Prepared for:

Federal Communications Commission

CMS Alliance to Modernize Healthcare
Federally Funded Research and Development Center

Building a Center of Expertise for the FCC's Telecommunications Relay Service (TRS) Health Improvement Activities

Contract No. HHSM-5000-2012-000081

Task Order No. FCC15D0002

VRS Auto-Routing Proof-of-Concept Prototype Cookbook

Final Draft

For Pubic Release – In Accordance with Data Rights

Version 0.5 [Cookbook Section Only]

September 29, 2015

MITRE
7515 Colshire Drive
McLean, Virginia 22102

Record of Changes

No.	Date	Reference	A=Add M=Modify D=Delete	Description of Change
0.1	May 29, 2015		A	Initial Annotated Table of Contents
0.2	July 21, 2015	All	M	Cookbook Section Only
0.3	August 19, 2015		M	Revised configuration files in subsections 2.4.1–9 Updated document based on comments from version 0.2
0.4	August 27, 2015	All	A/M/D	Incorporate comments to version 0.3
0.5	September 29, 2015 November 5, 2015	All	A/M/D	Final draft, with replacement of Executive Summary by Foreword; revisions throughout. Approved for Public Release

Foreword

The FCC Telecommunications Relay Service (TRS) COE Project promotes the Commission's goal to foster innovations that advance functionally equivalent telecommunications. Toward that end, the project ensures that the Telecommunications Relay Service employs improved technology for persons who are deaf, hard of hearing, deaf-blind, and/or have speech disabilities.

The FCC has embraced a research-based approach to achieve this goal by engaging The MITRE Corporation to conduct independent engineering assessments that promote and demonstrate TRS's functional equivalence. MITRE is independently assessing voice telephone services, video access services, and Internet Protocol (IP)-based captioning technology; improvements to TRS efficiency; solutions for direct communication between people with communication disabilities and other telephone users; and the effectiveness, efficiency, and consumer response to current and future approaches for delivering TRS.

Recently, the FCC directed MITRE to prototype an innovation for auto call routing (ACR): enabling direct calling from a Video Relay Service (VRS) customer to an American Sign Language (ASL)-trained agent for call handling. MITRE is testing and validating that the ACR prototype can scale from a single ASL-trained agent to call centers serving larger agencies. MITRE has successfully demonstrated the feasibility of auto-routing a VRS user's call into a federal call center to reach a customer service representative with VRS capabilities. The demonstration considered a call center scale equivalent to the Social Security Administration (SSA), FCC, Equal Employment Opportunity Commission (EEOC), Internal Revenue Service (IRS), and Centers for Medicare & Medicaid Services (CMS).

Specifically, the FCC tasked MITRE to:

1. Develop a proof-of-concept, or prototype, to provision Auto Call Routing (ACR) at the FCC (single agent) and then a larger agency effort (for example SSA, supporting 10+ agents), and identify relevant technologies, protocols, and/or procedures that enable a VRS user to:
    a. Call directly into the agency using a standard customer 1-800 service number
    b. Be routed automatically to an ASL-trained agent for call handling
    c. Manage a queue of multiple incoming calls routed to multiple ASL-trained agents
    d. Support the successful handling of that call (formal Demonstration)
2. Identify specific emerging technologies, trends, and approaches that:
    a. Support recognizing the source of the incoming call (from a VRS)
    b. Include the capability to automatically route that incoming call to a VRS-trained handler
    c. Identify direct communication alternatives for incorporation into technology challenges and other acquisition strategies

3. Conduct demonstrations for the FCC and other federal agencies, such as the EEOC and SSA, and prepare a demonstration of the Auto Call Routing operational prototype for several federal agencies at the White House Conference Center in the Eisenhower Executive Office Building.

For the ACR, MITRE developed and tested a core function of call center Private Branch Exchange (PBX) functions, including the auto call routing based on agent queues, incoming dialed number, and call management. MITRE also built an Integrated ASL Video Response (IAVR) capability. MITRE researched, edited configuration files, scripted the dial plan for incoming call flow, and tested currently available open source video telephony and call management technologies needed to support direct call center response and queue management for up to ten (10) simultaneous direct video calls. These calls originate from deaf and hard-of-hearing individuals using a standard videophone with real-time video connection to an ASL-trained agent within an organization's call center.

MITRE is also testing multiple vendor and open source products across current platforms, including computer-based video, desktop integration, and handheld mobile video over secure wireless, hotspot, and traditional secure network connections. Testing of core ACR functions covers provider endpoint video phones, open source softphones, commercially licensed softphones, and desktop telephony applications.

In addition, MITRE drafted this implementation guide to provide a specific technical "cookbook" section for guiding and assisting developer teams with the setup, configuration, and test of the ACR proof of concept. The primary audience for this document is the standard call center technical team. Specific configuration sections are included for the senior systems engineer and identifies core network, programming, and computing infrastructure skills, experience, and competencies that should also form part of the audience and team.

The ACR cookbook's detailed instructions address the configuration of an open source Session Initiation Protocol (SIP) PBX using the MITRE-developed direct video call (DVC) ACR Proof of Concept. The cookbook facilitates replicating the auto-routing configuration, which enables a direct video call from a vendor endpoint device to an ASL-proficient customer service representative's desktop. The ACR cookbook describes how to integrate the ACR prototype with current call center workflow to provide an independent, on-demand service.

The ACR cookbook provides the context and technical descriptions of the components (open source, off the shelf, licensed, and/or uniquely built/configured) to support the on-demand service. Its examples of the setup/configuration steps, scripts, programs, and command sequences will support a specific set of use cases and tests. The cookbook also supplies documentation for implementing the recommended approach successfully within the agency call center.

MITRE and the FCC jointly decided to make the ACR cookbook a publicly released document. It may be downloaded from the TRS-COE.org portal for use by commercial entities and in open source forums.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Background

At the request of the Federal Communications Commission (FCC), The MITRE Corporation (MITRE) drafted a Direct Video Connectivity (DVC) Auto-Routing Proof of Concept in support of the FCC's Telecommunications Relay Service (TRS) Health Improvement Activities. This auto-routing proof of concept enables direct calling from deaf and hard of hearing individuals to an American Sign Language (ASL)-trained agent within the organization's call center for call handling using a video capable phone with real-time video connection.

The MITRE Project Team explored, researched, and tested a variety of current video telephony technologies as a functionally equivalent means of two-way communication with real-time video. MITRE is testing the multiple vendor and open source products across current platforms, including computer-based video, desktop, and handheld mobile video over wireless, Wi-Fi, and traditional network connections.

1.2 Purpose

The purpose of this document is to provide a detailed guide on the configuration of an open source Session Initiation Protocol (SIP) Private Branch Exchange (PBX) to develop a Direct Video Connectivity (DVC) Auto-Routing Proof of Concept, and support a direct call from deaf and hard of hearing individuals using a video capable phone with real-time video connection directly to an ASL-trained agent within an agency, company, or organization's call center for call handling.

MITRE has prepared this proof of concept cookbook to describe the recommended approach for explaining a specific configuration and implementation of auto-routing a direct video call from a user endpoint to a customer service representative (CSR). It contains examples of the setup/configuration steps, scripts, programs, and command sequences used to run a specific set of use cases and tests.

1.3 Goals

The primary goals for creating this proof of concept and the associated cookbook are:

- Identify specific emerging technologies, trends, and approaches that support the expected types of incoming calls; be able to automatically route the incoming call as a direct communication to an ASL-trained point of contact within an agency, organization, or company
- Acquire and configure relevant enabling technologies in the MITRE test laboratory, and develop a prototype using the Open Source Asterisk PBX software as a baseline
- Demonstrate an alternative direct contact provision for deaf and hard of hearing video users to a designated ASL agent at an agency, organization, or company; this technology

will support a new service and provide more efficient and additional access to existing methods.

- Research and identify relevant technologies, protocols, and/or procedures that enable a deaf or hard of hearing user to call directly into the agency using a standard customer service number and automatically route the call to an ASL-trained agent at an agency, organization, or company
- Analyze the results of the innovations through iterative usability testing to identify alternatives and make assessments based on the testing results
- Provide technology, security, and process recommendations
- Document results and proof-of-concept lessons learned from initial requirements and use case analysis, and deliver the documented results to the FCC for review and comment in the form of a "cookbook"

1.4 Development Approach

To ensure efficient use of resources, it is essential to understand what activities are necessary and how they affect the product and each other. The proof of concept is based on open source as a development model and promotes universal access via a free license to a product's design or blueprint, and universal redistribution of that design or blueprint, including subsequent improvements to it.[1]

Open source refers to a computer program in which the source code is available to the general public for use and/or modification from its original design. Open-source code is meant to be a collaborative effort, where programmers improve upon the source code and share the changes within the community. The open-source model is based on a decentralized model of production. A main principle of open source software development is peer production, with products such as source code, "blueprints", and documentation available to the public at no cost.

As part of the development process for the Cookbook, the MITRE team researched and reviewed extensive materials, including:

- Open Source Asterisk PBX documentation
- Existing Open Source SIP and PBX operations and maintenance (O&M) policies
- Procedures documentation of production systems
- Existing operational documents for direct video calls and call center operations

MITRE also interviewed several Open Source SIP and PBX subject matter experts (SME) to gain insight into the policies and procedures for operating and maintaining production systems.

MITRE based the proof of concept auto-routing prototype and demonstration on unified communications and the Internet Protocol (IP) PBX architecture[2]. MITRE extended the baseline

1 http://opensource.org/
2 http://www.asterisk.org/get-started/applications/pbx

using modular functions such as the automated attendant, video and voice messaging, and call queuing. The communications platforms continue to expand beyond email, instant messaging, video conferencing, desktop sharing, Short Message Service (SMS), and mobile telephony.

MITRE implemented an Integrated Product Team (IPT) and Integrated Process and Process Development (IPPD) approach to the proof of concept. The approach includes the following activities (further defined and explained in Appendix H. , *Essentials of the IPT and IPPD*):

- Understanding the requirements
- Outlining the approach
- Planning the effort
- Allocating resources
- Executing and tracking the plan

1.5 Supplemental Documentation

MITRE relied on the following supplemental documentation to support the recommended approach detailed in this cookbook:

- A Business Requirements Document (BRD) helps to shape the business requirements. Developed in concert with the cookbook, the BRD includes diagrams of workflows, a responsibility matrix, and detailed mapping of the As-Is work stream to the best practice. (Please see subsection 2.2, Requirements.)
- Asterisk™: The Definitive Guide, Fourth Edition, by Russell Bryant, Leif Madsen, and Jim Van Meggelen, Copyright 2013
- The Asterisk Cookbook, by Russell Bryant, Leif Madsen, Copyright 2013
- The Asterisk Guru, http://www.asteriskguru.com/
- Asterisk, The Definitive Guide, http://asteriskdocs.org/en/3rd_Edition/asterisk-book-html-chunk/index.html
- **Asterisk Quick Start Guide:** http://www.asterisk.org/sites/asterisk/files/mce_files/documents/asterisk_quick_start_guide.pdf
- Asterisk Step-by-step Installation, http://www.voip-info.org/wiki/view/Asterisk+Step-by-step+Installation
- Asterisk, http://www.asterisk.org/
- iTRS ENUM Database, https://www.neustar.biz/services/network-routing-and-addressing/itrs-enum-database
- Asterisk Project Documentation, https://wiki.asterisk.org/wiki/display/AST/Include+Statements
- Current Asterisk News, http://www.venturevoip.com/news.php
- Appendix A - Internet Engineering Task Force (IETF) Request for Comments (RFC)

1.6 Dependencies

MITRE recommends addressing the following dependencies to implement the proof of concept properly:

- **Executive Support** – Sponsor leadership must be solidly behind an implementation, or needed changes will not happen at the business unit, program, or system levels.
- **Cultural Change** – The Sponsor must understand and agree that a change in strategy or culture is needed. Changing an organization requires shifting resources from some areas into others. Stakeholders must be willing to make the change.[3]
- **Business Process Reengineering** – Effective redesign of business processes requires optimizing business processes to achieve dramatic improvements in productivity, cycle times, and quality.[4]
- **Technology Enablement** – Robust connectivity and the use of agile computing services are two primary technical elements essential to success of this proof of concept.
- **Continuous Improvement** – A project team should continually seek opportunities to improve services and meet the needs of its customers.
- **Resource Realignment** – Information technology (IT) requirements continue to grow. Accommodating these requirements typically means moving financial and personnel resources from duplicated programs, systems, and workflows to new shared approaches within and between agencies.

  Adoption Strategy – The IPT should clearly state who is accountable, what the performance targets are, and how each transition will occur by using a roles and responsibility (RACI) matrix with assignments tasks.

1.7 Document Organization

This document is organized as follows:

Section	Description
Section 2: Primary Cookbook Sections	Provides a high-level description of the Conceptual Review, Requirements, Design, Configuration, Testing, Release Plan, and Shared Services sections of the Cookbook.
Section 3: Proposed Next Steps	Provides a high-level summary of the recommendations in the Cookbook.
Appendix A: IETF Request for Comment	Presents a list of Internet Engineering Task Force resources.
Appendix B: SIP.CONF	Presents the Session Initiation Protocol configuration file.

3 http://guides.wsj.com/management/innovation/how-to-change-your-organizations-culture/
4 http://www.bain.com/publications/articles/management-tools-business-process-reengineering.aspx

Section	Description
Appendix C: Extension Files	Presents the Extensions configuration files for the video auto-routing proof of concept use case.
Appendix D: Queues	Presents the Queues code file.
Appendix E: Detailed Recommendations by Topic	Presents requirements recommendations in the areas of Requirements and Design, Security, Connectivity, and Disaster Recovery and Business Continuity.
Appendix F: Proposed Stacks	Provides an overview of the development and test environment stacks for the use case.
Appendix G: Notional Levels of Effort	Provide a notional level of effort example on previous experience using open source software, subject matter experts, and access to development and test environments.
Appendix H: Essentials of the IPT and IPPD	Presents a description of the IPT and IPPD.
Appendix I: Test Strategy	Provides a test strategy for the video auto-routing proof of concept.
Appendix J: Sample Test Matrix	Provides a sample text matrix for the video auto-routing proof of concept use case.
Acronyms	Defines the acronyms used in this document
List of References	Lists the sources used in preparing this document

2. Primary Cookbook Sections

There are seven primary subsections to the core cookbook:

1. **Conceptual Review:** Develop an overall understanding of the Open Source Asterisk PBX software, operating system, endpoint devices, and call center workflow
2. **Requirements:** Develop a core set of realistic, accurate, and traceable requirements for the auto-routing proof of concept
3. **Design:** Draft an initial design of the workflow based on the core set of requirements
4. **Configuration:** Configure the key variables, settings, and control logic
5. **Test Planning:** Draft test plan and use cases
6. **Release Plan:** Draft plan for release and implementation
7. **Shared Services:** Develop tactical plans for managing the impact of the development and test infrastructure and the call center facilities

The project team is the primary audience for this document. There are specific configuration sections for the senior systems engineer with core network, programming, and computing infrastructure skills, experience, and competencies. The proof of concept is based on the open source development model, which promotes universal access via a free license to a product's design or blueprint, and universal redistribution of that design or blueprint, including subsequent improvements to it.[5]

Appendix F, *Proposed Stacks*, provides an overview of the development and test environment stacks that should be reviewed to aid in the decision to build the development and test platform for the proof of concept.

2.1 Conceptual Review

The conceptual review phase includes additional research, planning, and recommendations about the following key areas:

- Concept of Channels and Dial Plans
- Core Modules
- End User Devices
- Call Center Workflow
- User ASL agent Desktop
- Visual Interactive Voice Response (IVR)
- Automatic Call Distribution (ACD)/IVR/Dialer Infrastructure Systems

5 http://opensource.org/

- Application Design
  - Design Business Rules, Workflows, and Routing Strategies
- Availability
- Scalability
- Geographic Coverage

The primary setup includes a basic dial plan and support for a variety of endpoint destinations (e.g., cell phone, VoIP endpoint, remote PBX)

2.1.1 Conceptual System Overview

A traditional implementation of the Asterisk PBX (as shown in Figure 1) supports an out-of-the box installation and configuration. The primary setup includes a basic dial plan and endpoint devices. [6]

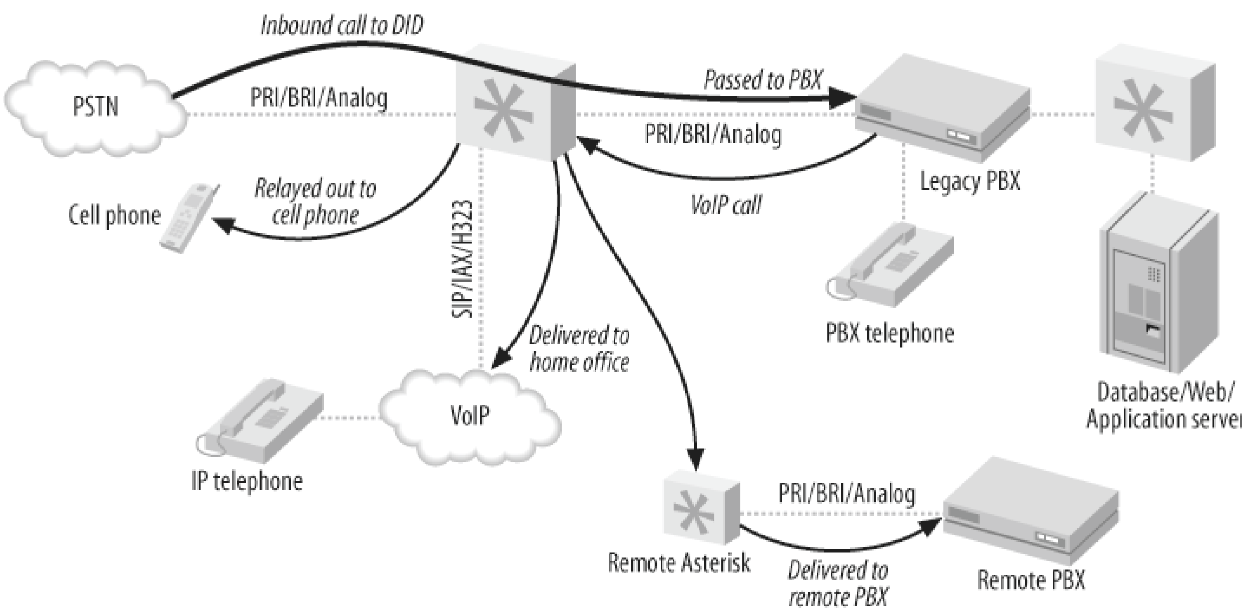

Figure 1. Standard Asterisk Overview

As depicted in Figure 2, the current POC baseline builds off of the standard Asterisk SIP flow and supports a SIP Proxy Server (e.g., Kamailio, http://www.kamailio.org) to manage the SIP traffic. The Asterisk PBX servers are configured with specific inbound call queues and extensions.

The POC baseline establishes a direct video communication workflow that introduces multiple devices that communicate primarily over the SIP protocol. “The SIP proxy acts as a router between the external peers, internal peers and the soft PBX. The soft PBX is a server running

6 http://astbook.asteriskdocs.org/en/2nd_Edition/asterisk-book-html/figs/web/ast2_1501.png

Asterisk. It is important to note that the soft PBX does not connect directly to the public Internet and none of the internal users connect directly to the soft PBX.

SIP proxy servers are generally more stable and more secure than a software version of a PBX, and provide more connectivity options, including support for IPv6, TLS and WebRTC."[7]

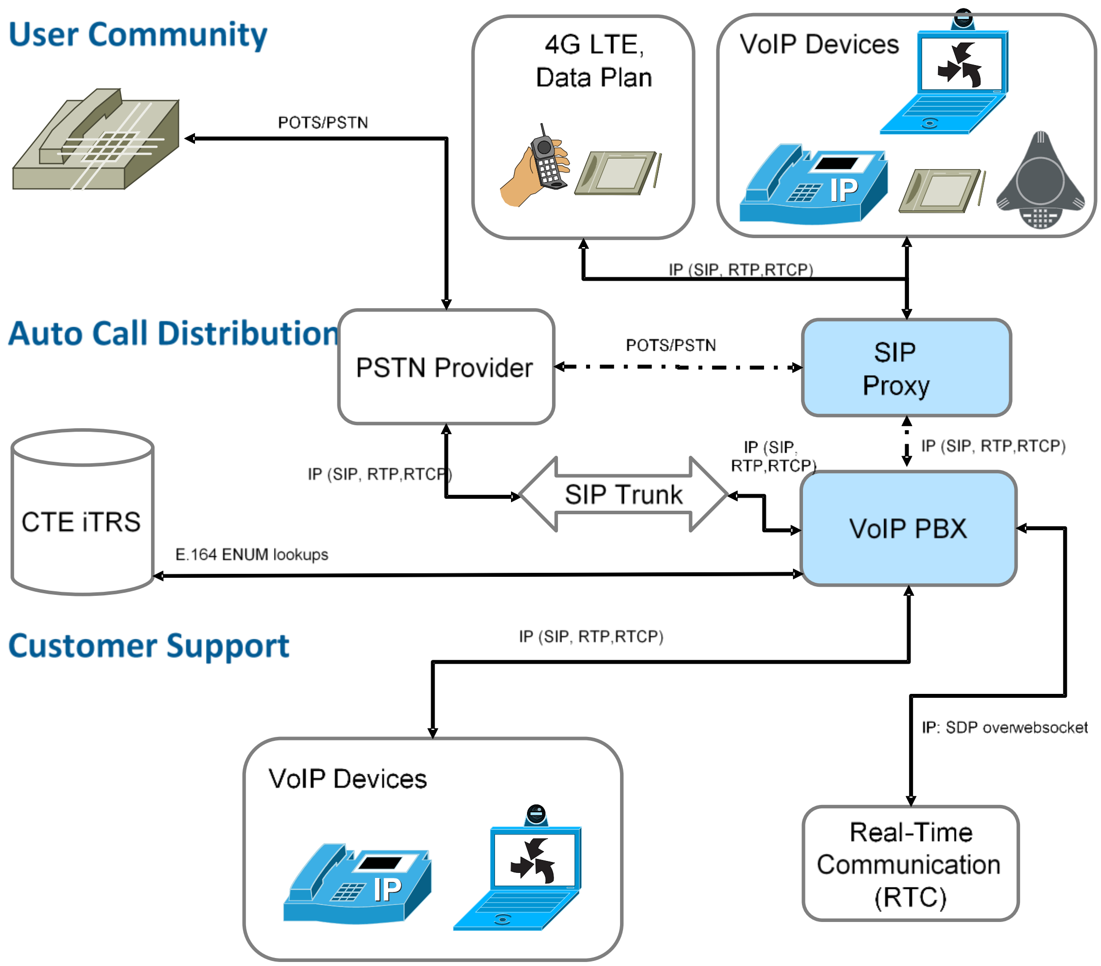

Figure 2. Notional POC Baseline

After testing an initial setup, the secure configuration (e.g., port 5061) and Secure RTP (SRTP) are added and tested to produce a more secure configuration as part of a supporting confidentiality, integrity, and privacy.

---

7 http://rtcquickstart.org/guide/multi/rtc-architecture.html

2.1.1.1 Performance Characteristics

The performance characteristics include such topics as bandwidth, computer or video phone CPU performance, resolution quality, and frame rate.[8] A primary objective is to set a minimum standard for connectivity and preempt challenges from low-quality wireless or remote communication channels that do not provide sufficient bandwidth.

During the test phase, an engineer can capture traffic traversing a network connection from an endpoint device to the PBX. As congestion is most likely to occur in infrastructure bottlenecks such as WAN interfaces, last-mile connections, and the end-user interfaces, the focus is to analyze the data captured in order to identify, classify, and mark the applications/protocols. The test engineer will also identify strategic locations to capture the data to provide the best current view of traffic flow. The data collected can also be used for capacity planning.

2.1.1.2 Availability and Reliability

The standard definition of *availability* is the percentage of time a system is considered ready to use when tasked, while *reliability* is the probability of zero failures over a defined time interval.[9]

Evaluating and updating performance metrics for availability and reliability depends on several considerations:

- Leveraging government/industry best practices
- Expectations of increased effectiveness in service delivery
- Increased availability/reliability based on improved technology
- Performance history

Table 1 presents a notional table used to document availability and reliability metrics, and their relative priority High = H, Medium = M, and L = Low) based on notional services and their current requirements

Table 1. Notional Example: Service Performance Metrics for Availability and Reliability

Service (Notional)	Priority (H,M,L)	Current Requirement	Availability (New Metric)	Reliability (New Metric)
Connectivity Services – Authentication	H			
Connectivity Services – DNS	H			

8 http://eeweb.poly.edu/faculty/yongliu/docs/infocom2014.pdf

9 http://www.mitre.org/publications/systems-engineering-guide/acquisition-systems-engineering/integrated-logistics-support/reliability-availability-and-maintainability

Service (Notional)	Priority (H,M,L)	Current Requirement	Availability (New Metric)	Reliability (New Metric)
Connectivity Services – Dynamic Host Configuration Protocol (DHCP)	M			
Connectivity Services – Network & Internet Access	M			
Connectivity Services – Network Performance (Tier I and II)	M			
Connectivity Services – Remote Access VPN	M			
Connectivity Services – Site Surveys	L			

The typical performance measures for assessing and reporting the availability of a service or component operate on the following assumptions:

- Priority (and associated metrics) for core services across different networks will vary based on Providers and the end user's connection.
- Providers will be measured at the same priority as services they feed downstream to endpoints.

