at the time the decision had to be taken (~ 1 year ago) not all Browsers supported it; for this reason it was discarded.
- JRB (Java Reflective Broker) was chosen; however many others are available; no particular assumptions are done upon them.



Overview of Lite

[Computational V.P.]

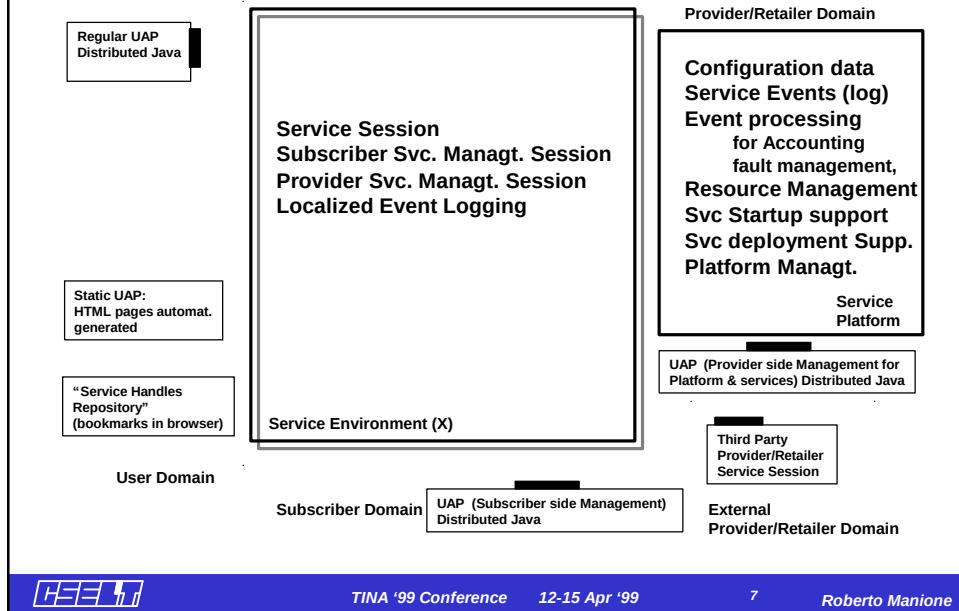
- Use of Object Distribution concepts
- Use of a common Distributed Processing Environment
- Definition of “standard” architectural modules, to be (re)used within different services

[Engineering V.P.]

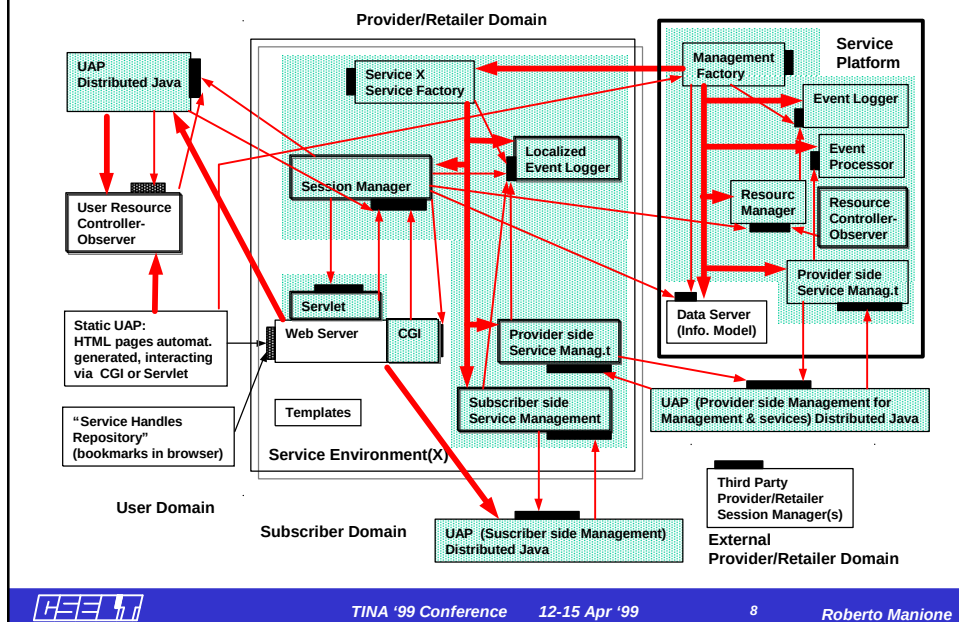
- Introduction of the concept of Service Platform (a cluster of Computational Objects which encapsulate Service Independent functions
- Introduction of three different deployment schemes for services, according to the foreseen traffic needs.
- Java native ORB, connecting all the Service Components



