Open Source Telephony

Integrating Asterisk & ooRexx
To Transition An IVR Platform
Into the 21st Century
Original Solution

- OS/2 Based
  - No Support
- Classic Rexx
  - Prevented Progress
- Closed Source
  - No Enhancement
Potential Solution 1

- Solution From Current Vendor
  - Pros
    - No Change In IVR Applications
    - ooRexx Support
  - Cons
    - Closed Source
    - Lack Of Support
Potential Solution 2

Solution From Current Vendor

• Pros
  • TCP/IP Socket Based
  • Provided Some Redundancy Options

• Cons
  • Closed Source
  • Required Rewrite or Abstraction Interface
  • Lack of Documentation & Support
Potential Solution 3

- Asterisk
  - Pros
    - Open Source
    - Linux Based
    - Scalable
    - Very Well Supported
  - Cons
    - Required Abstraction Interface
Selected Solution

- Asterisk
  - Implementation Goals
    - Minimize IVR Developmental Changes
    - Seamless Conversion
    - Redundancy
    - Scalability
Solution Goals

- **Call Handling**
  - Route To Least Utilized Asterisk Server
  - Bridged To Least Utilized IVR Server

- **IVR Execution**
  - IVR Executes Unchanged
  - Any IVR <-> Any Server
Solution Goals

- **Image Based Server Deployment**
  - Minimize Configuration Needed To Bring New or Replacement Servers On-line
    - (No Configuration Needed For Asterisk Server)
  - Ease Provisioning of New Servers
    - Enhances Scalability
Solution Goals

- **Web Based Management/Monitoring**
  - Comprehensive Monitor & Management
    - Enable/Disable Any Individual Server
      - Allows Removing Server From Service
    - High-level & Detailed Call Information
    - Ongoing Development
Solution Development Notes

- Everything Other Then Asterisk
  Written In ooRexx!!
  - From the server components to the web based interface
- Inter-Server Communication
  - TCP/IP Sockets
ooRexx interfaces to Asterisk

- AGI
  - Conceptually Similar To CGI
    - Redirects stdin, stdout, stderr
- ExternalIVR
  - Not As Flexible, But Buffers TouchTones
Solution Components

- **Asterisk Main (Linux)**
  - Converts Calls to VoIP
  - Routes to Least Used Asterisk Node
    - ooRexx AGI Script
    - Gets Least Used Node From MServer
Solution Components

- **MServer (Linux/Windows)**
  - Monitors Load on Asterisk Cluster Nodes
    - Connects to Asterisk Management Interface
  - Activates/Deactivates Asterisk Nodes
    - Driven From Website
Solution Components

- **Asterisk Cluster Node (Linux)**
  - Starts UCS Client
    - ooRexx ExternalIVR Script
  - UCS Client
    - Gets Least Used UCS Node From UServer
    - Connects to UCS Node
    - Fails over to Next Available UCS Node
Solution Components

- **UServer (Linux/Windows)**
  - UCS Servers Connect and Update Load
    - Doesn’t Route to Disconnected Nodes
  - Activates/Deactivates UCS Nodes
    - Driven From Website
Solution Components

- UCS Server (Windows)
  - Executes IVR Scripts
  - Modular
    - Support New Development
    - IIR Clients
    - AJAX Clients
    - Etc
Process Communication
Design

- Modified IVRs
  - Method Objects Instead Of Files
- Inter-thread Communication
  - Global Variables (.local)
  - Methods
  - Queues
    - Allows Blocking
- Inter-Process Communication
  - TCP/IP Sockets
call on user setvar

myroutine: procedure

....

raise user setvar additional (vars) return

setvar:

  do i = 1 to condition('a')~words() by 2
    call value condition('a')~word(i), condition('a')~word(i + 1)
  end

return
Tips & Tricks

/* required ooRexx file */

.local~myobject = .myobject

::class myobject

...
References

- Asterisk (Telephony Software)
  - http://www.asterisk.org

- Sangoma (Telephony Hardware)
  - http://www.sangoma.com