2.1.1.3 Interoperability, System Interconnection / Information Sharing

Endpoint devices that use the Internet for network transport allow users to connect from various access points for Voice over Internet Protocol (VoIP) voice services. An encoder converts the network user's voice, data, or even video to a digital IP -based packet format for transmission. The IP-based formatting and addressing structure provides interoperability. The FCC requirement specifies that all of the appropriate users are addressed correctly, and traffic passes through the router to the appropriate end-user devices on the Internet.[10]

The distinction between Communications Services and Enterprise Services can create ambiguity. To clarify, Communications Services consist of such basic capabilities as access, connection establishment, and packet routing. Communications Services typically are offered in voice, video, and data networks... Specific elements such as switches, gateways, routers, and firewalls support the provision of Communications Services. Directories, ACDs, and IVRs are examples of elements that enable Enterprise Services.

10 https://transition.fcc.gov/pshs/techtopics/tech-ip-interop.html

2.1.1.3.1 Communications Services

Communications Services provide interconnectivity among applications, systems, and people. This layer is implemented both inside the sponsor enterprise—for example, the Local Area Network (LAN)—and external to the sponsor enterprise (for example, a public telephone network carrier). Included in this layer are data networks [metropolitan area networks (MAN) and wide area networks (WAN)], voice networks, video networks, and wireless networks. This layer also includes telephone switches, gateways, automatic call distribution, network routers and switches, firewalls, and network security monitoring systems.

The focus of Communications Services is on the secure, reliable transport of voice, video, or data between endpoints. It includes the protection and integrity of information in transit, the protection of communication systems from external disruptions, and response to network issues and incidents.

2.1.1.3.2 Enterprise Services

Enterprise Services comprise a set of common capabilities deployed across the enterprise. These services are shared IT infrastructure components that mediate relationships between applications throughout the enterprise.

Higher-level communications support functions are classified as Enterprise Services rather than Communication Services. These support functions include call routing based on caller information and the location of employees with the necessary skills to handle the call. Similarly, computer telephony integration functions, such as interfaces to databases to support screen pops, are also classified as Enterprise Services. The rationale for these classifications is that the higher-level functions require platforms and data to operate, while the communications transport does not.

Web Application Services provide a Web-based application execution environment that allows implementation and management of information pages and call center business applications modules.

Distributed application services provide capabilities that allow call center applications to communicate via synchronous or asynchronous mechanisms, including distributed objects such as Enterprise JavaBeans (EJB), .NET remoting, and remote procedure call.

Directory services provide call center-centric repositories of information that can be accessed through a common application-programming interface (API) such as Lightweight Directory Access Protocol (LDAP).

Messaging services provide mechanisms for the reliable and secure transport of commands and data between call center application programs. Messaging service capabilities include message queuing, guaranteed delivery, transaction support, and character code translation. Messaging services also provide for load balancing and fault tolerance among server processes and systems.

Workflow services provide a means for defining and controlling the flow of call center information among users and automated functions in support of business processes.

Logging services provide mechanisms to record nominal and exceptional call center events as required by legislation, sponsor business rules, and enterprise systems management.

Telephony and computer-telephony integration (CTI) services provide the ability to exchange control information among telephone switches, voice response units, and application programs to support customer service and telephone call center self-service systems.

Data access services define interfaces and formats for exchanging call center information among applications, including but not limited to, markup languages.

Software engineering services cover the technology associated with building/changing call center software systems as well as technical solutions supporting management issues, such as testing, modeling, and versioning.

2.1.1.4 Provisions for Security, Privacy, Integrity, and Contingency

The security requirements, including the Information Security Acceptable Risks Safeguards and any security requirements in the design, should provide information on system access, generally accepted security principles, and specific application security requirements. The design should also provide security requirements that include encryption of call setup and payload.

Because this is part of the effort that extends beyond the initial release of the POC, there are several areas to emphasize:

- The Open Source technical architecture should have a cross-reference for each of the configurations denoting security requirements and providing information on how the design complies with the requirements for a vulnerability assessment. The cross-reference can provide a security compliance checklist that can be reused with additional implementations of the technical architecture.
- The design should specify POC configuration information to cross-reference the configuration settings necessary to maintain all hardware and software proposed in the technical architecture. A baseline configuration can provide a reliable starting point for guiding an installation or making updates to a baseline release.
- The design should include a processing view that provides operational requirements that include intrusion detection and monitoring specifications.
- The design should include a Protocol view that depicts all port assignments on all hardware for each processing layer (e.g., SIP Proxy, Asterisk PBX, Directory Services, and Database Services). Each open source application and server uses protocol ports at the network (transport) layer to manage communication channels between programs. The design should depict all ports used by hardware and software to help with the installation and configuration, and to assist with determining risk.

2.2 Requirements

In the requirements phase, the project team researches each of the following areas, and prepares a statement of objectives and requirements. The requirements include the technical, operational, and functional needs of the specific use case in the proof of concept as follows:

- Multi-purpose signaling SIP server
  - SIP Router/Switch, SIP Registrar

  - Application Server
  - Redirect Server
  - Load Balancer / Dispatcher
  - Back-to-Back User Agent
  - Presence Server, IM Server
  - Session Border Controller
  - SIP Front-End
  - Network Address Translation (NAT) Traversal Server
  - IP Gateway [(Short Message Service (SMS), Extensible Messaging and Presence Protocol (XMPP)]
- User, Administration, and Provisioning Portal
- PSTN (public switched telephone network) Gateway
- Media Server
- Media Proxy or Real-Time Protocol (RTP) Proxy for NAT Traversal
- Fundamental Accounting
- Call Detail Records (CDR) (storing and reporting)
- Monitoring Tools [OrderlyQ (Linux based) (http://www.orderlyq.com/) and AstChannelIsLive (Windows) (http://sourceforge.net/projects/astchannelslive/files/AstChannelsLive_5.0.2/)]
- User End Point Device and Softphone configurations
- Firewall Setting (Firewall and NAT devices must be SIP Aware)
- An understanding of Network Address Translation, Session Traversal Utilities for NAT (STUN), Traversal Using Relays around NAT (TURN), and Interactive Connectivity Establishment (ICE)

For the current release of the POC, the source code and scripts are updated to support the following requirements:

- Standard PBX functions associated with the out-of-the-box Asterisk configuration files, including:
  - Call queuing
  - Call routing
  - Database integration
  - Number discovery
  - Local and remote agents
  - Video message on hold
- Adapted to support Provider endpoint device (e.g., ZVRS Z-20)
- Interactive Voice and Video Response: Video IVR Navigation via DTMF dial pad

- Dial by name: Option to allow direct dialing to a specific ASL agent via extension
- Call detail records: Aggregate information on completed/abandoned calls (call time, length, call-back info, video mail, location, etc.)
- Basic monitoring: Dashboard overview for supervisors with call center metrics / performance management tools (such as queue info, interactions in progress, ASL agent statistics, etc.)

Table 2 presents an initial capture of workflow requirements that demand additional planning, scoping, and discussion to guide a representative level of effort to extend the functionality of the POC.

Table 2. Workflow Requirements

Count	Type	Requirement
1	Contact Center ASL agent UX	Transfer customer from one CSR to a specified queue. (A companion requirement, the transfer of a customer from a specific queue to a specific CSR, is premised on skills-based routing.)
2	Contact Center ASL agent UX	Transfer customer into a voice queue inserting an interpreter
3	Contact Center ASL agent UX	Visual alerting options for when receiving an incoming call [screen flash, toaster-style pop-up, Universal Serial Bus (USB) strobe light, etc.]
4	Contact Center ASL agent UX	Switch outgoing video inputs among different feeds—webcam, screen sharing, image library
5	Contact Center ASL agent UX	Enhanced Caller ID with info pulled from iTRS database / telco carrier / IP address / Customer Relationship Management (CRM) (caller's name, phone number, city/state, etc.)
6	Contact Center ASL agent UX	Video Whiteboard—ability to superimpose text/numbers onto outgoing video feed
7	Contact Center ASL agent UX	Ability to transfer/escalate calls to another ASL agent or number (Warm, Tepid & Cold transfers)
8	Contact Center ASL agent UX	Text chat with callers and fellow ASL agents ([IM or Real Time Text (RTT)]
9	Contact Center ASL agent UX	3-way video calling (VRS interpreter, supervisor, etc.) / A-B-C line switching
10	Contact Center ASL agent UX	Accept Dual Tone Multi-Frequency (DTMF) dial pad input [confirmation numbers, Social Security Numbers (SSN) etc.]
11	Contact Center ASL agent UX	Ability to quickly search for and pull up reference info (factsheets, call scripts, etc.)
12	Caller UX	Visual IVR Navigation via DTMF dial pad
13	Caller UX	Dynamic on-hold screen, including ACD wait time estimates, etc.

Count	Type	Requirement
14	Caller UX	Ability to select and view videos via Video IVR—e.g., FCC video announcements made in ASL
15	Caller UX	Display ASL agent's name/ID/email address onscreen for caller
16	Caller UX	Option to allow direct dialing to a specific ASL agent via extension
17	Caller UX	Smart queueing, allows caller to leave video mail and/or request call-back
18	Contact Center Administration	Visual IVR with ability to display different elements in rotation (images, videos, etc.)
19	Contact Center Administration	An ability to support customization of ACD call flow routing rules
20	Contact Center Administration	Automatic/predictive dial-out for idle ASL agents to fulfill call-back requests
21	Contact Center Administration	Aggregate info on completed/abandoned calls (call time, length, call-back info, video mail, location, etc.)
22	Contact Center Administration	Call recording/monitoring capabilities (incl. date/time stamp and caller ID information)
23	Contact Center Administration	Computer/telephony integration API so Caller ID info "pops" into CRM automatically—displaying previous calls, caller information and notes as needed
24	Contact Center Administration	Supervisor notification when callers have to wait beyond a configurable amount of time
25	Contact Center Administration	Dashboard overview for supervisors with call center metrics / performance management tools (queue info, interactions in progress, ASL agent statistics, etc.)
26	Contact Center Administration	Capability to listen in on calls, "whisper" to ASL agents (via translucent/separate video feed?), park calls and jump into calls if needed
27	Contact Center Administration	Call disposition coding so calls can easily be categorized (e.g., info requests, complaints, referrals, and crazy individuals)
28	Contact Center Administration	Video Conference bridge to allow for multi-party calls
29	Contact Center Administration	RTT modem gateway to accept TTY/TDD calls, for ASL agent cross-utilization
30	Contact Center Administration	Bridge to VRS to accommodate hearing callers if necessary (i.e., 3-way call with telecom carrier support)

2.3 Design

The design phase requires the project team to define each Asterisk module, function, and capability that is active for the specific use case in this proof of concept. The modular and functional decomposition activities are integral to provide a cross-reference and traceability to the requirements and objectives.

From an architectural standpoint, Asterisk consists of many different modules. This modularity provides flexibility in the design of a SIP Auto-Routing system. From a system administration perspective, each module selected and loaded provides different capabilities to the system. For example, one module might allow the system to communicate with SIP, while another might add call-reporting capabilities.

Asterisk has a core that interacts with many modules. Modules called channel drivers provide channels that follow a specific dial plan to execute programmed logic and facilitate communication between devices or programs external to the core. The channels often use bridging infrastructure to interact with other channels.

The PBX core is the essential component that provides the primary infrastructure. The PBX core has many functions: it reads the configuration files, including dial plan, and loads all the other modules and distinct components that provide greater functionality.

The dial plan contains a list of instructions to handle incoming and outgoing calls on the system.

The following design checklist can be used to evaluate the success of a modular design. Each question should be answered in the affirmative.[11]

- Does the design identify clearly defined modules?
- Does each module have a clearly defined purpose? (Can you summarize it in one sentence?)
- Is each module's interface sufficiently abstract that you do not need to think about its implementation in order to understand it? Does it hide its implementation details from other modules?
- Have you subdivided modules as far as usefully possible?
- Have you verified that different modules do not replicate functionality?
- Have you isolated those aspects of the design that are most hardware specific, complex, or otherwise likely to change?

Appendix E, *Detailed Recommendations by Topic Area*, provides detail of design, and to some degree, operational, requirements.

2.4 Configuration

The configuration section will specifically deal with configuration files needed to operate the Asterisk PBX server. These files are in the following appendices:

- Appendix B – SIP.CONF
- Appendix C – Extensions File
- Appendix D – Queues

[11] http://www.mcs.anl.gov/~itf/dbpp/text/node40.html

The reader should refer to the Conceptual Review section for an overview of the main architecture components; however, the majority of Asterisk's features and functionality are separated outside of the core into various modules. Each module has distinct functionality, but sometimes relies on another module or modules.

Asterisk provides capability to load modules automatically and manually. Module load order can be configured before load-time, or modules may be loaded and unloaded during run-time.

The configuration file for Asterisk's module loader is modules.conf. It is read from the typical Asterisk configuration directory.

The configuration consists of one large section called "modules" with possible directives configured within, including:

- Timing Interfaces
- Asterisk Built-in mini-HTTP Server
- Logging Configuration
- Asterisk Command Line Interface (CLI) Configuration
- Configuring the Asterisk Module Loader
- Configuring Localized Tone Indications
- Video Telephony

Table 3 presents a summary of the specific proof-of-concept settings for the primary use case. It is important to research each Variable Assignment and assign a Variable Value for the test setup, and eventually for a live test from different endpoint devices and sponsor's customer service desktops.

Table 3. Specific Use Case Settings

Variable Description	Variable Assignment Examples
Vendor Call Center registered number	703-570-1234 (replace with assigned number)
Numbers to be used for softphone end user devices	703-265-5710 to 703-265-5720 (replace with selected numbers)
Public Domain Name Service (DNS)	ec2-54-86-14-195.compute-1.cloudserver.com (an example of a Linux server domain name, replace)
Transport	UDP
Video Codec	H.263, H.264
Audio Codec	Ulaw/PCMA (µ-law Algorithm/Pulse Code Modulation alaw)
Customer Service Representative (CSR) designated number	703-265-5721 to 703-265-5725 (replace)
To register device as a video capable CRS	dial [*10], to unregister dial [*15]

Variable Description	Variable Assignment Examples
To register device as an audio-capable CSR	dial [*20], to unregister dial [*25]
Google Stun Server	stun.l.google.com:19302 (an example of a production STUN server, replace)

The following subsections are detailed examples of specific configuration files and the configuration variable assignments that should be edited by the project team system engineer SME. This section of the document is used in a software development and testing life-cycle process. It is intended as part of the research, experiment, discovery, and test phase; it is not, however, intended as a production guide or to edit or update operational or live configuration files to an active production system.

This section of the cookbook assumes the following tasks and supporting activities are in place prior to editing the configuration files:

- Read and review the resources in the previous section, Supplemental Documentation; these documents have the required research, lessons learned, commands, and code examples to create, test, and run a baseline instance of the Operating System, the Asterisk Server, and the endpoint devices.
- Design and document a dial plan workflow with realistic phone extensions. This draft dial plan and the associated workflow with extensions have the primary steps necessary to establish the baseline call routes.
- Establish a working and replicable development and test (dev/test) environment. The dev/test environments require simple, working change management and configuration management processes. These processes are useful to track and review code changes, maintain traceability to the test cases and requirements, and manage the level of effort associated with changes.
  - These environments require minimum connectivity and bandwidth requirements to support endpoint devices, primary proxy and PBX servers, and CSR desktops. The specifications are documented in the supplemental reading.
  - Prior to editing the following files, make a backup copy, and use the comment (‘;’) liberally to document your inline code changes. There are eight (8) configuration files that need changes to implement and test the auto-routing workflow.

2.4.1 Filename: enum.conf:

ENUM: The bridge between the switched telephony network and the Internet (E.164 Number to URI Mapping) translates telephone numbers into Internet addresses.

ENUM Configuration for resolving phone numbers over DNS, set the following:

search => itrs.us

Reference the following iTRS documentation:

- Internet-based Telecommunications Relay Service (iTRS) Customer User Guide

  - Document: iTRS_User_Guide_5.0_v1.1
- Neustar™ Addressing & Routing iTRS Provisioning Interface Guide
  - Document: iTRS_PI_Guide_5.0_v1.1
- Neustar™ Addressing & Routing iTRS Query Interface Guide
  - Document: iTRS_QI_Guide_5.0_v1.1

2.4.2 Filename: rtp.conf:

The configuration of Asterisk Real Time Protocol (RTP) media channels. RTP is used for SIP communications.

RTP start and RTP end configure start and end addresses:

```
rtpstart=10000
rtpend=20000
```

Interactive Connectivity Establishment (ICE) is used to facilitate Network Address Traversal and communications involving hosts on private network installations located behind firewalls.

Enable ICE support:

```
icesupport=true
icesupport=yes
```

2.4.3 Filename: http.conf:

To provide secure communications, enable the HTTP/HTTPS interface:

```
enabled=yes
```

In addition to enabled=yes, you need to explicitly enable Transport Layer Security (TLS), define the port to use, and have a certificate stored.

The CA certificate must be known to all the clients to ensure certificate legitimacy.

Set the path to the certificate file (*.pem):

```
tlscertfile=/etc/ssl/certs/asterisk.pem
```

Set the path to private key file (*.pem):

```
tlsprivatekey=/etc/ssl/certs/asterisk.pem
```

2.4.4 Filename: queues.conf:

These are the global settings for call queues, Queue Timing Options and Weight Queue Options.

The frequency to announce queue position and/or estimated holdtime to caller (0=off):

```
announce-frequency = 5
```

The absolute minimum time between the start of each queue position and/or estimated holdtime announcement

```
min-announce-frequency = 2
```

How often to make any periodic announcement (see periodic-announce)

periodic-announce-frequency=10 ;

To announce the Queue position, set to “yes":

announce-position = yes

Each member of this call queue is listed on a separate line in the form technology/dial string; “member” means a normal member of a queue. A Priority Queue (Queue Weighting) is needed to add people to a queue at a higher priority than that given to other callers.

Perhaps the caller has already spent time waiting in a queue, and an ASL agent has taken some information but realized the caller needed to be transferred to another queue.

In this case, to minimize the caller’s overall wait time, it is desirable to transfer the call to a priority queue that has a higher weight (and thus a higher preference), so it will be answered quickly.

An optional penalty may be specified after a comma, such that entries with higher penalties are considered last. An optional member name may also be specified after a second comma, which is used in log messages as a “friendly name”. Multiple interfaces may share a single member name. An optional state interface may be specified after a third comma.

The sample configuration section name is:

[**StandardQueue**](!)(general)

[**VideoQ_GenQuest_1**](StandardQueue)

[**VideoQ_Complaint_1**](StandardQueue)

2.4.5 Filename: manager.conf

This is a reference to the Asterisk Manager Interface (AMI), and provides third party application call management support and PBX event supervision.

Use the "manager show commands" at the CLI to list available manager commands and their authorization levels.

Note that you should not enable the AMI on a public IP address. If needed, block this TCP port with iptables (or another FW software) and reach it with IPsec, SSH, or SSL vpn tunnel.

To make the manager interface available over http/https, set the Asterisk http server to enabled in http.conf:

"enabled"=yes

"webenabled"=yes.

2.4.6 Filename: extensions.conf

The Asterisk dial plan includes statements to allow a developer to split up the functionality into smaller sections, and then have Asterisk search multiple contexts for a dialed extension. Most commonly, this functionality is used to provide security boundaries between different classes of callers. It is important to remember that when calls come into the Asterisk dial plan, they are directed to a particular context by the channel driver. Asterisk then begins looking for the dialed extension in the context specified by the channel driver. By using include statements, other contexts in the search for the dialed extension are included.[12]

```
include => record_studio

include => Company1_QueueLoginLogout
```

Set the call workflow to Company 1, edit the proxy numbers and provide an ENUM lookup:

```
exten => _17034369338,n,Set(sipuri=${ENUMLOOKUP(${CALLERID(num)},,,,itrs.us)})

exten => _17034369338,n,GotoIf($["${sipuri}" = ""
]?VoiceQ_GenQuest_1,start,1:VideoQ_GenQuest_1,start,1)

exten => _17034369339,n,AGI(queue_feedback.agi)
```

Set the play_key_video:

```
exten => num1,1,Playback(number_1-recording0)

exten => num1,n,Goto(company1_video_caller_query,start,100)
```

Set the IVR and determination or proper Video Q:

```
[company1_video_caller_query]

exten => start,1,(start)

exten => start,n,Verbose(2,${CALLERID(num)} entering the Company1 skill query for proper queue.)

exten => start,n,Wait(3)

exten => start,n,Playback(quiet_1sec)

exten => start,20,Playback(welcome-recording1)

exten => start,21,Playback(4_GeneralQuestions-recording1&5_Complaints-
recording0&9_to_Repeat_menu-recording0)
```

Note: there is a separate section that explains the video sources and the video playback

```
exten => start,n,SendImage(/var/lib/asterisk/images/asterisk-intro)
```

Echo the key press back to caller:

```
exten => start,n,GotoIf($["${digito}" = "0" ]?play_key_for_video,num10,1)
```

Set the Company1 Q details for VIDEO Queue -1:

```
[VideoQ_GenQuest_1]

exten => start,1,Verbose(2,${CALLERID(num)} entering the VIDEO queue)

exten => start,n,Set(qinfo=${QUEUE_VARIABLES(VideoQ_GenQuest_1)}) ; get the QUEUE information.
returns 0 f successful

exten => start,n,Set(CALLERID(num)=${CALLERID(num):0:40}) ; to cover for a bug that only allowed for 40
bytes

exten => start,n,Set(CALLERID(name)=${CALLERID(name):0:40})

exten => start,n,Set(ACTUALTO=sip:${CALLERID(num)})

exten => start,n,Set(ACTUALFROM=sip:${EXTEN})
```

---

12 https://wiki.asterisk.org/wiki/display/AST/Include+Statements

```
exten => start,n,Macro(sendIMmacro,"You are in the General Questions (video) Queue. There are
$[${QUEUECALLS}+1] calls ahead of you. The average wait is about ${QUEUEHOLDTIME}
minutes",${ACTUALTO},${ACTUALFROM})
```

Load up the variables that will be accessed from the queue app by the macro that is passed:

```
exten => start,n,Set(_MYARG1="You are now connected to an (video) agent who can handle your
questions.  Thank you.")
```

Execute the queue and pass the macro:

```
exten => start,n,Queue(VideoQ_GenQuest_1,t,,,,,sendIM_Q_macro,video-gosub)
```

Establish VIDEO Queue -2:

```
[VideoQ_Complaint_1]

exten => start,1,Verbose(2,${CALLERID(num)} entering the VIDEO queue)

exten => start,n,Set(qinfo=${QUEUE_VARIABLES(VideoQ_GenQuest_1)}) ; get the QUEUE information.
returns 0 f successful

exten => start,n,Macro(sendIMmacro,"You are in the Complaints (video) Queue. There are
$[${QUEUECALLS}+1] calls ahead of you. The average wait is about ${QUEUEHOLDTIME}
minutes",${ACTUALTO},${ACTUALFROM})
```

Define the subroutines that are passed to the queue application calls.

GoSub allows you to execute a specific block (context or section) of dialplan as well as pass and return information via arguments to/from the scope of the block. Whereas Macro has issues with nesting, GoSub does not and GoSub should be used wherever you would have used a Macro.

```
[video-gosub]

exten => s,1,Verbose("Here we are in a subroutine GeneralQ! Playback audio and video")

exten => s,n,Answer()

exten => s,n,Wait(3)

exten => s,n,Playback(customer_selected_GeneralQ-recording1)

exten => s,n,Return()
```

Set the Asterisk command AddQueueMember

Synopsis: Dynamically adds queue members

Options:

queuename - The name of the queue to add a member to

interface - The interface to add to the queue, if not specified, uses the current interface

penalty - Integer greater than or equal to 0, available members with lower penalties will get calls first

Establish Company1 agent registration:

```
[Company1_QueueLoginLogout]

exten => *10,1,Verbose(2,Logging in skill-1 VIDEO General queue member)

exten => *10,n,Set(MemberChannel=${CHANNEL(channeltype)}/${CALLERID(num)})

exten => *10,n,AddQueueMember(VideoQ_GenQuest_1,${MemberChannel})

exten => *10,n,Wait(2)

exten => *10,n,Playback(quiet_1sec)
```

```
exten => *10,n,Playback(your_are_the_rep_in_general-recording0) ;,skip
exten => *10,n,Hangup()
```

Define the Demonstration with a provider assigned number. These are the numbers that a person will call from an endpoint device to a CSR desktop:

```
[incoming_from_provider]
include => VideoQ_GenQuest_1
exten => _7035705868,1,Answer()
exten => _7035705868,n,Goto(company1_video_caller_query,start,2)
include => VideoQ_company2
```

TEST the IVR Q skill-based:

```
exten => _7035705174,n,Goto(company1_video_caller_query,start,2)
```

2.4.7 Filename: sip.conf

It is useful to visualize the relationship between the channel configuration files (sip.conf) and the dial plan (extensions.conf). The dial plan is the heart of an Asterisk system: it controls how call logic is applied to any connection from any channel. Both the relevant channel configuration file and the extensions.conf file play a role in most calls routed through the system.

When a call comes into Asterisk, the identity of the incoming call is matched in the channel configuration file for the protocol in use (e.g., sip.conf). The channel configuration file also handles authentication and defines where that channel will enter the dial plan.