The Lite Service Architecture



The Lite Architecture - expanded



Lite Information Model

Basic Information Model (Platform Data): such Information Model will be augmented with new service-specific classes

- Service Descriptor (URL of factory, Options to factory, rates)
- Identity Descriptor (Personal data)
- Subscriber Descriptor
- Subscription Descriptor (binds Services to Subscribers and adds service personalization data)
- Actor Descriptor (any kind of user of the platform)
- Interest Descriptor (related to either leisure or business)
- Group Descriptor (set of users which share any interest)
- Terminal Descriptor (characteristics of terminals used within a subscription)
- Resource Descriptor (IP address, cardinality, etc.)



Service Environment

Contains the Service Components which implement one service

According to the usage volume of services, 3 deployment models are supported by Lite (with no change in the Service Code):

- (**Small size**) Service Platform + all Service Environments upon one single server
- (**Medium size**) Service Platform upon one server + one server per service
- (**Large size**) Service Platform upon one server + more than one server per service
- Service Sessions: encapsulate the state of every instance of service - can be single-party, two-party, multi-party
- Service Management Sessions: allow for the management of the service (both for Provider side Manager and for Subscriber side Manager)



The WebCentric® Web Call Center

Basic definition of a Web Call Center Service:

“allows subscribers to publish Web sites which enable their visitors to speak to agents while still browsing pages, in particular those sent by agents, as the conversation goes on”

(Human assisted Electronic Commerce)

WebCentric adds the following features:

- the service is hosted on servers located within the Provider premises, freeing the subscribers from buying, running and managing specialized hardware.
- On the final user side, the voice connection can use either Voice over IP techniques, in order to use one single phone line for both voice and web, or a second phone line (possibly mobile).
- Teleworking comes very easy, all what is needed for an agent to operate is one PC and 2 phone lines (one for IP connection, one for Voice)

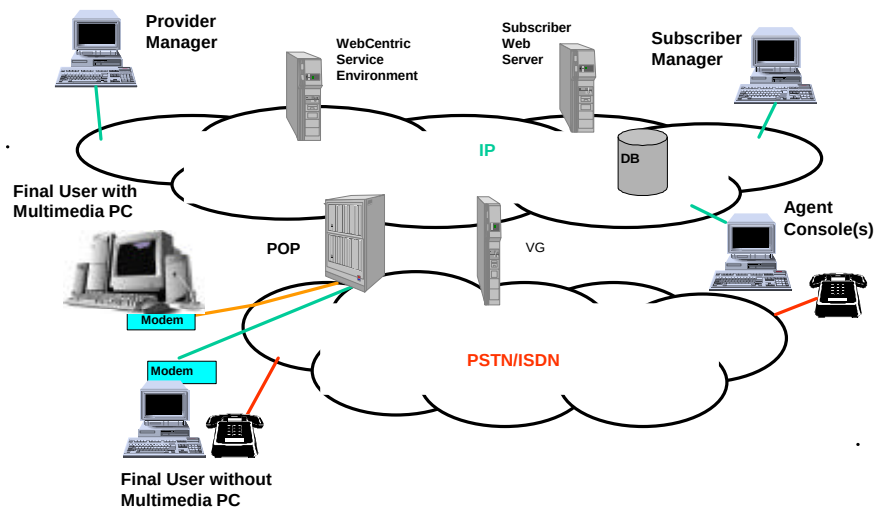


TINA '99 Conference 12-15 Apr '99

11

Roberto Manione

Network setting of WebCentric®



TINA '99 Conference 12-15 Apr '99

12

Roberto Manione

Design of WebCentric® Within LITE

User Domain

- Final users as Anonymous users
 - Service accessed via links in the subscribers Web pages
 - UAP: static, in order to open to the widest population of users
 - VoIP client started via UAP, by means of the Helper Application mechanism (via MIME)
- Agents as Named Users within one subscription
 - UAP: active
 - Agent authentication, profile lookup
 - Agent Gateway (interface to Voice Modem and Browser) started via UAP, by means of the Helper Application mechanism (via MIME)