Once Asterisk has determined how to handle the channel, it will pass call control to the correct context in the dial plan. The context parameter in the channel configuration file tells the channel where it will enter the dial plan (which contains all the information about how to handle and route the call).[13]

ACD demo- support the SIP trunk to our SIP proxy server:

```
transport=tcp,udp          ; Set the default transports.  The order determines the primary default transport.
srvlookup=yes               ; Enable DNS SRV lookups on outbound calls

localnet=172.31.16.0/255.255.240.0 ; RFC 1918 addresses
externaddr=54.86.14.195         ; use this address.
nat=force_rport
```

CODECs represent mathematical algorithms for encoding (compression) and decoding (decompression) media streams. Asterisk uses CODEC modules to both send and receive media (audio and video). Asterisk also uses CODEC modules to convert (or transcode) media streams between different formats.

[13] http://www.asteriskdocs.org/en/3rd_Edition/asterisk-book-html-chunk/DeviceConfig_id216341.html

Asterisk supports video telephony in the core infrastructure. Internally, it's one audio stream and one video stream in the same call. Some channel drivers and applications have video support, but not all.

Asterisk supports the following video codecs and file formats.

Establish a template for the preferred codecs:

```
[my-codecs](!)                    ;
    disallow=all
    allow=ilbc
    allow=g729
    allow=gsm
    allow=g723
    allow=ulaw
```

Establish one for ulaw-only:

```
[ulaw-phone](!)
    disallow=all
    allow=ulaw
```

Establish Company 1 assignments:

```
[company1_phone](!)
       type=peer
       transport=udp
       secret=moie
       host=dynamic               ; This device needs to register
       directmedia=no             ; Typically set to NO if behind NAT

       allow=all
       allow=ulaw
       allow=h263
       allow=h263p
       allow=vp8
       allow=h264
       registertrying=yes         ; Send a 100 Trying when the device registers.
       context=company1
       callcounter = yes          ; Enable call counters on devices. This can be set per
       qualify=yes
       busydetect=yes
       videosupport=yes           ; Turn on support for SIP video. You need to turn this
       dtmfmode=rfc2833
```


For Pubic Release – In Accordance with Data Rights

Define the following support for instant messaging:

```
accept_outofcall_message=yes
outofcall_message_context=IM
auth_message_requests=yes
[7032655700](company1_phone)
regexten=7032655700            ; When they register, create extension 1234
callerid=<7032655700>

[7032655701](company1_phone)
regexten=7032655701            ; When they register, create extension 1234
callerid=<7032655701>
```

Establish a TEST path from provider number:

```
[provider_test_phone_tcp](!)
        type=friend
        transport=tcp
        directmedia=no
         host=208.94.16.194
        nat=force_rport
        directmedia=no          ; Typically set to NO if behind NAT
        disallow=all
        allow=ulaw
        allow=h263
        allow=h264
        context=incoming_from_provider
        videosupport=yes        ; Turn on support for SIP video. You need to turn this
        dtmfmode=rfc2833

        accept_outofcall_message=yes
        outofcall_message_context=IM
        auth_message_requests=yes
        [7035705868](provider_test_phone_tcp)
        regexten=7035705868          ; When they register, create extension
        [7035705174](provider_test_phone_tcp)
        regexten=7035705174          ; When they register, create extension
```

2.4.8 Filename: [network:interfaces.d] eth0.cfg:

Define a primary network interface so that Asterisk can connect and communicate through the operating system:

```
auto eth0
iface eth0 inet dhcp
dns-nameservers 156.154.59.227
dns-nameservers 172.31.0.2
dns-search ec2.internal
```

2.4.9 SIP Proxy Files

Define SIP Proxy host specific configuration files and initialization parameters to start and stop services:

```
# Kamailio startup options
#
# Set to yes to enable kamailio, once configured properly.
RUN_KAMAILIO=yes
```

Review user and group assignments to meet security requirements:

```
# User to run as
#USER=kamailio
# Group to run as
#GROUP=kamailio
```

Updates to the Asterisk SIP.CONF file:

```
[Kam-ACD-sip-proxy]
type=friend
host=52.2.224.220
fromdomain=54.86.14.195
port=5060
transport=udp
videosupport=yes
disallow=all
allow=ulaw
allow=h263
allow=h264
context=company1
directmedia=yes
insecure=port,invite
dtmfmode=rfc2833
nat=force_rport
qualify=yes
```

Maintain the ICE and STUN to address issues with NAT:

;icesupport=true

;stunaddr=stun.l.google.com:19302

2.5 Test Planning

The following are the key objectives and guiding principles of the Test Planning Phase:

- Leverage existing facilities resources for the Proof of Concept application projects
- Initiate infrastructure enhancements, including virtualization
- Initiate test data preparation activities
- Deploy core service management processes and enable support tools
- Design standard service offerings to customers
- Establish a presence in a Vendor/Infrastructure as a Service (IaaS) Cloud Computing Facility as a short-term measure
- Support for the development of the Acquisition Strategy to support IaaS

To achieve these objectives, significant actionable items have been identified and are presented in the following subsections. These actionable items are organized by technical area referenced in Appendix I, *Sample Test Matrix*.

2.5.1 Test Planning: Actionable Items

The test plan action items can provide baseline activities for the development of a test project plan. This test plan can guide the approach and focus on specific test areas and issues. The actionable items can also be used to connect specific areas of the test scope and objectives to improve "the testability of the item under test".[14]

The detailed subsections comprise the actionable items for guiding the IPT and are referenced in Appendix H, *Test Strategy*.

2.6 Release Plan

The project team will draft a release plan with the following content:

1. Project Context
    a. Compressed Schedule
    b. Integration Testing done in Production Environment
    c. No real "production" implementation
2. Status of Operational Readiness Review (ORR) artifacts

[14] http://istqbexamcertification.com/what-is-the-purpose-and-importance-of-test-plans/

   a. Operations Manual
   b. Test Summary Report
   c. System Accreditation Form
   d. Companion Guide
   e. Version Description Document
   f. Implementation Plan
3. Project Timeline
   a. Snapshot of major milestones
   b. Impact of current timeline on various activities – Stress Testing, Integration Testing, etc.
4. Status of Integration Testing
   a. What has been tested and results (high level)
   b. What has not been tested (high level)
   c. Review of open Change Requests – identify showstoppers and anticipated time to resolve
5. Status of remaining development activities and system testing
6. Help Desk Readiness
7. Operations Readiness
   a. Stability of system (components staying up?)
   b. Availability of monitoring tools and trained resources
   c. Operations and Maintenance (O&M) documentation status
   d. Escalation procedures defined (near-term)
   e. Stress Test results/decisions
8. Security Readiness
   a. Administrative Safeguards
   b. Physical Safeguards
   c. Technical Safeguards
9. Outstanding Risks and Concerns
   a. Finishing integration and stress testing
   b. How to elevate changes to production without impact to production users
   c. Lack of proactive monitoring tools and automated responses
10. Next Steps

2.7 Shared Services

The shared services approach to an auto-call routing workflow is to provide the right services to the right subject matter expert at the right time. This approach supports ASL-qualified subject matter experts to support multiple domains across agencies, companies, and organizations from geographically diverse locations. Customers can leverage this approach to optimize resources and drive better end-user experience (e.g., equivalency for deaf and hard of hearing users, shorten wait times during periods of heavy call loads).

2.7.1 Develop the Near-term Tactical Plans

This activity develops the near-term, tactical plans for managing the impact of the development and test infrastructure and the call center facilities on the entire computing environment. The plans should include current, strategic, transitional, business impacts, financial, and fund source requirements. These requirements may include:

- Hardware Plan
- Software Plan
- Business Applications Plan
- Facilities Plan
- Network Connectivity and Traffic Plan
- Human Resource Allocation Plan
- Communications Plan
- Security Plan
- Business Requirements/Impact Analysis Plan
- Transition/Migration Plan
- Performance Management and Auditing Plan
- Business Continuity Plan
- Data Management/Data Quality Plan

To manage the risk for this activity, the effort should coordinate the business leadership and the participation for informed IT planning to reduce the chance for the interruption of services.

2.7.2 Task-Specific Prioritization and Baseline Metrics

The first activity should develop a detailed task-specific plan for the development and test infrastructure and the call center facilities. The plan should include sections that delineate roles to:

- Develop program policy, strategic, and tactical plans
- Define performance expectations and performance measures, and manage vendor contractors
- Define project/communication plans to manage infrastructure and call center facilities transition/cost, the associated vendor contractors, and the business partners

The prioritization plan should include a methodology to account for the performance measures and scorecards. The results should become part of a Service Delivery Portfolio to include:

- A ”lean” Program Management Office (PMO) for the transition period
- A problem reporting process and trouble ticket system
- Operational procedures
- A Communications Plan
- Monitoring tools
- A cross-reference to applicable Service Delivery Targets
- Network standards and topology, including call center connectivity, Network Operation Center (NOC) administration and remote management
- Provisioning interim (temporary) services
- Equipment redirection and retirement plans
- A service continuity and contingency plan

To manage the risk for this activity, the effort should account for the full and adequate understanding of the complexities and interdependencies of provisioning equipment for virtualization and consolidation of development and test infrastructures.

The second activity should consider that the decision-makers are well informed on agency impacts and service delivery (e.g., network connectivity limitations or delays). A governance structure should be leveraged or established to ensure key decisions on infrastructure, architecture, networking, and strategic goals for implementation, culture, and funding.

The expected outcome for this activity is a task-specific prioritization plan for the development and test infrastructures and call center facilities.

To manage the risk for this activity, the effort should provide a continuous communication loop to the key decision-makers to ensure reporting, assessment, and mitigation of risk. Decision-makers and the governance structure should be kept informed of the risks, impact, and exposure if one or more sections of the contingency plan are executed.

The third activity is to draft a business continuity plan. A business continuity plan is vital because the ACR proof of concept introduces new workflow and dependencies on ASL-qualified subject matter experts to handle the direct video calls.

The expected outcome for this activity is a business continuity plan to support the development and test infrastructures and call center facilities.

The risk to an agency, company, or organization is a gap in ASL-qualified SMEs during an incident or event that causes a disruption in service.

3. Proposed Next Steps

This document provides an approach and guide to replicate the ACR configuration. Its guidance helps the engineering team tailor the resultant configuration to the customer's/agency's requirements and needs (i.e., the outcome results in more than a replication of the MITRE ACR POC).

MITRE recommends the following actions to implement the proof of concept with other agencies, organizations, and companies:

- Require change control for development and test environment configurations
- Develop a prioritized list of tool changes that would integrate capabilities, reduce risk, and improve team productivity
- Develop regression testing and mature manual test processes
- Formalize a Change Control Board (CCB) structure and operation. Implement a Change Advisory Board (CAB) to analyze each proposed change and provide extensive information about the background, potential problems, and possible solutions with pros and cons.
- Define and document the Change Management Process to generate accurate and complete information about change dependencies
- Manage and track Change Requests (CR) to ensure that the impact of a proposed change is clearly understood across other projects that are related to the changes requested (such as infrastructure upgrades to current call center operations)
- Conduct Impact Analysis to place greater emphasis on the impact of proposed changes and ensure the interdependencies are clearly listed for review
- Update or draft a Concept of Operations to introduce the POC to production services and review interdependencies and impact to schedules
- Modify release cycle and documentation. Assign appropriate personnel to implement a release management plan and template for all releases.
- Create a document repository and designate ownership. Develop a SharePoint site to serve as the template repository.
- Create a communications and shakeout plan. Formalize the release process with a standard deployment and shakeout plan.
- Perform root cause analysis. Develop a methodology for root cause analysis and assist the testing team with root cause analysis after a release.
- Create a schedule of regular meetings, working sessions, and facilitated technical exchanges to define, update, and document user needs and system requirements

- Capture design, development, and testing lessons learned from each release
- Develop a test and integration process to streamline, measure, and compare for optimized user work processes
- Develop an interim process to assess, compare, and document the end-user workflow and end-user input with regard to new requirements

Appendix A. IETF RFC

The following Requests for Comments (RFC) are associated with Session Initiation Protocol (SIP) but are not listed in this Appendix: 3261, 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954, 6026, 6141, 6665, 6878, 7462, 7463.

RFC 3428: Session Initiation Protocol (SIP) Extension for Instant Messaging
https://tools.ietf.org/html/rfc3428

RFC 2833: RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals
https://tools.ietf.org/html/rfc2833

RFC 2543: SIP: Session Initiation Protocol
https://tools.ietf.org/html/rfc2543

RFC 3550: RTP: A Transport Protocol for Real-Time Applications
https://tools.ietf.org/html/rfc3550

RFC 3605: Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)
https://tools.ietf.org/html/rfc3605

RFC 3986: Uniform Resource Identifier (URI): Generic Syntax
https://tools.ietf.org/html/rfc3986

RFC 4904: Representing Trunk Groups in TEL/SIP Uniform Resource Identifiers (URIs)
https://tools.ietf.org/html/rfc4904

RFC 4961: Symmetric RTP/RTP Control Protocol (RTCP)
https://tools.ietf.org/html/rfc4961

RFC 5245: Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols
https://tools.ietf.org/html/rfc5245

RFC 5527: Combined User and Infrastructure ENUM in the e164.arpa Tree
https://tools.ietf.org/html/rfc5527

RFC 5630: The Use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)
https://tools.ietf.org/html/rfc5630

RFC 6116: The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)
https://tools.ietf.org/html/rfc6116

RFC 3551: RTP Profile for Audio and Video Conferences with Minimal Control

Appendix B. Extensions File

```
; extensions.conf - the Asterisk dial plan
;
; Static extension configuration file, used by
; the pbx_config module. This is where you configure all your
; inbound and outbound calls in Asterisk.
;
; This configuration file is reloaded
; - With the "dialplan reload" command in the CLI
; - With the "reload" command (that reloads everything) in the CLI

;
; The "General" category is for certain variables.
;
[general]
;
; If static is set to no, or omitted, then the pbx_config will rewrite
; this file when extensions are modified.  Remember that all comments
; made in the file will be lost when that happens.
;
; XXX Not yet implemented XXX
;
static=yes
;
; if static=yes and writeprotect=no, you can save dial plan by
; CLI command "dial plan save" too
;
writeprotect=yes ;was no
;
; If autofallthrough is set, then if an extension runs out of
; things to do, it will terminate the call with BUSY, CONGESTION
; or HANGUP depending on Asterisk's best guess. This is the default.
;
; If autofallthrough is not set, then if an extension runs out of
; things to do, Asterisk will wait for a new extension to be dialed
; (this is the original behavior of Asterisk 1.0 and earlier).
;
;autofallthrough=no
;
;
;
; If extenpatternmatchnew is set (true, yes, etc), then a new algorithm that uses
; a Trie to find the best matching pattern is used. In dialplans
; with more than about 20-40 extensions in a single context, this
; new algorithm can provide a noticeable speedup.
; With 50 extensions, the speedup is 1.32x
```

```
; with 88 extensions, the speedup is 2.23x
; with 138 extensions, the speedup is 3.44x
; with 238 extensions, the speedup is 5.8x
; with 438 extensions, the speedup is 10.4x
; With 1000 extensions, the speedup is ~25x
; with 10,000 extensions, the speedup is 374x
; Basically, the new algorithm provides a flat response
; time, no matter the number of extensions.
;
; By default, the old pattern matcher is used.
;
; ****This is a new feature! *********************
; The new pattern matcher is for the brave, the bold, and
; the desperate. If you have large dialplans (more than about 50 extensions
; in a context), and/or high call volume, you might consider setting
; this value to "yes" !!
; Please, if you try this out, and are forced to return to the
; old pattern matcher, please report your reasons in a bug report
; on https://issues.asterisk.org. We have made good progress in providing
; something compatible with the old matcher; help us finish the job!
;
; This value can be switched at runtime using the cli command "dial plan set extenpatternmatchnew true"
; or "dial plan set extenpatternmatchnew false", so you can experiment to your hearts content.
;
;extenpatternmatchnew=no
;
; If clearglobalvars is set, global variables will be cleared
; and reparsed on a dialplan reload, or Asterisk reload.
;
; If clearglobalvars is not set, then global variables will persist
; through reloads, and even if deleted from the extensions.conf or
; one of its included files, will remain set to the previous value.
;
; NOTE: A complication sets in, if you put your global variables into
; the AEL file, instead of the extensions.conf file. With clearglobalvars
; set, a "reload" will often leave the globals vars cleared, because it
; is not unusual to have extensions.conf (which will have no globals)
; load after the extensions.ael file (where the global vars are stored).
; So, with "reload" in this particular situation, first the AEL file will
; clear and then set all the global vars, then, later, when the extensions.conf
; file is loaded, the global vars are all cleared, and then not set, because
; they are not stored in the extensions.conf file.
;
clearglobalvars=no
;
; User context is where entries from users.conf are registered.  The
; default value is 'default'
```

```
;
;userscontext=default
;
; You can include other config files, use the #include command
; (without the ';'). Note that this is different from the "include" command
; that includes contexts within other contexts. The #include command works
; in all asterisk configuration files.
;#include "filename.conf"
;#include <filename.conf>
;#include filename.conf
;
; You can execute a program or script that produces config files, and they
; will be inserted where you insert the #exec command. The #exec command
; works on all asterisk configuration files.  However, you will need to
; activate them within asterisk.conf with the "execincludes" option.  They
; are otherwise considered a security risk.
;#exec /opt/bin/build-extra-contexts.sh
;#exec /opt/bin/build-extra-contexts.sh --foo="bar"
;#exec </opt/bin/build-extra-contexts.sh --foo="bar">
;#exec "/opt/bin/build-extra-contexts.sh --foo=\"bar\""
;

; The "Globals" category contains global variables that can be referenced
; in the dialplan with the GLOBAL dialplan function:
; ${GLOBAL(VARIABLE)}
; ${${GLOBAL(VARIABLE)}} or ${text${GLOBAL(VARIABLE)}} or any hybrid
; Unix/Linux environmental variables can be reached with the ENV dial plan
; function: ${ENV(VARIABLE)}
;
[globals]
CONSOLE=Console/dsp                             ; Console interface for demo
;CONSOLE=DAHDI/1
;CONSOLE=Phone/phone0
IAXINFO=guest                                   ; IAXtel username/password
;IAXINFO=myuser:mypass
TRUNK=DAHDI/G2                                          ; Trunk interface
;
; Note the 'G2' in the TRUNK variable above. It specifies which group (defined
; in chan_dahdi.conf) to dial, i.e. group 2, and how to choose a channel to use
; in the specified group. The four possible options are:
;
; g: select the lowest-numbered non-busy DAHDI channel
;    (aka. ascending sequential hunt group).
; G: select the highest-numbered non-busy DAHDI channel
;    (aka. descending sequential hunt group).
; r: use a round-robin search, starting at the next highest channel than last
;    time (aka. ascending rotary hunt group).
```

```
; R: use a round-robin search, starting at the next lowest channel than last
;    time (aka. descending rotary hunt group).
;
TRUNKMSD=1                                   ; MSD digits to strip (usually 1 or 0)
;TRUNK=IAX2/user:pass@provider

;FREENUMDOMAIN=mydomain.com                  ; domain to send on outbound
                                             ; freenum calls (uses outbound-freenum
                                             ; context)


;
; WARNING WARNING WARNING WARNING
; If you load any other extension configuration engine, such as pbx_ael.so,
; your global variables may be overridden by that file.  Please take care to
; use only one location to set global variables, and you will likely save
; yourself a ton of grief.
; WARNING WARNING WARNING WARNING
;
; Any category other than "General" and "Globals" represent
; extension contexts, which are collections of extensions.
;
; Extension names may be numbers, letters, or combinations
; thereof. If an extension name is prefixed by a '_'
; character, it is interpreted as a pattern rather than a
; literal.  In patterns, some characters have special meanings:
;
;   X - any digit from 0-9
;   Z - any digit from 1-9
;   N - any digit from 2-9
;   [1235-9] - any digit in the brackets (in this example, 1,2,3,5,6,7,8,9)
;   . - wildcard, matches anything remaining (e.g. _9011. matches
;           anything starting with 9011 excluding 9011 itself)
;   ! - wildcard, causes the matching process to complete as soon as
;       it can unambiguously determine that no other matches are possible
;
; For example, the extension _NXXXXXX would match normal 7 digit dialings,
; while _1NXXNXXXXXX would represent an area code plus phone number
; preceded by a one.
;
; Each step of an extension is ordered by priority, which must always start
; with 1 to be considered a valid extension.  The priority "next" or "n" means
; the previous priority plus one, regardless of whether the previous priority
; was associated with the current extension or not.  The priority "same" or "s"
; means the same as the previously specified priority, again regardless of
; whether the previous entry was for the same extension.  Priorities may be
; immediately followed by a plus sign and another integer to add that amount
; (most useful with 's' or 'n').  Priorities may then also have an alias, or
```

```
; label, in parentheses after their name which can be used in goto situations.
;
; Contexts contain several lines, one for each step of each extension.  One may
; include another context in the current one as well, optionally with a date
; and time.  Included contexts are included in the order they are listed.
; Switches may also be included within a context.  The order of matching within
; a context is always exact extensions, pattern match extensions, includes, and
; switches.  Includes are always processed depth-first.  So for example, if you
; would like a switch "A" to match before context "B", simply put switch "A" in
; an included context "C", where "C" is included in your original context
; before "B".
;
;[context]
;exten => someexten,{priority|label{+|-}offset}[(alias)],application(arg1,arg2,...)
;
; Timing list for includes is
;
;   <time range>,<days of week>,<days of month>,<months>[,<timezone>]
;
; Note that ranges may be specified to wrap around the ends.  Also, minutes are
; fine-grained only down to the closest even minute.
;
;include => daytime,9:00-17:00,mon-fri,*,*
;include => weekend,*,sat-sun,*,*
;include => weeknights,17:02-8:58,mon-fri,*,*
;
; ignorepat can be used to instruct drivers to not cancel dialtone upon receipt
; of a particular pattern.  The most commonly used example is of course '9'
; like this:
;
;ignorepat => 9
;
; so that dialtone remains even after dialing a 9.  Please note that ignorepat
; only works with channels which receive dialtone from the PBX, such as DAHDI,
; Phone, and VPB.  Other channels, such as SIP and MGCP, which generate their
; own dialtone and converse with the PBX only after a number is complete, are
; generally unaffected by ignorepat (unless DISA or another method is used to
; generate a dialtone after answering the channel).
[record_studio]
; used to record prompts
exten => 205,1,Answer
exten => 205,n,Wait(2)
;exten => 205,n,Record(asterisk-recording%d:ulaw,,5) ; for 5 sec otherwise wait for # or hangup to stop recording
exten => 205,n,Record(TEST-recording%d:ulaw)
;exten => 205,n,Record(asterisk-recording%d)
exten => 205,n,Wait(2)
exten => 205,n,Playback(${RECORDED_FILE})
```

```
exten => 205,n,Wait(1)
exten => 205,n,Playback(${RECORDED_FILE})
exten => 205,n,Wait(1)
exten => 205,n,Hangup

exten => 305,1,Answer()
exten => 305,n,Wait(1)
exten => 305,n,Playback(quiet_1sec)
exten => 305,n,Wait(1)
exten => 305,n,Playback(welcome-recording1,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(4_GeneralQuestions-recording1,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(5_Complaints-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(9_to_Repeat_menu-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(customer_selected_GeneralQ-recording1,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(customer_selected_ComplaintsQ-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(number_1-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(number_2-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(number_3-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(number_4-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(number_5-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(number_6-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(number_7-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(number_8-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(number_9-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(number_0-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(number_star-recording1,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(number_pound-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Playback(you_are_now_in_complaintsQ-recording0,skip)
exten => 305,n,Wait(1)
```

```
exten => 305,n,Playback(you_are_now_in_generalQ-recording0,skip)
exten => 305,n,Wait(1)
exten => 305,n,Hangup

exten => 405,1,Answer()
exten => 405,n,Wait(1)
exten => 405,n,Playback(quiet_1sec)
exten => 405,n,Wait(1)
exten => 405,n,Playback(EnterYourSelection-recording1,skip)
exten => 405,n,Wait(1)
exten => 405,n,Playback(rep_not_available-recording1,skip)
exten => 405,n,Wait(1)
exten => 405,n,Playback(your_are_the_rep_in_general-recording0,skip)
exten => 405,n,Wait(1)
exten => 405,n,Playback(your_are_the_rep_in_complaintsQ-recording1,skip)
exten => 405,n,Wait(1)
exten => 405,n,Playback(you_are_removed_as_rep_from_general-recording0,skip)
exten => 405,n,Wait(1)
exten => 405,n,Playback(you_are_removed_as_rep_from_complaintsQ-recording0,skip)
exten => 405,n,Wait(1)
exten => 405,n,Playback(GoodBye-recording0,skip)
exten => 405,n,Wait(1)
exten => 405,n,Hangup


[onsip_softphones]
;[default]
exten => s,1,Answer()
exten => s,n,DumpChan() ; dumps all available vars for the given channel
exten => s,n,NoOp("Caller ID IS: ${CALLERID(number)}")  ; just for informational purposes
exten => s,n,Playback(demo-echotest) ; Let them know what's going on
exten => s,n,Echo                  ; Do the echo test
exten => s,n,Playback(demo-echodone) ; Let them know it's over


include => record_studio
include => Company1_QueueLoginLogout
; 1800 CALL to Company1
exten => _17034369338,1,Set(CHANNEL(musicclass)=moh)
exten => _17034369338,n,DumpChan() ; dumps all available vars for the given channel
exten => _17034369338,n,Answer()
exten => _17034369338,n,Playback(queue-callswaiting) ; Audio only
exten => _17034369338,n,NoOp("Caller ID is: ${CALLERID(number)}")  ; just for informational purposes
exten => _17034369338,n,Set(sipuri=${ENUMLOOKUP(${CALLERID(num)},,,,itrs.us)})
exten => _17034369338,n,NoOp("sipuri: ${sipuri}")  ; just for informational purposes
exten => _17034369338,n,GotoIf($["${sipuri}" = "" ]?VoiceQ_GenQuest_1,start,1:VideoQ_GenQuest_1,start,1)
;;exten => _17034369338,n,GotoIf($["${sipuri}" = "" ]?VideoQ_GenQuest_1,start,1:VoiceQ_GenQuest_1,start,1)
```

```
;exten => _17034369338,n,Goto(VideoQ_GenQuest_1,start,1)
exten => _17034369338,n,Hangup()

exten => _17034369339,1,NoOp("Caller ID IS: ${CALLERID(number)}")  ; just for informational purposes
exten => _17034369339,n,DumpChan() ; dumps all available vars for the given channel
exten => _17034369339,n,AGI(queue_feedback.agi)
exten => _17034369339,n,Playback(demo-echotest)  ; Let them know what's going on
exten => _17034369339,n,Echo                      ; Do the echo test
exten => _17034369339,n,Playback(demo-echodone)           ; Let them know it's over

[incoming_from_onsip_9338]
exten => s,1,Answer()
exten => s,n,DumpChan() ; dumps all available vars for the given channel
exten => s,n,NoOp("Caller ID IS: ${CALLERID(number)}")  ; just for informational purposes
exten => s,n,Playback(demo-echotest) ; Let them know what's going on
exten => s,n,Echo          ; Do the echo test
exten => s,n,Playback(demo-echodone) ; Let them know it's over

;exten => start,n,Set(CALLERID(num)=${CALLERID(num):0:40}) ; to cover for a bug that only allowed for 40 bytes
;exten => start,n,Set(CALLERID(name)=${CALLERID(name):0:40})
;exten => start,n,Set(ACTUALTO=sip:${CALLERID(num)})
;exten => start,n,Set(ACTUALFROM=sip:${EXTEN})
;exten => start,n,Macro(sendIMmacro,"You are in the (video) agent (sales) Queue. There are $[${QUEUECALLS}+1]
calls ahead of you. The average wait is about ${QUEUEHOLDTIME} minutes",${ACTUALTO},${ACTUALFROM})

;exten => s,n,Hangup()
; 1800 CALL to Company1
exten => _17034369338,1,Set(CHANNEL(musicclass)=moh)
exten => _17034369338,n,Answer()
;exten => _17034369338,n,Playback(queue-callswaiting) ; Audio only
exten => _17034369338,n,DumpChan() ; dumps all available vars for the given channel
exten => _17034369338,n,NoOp("Caller ID is: ${CALLERID(number)}")  ; just for informational purposes
exten => _17034369338,n,Set(sipuri=${ENUMLOOKUP(${CALLERID(num)},,,,itrs.us)})
exten => _17034369338,n,NoOp("sipuri: ${sipuri}")  ; just for informational purposes
exten => _17034369338,n,GotoIf($["${sipuri}" = ""
]?company1_voice_caller_query,start,2:company1_video_caller_query,start,2)
;exten => _17034369338,n,Goto(VoiceQ_GenQuest_1,start,1)
exten => _17034369338,n,Hangup()

; 1800 CALL to Company2
;exten => _17034369339,1,Set(CHANNEL(musicclass)=moh)
;exten => _17034369339,n,Answer()
;exten => _17034369339,n,Playback(queue-callswaiting) ; Audio only
;exten => _17034369339,n,NoOp("Caller ID is: ${CALLERID(number)}")  ; just for informational purposes
;exten => _17034369339,n,Goto(VoiceQ_company2,start,1)
;exten => _17034369339,n,Hangup()
```

```
exten => _17034369339,1,NoOp("Caller ID IS: ${CALLERID(number)}")  ; just for informational purposes
exten => _17034369339,n,DumpChan() ; dumps all available vars for the given channel
exten => _17034369339,n,Playback(demo-echotest)  ; Let them know what's going on
exten => _17034369339,n,Echo                              ; Do the echo test
exten => _17034369339,n,Playback(demo-echodone)             ; Let them know it's over

;exten => _X.,1,Playback(demo-echotest)      ; Let them know what's going on
;exten => _X.,n,NoOp("Caller ID IS: ${CALLERID(number)}")  ; just for informational purposes
;exten => _X.,n,DumpChan() ; dumps all available vars for the given channel
;exten => _X.,n,Echo                          ; Do the echo test
;exten => _X.,n,Playback(demo-echodone)    ; Let them know it's over
[incoming_from_onsip_9339]
exten => s,1,Answer()
exten => s,n,DumpChan() ; dumps all available vars for the given channel
exten => s,n,NoOp("Caller ID IS: ${CALLERID(number)}")  ; just for informational purposes
exten => s,n,Playback(demo-echotest) ; Let them know what's going on
exten => s,n,Echo                 ; Do the echo test
exten => s,n,Playback(demo-echodone) ; Let them know it's over

exten => _17034369338,1,Set(CHANNEL(musicclass)=moh)
exten => _17034369338,n,Answer()
exten => _17034369338,n,Playback(queue-callswaiting) ; Audio only
exten => _17034369338,n,DumpChan() ; dumps all available vars for the given channel
exten => _17034369338,n,Hangup() ; dumps all available vars for the given channel

exten => _17034369339,1,Set(CHANNEL(musicclass)=moh)
exten => _17034369339,n,Answer()
exten => _17034369339,n,Playback(queue-callswaiting) ; Audio only
exten => _17034369339,n,DumpChan() ; dumps all available vars for the given channel
exten => _17034369339,n,Hangup() ; dumps all available vars for the given channel

;*****************************************************************************
; Company1  703-436-9338
;*****************************************************************************

[company1]
;
; By default we include the demo.  In a production system, you
; probably don't want to have the demo there.
;
include => record_studio
;include => demo
exten => 600,1,NoOp("Caller ID IS: ${CALLERID(number)}")  ; just for informational purposes
exten => 600,n,DumpChan() ; dumps all available vars for the given channel
exten => 600,n,Playback(demo-echotest)      ; Let them know what's going on
exten => 600,n,Echo                          ; Do the echo test
exten => 600,n,Playback(demo-echodone)    ; Let them know it's over
```

```
;Handle internal callers
;exten => _70326557XX,1,Set(CHANNEL(musicclass)=music)
exten => _70326557XX,1,Dial(SIP/${EXTEN})
exten => _70326557XX,n,HangUp()

exten => _72426557XX,1,Dial(SIP/${EXTEN})
exten => _72426557XX,n,HangUp()

;***********************************************************
;******************** ACD Demo***********************
;***********************************************************
include => Company1_QueueLoginLogout
exten => _7034369338,1,Answer()
;exten => _7034369338,n,AGI(VideoEnabled_lookupphone.agi)
exten => _7034369338,n,DumpChan() ; dumps all available vars for the given channel
exten => _7034369338,n,NoOp("Caller ID is: 1${CALLERID(number)}")  ; just for informational purposes
exten => _7034369338,n,Set(sipuri=${ENUMLOOKUP(+1${CALLERID(num)},,,,itrs.us)})
exten => _7034369338,n,NoOp("sipuri: ${sipuri}")  ; just for informational purposes
exten => _7034369338,n,GotoIf($["${sipuri}" = ""
]?company1_voice_caller_query,start,2:company1_video_caller_query,start,2)
exten => _7034369338,n,Hangup()


; Z number
exten => _7035705868,1,Answer()
exten => _7035705868,n,DumpChan() ; dumps all available vars for the given channel
exten => _7035705868,n,NoOp("Caller ID is: 1${CALLERID(number)}")  ; just for informational purposes
exten => _7035705868,n,Set(sipuri=${ENUMLOOKUP(+1${CALLERID(num)},,,,itrs.us)})
exten => _7035705868,n,NoOp("sipuri: ${sipuri}")  ; just for informational purposes
exten => _7035705868,n,GotoIf($["${sipuri}" = ""
]?company1_voice_caller_query,start,2:company1_video_caller_query,start,2)
exten => _7035705868,n,Hangup()


; send all outgoing calls to onsip
exten => _1X.,1,Dial(SIP/000${EXTEN}@onsip-9338)
exten => 8000,1,Dial(SIP/${EXTEN}@onsip)-9338

; all users can call all companies
include => company2
exten => _7034369339,n,Goto(company2,_7034369339,1)

;******************************************************************************
; Company2  703-436-4339
;******************************************************************************
```

```
[company2]
exten => 600,1,NoOp("Caller ID IS: ${CALLERID(number)}")  ; just for informational purposes
exten => 600,n,DumpChan() ; dumps all available vars for the given channel
exten => 600,n,Playback(demo-echotest)  ; Let them know what's going on
exten => 600,n,Echo                  ; Do the echo test
exten => 600,n,Playback(demo-echodone)  ; Let them know it's over

;Handle internal callers
;exten => _72426557XX,1,Set(CHANNEL(musicclass)=music)
exten => _72426557XX,1,Dial(SIP/${EXTEN})
exten => _72426557XX,n,HangUp()

exten => _70326557XX,1,Dial(SIP/${EXTEN})
exten => _70326557XX,n,HangUp()

; send all outgoing calls to onsip
exten => _1X.,1,Dial(SIP/000${EXTEN}@onsip-9338)
exten => 8000,1,Dial(SIP/${EXTEN}@onsip)-9338

;***********************************************************
;********************* ACD Demo stuff ***********************
;***********************************************************
; all users can call all companies
include => company1
exten => _7034369338,n,Goto(company1,_7034369338,1)
; Company1 IVR
exten => _7035705868,n,Goto(company1,_7035705868,1)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
include => Company1_QueueLoginLogout
exten => _7034369339,1,Answer()
exten => _7034369339,n,Playback(queue-callswaiting) ; Audio only
exten => _7034369339,n,NoOp("Caller ID is: 1${CALLERID(number)}")  ; just for informational purposes
exten => _7034369339,n,Set(sipuri=${ENUMLOOKUP(+1${CALLERID(num)},,,,itrs.us)})
exten => _7034369339,n,NoOp("sipuri: ${sipuri}")  ; just for informational purposes
; TODO add the Company2 querys
;exten => _7034369339,n,GotoIf($["${sipuri}" = ""
]?company1_voice_caller_query,start,2:company1_video_caller_query,start,2)
exten => _7034369339,n,Hangup()



;***********************************************************
;********************* ACD Demo stuff ***********************
;***********************************************************
include => Company2_QueueLoginLogout
;exten => _7034369339,1,Set(CHANNEL(musicclass)=moh)
```

```
exten => _7034369339,1,Answer()
exten => _7034369339,n,Playback(queue-callswaiting) ; Audio only
;exten => _7034369339,n,AGI(VideoEnabled_lookupphone.agi)
;exten => _7034369339,n,NoOp("Caller ID is: 1${CALLERID(number)}")  ; just for informational purposes
exten => _7034369339,n,Set(sipuri=${ENUMLOOKUP(+1${CALLERID(num)},,,,itrs.us)})
;exten => _7034369339,n,NoOp("sipuri: ${sipuri}")  ; just for informational purposes
exten => _7034369339,n,GotoIf($["${sipuri}" = "" ]?VoiceQ_company2,start,1:VideoQ_company2,start,1)
exten => _7034369339,n,Hangup()

; send all outgoing calls to onsip
exten => _1X.,1,Dial(SIP/000${EXTEN}@onsip)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;	********** I V R for Z NUMBERS **********
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; play keypad hits
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
[play_key_for_video]
exten => num1,1,Playback(number_1-recording0)
exten => num1,n,Goto(company1_video_caller_query,start,100)
exten => num2,1,Playback(number_2-recording0)
exten => num2,n,Goto(company1_video_caller_query,start,100)
exten => num3,1,Playback(number_3-recording0)
exten => num3,n,Goto(company1_video_caller_query,start,100)
exten => num4,1,Playback(number_4-recording0)
exten => num4,n,Goto(company1_video_caller_query,start,100)
exten => num5,1,Playback(number_5-recording0)
exten => num5,n,Goto(company1_video_caller_query,start,100)
exten => num6,1,Playback(number_6-recording0)
exten => num6,n,Goto(company1_video_caller_query,start,100)
exten => num7,1,Playback(number_7-recording0)
exten => num7,n,Goto(company1_video_caller_query,start,100)
exten => num8,1,Playback(number_8-recording0)
exten => num8,n,Goto(company1_video_caller_query,start,100)
exten => num9,1,Playback(number_9-recording0)
exten => num9,n,Goto(company1_video_caller_query,start,100)
exten => num10,1,Playback(number_0-recording0)
exten => num10,n,Goto(company1_video_caller_query,start,100)
exten => num11,1,Playback(number_star-recording1)
exten => num11,n,Goto(company1_video_caller_query,start,100)
exten => num12,1,Playback(number_pound-recording0)
exten => num12,n,Goto(company1_video_caller_query,start,100)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
; IVR and determination for proper Video Q
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

[company1_video_caller_query]
exten => start,1,(start)
exten => start,n,Verbose(2,${CALLERID(num)} entering the Company1 skill query for proper queue.)
exten => start,n,Wait(3)
exten => start,n,Playback(quiet_1sec)
;exten => start,n,Goto(company1_video_caller_query,start,20)
exten => start,n,Goto(start,20)
exten => start,20,Playback(welcome-recording1)
exten => start,21,Playback(4_GeneralQuestions-recording1&5_Complaints-recording0&9_to_Repeat_menu-
recording0)

exten => start,n,SendImage(/var/lib/asterisk/images/asterisk-intro)
;exten => start,n,NoOp(SendImage(/var/lib/asterisk/images/asterisk-intro))
;exten => start,n,NoOp(image_rc = ${image_rc} )

exten => start,n,Goto(start,30)

;INSERT MESSAGE -WAITING FOR SELECTION
exten => start,30,Read(digito,,1,,5,5)
;exten => start,n,SayDigits(${digito})

; Echo the key press back to caller
exten => start,n,GotoIf($["${digito}" = "0" ]?play_key_for_video,num10,1)
exten => start,n,GotoIf($["${digito}" = "1" ]?play_key_for_video,num1,1)
exten => start,n,GotoIf($["${digito}" = "2" ]?play_key_for_video,num2,1)
exten => start,n,GotoIf($["${digito}" = "3" ]?play_key_for_video,num3,1)
exten => start,n,GotoIf($["${digito}" = "4" ]?play_key_for_video,num4,1)
exten => start,n,GotoIf($["${digito}" = "5" ]?play_key_for_video,num5,1)
exten => start,n,GotoIf($["${digito}" = "6" ]?play_key_for_video,num6,1)
exten => start,n,GotoIf($["${digito}" = "7" ]?play_key_for_video,num7,1)
exten => start,n,GotoIf($["${digito}" = "8" ]?play_key_for_video,num8,1)
exten => start,n,GotoIf($["${digito}" = "9" ]?play_key_for_video,num9,1)
exten => start,n,GotoIf($["${digito}" = "*" ]?play_key_for_video,num11,1)
exten => start,n,GotoIf($["${digito}" = " " ]?play_key_for_video,num12,1) ; was # but # shows up as "User entered
nothing" at read()

; Caller didnt push any key or an unsupported DTMF was sent
; So we just jump to the next point in this IVR
exten => start,n,Goto(start,100)

; Pointer to where we want to return to in this IVR context
exten => start,100,NoOp(Return from play_key_for_video)
exten => start,n,NoOp(NativeFormat = ${CHANNEL(videonativeformat)})
```

```
; handle all DTMF inputs for the IVR
exten => start,n,GotoIf($["${digito}" = "*" ]?bye,1)
exten => start,n,GotoIf($["${digito}" = "9" ]?start,21)
; Handle invalid options
exten => start,n,GotoIf($["${digito}" < "4" ]?start,30)
exten => start,n,GotoIf($["${digito}" > "5" ]?start,30)
; Handle valid options
exten => start,n,GotoIf($["${digito}" = "4" ]?VideoQ_GenQuest_1,start,1)
exten => start,n,GotoIf($["${digito}" = "5" ]?VideoQ_Complaint_1,start,1)

; If we made it here, then its bye bye
exten => start,n,Goto(bye,1)

; Goodbye
exten => bye,1,Playback(quiet_1sec)
exten => bye,2,Playback(GoodBye-recording0)
exten => bye,3,Hangup()

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; play keypad hits
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
[play_key_for_voice]
exten => num1,1,Playback(number_1-recording0)
exten => num1,n,Goto(company1_voice_caller_query,start,100)
exten => num2,1,Playback(number_2-recording0)
exten => num2,n,Goto(company1_voice_caller_query,start,100)
exten => num3,1,Playback(number_3-recording0)
exten => num3,n,Goto(company1_voice_caller_query,start,100)
exten => num4,1,Playback(number_4-recording0)
exten => num4,n,Goto(company1_voice_caller_query,start,100)
exten => num5,1,Playback(number_5-recording0)
exten => num5,n,Goto(company1_voice_caller_query,start,100)
exten => num6,1,Playback(number_6-recording0)
exten => num6,n,Goto(company1_voice_caller_query,start,100)
exten => num7,1,Playback(number_7-recording0)
exten => num7,n,Goto(company1_voice_caller_query,start,100)
exten => num8,1,Playback(number_8-recording0)
exten => num8,n,Goto(company1_voice_caller_query,start,100)
exten => num9,1,Playback(number_9-recording0)
exten => num9,n,Goto(company1_voice_caller_query,start,100)
exten => num10,1,Playback(number_0-recording0)
exten => num10,n,Goto(company1_voice_caller_query,start,100)
exten => num11,1,Playback(number_star-recording1)
exten => num11,n,Goto(company1_voice_caller_query,start,100)
exten => num12,1,Playback(number_pound-recording0)
exten => num12,n,Goto(company1_voice_caller_query,start,100)
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; IVR and determination for proper Voice Q
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

[company1_voice_caller_query]
exten => start,1,(start)
exten => start,n,Verbose(2,${CALLERID(num)} entering the Company1 skill query for proper queue.)
;exten => start,n,Wait(3)
;exten => start,n,Playback(quiet_1sec)
;exten => start,n,Goto(company1_video_caller_query,start,20)
exten => start,n,Goto(start,20)
exten => start,20,Playback(welcome-recording1)
exten => start,21,Playback(4_GeneralQuestions-recording1&5_Complaints-recording0&9_to_Repeat_menu-
recording0)

exten => start,n,SendImage(/var/lib/asterisk/images/asterisk-intro)

exten => start,n,Goto(start,30)

;INSERT MESSAGE -WAITING FOR SELECTION
exten => start,30,Read(digito,,1,,5,5)
;exten => start,n,SayDigits(${digito})

; Echo the key press back to caller
exten => start,n,GotoIf($["${digito}" = "0" ]?play_key_for_voice,num10,1)
exten => start,n,GotoIf($["${digito}" = "1" ]?play_key_for_voice,num1,1)
exten => start,n,GotoIf($["${digito}" = "2" ]?play_key_for_voice,num2,1)
exten => start,n,GotoIf($["${digito}" = "3" ]?play_key_for_voice,num3,1)
exten => start,n,GotoIf($["${digito}" = "4" ]?play_key_for_voice,num4,1)
exten => start,n,GotoIf($["${digito}" = "5" ]?play_key_for_voice,num5,1)
exten => start,n,GotoIf($["${digito}" = "6" ]?play_key_for_voice,num6,1)
exten => start,n,GotoIf($["${digito}" = "7" ]?play_key_for_voice,num7,1)
exten => start,n,GotoIf($["${digito}" = "8" ]?play_key_for_voice,num8,1)
exten => start,n,GotoIf($["${digito}" = "9" ]?play_key_for_voice,num9,1)
exten => start,n,GotoIf($["${digito}" = "*" ]?play_key_for_voice,num11,1)
exten => start,n,GotoIf($["${digito}" = " " ]?play_key_for_voice,num12,1) ; was # but # shows up as "User entered
nothing" at read()

; Caller didnt push any key or an unsupported DTMF was sent
; So we just jump to the next point in this IVR
exten => start,n,Goto(start,100)

; Pointer to where we want to return to in this IVR context
exten => start,100,NoOp(Return from play_key_for_voice)

; handle all DTMF inputs for the IVR
exten => start,n,GotoIf($["${digito}" = "*" ]?bye,1)
```

```
exten => start,n,GotoIf($["${digito}" = "9" ]?start,21)
; Handle invalid options
exten => start,n,GotoIf($["${digito}" < "4" ]?start,30)
exten => start,n,GotoIf($["${digito}" > "5" ]?start,30)
; Handle valid options
exten => start,n,GotoIf($["${digito}" = "4" ]?VoiceQ_GenQuest_1,start,1)
exten => start,n,GotoIf($["${digito}" = "5" ]?VoiceQ_Complaint_1,start,1)

; If we made it here, then its bye bye
exten => start,n,Goto(bye,1)

; Goodbye
exten => bye,1,Playback(goodbye)
exten => bye,2,Playback(goodbye-recording0)
exten => bye,3,Hangup()

;*************
; Company1 Q stuff
;*************
; VIDEO Queue -1
[VideoQ_GenQuest_1]
exten => start,1,Verbose(2,${CALLERID(num)} entering the VIDEO queue)
exten => start,n,Set(qinfo=${QUEUE_VARIABLES(VideoQ_GenQuest_1)}) ; get the QUEUE information. returns 0 f
successful
exten => start,n,Set(CALLERID(num)=${CALLERID(num):0:40}) ; to cover for a bug that only allowed for 40 bytes
exten => start,n,Set(CALLERID(name)=${CALLERID(name):0:40})
exten => start,n,Set(ACTUALTO=sip:${CALLERID(num)})
exten => start,n,Set(ACTUALFROM=sip:${EXTEN})
exten => start,n,Macro(sendIMmacro,"You are in the General Questions (video) Queue. There are
$[${QUEUECALLS}+1] calls ahead of you. The average wait is about ${QUEUEHOLDTIME}
minutes",${ACTUALTO},${ACTUALFROM})

; load up the variables that will be accessed from the queue app by he macro that is passed
exten => start,n,Set(_MYARG1="You are now connected to an (video) agent who can handle your questions.
Thank you.")
exten => start,n,Set(_MYARG2=${ACTUALTO})
exten => start,n,Set(_MYARG3=${ACTUALFROM})

;VIDEO notice -Welcome and instructions for Callback and more
exten => start,n,Playback(you_are_now_in_generalQ-recording0) ;,skip
;exten => start,n,Background(asterisk205-recording1) ;,skip

; execute the queue and pass the macro
exten => start,n,Queue(VideoQ_GenQuest_1,t,,,,,sendIM_Q_macro,video-gosub)

; in case no agent is registered with the Q
```

```
exten => start,n,Macro(sendIMmacro,"Sorry. No one is available to take your call.
Goodbye.",${ACTUALTO},${ACTUALFROM})

;VIDEO notice -No agent available
exten => start,n,Playback(rep_not_available-recording1) ;,skip

exten => start, n,Playback(vm-nobodyavail)
exten => start, n,Playback(vm-goodbye)
exten => start, n,Hangup()
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; VIDEO Queue -2
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
[VideoQ_Complaint_1]
exten => start,1,Verbose(2,${CALLERID(num)} entering the VIDEO queue)
exten => start,n,Set(qinfo=${QUEUE_VARIABLES(VideoQ_GenQuest_1)}) ; get the QUEUE information. returns 0 f
successful
exten => start,n,Set(CALLERID(num)=${CALLERID(num):0:40}) ; to cover for a bug that only allowed for 40 bytes
exten => start,n,Set(CALLERID(name)=${CALLERID(name):0:40})
exten => start,n,Set(ACTUALTO=sip:${CALLERID(num)})
exten => start,n,Set(ACTUALFROM=sip:${EXTEN})
exten => start,n,Macro(sendIMmacro,"You are in the Complaints (video) Queue. There are $[${QUEUECALLS}+1]
calls ahead of you. The average wait is about ${QUEUEHOLDTIME} minutes",${ACTUALTO},${ACTUALFROM})

; load up the variables that will be accessed from the queue app by he macro that is passed
exten => start,n,Set(_MYARG1="You are now connected to a (video) agent who can handle your call.  Thank you.")
exten => start,n,Set(_MYARG2=${ACTUALTO})
exten => start,n,Set(_MYARG3=${ACTUALFROM})

;VIDEO notice -Welcome and instructions for Callback and more
exten => start,n,Playback(you_are_now_in_complaintsQ-recording0) ;,skip
;exten => start,n,Background(asterisk205-recording1) ;,skip

; execute the queue and pass the macro
exten => start,n,Queue(VideoQ_Complaint_1,t,,,,,sendIM_Q_macro,video2-gosub)

; in case no agent is registered with the Q
exten => start,n,Macro(sendIMmacro,"Sorry. No one is available to take your call.
Goodbye.",${ACTUALTO},${ACTUALFROM})

;VIDEO notice -No agent available
exten => start,n,Playback(rep_not_available-recording1) ;,skip

exten => start, n,Playback(vm-nobodyavail)
exten => start, n,Playback(vm-goodbye)
exten => start, n,Hangup()

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
; GOSUB that are passed to the queue application calls
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
[video-gosub]
exten => s,1,Verbose("Here we are in a subroutine GeneralQ! Playback audio and video")
;exten => s,n,Playback(asterisk205-recording3)
exten => s,n,Answer()
exten => s,n,Wait(3)
exten => s,n,Playback(customer_selected_GeneralQ-recording1)
;exten => s,n,Background(asterisk205-recording0)
exten => s,n,Return()

[video2-gosub]
exten => s,1,Verbose("Here we are in a subroutine ComplaintsQ! Playback audio and video")
;exten => s,n,Playback(asterisk205-recording3)
exten => s,n,Wait(3)
exten => s,n,Playback(customer_selected_ComplaintsQ-recording0)
;exten => s,n,Background(asterisk205-recording1)
exten => s,n,Return()


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;The following uses Variable Inheritance preced the var with one
;or two underbar chars.
;reference:  https://wiki.asterisk.org/wiki/display/AST/Variable+Inheritance
;n,DumpChan() ; dumps all available vars for the given channel
;exten => start,n,DumpChan() ; dumps all available vars for the given channel

;QUEUEHOLDTIME for time expected in queue
; NonVIDEO Queue
[VoiceQ_GenQuest_1]
exten => start,1,Verbose(2,${CALLERID(num)} entering the Non-VIDEO queue)
exten => start,n,Set(qinfo=${QUEUE_VARIABLES(VoiceQ_GenQuest_1)}) ; get the QUEUE information. returns 0 f
successful
exten => start,n,Set(CALLERID(num)=${CALLERID(num):0:40}) ; to cover for a bug that only allowed for 40 bytes
exten => start,n,Set(CALLERID(name)=${CALLERID(name):0:40})
exten => start,n,Set(ACTUALTO=sip:${CALLERID(num)})
exten => start,n,Set(ACTUALFROM=sip:${EXTEN})
exten => start,n,Macro(sendIMmacro,"You are in the General Questions Queue. There are $[${QUEUECALLS}+1]
calls ahead of you. The average wait is about ${QUEUEHOLDTIME} minutes",${ACTUALTO},${ACTUALFROM})
; load up the variables that will be accessed from the queue app by he macro that is passed
exten => start,n,Set(_MYARG1="You are now connected to an agent who can handle your call.  Thank you.")
exten => start,n,Set(_MYARG2=${ACTUALTO})
exten => start,n,Set(_MYARG3=${ACTUALFROM})
; execute the queue and pass the macro
exten => start,n,Queue(VoiceQ_GenQuest_1,,,,,,sendIM_Q_macro,video-gosub)
; in case no agent is registered with the Q
```

```
exten => start,n,Macro(sendIMmacro,"Sorry. No one is available to take your call.
Goodbye.",${ACTUALTO},${ACTUALFROM})
exten => start,n,Playback(vm-nobodyavail)
exten => start,n,Playback(vm-goodbye)
exten => start,n,Hangup()

[VoiceQ_Complaint_1]
exten => start,1,Verbose(2,${CALLERID(num)} entering the Non-VIDEO VoiceQ_Complaint_1 queue)
exten => start,n,Set(qinfo=${QUEUE_VARIABLES(VoiceQ_GenQuest_1)}) ; get the QUEUE information. returns 0 f
successful
exten => start,n,Set(CALLERID(num)=${CALLERID(num):0:40}) ; to cover for a bug that only allowed for 40 bytes
exten => start,n,Set(CALLERID(name)=${CALLERID(name):0:40})
exten => start,n,Set(ACTUALTO=sip:${CALLERID(num)})
exten => start,n,Set(ACTUALFROM=sip:${EXTEN})
exten => start,n,Macro(sendIMmacro,"You are in the Complaints Queue. There are $[${QUEUECALLS}+1] calls
ahead of you. The average wait is about ${QUEUEHOLDTIME} minutes",${ACTUALTO},${ACTUALFROM})
; load up the variables that will be accessed from the queue app by he macro that is passed
exten => start,n,Set(_MYARG1="You are now connected to an agent who can handle your call.  Thank you.")
exten => start,n,Set(_MYARG2=${ACTUALTO})
exten => start,n,Set(_MYARG3=${ACTUALFROM})
; execute the queue and pass the macro
exten => start,n,Queue(VoiceQ_Complaint_1,,,,,,sendIM_Q_macro,video2-gosub)
; in case no agent is registered with the Q
exten => start,n,Macro(sendIMmacro,"Sorry. No one is available to take your call.
Goodbye.",${ACTUALTO},${ACTUALFROM})
exten => start,n,Playback(vm-nobodyavail)
exten => start,n,Playback(vm-goodbye)
exten => start,n,Hangup()

;***************
; Company2 Q stuff
;***************
; VIDEO Queue
[VideoQ_company2]
exten => start,1,Verbose(2,${CALLERID(num)} entering the VIDEO queue)
exten => start,n,Set(qinfo=${QUEUE_VARIABLES(VideoQ_GenQuest_1)}) ; get the QUEUE information. returns 0 f
successful
exten => start,n,Set(CALLERID(num)=${CALLERID(num):0:40}) ; to cover for a bug that only allowed for 40 bytes
exten => start,n,Set(CALLERID(name)=${CALLERID(name):0:40})
exten => start,n,Set(ACTUALTO=sip:${CALLERID(num)})
exten => start,n,Set(ACTUALFROM=sip:${EXTEN})
exten => start,n,Macro(sendIMmacro,"You are in the (video) agent Queue. There are $[${QUEUECALLS}+1] calls
ahead of you. The average wait is about ${QUEUEHOLDTIME} minutes",${ACTUALTO},${ACTUALFROM})
; load up the variables that will be accessed from the queue app by he macro that is passed
exten => start,n,Set(_MYARG1="You are now connected to an (video) agent who can handle your call.  Thank
you.")
exten => start,n,Set(_MYARG2=${ACTUALTO})
```

```
exten => start,n,Set(_MYARG3=${ACTUALFROM})
; execute the queue and pass the macro
exten => start,n,Queue(VideoQ_company2,,,,,,sendIM_Q_macro)
; in case no agent is registered with the Q
exten => start,n,Macro(sendIMmacro,"Sorry. No one is available to take your call.
Goodbye.",${ACTUALTO},${ACTUALFROM})
exten => start, n,Playback(vm-nobodyavail)
exten => start, n,Playback(vm-goodbye)
exten => start, n,Hangup()


;The following uses Variable Inheritance preced the var with one
;or two underbar chars.
;reference:  https://wiki.asterisk.org/wiki/display/AST/Variable+Inheritance
;n,DumpChan() ; dumps all available vars for the given channel
;exten => start,n,DumpChan() ; dumps all available vars for the given channel

;QUEUEHOLDTIME for time expected in queue
; NonVIDEO Queue
[VoiceQ_company2]
exten => start,1,Verbose(2,${CALLERID(num)} entering the Non-VIDEO queue)
exten => start,n,Set(qinfo=${QUEUE_VARIABLES(VoiceQ_GenQuest_1)}) ; get the QUEUE information. returns 0 f
successful
exten => start,n,Set(CALLERID(num)=${CALLERID(num):0:40}) ; to cover for a bug that only allowed for 40 bytes
exten => start,n,Set(CALLERID(name)=${CALLERID(name):0:40})
exten => start,n,Set(ACTUALTO=sip:${CALLERID(num)})
exten => start,n,Set(ACTUALFROM=sip:${EXTEN})
exten => start,n,Macro(sendIMmacro,"You are in the agent Queue. There are $[${QUEUECALLS}+1] calls ahead of
you. The average wait is about ${QUEUEHOLDTIME} minutes",${ACTUALTO},${ACTUALFROM})
; load up the variables that will be accessed from the queue app by he macro that is passed
exten => start,n,Set(_MYARG1="You are now connected to an agent who can handle your call.  Thank you.")
exten => start,n,Set(_MYARG2=${ACTUALTO})
exten => start,n,Set(_MYARG3=${ACTUALFROM})
; execute the queue and pass the macro
exten => start,n,Queue(VoiceQ_company2,,,,,,sendIM_Q_macro)
; in case no agent is registered with the Q
exten => start,n,Macro(sendIMmacro,"Sorry. No one is available to take your call.
Goodbye.",${ACTUALTO},${ACTUALFROM})
exten => start,n,Playback(vm-nobodyavail)
exten => start,n,Playback(vm-goodbye)
exten => start,n,Hangup()


[macro-sendIM_Q_macro]
exten => s,1,Set(MESSAGE(body)=${MYARG1})
exten => s,n,Macro(sendIMmacro,${MYARG1},${MYARG2},${MYARG3})
```

```
[macro-sendIMmacro]
;ARG1 -message to send
;ARG2 -SIP recipient
;ARG3 -SIP sender
exten => s,1,Set(MESSAGE(body)=${ARG1})
exten => s,n,MessageSend(${ARG2},${ARG3})

;***************************
;Company1 agent registration
;***************************
[Company1_QueueLoginLogout]
exten => *10,1,Verbose(2,Logging in skill-1 VIDEO General queue member)
exten => *10,n,Set(MemberChannel=${CHANNEL(channeltype)}/${CALLERID(num)})
exten => *10,n,AddQueueMember(VideoQ_GenQuest_1,${MemberChannel})
exten => *10,n,Wait(2)
exten => *10,n,Playback(quiet_1sec)
exten => *10,n,Playback(your_are_the_rep_in_general-recording0) ;,skip
exten => *10,n,Hangup()

exten => *15,1,Verbose(2,Logging out skill-1 VIDEO General queue member)
exten => *15,n,Set(MemberChannel=${CHANNEL(channeltype)}/${CALLERID(num)})
exten => *15,n,RemoveQueueMember(VideoQ_GenQuest_1,${MemberChannel})
exten => *15,n,Wait(2)
exten => *15,n,Playback(quiet_1sec)
exten => *15,n,Playback(you_are_removed_as_rep_from_general-recording0) ;,skip
exten => *15,n,Hangup()


exten => *20,1,Verbose(2,Logging in VIDEO Compalints queue member)
exten => *20,n,Set(MemberChannel=${CHANNEL(channeltype)}/${CALLERID(num)})
exten => *20,n,AddQueueMember(VideoQ_Complaint_1,${MemberChannel})
exten => *20,n,Wait(2)
exten => *20,n,Playback(quiet_1sec)
exten => *20,n,Playback(your_are_the_rep_in_complaintsQ-recording1) ;,skip
exten => *20,n,Hangup()

exten => *25,1,Verbose(2,Logging out VIDEO Complaints queue member)
exten => *25,n,Set(MemberChannel=${CHANNEL(channeltype)}/${CALLERID(num)})
exten => *25,n,RemoveQueueMember(VideoQ_Complaint_1,${MemberChannel})
exten => *25,n,Wait(2)
exten => *25,n,Playback(quiet_1sec)
exten => *25,n,Playback(you_are_removed_as_rep_from_complaintsQ-recording0) ;,skip
exten => *25,n,Hangup()

; voice only agents
exten => *50,1,Verbose(2,Logging in Non-VIDEO General queue member)
exten => *50,n,Set(MemberChannel=${CHANNEL(channeltype)}/${CHANNEL(peername)})
exten => *50,n,AddQueueMember(VoiceQ_GenQuest_1,${MemberChannel})
```

```
exten => *50,n,Wait(1)
exten => *50,n,Playback(your_are_the_rep_in_general-recording0) ;,skip
exten => *50,n,Hangup()

exten => *55,1,Verbose(2,Logging out Non-VIDEO General queue member)
exten => *55,n,Set(MemberChannel=${CHANNEL(channeltype)}/${CHANNEL(peername)})
exten => *55,n,RemoveQueueMember(VoiceQ_GenQuest_1,${MemberChannel})
exten => *55,n,Wait(1)
exten => *55,n,Playback(you_are_removed_as_rep_from_general-recording0) ;,skip
exten => *55,n,Hangup()

exten => *60,1,Verbose(2,Logging in Non-VIDEO Complaint queue member)
exten => *60,n,Set(MemberChannel=${CHANNEL(channeltype)}/${CHANNEL(peername)})
exten => *60,n,AddQueueMember(VoiceQ_Complaint_1,${MemberChannel})
exten => *60,n,Wait(1)
exten => *60,n,Playback(your_are_the_rep_in_complaintsQ-recording1) ;,skip
exten => *60,n,Hangup()

exten => *65,1,Verbose(2,Logging out Non-VIDEO Complaint queue member)
exten => *65,n,Set(MemberChannel=${CHANNEL(channeltype)}/${CHANNEL(peername)})
exten => *65,n,RemoveQueueMember(VoiceQ_Complaint_1,${MemberChannel})
exten => *65,n,Wait(1)
exten => *65,n,Playback(you_are_removed_as_rep_from_complaintsQ-recording0) ;,skip
exten => *65,n,Hangup()


;***************************
;Company2 agent registration
;***************************
[Company2_QueueLoginLogout]
exten => *50,1,Verbose(2,Logging in VIDEO queue member)
exten => *50,n,Set(MemberChannel=${CHANNEL(channeltype)}/${CALLERID(num)})
exten => *50,n,AddQueueMember(VideoQ_company2,${MemberChannel})
exten => *50,n,Hangup()

exten => *55,1,Verbose(2,Logging out VIDEO queue member)
exten => *55,n,Set(MemberChannel=${CHANNEL(channeltype)}/${CALLERID(num)})
exten => *55,n,RemoveQueueMember(VideoQ_company2,${MemberChannel})
exten => *55,n,Hangup()

exten => *60,1,Verbose(2,Logging in Non-VIDEO queue member)
exten => *60,n,Set(MemberChannel=${CHANNEL(channeltype)}/${CHANNEL(peername)})
exten => *60,n,AddQueueMember(VoiceQ_company2,${MemberChannel})

exten => *65,1,Verbose(2,Logging out Non-VIDEO queue member)
exten => *65,n,Set(MemberChannel=${CHANNEL(channeltype)}/${CHANNEL(peername)})
exten => *65,n,RemoveQueueMember(VoiceQ_company2,${MemberChannel})
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; June demonstration with Z PROVIDER
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
[incoming_from_provider]
; all users can call all companies
include => VideoQ_GenQuest_1
exten => _7035705868,1,Answer()
exten => _7035705868,n,DumpChan() ; dumps all available vars for the given channel
exten => _7035705868,n,Goto(company1_video_caller_query,start,2)
; all users can call all companies
include => VideoQ_company2
;exten => _7035705174,1,Goto(VideoQ_company2,start,1)
;TEST the IVR Q skill-based
exten => _7035705174,n,Goto(company1_video_caller_query,start,2)

;TEST to QUEUE
;exten => _7035705174,1,Dial(SIP/0008000@onsip)
```