Subscriber Domain

- SuSM Operator as Named User within one subscription
 - UAP: active
 - Manager authentication



Specialization of the Platform side

Information Model (few additions to the basic Lite model):

- added service specific descriptors
- WCCSubscription Descriptor: derived from Subscription
- WCCCampaign Descriptor: derived from Interest: each campaign is characterized by Groups of Agents and by a Dispatching Scheme; each scheme is made of a set of Or-And Conditions.

Resource Manager:

- Handles Voice Gateways and Remote Access Servers

Event Logger and Processor:

- Added Service specific Event types



WebCentric Service Environment

WebCentric Service Sessions

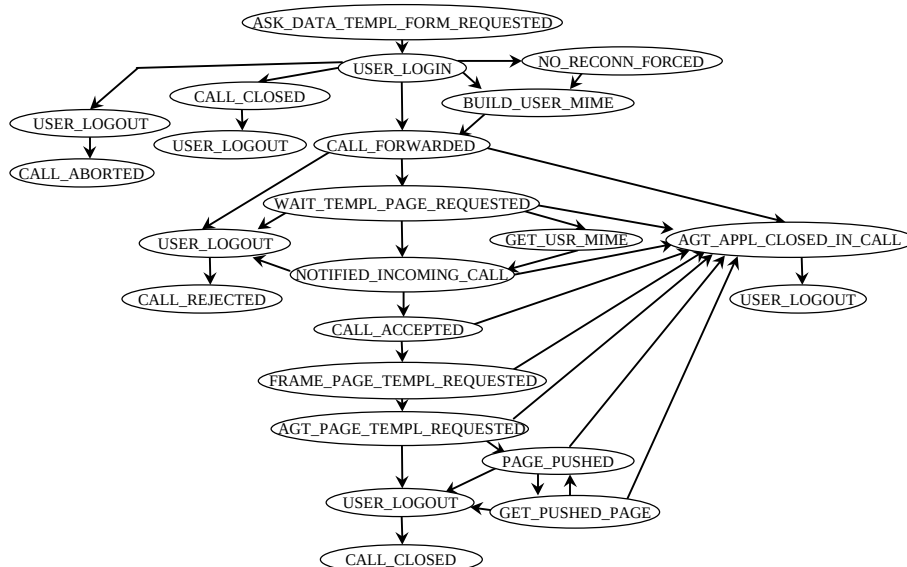
- one Service Session Manager for each subscriber: every SSM handles all the sessions pertaining to that subscriber (maximum number of Sessions found in Subscriber descriptor)
- WCC sessions are two-party, asymmetrical: one final user, one agent
 - hold state of Agents (e.g. free, busy, suspended)
 - dispatch incoming calls (Web-ACD Automatic Contact Dispatcher) to the proper agent, according to interests of the final user and to the routing plan
 - hold the state of sessions

WebCentric Management Session (subscriber side)

- one-party: allows the Subscriber Manager to personalize the configuration of his service (Agent Profiles, Dispatching Plan, messages, etc.) and obtain accounting information