Appendix C. SIP.CONF

```
[general]
;***************************************************
; ACD demo- support the SIP trunk to our OnSIP server
;***************************************************
register => vmdigioia_jcloud861@vmdigioia.onsip.com:xyz:vmdigioia_jcloud861@sip.onsip.com
register => jcloud861.6@vmdigioia.onsip.com:xyz:vmdigioia_jcloud861_6@sip.onsip.com
;register => prod-vcs-11.dmz.reston.champvrs.com::prod-vcs-11.dmz.reston.champvrs.com

;context=default          ; Default context for incoming calls. Defaults to 'default'
                          ; 'username' field from the authentication line
                          ; instead of the From: field.
allowoverlap=no           ; Disable overlap dialing support. (Default is yes)

udpbindaddr=0.0.0.0       ; IP address to bind UDP listen socket to (0.0.0.0 binds to all)
                          ; Optionally add a port number, 192.168.1.1:5062 (default is port 5060)
realm=54.86.14.195

tcpenable=yes             ; Enable server for incoming TCP connections (default is no)
tcpbindaddr=0.0.0.0       ; IP address for TCP server to bind to (0.0.0.0 binds to all interfaces)
                          ; Optionally add a port number, 192.168.1.1:5062 (default is port 5060)

;transport=tcp,udp,ws,wss,tls          ; Set the default transports.  The order determines the primary default
transport.
;transport=tcp,udp,ws,wss          ; Set the default transports.  The order determines the primary default
transport.
transport=tcp,udp         ; Set the default transports.  The order determines the primary default transport.
                          ; If tcpenable=no and the transport set is tcp, we will fallback to UDP.
srvlookup=yes             ; Enable DNS SRV lookups on outbound calls
                          ; Note: Asterisk only uses the first host
                          ; in SRV records
                          ; Disabling DNS SRV lookups disables the
                          ; ability to place SIP calls based on domain
                          ; names to some other SIP users on the Internet
                          ; Specifying a port in a SIP peer definition or
                          ; when dialing outbound calls will supress SRV
                          ; lookups for that peer or call.

localnet=172.31.16.0/255.255.240.0 ; RFC 1918 addresses
externaddr=54.86.14.195   ; use this address.
;nat=force_rport,comedia
nat=force_rport

[authentication]

[basic-options](!)        ; a template
```

```
    dtmfmode=rfc2833
    context=from-office
    type=friend

[natted-phone](!,basic-options)   ; another template inheriting basic-options
    directmedia=no
    host=dynamic

[public-phone](!,basic-options)   ; another template inheriting basic-options
    directmedia=no

[my-codecs](!)                  ; a template for my preferred codecs
    disallow=all
    allow=ilbc
    allow=g729
    allow=gsm
    allow=g723
    allow=ulaw
    ; Or, more simply:
    ;allow=!all,ilbc,g729,gsm,g723,ulaw

[ulaw-phone](!)                 ; and another one for ulaw-only
    disallow=all
    allow=ulaw
    ; Again, more simply:
    ;allow=!all,ulaw

;***************************************************
; ACD demo- support the SIP trunk to our OnSIP server
;***************************************************


[company1_phone](!)               ; and another one for ulaw-only
; Turn off silence suppression in X-Lite ("Transmit Silence"=YES)!
; Note that Xlite sends NAT keep-alive packets, so qualify=yes is not needed
        ;media_address = 192.168.100.200 ; this forces the rtp to use this address for the server
        ;type=friend
        type=peer
        transport=udp
        secret=moie
        host=dynamic                  ; This device needs to register
        directmedia=no                ; Typically set to NO if behind NAT
        disallow=all
        allow=all
;allow=gsm                    ; GSM consumes far less bandwidth than ulaw
        allow=ulaw
        ;allow=alaw
```

```
;	;allow=h261
	allow=h263
	allow=h263p
	allow=vp8
	allow=h264
	registertrying=yes		; Send a 100 Trying when the device registers.
	context=company1
	callcounter = yes		; Enable call counters on devices. This can be set per
	qualify=yes
	busydetect=yes
	videosupport=yes		; Turn on support for SIP video. You need to turn this
	dtmfmode=rfc2833
	; THE FOLLOWING IS SUPPORT FOR INSTANT MESSAGING
	accept_outofcall_message=yes
	outofcall_message_context=IM
	auth_message_requests=yes

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

[7032655700](company1_phone)
regexten=7032655700		; When they register, create extension 1234
callerid=<7032655700>

[7032655701](company1_phone)
regexten=7032655701		; When they register, create extension 1234
callerid=<7032655701>

[7032655702](company1_phone)
regexten=7032655702		; When they register, create extension 1234
callerid=<7032655702>

[7032655703](company1_phone)
regexten=7032655703		; When they register, create extension 1234
callerid=<7032655703>

[7032655704](company1_phone)
regexten=7032655704		; When they register, create extension 1234
callerid=<7032655704>

[7032655705](company1_phone)
regexten=7032655705		; When they register, create extension 1234
callerid=<7032655705>

[7032655706](company1_phone)
regexten=7032655706		; When they register, create extension 1234
callerid=<7032655706>
```

```
[7032655707](company1_phone)
regexten=7032655707           ; When they register, create extension 1234
callerid=<7032655707>

[7032655708](company1_phone)
regexten=7032655708           ; When they register, create extension 1234
callerid=<7032655708>

[7032655709](company1_phone)
regexten=7032655709           ; When they register, create extension 1234
callerid=<7032655709>

[7032655710](company1_phone)
regexten=7032655710           ; When they register, create extension 1234
callerid=<7032655710>

[7032655711](company1_phone)
regexten=7032655711           ; When they register, create extension 1234
callerid=<7032655711>

[7032655712](company1_phone)
regexten=7032655712           ; When they register, create extension 1234
callerid=<7032655712>

[7032655713](company1_phone)
regexten=7032655713           ; When they register, create extension 1234
callerid=<7032655713>

[7032655714](company1_phone)
regexten=7032655714           ; When they register, create extension 1234
callerid=<7032655714>

[7032655715](company1_phone)
regexten=7032655715           ; When they register, create extension 1234
callerid=<7032655715>

[7032655716](company1_phone)
regexten=7032655716           ; When they register, create extension 1234
callerid=<7032655716>

[7032655717](company1_phone)
regexten=7032655717           ; When they register, create extension 1234
callerid=<7032655717>

[7032655718](company1_phone)
regexten=7032655718           ; When they register, create extension 1234
callerid=<7032655718>
```

```
[7032655719](company1_phone)
regexten=7032655719             ; When they register, create extension 1234
callerid=<7032655719>

[7032655720](company1_phone)
regexten=7032655720             ; When they register, create extension 1234
callerid=<7032655720>

[7032655721](company1_phone)
regexten=7032655721             ; When they register, create extension 1234
callerid=<7032655721>

[7032655722](company1_phone)
regexten=7032655722             ; When they register, create extension 1234
callerid=<7032655722>

[7032655723](company1_phone)
regexten=7032655723             ; When they register, create extension 1234
callerid=<7032655723>

[7032655724](company1_phone)
regexten=7032655724             ; When they register, create extension 1234
callerid=<7032655724>

[7032655725](company1_phone)
regexten=7032655725             ; When they register, create extension 1234
callerid=<7032655725>

[company2_phone](!)             ; and another one for ulaw-only
; Turn off silence suppression in X-Lite ("Transmit Silence"=YES)!
; Note that Xlite sends NAT keep-alive packets, so qualify=yes is not needed
        ;media_address = 192.168.100.200 ; this forces the rtp to use this address for the server
        ;type=friend
        type=peer
        transport=udp
        secret=moie
        host=dynamic            ; This device needs to register
        directmedia=no          ; Typically set to NO if behind NAT
        disallow=all
;       allow=all
        ;allow=gsm              ; GSM consumes far less bandwidth than ulaw
        allow=ulaw
        ;allow=alaw
;       ;allow=h261
        allow=h263
        allow=h263p
```

```
    allow=vp8
    allow=h264
    registertrying=yes         ; Send a 100 Trying when the device registers.
    context=company2
    callcounter = yes          ; Enable call counters on devices. This can be set per
    qualify=yes
    busydetect=yes
    videosupport=yes           ; Turn on support for SIP video. You need to turn this
    dtmfmode=rfc2833
    ; THE FOLLOWING IS SUPPORT FOR INSTANT MESSAGING
    accept_outofcall_message=yes
    outofcall_message_context=IM
    auth_message_requests=yes

[7242655700](company2_phone)
regexten=7242655700
callerid=<7242655700>

[7242655701](company2_phone)
regexten=7242655701
callerid=<7242655701>

[7242655702](company2_phone)
regexten=7242655702
callerid=<7242655702>

[7242655703](company2_phone)
regexten=7242655703
callerid=<7242655703>

[7242655704](company2_phone)
regexten=7242655704
callerid=<7242655704>

[7242655705](company2_phone)
regexten=7242655705
callerid=<7242655705>

[7242655706](company2_phone)
regexten=7242655706
callerid=<7242655706>

[7242655707](company2_phone)
regexten=7242655707
callerid=<7242655707>

[7242655708](company2_phone)
```

```
regexten=7242655708
callerid=<7242655708>

[7242655709](company2_phone)
regexten=7242655709
callerid=<7242655709>

[7242655710](company2_phone)
regexten=7242655710
callerid=<7242655710>

[7242655711](company2_phone)
regexten=7242655711
callerid=<7242655711>

[7242655712](company2_phone)
regexten=7242655712
callerid=<7242655712>

[7242655713](company2_phone)
regexten=7242655713
callerid=<7242655713>

[7242655714](company2_phone)
regexten=7242655714
callerid=<7242655714>

[7242655715](company2_phone)
regexten=7242655715
callerid=<7242655715>

[7242655716](company2_phone)
regexten=7242655716
callerid=<7242655716>

[7242655717](company2_phone)
regexten=7242655717
callerid=<7242655717>

[7242655718](company2_phone)
regexten=7242655718
callerid=<7242655718>

[7242655719](company2_phone)
regexten=7242655719
callerid=<7242655719>
```

```
[7242655720](company2_phone)
regexten=7242655720
callerid=<7242655720>

[7242655721](company2_phone)
regexten=7242655721
callerid=<7242655721>

[7242655722](company2_phone)
regexten=7242655722
callerid=<7242655722>

[7242655723](company2_phone)
regexten=7242655723
callerid=<7242655723>

[7242655724](company2_phone)
regexten=7242655724
callerid=<7242655724>

[7242655725](company2_phone)
regexten=7242655725
callerid=<7242655725>


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; TEST from provider numbers
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;[provider_test_phone](!)		; and another one for ulaw-only
[provider_test_phone_tcp](!)
; prod-vcs-11.dmz.reston.champvrs.com
;	type=peer
	type=friend
	transport=tcp
    directmedia=no
;	host=prod-vcs-11.dmz.reston.champvrs.com
;	host=208.94.16.126
    host=208.94.16.194
	nat=force_rport
	directmedia=no		; Typically set to NO if behind NAT
	disallow=all
	allow=ulaw
	allow=h263
	allow=h264
;	allow=h263p
;	allow=vp8
```

```
;           registertrying=yes          ; Send a 100 Trying when the device registers.
            context=incoming_from_provider
            videosupport=yes            ; Turn on support for SIP video. You need to turn this
            dtmfmode=rfc2833

;           qualify=yes
            ; THE FOLLOWING IS SUPPORT FOR INSTANT MESSAGING
            accept_outofcall_message=yes
            outofcall_message_context=IM
            auth_message_requests=yes

[7035705868](provider_test_phone_tcp)
regexten=7035705868             ; When they register, create extension 1234
;callerid=<7035705868>

[7035705174](provider_test_phone_tcp)
regexten=7035705174             ; When they register, create extension 1234
;callerid=<7035705174>


;[provider_test_phone](!)              ; and another one for ulaw-only
[onsip_registered_softphones](!)
; prod-vcs-11.dmz.reston.champvrs.com
;     type=peer
      type=friend
      transport=udp
      directmedia=no
;     host=prod-vcs-11.dmz.reston.champvrs.com
;     host=208.94.16.126
      host=dynamic
      context=onsip_softphones
      nat=force_rport
;     host=208.94.16.126               ; This device needs to register
      disallow=all
            allow-all
      ;allow=gsm                 ; GSM consumes far less bandwidth than ulaw
      allow=ulaw
      ;allow=alaw
;     ;allow=h261
      allow=h263
      allow=h263p
      allow=vp8
      allow=h264
;     registertrying=yes          ; Send a 100 Trying when the device registers.
      videosupport=yes            ; Turn on support for SIP video. You need to turn this
      dtmfmode=rfc2833
```

```
;       qualify=yes
        ; THE FOLLOWING IS SUPPORT FOR INSTANT MESSAGING
        accept_outofcall_message=yes
        outofcall_message_context=IM
        auth_message_requests=yes
;[jcloud861.1](onsip_registered_softphones)
[jcloud861.1](onsip_registered_softphones)
regexten=jcloud861.1
callerid=7032659999
[jcloud861.2](onsip_registered_softphones)
regexten=jcloud861.2
;callerid=jcloud861.2
[jcloud861.3](onsip_registered_softphones)
regexten=jcloud861.3
;callerid=jcloud861.3
[jcloud861.4](onsip_registered_softphones)
regexten=jcloud861.4
;callerid=jcloud861.4
[jcloud861.5](onsip_registered_softphones)
regexten=jcloud861.5
;callerid=jcloud861.5
[5702655710](onsip_registered_softphones)
regexten=5702655710
;callerid=5702655710
[15702655711](onsip_registered_softphones)
regexten=15702655711
;callerid=15702655711
[5702655712](onsip_registered_softphones)
regexten=5702655712
;callerid=5702655712
[15702655713](onsip_registered_softphones)
regexten=15702655713
;callerid=15702655713
[15702655714](onsip_registered_softphones)
regexten=15702655714
;callerid=15702655714

;***************************************************
; ACD demo- support the SIP trunk to our OnSIP server
;***************************************************
[onsip-9338]
type=peer
transport=udp
directmedia=no
host=sip.onsip.com
fromdomain=vmdigioia.onsip.com
username=vmdigioia_jcloud861
```

```
fromuser=vmdigioia_jcloud861
secret=WtETDxAKTsrJpdiZ
dtmfmode=RFC2833
context=incoming-context
insecure=invite
videosupport=yes
disallow=all
allow=all
context=incoming_from_onsip_9338
qualify=yes
;qualify=5000

; Trunkless Trunk for the 2 PSTN numbers from onSIP 703-436-9338/9
[vmdigioia_jcloud861](onsip-9338)
regexten=vmdigioia_jcloud861
callerid=7034369338

[onsip-9339]
type=peer
transport=udp
directmedia=no
host=sip.onsip.com
fromdomain=vmdigioia.onsip.com
username=vmdigioia_jcloud861_6
fromuser=jcloud861.6
secret=FcyrmvPTh7hiucgd
dtmfmode=RFC2833
context=incoming-context
insecure=invite
videosupport=yes
disallow=all
allow=all
context=incoming_from_onsip_9339
qualify=yes
;qualify=5000

; Trunkless Trunk for the 2 PSTN numbers from onSIP 703-436-9338/9
[vmdigioia_jcloud861](onsip-9339)
;regexten=vmdigioia_jcloud861_6
regexten=jcloud861.6
callerid=7034369339
```

Appendix D. Queues

```
;***************************************************
;********* ACD Demo *********
;***************************************************

[general]
;
; Global settings for call queues
;
; Persistent Members
;    Store each dynamic member in each queue in the astdb so that
;    when asterisk is restarted, each member will be automatically
;    read into their recorded queues. Default is 'no'.
;
persistentmembers = yes
;
; AutoFill Behavior
;    The old behavior of the queue (autofill=no) is to have a serial type behavior
;    in that the queue will make all waiting callers wait in the queue
;    even if there is more than one available member ready to take
;    calls until the head caller is connected with the member they
;    were trying to get to. The next waiting caller in line then
;    becomes the head caller, and they are then connected with the
;    next available member and all available members and waiting callers
;    waits while this happens. The new behavior, enabled by setting
;    autofill=yes makes sure that when the waiting callers are connecting
;    with available members in a parallel fashion until there are
;    no more available members or no more waiting callers. This is
;    probably more along the lines of how a queue should work and
;    in most cases, you will want to enable this behavior. If you
;    do not specify or comment out this option, it will default to yes.
;
;autofill = no
;
; Monitor Type
;    By setting monitor-type = MixMonitor, when specifying monitor-format
;    to enable recording of queue member conversations, app_queue will
;    now use the new MixMonitor application instead of Monitor so
;    the concept of "joining/mixing" the in/out files now goes away
;    when this is enabled. You can set the default type for all queues
;    here, and then also change monitor-type for individual queues within
;    queue by using the same configuration parameter within a queue
;    configuration block. If you do not specify or comment out this option,
;    it will default to the old 'Monitor' behavior to keep backward
;    compatibility.
```

```
;
monitor-type = MixMonitor
;
; UpdateCDR behavior.
;    This option is implemented to mimic chan_agents behavior of populating
;    CDR dstchannel field of a call with an agent name, which you can set
;    at the login time with AddQueueMember membername parameter.
;
; updatecdr = no

;
; Note that a timeout to fail out of a queue may be passed as part of
; an application call from extensions.conf:
; Queue(queuename,[options],[optionalurl],[announceoverride],[timeout])
; example: Queue(dave,t,,,45)

; shared_lastcall will make the lastcall and calls received be the same in
; members logged in more than one queue.  This is useful to make the queue
; respect the wrapuptime of another queue for a shared member.
; The default value is no.
;
;shared_lastcall=no
;
; Negative_penalty_invalid will treat members with a negative penalty as logged off
;
;negative_penalty_invalid = no
;
; log_membername_as_agent will cause app_queue to log the membername rather than
; the interface for the ADDMEMBER and REMOVEMEMBER events when a state_interface
; is set.  The default value (no) maintains backward compatibility.
;
;log_membername_as_agent = no
;
;[markq]
;
; A sample call queue
;
; Musicclass sets which music applies for this particular call queue.
; The only class which can override this one is if the MOH class is set
; directly on the channel using Set(CHANNEL(musicclass)=whatever) in the
; dialplan.
;
;musicclass = default
;
; An announcement may be specified which is played for the member as
; soon as they answer a call, typically to indicate to them which queue
; this call should be answered as, so that agents or members who are
```

```
; listening to more than one queue can differentiated how they should
; engage the customer
;
;announce = queue-markq
;
; A strategy may be specified.  Valid strategies include:
;
; ringall - ring all available channels until one answers (default)
; leastrecent - ring interface which was least recently hung up by this queue
; fewestcalls - ring the one with fewest completed calls from this queue
; random - ring random interface
; rrmemory - round robin with memory, remember where we left off last ring pass
; rrordered - same as rrmemory, except the queue member order from config file
;           is preserved
; linear - rings interfaces in the order specified in this configuration file.
;         If you use dynamic members, the members will be rung in the order in
;         which they were added
; wrandom - rings random interface, but uses the member's penalty as a weight
;          when calculating their metric. So a member with penalty 0 will have
;          a metric somewhere between 0 and 1000, and a member with penalty 1 will
;                                       have a metric between 0 and 2000, and a member with penalty 2 will have
;          a metric between 0 and 3000. Please note, if using this strategy, the member
;          penalty is not the same as when using other queue strategies. It is ONLY used
;          as a weight for calculating metric.
;
;strategy = ringall
;
; Second settings for service level (default 0)
; Used for service level statistics (calls answered within service level time
; frame)
;servicelevel = 60
;
; A context may be specified, in which if the user types a SINGLE
; digit extension while they are in the queue, they will be taken out
; of the queue and sent to that extension in this context.
;
;context = qoutcon
;
; A limit can be set to disregard penalty settings when the queue has
; too few members.  No penalty will be weighed in if there are only X
; or fewer queue members. (default 0)
;
;penaltymemberslimit = 5
;
;----------------------QUEUE TIMING OPTIONS------------------------------------
; A Queue has two different "timeout" values associated with it. One is the
; timeout parameter configured in queues.conf. This timeout specifies the
```

```
; amount of time to try ringing a member's phone before considering the
; member to be unavailable. The other timeout value is the second argument
; to the Queue() application. This timeout represents the absolute amount
; of time to allow a caller to stay in the queue before the caller is
; removed from the queue. In certain situations, these two timeout values
; may clash. For instance, if the timeout in queues.conf is set to 5 seconds,
; the retry value in queues.conf is set to 4, and the second argument to Queue()
; is 10, then the following may occur:
;
; A caller places a call to a queue.
; The queue selects a member and attempts to ring that member.
; The member's phone is rung for 5 seconds and he does not answer.
; The retry time of 4 seconds occurs.
; The queue selects a second member to call.
;
; How long does that second member's phone ring? Does it ring for 5 seconds
; since the timeout set in app_queue is 5 seconds? Does it ring for 1 second since
; the caller has been in the queue for 9 seconds and is supposed to be removed after
; being in the queue for 10 seconds? This is configurable with the timeoutpriority
; option. By setting the timeoutpriority to "conf" then you are saying that you would
; rather use the time specified in the configuration file even if it means having the
; caller stay in the queue longer than the time specified in the application argument.
; For the scenario described above, timeoutpriority=conf would result in the second
; member's phone ringing for 5 seconds. By specifying "app" as the value for
; timeoutpriority, you are saying that the timeout specified as the argument to the
; Queue application is more important. In the scenario above, timeoutpriority=app
; would result in the second member's phone ringing for 1 second.
;
; There are a few exceptions to the priority rules. For instance, if timeoutpriority=appp
; and the configuration file timeout is set to 0, but the application argument timeout is
; non-zero, then the timeoutpriority is ignored and the application argument is used as
; the timeout. Furthermore, if no application argument timeout is specified, then the
; timeoutpriority option is ignored and the configuration file timeout is always used
; when calling queue members.
;
; In timeoutpriority=conf mode however timeout specified in config file will take higher
; priority than timeout in application arguments, so if config file has timeout 0, each
; queue member will be called indefineately and application timeout will be checked only
; after this call attempt. This is useful for having queue members with custom timeouts
; specified within Dial application of Local channel, and allows handling NO ANSWER which
; would otherwise be interrupted by queue destroying child channel on timeout.
;
; The default value for timeoutpriority is "app" since this was how previous versions of
; Asterisk behaved.
;
;timeout = 15
;retry = 5
```

```
;timeoutpriority = app|conf
;
;-----------------------END QUEUE TIMING OPTIONS---------------------------------
; Weight of queue - when compared to other queues, higher weights get
; first shot at available channels when the same channel is included in
; more than one queue.
;
;weight=0
;
; After a successful call, how long to wait before sending a potentially
; free member another call (default is 0, or no delay)
;
;wrapuptime=15
;wrapuptime=5
;
; Autofill will follow queue strategy but push multiple calls through
; at same time until there are no more waiting callers or no more
; available members. The per-queue setting of autofill allows you
; to override the default setting on an individual queue level.
;
;autofill=yes
;
; Autopause will pause a queue member if they fail to answer a call
; no: Member will not be paused
; yes: Member will be paused only in the queue where the timeout took place
; all: Memeber will be paused in all queues he/she is a member
;autopause=yes
;
; Autopausedelay delay autopause for autopausedelay seconds from the
; last call if a member has not taken a call the delay has no effect.
;autopausedelay=60
;
; Autopausebusy controls whether or not a queue member is set as paused
; automatically upon the member device reporting busy. The autopausedelay
; option applies. Defaults to 'no'.
;autopausebusy=no
;
; Autopauseunavail controls whether or not a queue member is set as paused
; automatically upon the member device reporting congestion. The autopausedely
; option applies. Defaults to 'no'.
;autopauseunavail=no
;
; Maximum number of people waiting in the queue (0 for unlimited)
;
;maxlen = 0
;
; Note: for below queue channel options (setinterfacevar, setqueueentryvar,
```

```
; setqueuevar) if the caller channel is a local channel and optimizations
; is enabled then after optimization has occurred only the queue member
; channel will contain the variables.
;
; If set to yes, just prior to the caller being bridged with a queue member
; the following variables will be set on the caller and queue member channels:
; MEMBERINTERFACE is the interface name (eg. Agent/1234)
; MEMBERNAME is the member name (eg. Joe Soap)
; MEMBERCALLS is the number of calls that interface has taken,
; MEMBERLASTCALL is the last time the member took a call.
; MEMBERPENALTY is the penalty of the member
; MEMBERDYNAMIC indicates if a member is dynamic or not
; MEMBERREALTIME indicates if a member is realtime or not
;
setinterfacevar=yes
;
; If set to yes, just prior to the caller being bridged with a queue member
; the following variables will be set on the caller and queue member channels:
; QEHOLDTIME callers hold time
; QEORIGINALPOS original position of the caller in the queue
;
setqueueentryvar=yes
;
; If set to yes, the following variables will be set
; just prior to the caller being bridged with a queue member (set on the
; caller and queue member channels) and just prior to the caller
; leaving the queue
; QUEUENAME name of the queue
; QUEUEMAX maxmimum number of calls allowed
; QUEUESTRATEGY the strategy of the queue;
; QUEUECALLS number of calls currently in the queue
; QUEUEHOLDTIME current average hold time
; QUEUECOMPLETED number of completed calls for the queue
; QUEUEABANDONED number of abandoned calls
; QUEUESRVLEVEL queue service level
; QUEUESRVLEVELPERF current service level performance
;
setqueuevar=yes

; if set, run this macro when connected to the queue member
; you can override this macro by setting the macro option on
; the queue application
;
;membermacro=macro_name[,arg1[,...][,argN]]

; if set, run this gosub when connected to the queue member
; you can override this gosub by setting the gosub option on
```

```
; the queue application
;
;membergosub=gosub_context_name[,arg1[,...][,argN]]

; How often to announce queue position and/or estimated
; holdtime to caller (0=off)
; Note that this value is ignored if the caller's queue
; position has changed (see min-announce-frequency)
;
announce-frequency = 5 ; was 90
;
; The absolute minimum time between the start of each
; queue position and/or estimated holdtime announcement
; This is useful for avoiding constant announcements
; when the caller's queue position is changing frequently
; (see announce-frequency)
;
;min-announce-frequency = 15
min-announce-frequency = 2 ;was 15
;
; How often to make any periodic announcement (see periodic-announce)
;
periodic-announce-frequency=10 ;was 60
;
; Should the periodic announcements be played in a random order? Default is no.
;
;random-periodic-announce=no
;
; If set to yes, the periodic announcment frequency will be timed from the end
; of each announcment rather than from the start of each announcment.  This
; defaults to off.
;
;relative-periodic-announce=yes
;
; Should we include estimated hold time in position announcements?
; Either yes, no, or only once.
; Hold time will be announced as the estimated time.
;
;announce-holdtime = yes|no|once
announce-holdtime = yes
;
; Queue position announce?
; Valid values are "yes," "no," "limit," or "more." If set to "no," then the caller's position will
; never be announced. If "yes," then the caller's position in the queue will be announced
; to the caller. If set to "more," then if the number of callers is more than the number
; specified by the announce-position-limit option, then the caller will hear that there
; are more than that many callers waiting (i.e. if a caller number 6 is in a queue with the
```

```
; announce-position-limit set to 5, then that caller will hear that there are more than 5
; callers waiting). If set to "limit," then only callers within the limit specified by announce-position-limit
; will have their position announced.
;
announce-position = yes
;
; If enabled, play announcements to the first user waiting in the Queue. This may mean
; that announcements are played when an agent attempts to connect to the waiting user,
; which may delay the time before the agent and the user can communicate. Disabled by
; default.
;
announce-to-first-user = yes
;
; If you have specified "limit" or "more" for the announce-position option, then the following
; value is what is used to determine what announcement to play to waiting callers. If you have
; set the announce-position option to anything else, then this will have no bearing on queue operation
;
;announce-position-limit = 5
;
; What's the rounding time for the seconds?
; If this is non-zero, then we announce the seconds as well as the minutes
; rounded to this value.
; Valid values are 0, 5, 10, 15, 20, and 30.
;
announce-round-seconds = 10
;
; Use these sound files in making position/holdtime announcements.  The
; defaults are as listed below -- change only if you need to.
;
; Keep in mind that you may also prevent a sound from being played if you
; explicitly set a sound to be an empty string. For example, if you want to
; prevent the queue from playing queue-thankyou, you may set the sound using
; the following line:
;
; queue-thankyou=
;
                              ;          ("You are now first in line.")
queue-youarenext = queue-youarenext
                              ;          ("There are")
queue-thereare   = queue-thereare
                              ;          ("calls waiting.")
queue-callswaiting = queue-callswaiting
                              ;          ("The current est. holdtime is")
queue-holdtime = queue-holdtime
                              ;          ("minute.")
queue-minute = queue-minute
                              ;          ("minutes.")
```

```
queue-minutes = queue-minutes
                                        ;            ("seconds.")
queue-seconds = queue-seconds
                                        ;            ("Thank you for your patience.")
queue-thankyou = queue-thankyou
                                        ;      ("Hold time")
queue-reporthold = queue-reporthold
                                        ;      ("All reps busy / wait for next")
periodic-announce = queue-periodic-announce
;
; A set of periodic announcements can be defined by separating
; periodic announcements to reproduce by commas. For example:
;periodic-announce = queue-periodic-announce,your-call-is-important,please-wait
;
; The announcements will be played in the order in which they are defined. After
; playing the last announcement, the announcements begin again from the beginning.
;
; Calls may be recorded using Asterisk's monitor/MixMonitor resource
; This can be enabled from within the Queue application, starting recording
; when the call is actually picked up; thus, only successful calls are
; recorded, and you are not recording while people are listening to MOH.
; To enable monitoring, simply specify "monitor-format";  it will be disabled
; otherwise.
;
; You can specify the monitor filename with by calling
;    Set(MONITOR_FILENAME=foo)
; Otherwise it will use MONITOR_FILENAME=${UNIQUEID}
;
; Pick any one valid extension for monitor format recording. If you leave
; monitor-format commented out, it will not record calls.
;
; monitor-format = gsm|wav|wav49
;
; Monitor Type
;    By setting monitor-type = MixMonitor, when specifying monitor-format
;    to enable recording of queue member conversations, app_queue will
;    now use the new MixMonitor application instead of Monitor so
;    the concept of "joining/mixing" the in/out files now goes away
;    when this is enabled. If you do not specify or comment out this option,
;    it will default to the old 'Monitor' behavior to keep backward
;    compatibility.
;
; monitor-type = MixMonitor
;
; ----------------------- TYPE MIXMONITOR OPTIONS -----------------------------
;
;
```

```
; You can specify the options supplied to MixMonitor by calling (from the dialplan)
;   Set(MONITOR_OPTIONS=av(<x>)V(<x>)W(<x>))
; The 'b' option for MixMonitor (only save audio to the file while bridged) is
; implied.
;
; You can specify a post recording command to be executed after the end of
; recording by calling (from the dialplan)
;
;   Set(MONITOR_EXEC=mv /var/spool/asterisk/monitor/^{MONITOR_FILENAME} /tmp/^{MONITOR_FILENAME})
;
; or
;
;   Set(MONITOR_EXEC=mv /var/spool/asterisk/monitor/^{MIXMONITOR_FILENAME}
/tmp/^{MIXMONITOR_FILENAME})
;
; If you choose to use the latter, you will not be able to switch the monitor-type back to Monitor
; without changing this in the dialplan.
;
;
; The command specified within the contents of MONITOR_EXEC will be executed when
; the recording is over. Any strings matching ^{X} will be unescaped to ${X} and
; all variables will be evaluated just prior to recording being started.
;
; The contents of MONITOR_FILENAME will also be unescaped from ^{X} to ${X} and
; all variables will be evaluated just prior to recording being started.
;
; ---------------------- Queue Empty Options ----------------------------------
;
; Asterisk has provided the "joinempty" and "leavewhenempty" options for a while
; with tenuous definitions of what they actually mean. The "joinempty" option controls
; whether a caller may join a queue depending on several factors of member availability.
; Similarly, then leavewhenempty option controls whether a caller may remain in a queue
; he has already joined. Both options take a comma-separated list of factors which
; contribute towards whether a caller may join/remain in the queue. The list of
; factors which contribute to these option is as follows:
;
; paused: a member is not considered available if he is paused
; penalty: a member is not considered available if his penalty is less than QUEUE_MAX_PENALTY
; inuse: a member is not considered available if he is currently on a call
; ringing: a member is not considered available if his phone is currently ringing
; unavailable: This applies mainly to Agent channels. If the agent is a member of the queue
;            but has not logged in, then do not consider the member to be available
; invalid: Do not consider a member to be available if he has an "invalid" device state.
;         This generally is caused by an error condition in the member's channel driver.
; unknown: Do not consider a member to be available if we are unable to determine the member's
;         current device state.
; wrapup: A member is not considered available if he is currently in his wrapuptime after
```

```
;       taking a call.
;
; For the "joinempty" option, when a caller attempts to enter a queue, the members of that
; queue are examined. If all members are deemed to be unavailable due to any of the conditions
; listed for the "joinempty" option, then the caller will be unable to enter the queue. For the
; "leavewhenempty" option, the state of the members of the queue are checked periodically during
; the caller's stay in the queue. If all of the members are unavailable due to any of the above
; conditions, then the caller will be removed from the queue.
;
; Some examples:
;
;joinempty = paused,inuse,invalid
;
; A caller will not be able to enter a queue if at least one member cannot be found
; who is not paused, on the phone, or who has an invalid device state.
;
;leavewhenempty = inuse,ringing
;
; A caller will be removed from the queue if at least one member cannot be found
; who is not on the phone, or whose phone is not ringing.
;
; For the sake of backwards-compatibility, the joinempty and leavewhenempty
; options also accept the strings "yes" "no" "strict" and "loose". The following
; serves as a translation for these values:
;
; yes - (empty) for joinempty; penalty,paused,invalid for leavewhenempty
; no - penalty,paused,invalid for joinempty; (empty) for leavewhenempty
; strict - penalty,paused,invalid,unavailable
; loose - penalty,invalid
;

; If you wish to report the caller's hold time to the member before they are
; connected to the caller, set this to yes.
;
; reportholdtime = no
;
; If you want the queue to avoid sending calls to members whose devices are
; known to be 'in use' (via the channel driver supporting that device state)
; uncomment this option. This can be controlled on a per member basis by
; setting 'ringinuse' on that member. This can be done in the member definition,
; in the 'ringinuse' field on a realtime member, via the QUEUE_MEMBER dialplan
; function, or with CLI/AMI. By default, the per member value will be the same
; as the queue's ringinuse value if it isn't set on the member deliberately.
; (Note: only the SIP channel driver currently is able to report 'in use'.)
; ringinuse = no
;
; If you wish to have a delay before the member is connected to the caller (or
```

```
; before the member hears any announcement messages), set this to the number of
; seconds to delay.
;
; memberdelay = 0
;
; If timeoutrestart is set to yes, then the timeout for an agent to answer is
; reset if a BUSY or CONGESTION is received.  This can be useful if agents
; are able to cancel a call with reject or similar.
;
; timeoutrestart = no
;
; If you wish to implement a rule defined in queuerules.conf (see
; configs/queuerules.conf.sample from the asterisk source directory for
; more information about penalty rules) by default, you may specify this
; by setting defaultrule to the rule's name
;
; defaultrule = myrule
;
; Each member of this call queue is listed on a separate line in
; the form technology/dialstring.  "member" means a normal member of a
; queue.  An optional penalty may be specified after a comma, such that
; entries with higher penalties are considered last.  An optional member
; name may also be specified after a second comma, which is used in log
; messages as a "friendly name".  Multiple interfaces may share a single
; member name. An optional state interface may be specified after a third
; comma. This interface will be the one for which app_queue receives device
; state notifications, even though the first interface specified is the one
; that is actually called.
;
; A hint can also be used in place of the state interface using the format
; hint:<extension>@<context>. If no context is specified then 'default' will
; be used.
;
; It is important to ensure that channel drivers used for members are loaded
; before app_queue.so itself or they may be marked invalid until reload. This
; can be accomplished by explicitly listing them in modules.conf before
; app_queue.so.  Additionally, if you use Local channels as queue members, you
; must also preload pbx_config.so and chan_local.so (or pbx_ael.so, pbx_lua.so,
; or pbx_realtime.so, depending on how your dialplan is configured).
;
; syntax: member => interface,[,penalty][,membername][,state_interface][,ringinuse]
;
;member => DAHDI/1
;member => DAHDI/2,10
;member => DAHDI/3,10,Bob Johnson
;member => Local/1001@agents,0,May Flowers,Agent:1001
;member => Local/1002@agents,0,John Doe,Agent:1002
```

```
;member => Local/1000@default,0,John Smith,SIP/1000
;member => Local/2000@default,0,Lorem Ipsum,SIP/2000,no



;[StandardQueue](general)(!)
[StandardQueue](!)(general)
;persistentmembers = yes
;monitor-type = MixMonitor
autofill=yes
;shared_lastcall=yes        ; This line causes an error
musicclass=moh
strategy=rrmemory
joinempty=no
leavewhenempty=yes
ringinuse=no
wrapuptime=1            ; When a member hangs up, how many seconds does queue wait before assigning another caller.  Default is 0.
setqueuevar=yes


[VideoQ_GenQuest_1](StandardQueue)
[VideoQ_Complaint_1](StandardQueue)

[VoiceQ_GenQuest_1](StandardQueue)
[VoiceQ_Complaint_1](StandardQueue)

[VideoQ_GenQuest_2](StandardQueue)
[VideoQ_Complaint_2](StandardQueue)

[VoiceQ_GenQuest_2](StandardQueue)
[VoiceQ_Complaint_2](StandardQueue)
```

Appendix E. Detailed Recommendations by Topic Area

MITRE offers the following recommendations to help mitigate issues that can delay implementation and operations. The recommendations are organized by topic area.