Event Traces - WebCentric user session



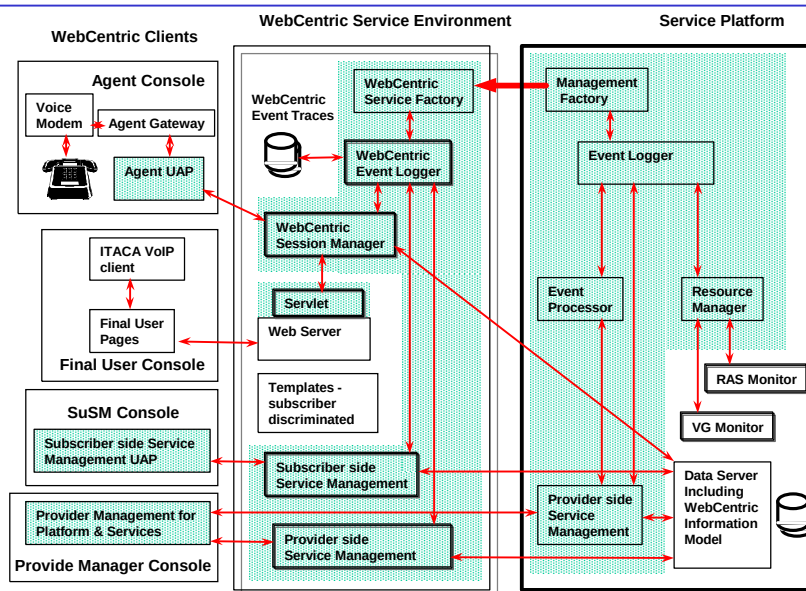
WebCentric Accounting

- Performed with information at service level, rather than using information obtained at network level (e.g. CDR)
- event traces generated by the Event Logger (on subscriber basis) are filtered by the Event Processor, searching for the proper patterns.
- Example patterns:
 - Voice_Call = <any>* Call_Accepted <any>* Call_Closed <any>*
 - Wait_in_queue = <any>* User_login <any>* Call_Forwarded <any>*
 - Page_Pushed = <any>* [Page_Pushed | Get_Pushed_Page]* <any>*
- Based upon the above patterns, one of the possible accounting policies is:

$$A_{sub} = K_v * \sum t(\text{Voice_Call}) + K_q * \sum t(\text{Wait_in_queue}) + K_p * \sum \#(\text{Page_Pushed})$$



WebCentric Structure (upon Lite)



Conclusions

- The Lite Service Architecture was developed to support the provision of services in the Internet-Telecom scenario
- Based upon the same principles of the TINA, although its Computational model is simplified wrt. To the one of TINA
- Such simplifications, in synergy with choices at the Engineering viewpoint have led to a Service Architecture which integrates well with the Web-Internet world.
- Lite was architected and designed thinking at all the aspects of the service and of its management.
- WebCentric®, the first “real” service implemented within Lite, is being trialed in the field since October 98 and is about to be introduced into the full commercial phase

www.webcentric.interbusiness.it/bank



Business Roles within Lite

- Provider/Retailer (unified)
- Subscriber: the holder of the relation with the Provider/Retailer
- User
 - Named: registered within the context of a subscription to a service;
 - service user
 - service manager (Subscriber side)
 - Anonymous: can only use freely available services
- Third party Provider/Retailer

Examples:

- Personal subscription: 1 person is subscriber, service user and subscriber service manager at the same time
- Corporate subscription: company subscribes to on service, authorizing employees as service users; one employee can operates as subscriber service manager



The Lite Architecture

- Platform Management (one)
 - Configuration data repository (Information Model)
 - Event Logger - Event Processor (Event Model)
 - Resource Manager
 - Platform Manager
- Service Environment (one per service)
 - Service session(s)
 - Subscriber side Service Management session(s)
 - Provider side Service Management session(s)
- Clients (per service)
 - User UAP (both active and static)
 - Subscriber side Service Manager UAP
 - Provider side Service Manager UAP
- Third Party Provider/Retailers (0 to ∞)



Factory - Resources - Events

Management Factory: supports the deployment of services:

- launches the Factories of all the deployed services environments

Resource Manager: handles the set of resources of the platform

- Resources are modeled as sets of channels
- Each channel can be in one of four states: Free, Committed, InUse, OutOfOrder

Event Logger: allows any Service component to open a log trace for Event logging

- Events are typed; each event type has a defined semantics, which allows further postprocessing (by the Event Processor)

Event Processor: extracts Event Patterns from Event traces

- Event patterns are specified using Regular Expressions; patterns represent particular kinds of sessions or capture particular features from selected kinds of sessions