E.1 Requirements and Design (RD)

Recommendation RD-1

Complete a lab simulation to demonstrate and verify that the design can support the workload stated and satisfy the performance requirements. Create a comparison of results using an end-to-end test.

Rationale: The design provides processing limits by zone and by hour. Without test or simulation information to support the Proof of Concept (POC) processing capabilities, the design does not inspire confidence that the system can meet the necessary performance expectations.

Questions about the scalability of the design will be raised from the future analysis of the POC integration products. To gain a return on investment (ROI) to justify the costs of the POC products suggested, the capacities of the production system should be based on a comprehensive analysis of the types and distributions of the workloads of the system. It is essential that ICE Systems Engineering team look at both daily and yearly distributions and at how the workloads are distributed geographically. Each unit of each type of workload needs to be converted to the required hardware resources, including Central Processing Units (CPU) and their capability in MIPS, Direct Access Storage Devices (DASD) and their capacity in Terabytes (TB), and Local Area and Wide Area Networks (LAN/WAN) and their speed in Gigabytes Per Second (GB/s). The results of the capacity projections should identify the required resource for handling steady and peak loads (in a virtualized environment).

Recommendation RD-2

The design workflow should take into account the logging and auditing requirements for projected performance.

Rationale: The design raises a question about managing the level of effort imposed on operations; must also consider the logging and auditing requirements and the corresponding constraints these activities will place on operations.

Recommendation RD-3

The design workflow should consider the initial sizing and scaling of the POC systems and their relation to projected performance. The results should show how the data are stored in the secure zone. Provide an Operational Requirements Matrix (ORM) corresponding to the design to address the operational infrastructure outside the processing boundary.

Rationale: The ORM can mature as sprints are reviewed, and can provide greater insight into the resource demands predicted by each sprint.

Recommendation RD-4

The design document should focus on the POC's constraints, limitations, or requirements in conjunction with the infrastructure products selected for the operating environment.

Rationale: There is a need to further review the selected POC within the bounds of the design to determine if they will satisfy the transaction demand by callers. The initial capacity projections place a burden on the enterprise integration server to process the request/response transactions in a timely manner, and to provide sufficient fault tolerance if the transaction fails.

E.2 Security (S)

The security requirements, including the Information Security Acceptable Risks Safeguards and those in the design, should provide information on system access, generally accepted security principles, and specific application security requirements.

Recommendation S-1

The Technical Architecture should cross-reference the section referring to security requirements and provide information on how the design complies with the requirements for a vulnerability assessment. The Technical Architecture document should state how the systems are configured to National Institute for Standards and Technology (NIST) security guidelines, and state where deviations to the guidelines are necessary.

Rationale: The NIST security configuration settings are very restrictive and could cause problems in running the application. These configurations can provide a security compliance checklist that can be reused with additional implementations of the technical architecture in similar projects.

Recommendation S-2

The design should specify POC configuration information to cross-reference the configuration settings necessary to maintain all hardware and software proposed in the technical architecture.

Rationale: A baseline configuration should be included in the design to provide a reliable starting point for the maintenance contractor.

Recommendation S-3

The design should include a backup workflow that depicts all data paths and specify the schedule and frequency of backups. The processing view should account for the system administration of backups

Rationale: The design documentation should reflect periodic computer system backups of mission-critical data and archives to ensure the data are adequately preserved and protected against data loss and destruction.

Recommendation S-4

The design should include a processing view that provides the paths to the Intrusion Detection System. The design should provide operational requirements that include IDS monitoring specifications.

Rationale: The design should depict the IDS interactions to make certain intrusion prevention is considered.

Recommendation S-5

The design should include a Protocol view that depicts all port assignments on all hardware for each zone.

Rationale: Every application and server in each zone uses the protocol "ports" at the network (transport) layer to manage communication channels between programs. The design should depict all ports used in each zone by hardware and software to help the maintenance contractor and also determine risk. This is especially important for any deviations from the standard.

E.3 Connectivity (C)

Recommendation C-1

The design should include connectivity, protocol, processing, and hardware/software views that depict the paths to the DNS for each region.

Rationale: The design document should provide Internet Protocol (IP) address and Fully Qualified Domain Name (FQDN) assignments for private network management. Application and system administrators should have adequate documentation of IP addresses or FQDNs in each zone.

Recommendation C-2

Specific waivers, exceptions, or addenda should provide an explanation for each POC product that connects directly from the Application Zone to the secure Data Zone and bypasses the message service.

Rationale: The configuration files specify a common environment for writing applications that access message queues. If the POC product does not provide a seamless interface to the message service, then the design should clearly depict the interface and services from the application region to the data region.

Recommendation C-3

The design should consider using a virtual storage strategy to conform to the connectivity and protocol views for firewalls, hardware devices, and system administration port assignments.

Rationale: The requirements and design include capacity projections that cover message creation, transaction logging, secure/private data, and error logging. The established standard enterprise Storage Area Network (SAN) strategy, architecture, and related components provide guidelines for the performance and interoperability between the systems and applications that utilize SAN storage in the secure data zone. The design document should include enterprise SAN technical information that meets the corresponding enterprise SAN product, architecture, and strategy requirements.

Recommendation C-4

The design should include connectivity, protocol, processing, and hardware/software views that depict all systems administration and support interfaces to the POC products (by network).

Rationale: The design integrates several applications from different vendors with systems administration requirements unique to the corresponding product. The systems administration interface is accessible from different paths, ports, and platforms in the Presentation, Application, and Secure Data Zones. Each system requires its own configuration and maintenance in addition to standard system administration policies. The requirements for the systems administration functions should be addressed in the design to account for non-compliant applications.

Recommendation C-5

The design should include the connectivity, protocol, processing, and hardware/software views that depict redundancy paths, failover requirements, load-balancing hardware, and other fault-tolerant or high-availability requirements; should provide a reference document that outlines the operational standards that are provided as a baseline to all applications.

Rationale: The design should highlight the business needs of the application and the minimum processing availability. Fault tolerance encapsulates a set of factors considered in the provision of each service in each zone. Documenting the fault-tolerance requirements for networked, multi-tier environments ensures uninterrupted access to the application.

E.4 Disaster Recovery and Business Continuity (DRBC)

Recommendation DRBC-1

The design document should cross-reference service level agreements (SLA) and provide a list of dependencies, restrictions, and constraints that can impact the POC products and their associated processing windows. This process includes supporting activities that can streamline the effort to research and document:

- Scope an agreement of the IT Service Continuity Management (ITSCM) process and the policies adopted
- Draft a Business Impact Assessment (BIA) to quantify the impact loss of IT service would have on the mission and end users
- Draft a risk assessment to identify potential threats to continuity and the likelihood of the threats becoming reality. This also includes taking measures to manage the identified threats where this can be cost-justified. The approach to managing these threats will form the core of the ITSCM strategy and plans.
- Draft an overall ITSCM strategy that must be integrated into the Business Continuity Management (BCM) strategy. This can be produced following the BIA and the

development of the risk assessment, and is likely to include elements of risk reduction as well as selection of appropriate and comprehensive recovery options.

- Production of an ITSCM Plan, which again must be integrated with the overall BCM plans
- Testing the plans
- Ongoing operation and maintenance of the plans
- Provide a DR/COOP that meets BIA requirements
- Test the DR/COOP Plan on a regularly scheduled basis and during DR/COOP exercises
- Ensure that the DR/COOP Plan is maintained and reflects changes that are made to the production systems

Rationale: The system utilization should account for negative impact to scheduled maintenance windows in all zones from front-end users because of peak loads. The standard POC and system maintenance requirements should be factored into system availability, redundancy, and restore design plans. This activity can support DR with the research, analysis, and planning necessary to design and implement an industry best practice, enterprise-wide DR infrastructure and its related processes.

Business (Service) Continuity Management encompasses DR and Continuity of Operations (COOP) planning. The purpose of Service Continuity Management is to support the overall Business Continuity Management process by ensuring that the required IT technical and services facilities (including computer systems, networks, applications, telecommunications, technical support, and service desk) can be recovered within required and agreed-upon mission critical timescales.

Recommendation DRBC-2

The design document should cross-reference the specific services in hardware configuration, network, security, performance, and disaster recovery guidelines.

Rationale: The system should provide a design that could be replicated and implemented by having similar, integrated platforms that must support secure data standards and requirements. The proposed processing for transactions can become the standard for all future transaction requirements. The design should include specific design requirements based on business requirements and on the selected architecture. These enterprise design requirements could lay the foundation for a standard set of guidelines to which future design documents should comply.

The As-Is and To-Be Process Analysis should develop any significant findings related to the initial requirements. Table 4 presents the significant risks and proposed actionable recommendations to resolve issues or mitigate risks based on that assessment.

Table 4. Proposed Risk Mitigation

Significant Risks	Mitigation
1. If there is inadequate planning at the start, then there will be missed tasks, delays, cost overruns, and rework.	• Draft an updated higher-level project plan supported by detailed lower-level tasks; the project plan should include: – Interdependent tasks that rely on external systems/applications, the corresponding data sources/data use agreements, and the maturity and stability of the available service offerings to include standardization, interoperability, portability, transition to/from – Multiple planned 'sprints' over the course of a 12-month cycle; the sprints provide a specific timeframe to implement the associated requirements, to test (including regression testing), and to review the performance – Realistic, accurate, and feasible task start dates and durations • Leverage existing recent relevant project plans for similar infrastructure dependencies • Prepare and provide a risk strategy and mitigation position to understand the scope and visibility, and the roles and responsibilities
2. If formal requirement are not gathered, documented, and traced, then the resulting delivery will not meet expectations.	• Document formal requirements that are integral to provisioning and sustaining IT Service Delivery functions that are associated with core enterprise infrastructure computing services • Promote a standard infrastructure and technical architecture in addition to prioritized business rules to support a diverse enterprise • Formal Configuration Management approach to preserve baseline configurations – Develop configuration management as the core process to establish and document baselines from which all other processes needing baseline information can operate – Establish a comprehensive approach and create a baseline to record and track assets and system configurations

<table>
<tr><th>Significant Risks</th><th>Mitigation</th></tr>
<tr><td>3. If there are missing and/or unavailable (undeveloped) application/system documents and artifacts to support design, configuration design and management, provisioning, and sustainment, then the resulting implementation will not be effective.</td><td>

Complete the documentation of the architecture and concepts; leverage prior artifacts / deliverables for similar systems/projects;

System Security Plan (SSP): a document that should provide an overview of the security requirements of the system and describe the controls in place (or planned), responsibilities, and expected behavior of all individuals who access the system.
Intrusion Detection System (IDS): a document that should provide an overview of a system used for detecting inappropriate, incorrect, or anomalous activity
Firewall rule-sets: a document that should provide specific rules for applications used by the design (web browsers, application servers, database servers, etc.)
Recommend as part of the design phase
System Configuration: a document that should provide information on the process of setting up the hardware devices and assigning resources
Recommend as part of the Statement of Work (SOW)
Domain Name Service (DNS) specifications: a document that should specify the rules for constructing domain names that satisfy both the rules of the domain system and the policies of system administration
Recommend as part of the design phase
Concept of Operations that describes the end-to-end solution
Business Requirements that describes the business and user requirements for the phase one implementation. A business requirement describes the business needs that will satisfy independently of how those needs will be met.
Assumptions that identify development constraints and assumptions in addition to selected activities and requirements that will be met by other contractors
Technical Proposal that describes the scope and approach of the development project
Call Center Standards and Procedures: a document that delineates call center access and operational procedures for hardware and software

</td></tr>
</table>

Significant Risks	Mitigation
4. If there is an incomplete analysis to determine and design the IT Service Continuity and Disaster Recovery (DR) risk assessment and approach, then recovery execution will not be possible.	• Assess the baseline and the future state system-level capability that supports a fully integrated set of components • Assess the DR requirements for rigorous support that requires synchronization of all databases and all types of transactions • Create an implementation plan that builds from a simpler level of DR for Initial Operating Capability (IOC) to a more complex, high availability configuration after IOC

Appendix F. Proposed Stacks

Stacks

A *Stack* is a set of programs that work together to form a complete suite of solutions to a business problem area. In other words, a Stack includes the platform, associated tools, and the information technology (IT) environment in which an application is developed, tested, and operated. Stacks are the supported application platforms and associated tools in the development, test, and production environments available to application projects. If one provides common tools and platforms, economies of scale can drive down the costs of application development, testing, and operations. Providing a virtual workspace for development and testing facilitates the collaborative development efforts.

The goal of introducing standard Stacks within the Integrated Project Team (IPT) is to promote standardized architecture, ensure interoperability, and provide and maintain a limited set of software development, testing, and operational tools and environments for optimized efficiency and effectiveness. The IPT has included a list of standard products for application projects to consider, especially if the application is going to run on or interface with the national IT environment.

Traditional stove-piped applications encountered interoperability and flexibility issues. Many business applications are becoming 3-tier, web-based applications that often have several parts residing in presentation, business logic, and data zones. The Stack of these application parts in different zones may vary depending on the advancement of technologies.

Without unlimited resources in funding and staff skills, organizations can only support a limited set of system platforms and middleware components for its applications. The current trends in the IT industry include simplification, standardization, consolidation, sharing the infrastructure, integration, and automation for more operational efficiency and effectiveness. Therefore, if a local or group application plans to run applications on or interface with the national IT environment, their application development projects will benefit from using the supported system platforms, middleware components, and associated tools.

The notionally defined stacks in this appendix include all supported platforms and associated tools for the future environment along with existing products in use. As the Stacks evolve over time, the list of products should shrink to reduce the complexity and costs.

Scope

The scope of any Stack consists of the system platforms and the middleware on which an application runs. An application operates throughout its life cycle phases in three environments: the development, test, and production environments.

As the collaboration environment matures, additional types of Stacks may be introduced to serve specific purposes, such as a training or a sandbox environment.

Initially, the IPT offers four major types of Stacks for different platform categories:

- Open Source on Linux
- Proprietary on Linux
- Proprietary on Windows
- Open Source on Windows

The community may specify the Stack environment and the Stack types simultaneously to assure a suitable environment for development, testing, and deployment of its applications.

Audience

The main audience for the Stack is all application project development teams. Application developers are encouraged to refer to the Stacks to determine which system platforms and middleware products support interoperability in the IT environment.

Elements in a Stack

The elements in a Stack include:

- System Platform services
- Enabling services (middleware)
- Application Management services
- Security Management services

In general, there are many aspects to consider when choosing Stacks, including:

- Application and System Architectures
- Product Functionality
- Product Quality / Reliability
- Interoperability
- Support
- Freedom of Choice
- Migration
- Capital and Operational Expenses
- Existing Investment
- Replacement Costs

When choosing a Stack for an application, the goal is to allow the application to evolve over a long period with optimized performance and the total cost of ownership. This goal includes having platform independence and not being locked in by specific vendor products.

The platform independence of the application can be achieved by adopting standards and using open source products can achieve platform independence of the application. A layered computing architecture can also accomplish platform independence at different levels. For example, at the hardware level, Xen Hypervisor and VMware can shield differences on vendor systems. At the operating system (OS) level, open source Linux, VMware, and Java can shield OS differences. The Open Database Connectivity (ODBC) or Java Database Connectivity (JDBC) can provide portability among RDBMS products. Java or C/C++ programming languages can provide application portability as well. At the application level, POC products from major vendors normally provide versions running on different platforms.

Open source software often provides freedom of choice and lower costs than the proprietary products; however, careful selection is necessary to ensure that open source products provide adequate quality and support.

Web-based applications normally have higher expectations or requirements on availability and performance. These applications require fast end-to-end (E2E) application management. This means that enterprise-wide, integrated Event Management is essential to meet those requirements. Accordingly, applications must be well instrumented with event detection, logging, and notification capabilities consistent with standard conventions. This is an important aspect in selecting the Stack.

Development Environment

The development tools and environment are provided within the IPT and supported by the sponsor. The objective is to share the Stacks among as many projects as possible to increase standardization and lower costs.

The choices of system platform include virtual machine (VM), *de facto* proprietary operating systems such as Windows Server, and open source systems such as Linux.

Middleware (or *enabling* services) operates between the system platform and the application. Middleware includes directory services, messaging services, database management systems, web services, and application management services.

Development tools include requirements management tools, programming and script language frameworks, Integrated Development Environment (IDE), collaborative services, source code version control system, change management system, problem management system, and capacity planning tools.

In addition to deciding on the system platforms and the middleware, an application development project also considers how to manage the application to sustain required availability, performance, and security, and how to properly implement the E2E application management in the development phase and operated in the production environment.

A well-managed, mission-critical application may require extensive software instrumentation for application management purposes. The instrumented software may generate event notifications. Event Management must be standardized in the environment for E2E application management. A guideline on standard event object format and conventions is necessary.

Configuration Management, Change Management, and Problem Management tools are all needed in the development environment.

For Pubic Release – In Accordance with Data Rights

Security and privacy requirements necessitate an enterprise approach for security management. It is essential to integrate the identity management, authentication, authorization, monitoring, and security audit of an application as components of the environment's enterprise security management.

Test Environment

The system platform and middleware that the application runs on in the test environment should be the same as those selected in the development environment. Other tools in test management, test automation, debugging, and modeling and simulation may also be used.

The test environment will also use automated software distribution tools and problem management systems.

Production Environment

In the production environment, the system platform and middleware that the application runs on in the production environment are the same as those in the development and test environments. The enterprise and application management and security management systems are managing the entire enterprise environment. Automated Software Distribution and Patch Management tools, Service Desk systems, Incident and Problem Management systems, Asset and Configuration Management systems, Change Management tools, Service Level Management tools, and a variety of Security Management tools will also be used.

Supported Stacks

Platform independence can be maintained by separate business domains. The underlying IT support for application development, testing, and production can be provided by the national environment. The national environment can take advantage of the economy of scale and achieve more effective use of IT resources in meeting its various business needs.

Table 5 lists the POC-supported system platforms and middleware components in the development environment.

Table 5. POC-supported System Platforms and Middleware Components in the Development Environment – NOTIONAL

Component	Proprietary on Windows	Open Source on Windows	Proprietary on Linux	Open Source on Linux
Programming Language	C#, C++	C++, Java, Perl, PHP	C#, C++	C++, Java, Perl, PHP, JSP
Framework	MS .NET	JBoss, Mono	Mono	JBoss, Mono
IDE	MS Visual Studio 2005	Eclipse	MonoDevelop	Eclipse
Web Server	MS IIS	Apache Web Server	IBM HTTP Server	Apache Web Server
Web Application Server	MS IIS	JBoss, Tomcat	IBM WebSphere Application Server	JBoss, Tomcat

Component	Proprietary on Windows	Open Source on Windows	Proprietary on Linux	Open Source on Linux
Messaging Service	IBM WebSphere MQ, MSMQ	Apache ActiveMQ	IBM WebSphere MQ	Apache ActiveMQ
Enterprise Service Bus	MS ESB Guidance, Progress Sonic ESB, IBM WebSphere ESB	Apache ServiceMix	Progress Sonic ESB, IBM WebSphere ESB	Apache ServiceMix
RDBMS	MS SQL Server, IBM DB2, Informix Dynamic Server	MySQL	IBM DB2, Informix Dynamic Server	MySQL
Directory Services	MS Active Directory	OpenLDAP+Kerberos, OpenDS		OpenLDAP+Kerberos, OpenDS
Operating System	MS Windows Server	MS Windows Server	Linux	Linux
Application Management Instrumentation	Tivoli Management, HP Operations Manager, BMC Performance Manager, MS Operations Manager	OpenNMS	Tivoli Management, HP Operations Manager, BMC Performance Manager	OpenNMS

Table 6 lists the POC-supported system platforms and middleware components in the test environment.

Table 6. POC-supported System Platforms and Middleware Components in the Test Environment – NOTIONAL

Component	Proprietary on Windows	Open Source on Windows	Proprietary on Linux	Open Source on Linux
Programming Language	C#, C++	C++, Java, Perl, PHP	C#, C++	C++, Java, Perl, PHP, JSP
Framework	MS .NET	JBoss, Mono	Mono	JBoss, Mono
IDE	MS Visual Studio 2005	Eclipse	MonoDevelop	Eclipse
Web Server	MS IIS	Apache Web Server	IBM HTTP Server	Apache Web Server
Web Application Serve	MS IIS	JBoss, Tomcat	IBM WebSphere Application Server	JBoss, Tomcat
Messaging Service	IBM WebSphere MQ, MSMQ	Apache ActiveMQ	IBM WebSphere MQ	Apache ActiveMQ
Enterprise Service Bus	MS ESB Guidance, Progress Sonic ESB, IBM WebSphere ESB	Apache ServiceMix	Apache ServiceMix	Apache ServiceMix

Component	Proprietary on Windows	Open Source on Windows	Proprietary on Linux	Open Source on Linux
RDBMS	MS SQL Server, IBM DB2, Informix Dynamic Server	MySQL	IBM DB2, Informix Dynamic Server	MySQL
Directory Services	MS Active Directory	OpenLDAP+Kerberos, OpenDS		OpenLDAP+Kerberos, OpenDS
Operating System	MS Windows Server	MS Windows Server	Linux	Linux

Table 7 lists the POC-supported system platforms and middleware components in the operational environment.

Table 7. POC-supported System Platforms and Middleware Components in the Operational Environment – NOTIONAL

Component	Proprietary on Windows	Open Source on Windows	Proprietary on Linux	Open Source on Linux
Framework	MS .NET	JBoss, Mono	Mono	JBoss, Mono
Web Server	MS IIS	Apache Web Server	IBM HTTP Server	Apache Web Server
Web Application Server	MS IIS	JBoss, Tomcat	IBM WebSphere Application Server	JBoss, Tomcat
Messaging Service	IBM WebSphere MQ, MSMQ	Apache ActiveMQ	IBM WebSphere MQ	Apache ActiveMQ
Enterprise Service Bus	MS ESB Guidance, Progress Sonic ESB, IBM WebSphere ESB	Apache ServiceMix	Apache ServiceMix	Apache ServiceMix
RDBMS	MS SQL Server, IBM DB2, Informix Dynamic Server	MySQL	IBM DB2, Informix Dynamic Server	MySQL
Directory Services	MS Active Directory	OpenLDAP+Kerberos, OpenDS		OpenLDAP+Kerberos, OpenDS
Operating System	MS Windows Server	MS Windows Server	Linux	Linux
Virtual System	MS Windows Server 2008 Hypervisor-V, VMware	MS Windows Server 2008 Hypervisor-V, KVM, VMware	KVM, Xen Hypervisor, VMware	KVM, Xen Hypervisor

Table 8 lists the support tools used in all environments.

Table 8. Supported Tools for All Environments – NOTIONAL

Component	Proprietary on Windows	Open Source on Windows	Proprietary on Linux	Open Source on Linux
Collaboration services	IBM Lotus Connections, MS SharePoint	NPJ	IBM Lotus Connections	Open-Xchange, NPJ
Source Code Control	MS Team Foundation Source Control, IBM Rational ClearCase	CVS, SVN	IBM Rational ClearCase	CVS, SVN, GIT
Bug & Issue Tracker	BMC Remedy, IBM ClearQuest, HP Service Manager Help Desk	Bugzilla, JIRA, BugTracker.NET, Gemini	BMC Remedy, IBM ClearQuest, HP Service Manager Help Desk	Bugzilla, JIRA
Test Tools	MS Visual Studio .NET, HP LoadRunner	SW Test Automation Framework	HP LoadRunner	SW Test Automation Framework
Replication / Backup / COOP/DR	MS Windows Backup, MS Exchange Server InterOrg Replication	Amanda, Bacula		Amanda, Bacula
Service & Resource Management	Tivoli Management, HP Operations Manager, BMC Performance Manager, MS Operations Manager	OpenNMS, SevenLayer	Tivoli Management, HP Operations Manager, BMC Performance Manager	Nagios, OpenNMS, SevenLayer, Tripwire
Security Management	MS Active Directory, ArcSight, Symantec Enterprise Security Manager, Norton AntiVirus, Altiris Patch Manager, IBM ISS Internet Scanner, Retina Scanner, Quest Single-Signon for Java	OSSIM, Snort, OpenSSH, Nmap	ArcSight, Symantec Enterprise Security Manager, Norton AntiVirus, Altiris Patch Manager, IBM ISS Internet Scanner, Retina Scanner, Quest Single-Signon for Java	OSSIM, Snort, OpenSSH, Nmap, Tripwire
Service Desk	BMC Remedy, IBM ClearQuest, HP Service Manager Help Desk, JIRA, FogBugz	BugTracker.NET, Bugzilla, Gemini	BMC Remedy, IBM ClearQuest, HP Service Manager Help Desk, JIRA, FogBugz	Best Practical RT
Remote Control	Tivoli Remote Control	EchoVNC	Tivoli Remote Control	EchoVNC

Appendix G. Notional Level of Effort

MITRE used an agile software development approach to create the proof of concept (POC) prototype for the auto-call routing (ACR) use case. MITRE factored into the system administration tasks the primary activities for creating a baseline operating platform with the standard configurations.

The project team includes a task leader, two (2) senior network engineers, two (2) lead systems engineers, and two (2) software developers. The examples included in this cookbook support a specific test plan and use case to demonstrate multiple vendor devices (up to 10) calling a common toll-free number through a SIP Proxy and PBX to a customer service representative in a queue.

The primary ACR use case and configuration is estimated to take 120 hours over a 5-day period to review the design and workflow, configure the initial operating platform, and to test the implementation of the dial plan.

A more complete ACR integration with desktop software, a variety of provider endpoint video devices, and additional dial plan workflow could require 2150 hours for the 7-member project team over a 12-week period. MITRE bases this Level of Effort example on previous experience using open source software, subject matter experts, and access to development and test environments.

For Pubic Release – In Accordance with Data Rights

Appendix H. Essentials of the IPT and IPPD

An IPPD Overview:[15]

Teams are central to the Integrated Process and Process Development (IPPD) process. Teams consist of everyone who has a stake in the outcome or product of the team, including the customer.

Collectively, team members should possess the necessary knowhow and capability to control the requisite resources to produce the proof of concept. Teams are organized and behave to seek the best value solution for the product development.

Development Processes are those activities that lead to both the end product and its associated processes. To ensure efficient use of resources, it is necessary to understand what activities are necessary and how they affect the product and each other. Examples include requirements analysis, configuration management, and detailed design drawings.

Product and Associated Processes include what is produced and provided to the customer. The ultimate measure of the team's success is customer satisfaction with the product's mission effectiveness, as well as operating and support aspects and costs.

Customer is the user and a team member. The customer is the ultimate authority regarding the product. Any changes to the formal requirements driving the product/process development must come through negotiation with the customer.

Figure 3[16] depicts the application of resources, including people, processes, funding, tools, and facilities. The IPPD process reorders decision-making, brings downstream issues to bear earlier and in concert with conceptual and detailed planning, and relies on applying functional expertise in a team-oriented manner. It is necessary to understand early the processes needed to develop, produce, operate, and support the product. Equally important are these processes' impacts on product design and development.[17]

---

15 IT Economics Corporation, 2010.

16 http://astbook.asteriskdocs.org/en/2nd_Edition/asterisk-book-html/figs/web/ast2_1501.png

17 IT Economics Corporation, 2010.

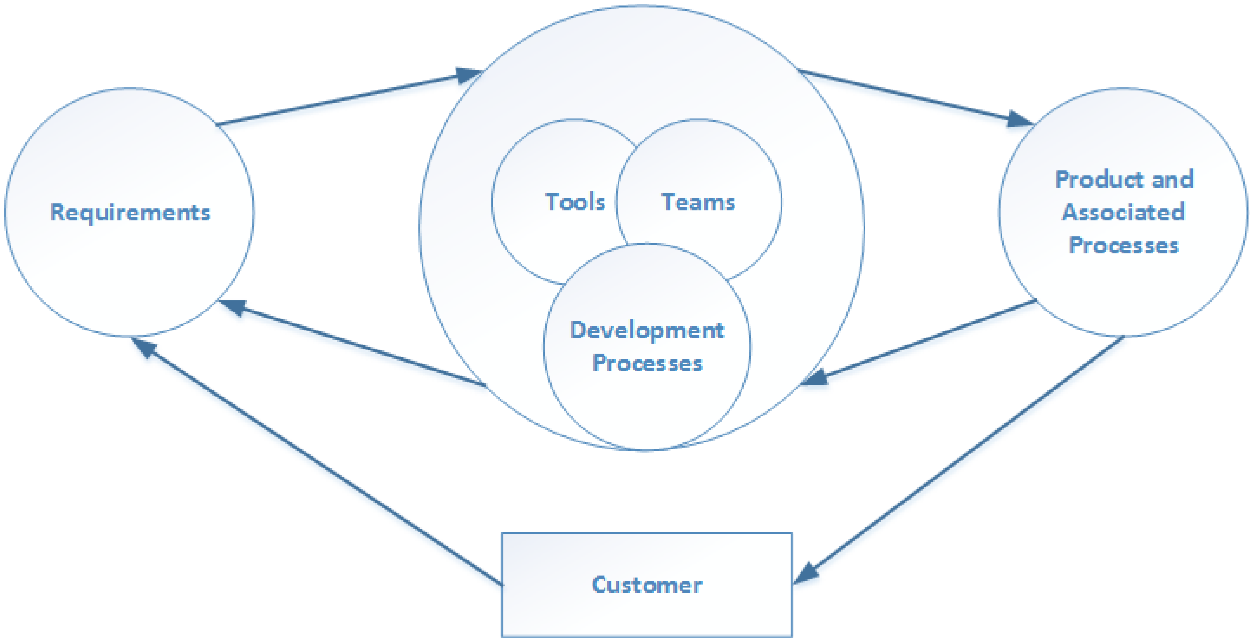

Figure 3. IPPD Iterative Process

Integrated Project Teams (IPT)

Integrated Project Teams are cross-functional teams formed for the specific purpose of delivering a product for an external or internal customer.

IPT members should have complementary skills and be committed to a common purpose, performance objectives, and approach for which they hold themselves mutually accountable.

IPTs are the means for implementing IPPD. IPT members represent technical, manufacturing, business, and support functions and organizations that are critical to developing, procuring, and supporting the product. Having these functions represented concurrently permits teams to consider more and broader alternatives quickly, and in a broader context, enables faster and better decisions.

Once an IPT member joins the team, the member's role changes from a member of a functional organization who focuses on a given discipline to a team member who focuses on a product and its associated processes. Each individual should offer his/her expertise to the team as well as understand and respect the expertise available from other members of the team. Team members work together to achieve the team's objectives.

The following elements are critical to the formation of a successful IPT:

1. All functional disciplines influencing the product throughout its lifetime should be represented on the team.
2. A clear understanding of the team's goals, responsibilities, and authority should be established among the business unit manager, program and functional managers, and the IPT.

3. Identification of resource requirements, such as staffing, funding, and facilities. The team charter can define these requirements and provide additional guidance.

Key Tenets of Integrated Process and Process Development

To implement IPPD effectively, it is important to understand the interrelated tenets inherent in IPPD. The following key tenets are consistent with those found in industry:

- **Customer Focus –** The primary objective of IPPD is to identify and satisfy the customer's needs better, faster, and cheaper. The customer's needs should determine the nature of the product and its associated processes.
- **Concurrent Development of Products and Processes –** Processes should be developed concurrently with the products they support. During product design and deployment, it is critical to consider the processes used to manage, develop, manufacture, verify, test, deploy, operate, support, train people, and eventually dispose of the product. Product and process design and performance should be kept in balance to achieve life-cycle cost and effectiveness objectives. Early integration of design elements can result in lower costs by requiring fewer costly changes late in the development process.
- **Early and Continuous Life Cycle Planning –** Planning for a product and its processes should begin early in the science and technology phase (especially advanced development) and extend throughout every product's life cycle. Early life-cycle planning, which includes customers, functions, and suppliers, lays a solid foundation for the various phases of a product and its processes. Key program activities and events should be defined to better track progress toward achievement of cost-effective targets, apply resources, and understand and manage the impact of problems, resource constraints, and requirements changes.
- **Maximize Flexibility for Optimization and Use of Contractor Approaches –** Requests for Proposals (RFP) and contracts should provide maximum flexibility for employment of IPPD principles and use of contractor processes and commercial specifications, standards, and practices. They should also accommodate changes in requirements and incentivize contractors to challenge requirements and offer alternative solutions that provide cost-effective solutions.
- **Encourage Robust Design and Improved Process Capability –**Advanced design and manufacturing techniques should encourage quality through (1) products with little sensitivity to variations in the manufacturing process (robust design); (2) a focus on process capability; and (3) continuous process improvement. Variability reduction tools, such as ultra-low variation process control similar to "Six Sigma", and lean/agile manufacturing concepts should be encouraged.
- **Event-Driven Scheduling –** A scheduling framework should be established to relate program events to their associated accomplishments and accomplishment criteria. An event is considered complete only when the accomplishments associated with that event have reached completion as measured by the accomplishment criteria. This event-driven scheduling reduces risk by ensuring that product and process maturity are incrementally demonstrated prior to beginning follow-on activities.

For Pubic Release – In Accordance with Data Rights

- **Multidisciplinary Teamwork –** Multidisciplinary teamwork is essential to the integrated and concurrent development of a product and its processes. The right people at the right place at the right time are required to make timely decisions. Team decisions that are the product of risk assessments should be based on the combined input of the entire team (technical, cost, manufacturing, and support functions and organizations) as well as customers and suppliers. Each team member must understand his/her role and support the roles of the other members, as well as understand the constraints under which team members operate. All team members must operate to seek global optima and targets.
- **Empowerment –** Decision-making should be driven to the lowest possible level commensurate with risk. Resources should be allocated to levels consistent with risk assessment authority, responsibility, and the ability of people. The team should be given the authority, responsibility, and resources to manage its product and its risk commensurate with the team's capabilities. The authority of team members should be defined and understood by the individual team members. The team should accept responsibility and be held accountable for the results of its efforts. Organization and team management practices must be team oriented rather than structurally, functionally, or individually oriented.
- **Seamless Management Tools –** A framework should be established that relates products and processes at all levels to demonstrate dependencies and interrelationships. A management system should be established that relates requirements, planning, resource allocation, execution, and program tracking over the product's life cycle. This integrated or dedicated approach helps ensure teams have all available information, thereby enhancing team decision-making at all levels. Capabilities should be provided to share technical, industrial, and business information throughout the product development and deployment life cycle through the use of acquisition and support shared information systems and software tools (including models) for accessing, exchanging, validating, and viewing information.
- **Proactive Identification and Management of Risk –** Critical cost, schedule, and technical parameters related to system characteristics should be identified from risk analyses and user requirements. Technical and business performance measurement plans, with appropriate metrics, should be developed and compared to best-in-class government and industry benchmarks to provide continuing verification of the effectiveness and degree of anticipated and actual achievement of technical and business parameters.

Appendix I. Test Strategy

I.1 Virtual Machine Configuration and Storage Infrastructure

Technical Area: VMC-SI-001

Activity: Determine the data center facility location to host the Proof of Concept activities

Description: An appropriate location for hosting the Proof of Concept must be determined. Options include a temporary solution such as a vendor IaaS cloud computing facility or an Outsourced Vendor Data Center Facility.

Implication: Failure to address this action item may result in an ineffective infrastructure environment to accommodate the Proof of Concept activities and cause more expenditures on Call Center infrastructure, either by adding what is missing or overcoming shortcomings within the site.

Dependency: None

Organization: Facilities Management, Infrastructure Engineering

Technical Area: VMC-SI-002

Activity: Architect and design the initial virtualized infrastructure environment

Description: The initial virtualized infrastructure environment must be designed and architected for the Proof of Concept phase. This virtualized infrastructure environment is designed to meet most of the organization's testing requirements. It must be designed with at least two physical servers and connected to a storage area network (SAN). Storage enhancements technologies, such as primary storage de-duplication, automated tiers, and replication of storage, should be evaluated and implemented. Finally, the virtualized environment should include standard operational tools and capabilities such as the use of agile tools for system deployments and configuration, patching, and backups.

Implication: If this action item is not implemented, there will be an impact to the provisioning of the test environment installation service offering. Specific impacts may include:

- Low degree of de-coupling between components (such as servers and operating systems, and servers and storage), producing less flexibility and preventing agility
- Inability to effectively leverage operational tools, such as agent-based monitoring
- Less than maximum system utilization, which increases the cost of the infrastructure
- Non-uniformity in managing the infrastructure, including:
  - Increased costs by decreasing management's efficiency
  - Increased time spent managing the environment because processes are not standardized, and must be repeated
- Decreased availability (virtualization promotes relatively quick and/or automated fail-over)

Dependency: None

Organization: Infrastructure Engineering, Server Administration

Technical Area: VMC-SI-003

Activity: Architect and design of the optimized infrastructure environment

Description: The optimized infrastructure environment must be designed and architected for the Proof of Concept Phase activities. This type of environment includes optimized testing applications, standard and custom test data, an optimized server infrastructure, and an optimized storage infrastructure. Optimization refers mainly to consolidation of server instances, database instances, and other instances. Optimization also refers to using a limited number of standard Commercial Off-the-Shelf (COTS) software versions running standard configurations.

Implication: Failure to address this action item may result in the inefficient management of testing activities that cannot be executed in an optimized infrastructure environment or non-virtualized environment. For example, virtualization may not be feasible for a vendor of a COTS product that does not support virtualization or the hypervisor of choice.

Dependency: None

Organization: Infrastructure Engineering, Server Administration

Technical Area: VMC-SI-004

Activity: Architect and design of the Production-like Environment

Description: Based on approaches similar to the construction of the virtualized and optimized infrastructure test environments, design and construct a production-like infrastructure environment. This environment mirrors production hardware and configuration characteristics. It is used only where performance of an application must be tested in a production-sized hardware environment. It is necessary to develop procedures to ensure that this testing environment is current with production. Key steps include:

- Architecture of the environment
- Detailed design
- Assessment of current capabilities and gap identification
- Preparation of a Bill of Materials (BOM)
- Acquisition of required hardware and software components
- Establishment of standards, including COTS tools, versions, and their configuration
- Construction and shake-out of the environment
- Deployment processes and procedures development
- Testing and rollout

Implication: Failure to address this action item may result in limited production-like testing support, which is a key component of the infrastructure.

Dependency: None

Organization: Infrastructure Management

Technical Area: VMC-SI-005

Activity: Analyze and allocate existing assets and resources for the Proof of Concept Phase

Description: After completing an analysis and design of both the virtualized infrastructure and optimized environments, an analysis of current IT assets and resources must be conducted to determine which of these assets can be allocated for the Proof of Concept Phase. These assets will be used to provision both the virtualized infrastructure and optimized infrastructure environments. If there is a shortage of IT assets for this buildout, a BOM will be issued (see next task).

Implication: If an accurate re-appropriation of assets is not performed, excessive IT assets and resources may be included in the BOM.

Dependency: None

Organization: Infrastructure Engineering, Server Administration

Technical Area: VMC-SI-006

Activity: Establish appropriate standards such as footprint sizes and technologies

Description: Sizing of the facilities refers to determining the quantities of the hardware, software, network, and other infrastructure components, as well as the size of the call center facility, power and cooling requirements, and other related components required to support the testing efforts. Standards for performing this sizing should be developed using established technology standards, baselined, and placed under configuration management. The organization should determine whether these standards follow the architecture guidelines. If not, the organization should initiate and manage the process to update the architecture guidelines to accommodate standards, or determine suitable alternatives.

Implication: Excess or insufficient capacity among various resources, thereby impacting delivery of service offerings.

Dependency: None

Organization: Infrastructure Engineering

Technical Area: VMC-SI-007

Activity: Create BOM for additional IT assets (if necessary)

Description: If necessary, generate a BOM to acquire additional IT assets and resources for provisioning the Proof of Concept Phase testing environments, i.e., the virtualized infrastructure, optimized infrastructure, and production-like hardware environments.

Implication: If this action item is not implemented, there may be insufficient IT assets for the Proof of Concept testing environment, thus preventing the efficient delivery of service offerings.

Dependency: VMC-SI-005

Organization: Infrastructure Engineering, Server Administration

Technical Area: VMC-SI-008

Activity: Initiate and manage the acquisition process for equipment, services (outsourcing, cloud services, etc.), and COTS tools (automation, service desk, configuration management, etc.)

Description: Execute the acquisition process to procure assets and resources identified in the BOM (see previous actionable item).

Implication: If this action item is not implemented, there may be insufficient IT assets for the Proof of Concept testing environment, thus preventing the efficient delivery of service offerings.

Dependency: VMC-SI-007

Organization: Infrastructure Engineering, Server Administration

Technical Area: VMC-SI-009

Activity: Provision server hardware

Description: Undertake the provisioning of the server hardware as designed for the following testing environments:

- Virtualized Environment
- Optimized Environment
- SAN storage hardware and software
- Networking equipment
- Infrastructure Management tools and products

Implication: If this action item is not implemented, there may be insufficient IT assets for the Proof of Concept testing environment, thus preventing the efficient delivery of service offerings.

Dependency: VMC-SI-002, VMC-SI-004, VMC-SI-005

Organization: Infrastructure Owner

Technical Area: VMC-SI-010

Activity: Develop and implement a training plan to ensure that the staff has the necessary skills for execution of the Proof of Concept (virtualization, storage enhancements, test data preparation, tools, etc.)

Description: Plan, develop, and implement a formal training plan to ensure that staff has the necessary skill sets to support Proof of Concept activities.

Implication: Improperly trained staff may impact the efficient and productive delivery of service offerings.

Dependency: None

Organization: Infrastructure Management

I.2 Test Data Approach

Technical Area: TDS-001

Activity: Identify initial systems to integrate with Proof of Concept

Description: An initial set of test systems must be identified for integration based on the following criteria:

- Business need and priority
- Alignment with Proof of Concept projects and testing activities plan
- Frequency of past systems integrations efforts

Implication: To test system interactions, coordinate the test data across all the systems.

Dependency: None

Organization: Data Administration

Technical Area: TDS-002

Activity: Architect and design the standard test data sets common across the test systems

Description: Architect and design the standard test data sets common across test systems based on the following:

- Memorandum of Understanding (MOU) negotiations and data acquisition
- Interface Control Documents (ICD) – specifications on the types of data exchanged and their formats
- Sized based on inclusion of statistical universes: Where sampling methods are used to select a subset of source data, statistical universes of data for multiple criteria should be considered to ensure a rich sample.
- Sized based on Bounds Inclusion: It is essential that sampled data include data that are at the edge of the allowable limits, and possible data sets outside of the allowable limits, in order to enable effective testing.
- Tools identification, i.e., data scrubbing (de-identification), sampling, simulation, patch level upgrades

Implication: Standard data sets are a prerequisite to providing the standard service offerings, i.e., the environment installation, Standard Integrated Test environment installation, etc.

Dependency: TDS-001

Organization: Data Administration

Technical Area: TDS-003

Activity: Create BOM for additional IT assets (if necessary)

Description: If necessary, generate a BOM to acquire additional IT assets and resources for provisioning the Proof of Concept Phase testing environments.

Implication: If this action item is not implemented, there may be insufficient IT assets for implementing the Proof of Concept test data strategy, thus preventing the efficient delivery of service offerings.

Dependency: TDS-002

Organization: Infrastructure Management

Technical Area: TDS-004

Activity: Initiate and manage the acquisition process for equipment, services (outsourcing, cloud services, etc.), and COTS tools (automation, service desk, configuration management, etc.)

Description: Execute the acquisition process to procure assets and resources identified in the BOM (see previous actionable item).

Implication: If this action item is not implemented, there may be insufficient IT assets for the Proof of Concept testing environment, thus preventing the efficient delivery of service offerings.

Dependency: TDS-003

Organization: Infrastructure Owner

Technical Area: TDS-005

Activity: Operationalize the designed architecture

Description: This activity involves the planning, build, configuration, and testing of the designed architecture (i.e., hardware and software) to operationalize a set of components for the testing environment. Activities include:

- Rollout planning
- Communication, preparation, and training
- Storage of controlled software in a Definitive Software Library
- Build and configure designed architecture
- Operational checklist of hardware and software prior to and following the implementation of:
  - Installation of new or upgraded hardware
  - Identity and access management
  - Security

  - Monitoring
  - Change and Configuration management processes
  - Sign-off of the build for implementation
- Testing to predefined acceptance criteria
- Release, distribution, and the installation of software

Implication: Impacts the ability of the testing facility to deliver service offerings in an efficient, repeatable manner.

Dependency: TDS-002, TDS-003

Organization: Infrastructure Engineering, Infrastructure Management, Server Administration, Data Administration

Technical Area: TDS-006

Activity: Develop virtual server and environment standards

Description: Develop virtual server configuration standards for the virtualized infrastructure and for the following testing environment configurations:

- Development
- SQA Testing
- Prototyping
- Research/Proofs of Concepts
- Modeling and Simulation
- Installation Testing
- Independent Verification and Validation (IV&V) Functional Testing
- IV&V Performance Testing
- Non-functional Testing
- Interoperability Testing
- Disaster Recovery Testing
- Patch Release Testing
- Performance Testing in virtualized environment
- Testing on Production-like Hardware
- Pre-production Testing

Implication: Impacts the ability to deliver service offerings in an efficient, repeatable, common manner.

Dependency: None.

Organization: Infrastructure Engineering

Technical Area: TDS-007

Activity: Develop standard test data sets for specific service offerings

Description: Develop and standardize test data repositories for the following standard service offerings:

- Standard Environment Installation
- Standard Integrated Testing Environment Installation

Implication: Standard data sets are a prerequisite to providing the standard service offerings, i.e., the environment installation, Standard Integrated Test environment installation, etc.

Dependency: None

Organization: Data Administration

Technical Area: TDS-008

Activity: Install testing applications that support the various testing activities

Description: Deploy the testing applications into the testing environment that supports the following testing activities:

- Development
- SQA Testing
- Prototyping
- Research and Proofs of Concept
- Modeling and Simulation
- Installation Testing
- IV&V Functional Testing
- IV&V Performance Testing
- Non-functional Testing
- Interoperability Testing
- Disaster Recovery Testing
- Patch Release Testing
- Testing of Production-Like Hardware
- Pre-production Testing

Implication: Testing applications are computer programs that are required to run in the testing environments to support various testing activities.

Dependency: None

Organization: Server Administration

I.3 Authorization and Authentication

Technical Area: AA-001

Activity: Design and Implement a Suitable Identity Management solution for the Development and Test Infrastructure environments

Description: The organization needs to define an identity management solution that (a) addresses the authorization and authentication of individuals and/or systems to the IT assets and resources, and (b) controls access to these resources within the call center facility. The solution needs to address the issues pertaining to deployments in multiple locations such as the vendor IaaS cloud computing facility and the vendor-outsourced call center facility.

Implication: An identify management solution is an essential component of the infrastructure environment to meet the security expectations.

Dependency: None

Organization: Infrastructure Engineering, Server Administration

I.4 Virtual Machine Capacity Growth

Technical Area: VMCG-001

Activity: Define, actively monitor, and baseline facility performance metrics and generate future forecasts

Description: It is important that the utilization of each resource and service is monitored on an ongoing basis to ensure the optimum use of the hardware and software resources, that all agreed service levels can be achieved, and that business volumes are as expected. The monitors should be specific to particular operating systems, hardware configurations, and applications. The infrastructure and tools need to be available and deployed to support capacity management. Typical monitored data includes:

- CPU, memory utilization
- % CPU per transaction type
- Input/Output (IO) rates (physical and buffer) and device utilization
- Transactions and transactions per second (maximum and average)
- Transaction response time

Implication: It is important that the monitors can collect all the data required to generate forecasts of resources for the Transition state.

Dependency: None

Organization: Infrastructure Management, Server Administration

I.5 Call Center Deployment Plan

Technical Area: P-CCDP-002

Activity: Explore options and establish feasibility for a Vendor Outsourced Call Center Facility

Description: In this task, outsourced vendor-provided call center facility options are explored. If a viable option is determined, a select set of projects will be deployed within the Vendor Outsourced Call Center (standard service offerings only) along with identifying and mitigating any Center (standard service offerings only) along with identifying and mitigating any operational issues discovered. The following tasks are necessary:

- Develop an initial list of vendors and their facilities based on broad market research
- Develop a final list of small number of vendors based on checklist assessment criterion
- Develop a deployment architecture for the facility
- Define outsourced call center performance metrics
- Conduct acquisition of suitable contracts, etc.
- Develop a project plan that addresses:
  - Define and negotiate Service Level Agreements (SLA)
  - Define Standard Operating Procedures (SOP)
  - Conduct Certification & Accreditation activities and ensure compliance with enterprise security standards, policy, and requirements
  - Conduct a post-implementation review
- Implement the plan and ensure that a presence is established that meets a subset of the requirements of test environments in a vendor call center facility
- Local Area Network (LAN)/Wide Area Network (WAN) bandwidth requirements (LAN should be upgraded to 1GB Ethernet as needed)
- Evaluate and deploy WAN accelerator technologies and workload partitioning techniques as described in the Physical/Logical Architecture document to support geographically distributed deployments, if necessary
- Evaluate additional vendor facility capabilities such as servers and storage

Implication: Direct impact of testing infrastructure call center performance and service delivery

Dependency: None

Organization: Infrastructure Management

Technical Area: P-CCDP-003

Activity: Explore options and establish feasibility for an Infrastructure as a Service Cloud Computing Facility

Description: Vendor-provided IaaS cloud computing facilities are investigated to support the testing infrastructure environments. Given the sensitive nature of some of the test data, security offered by the vendor IaaS clouds would have to be commensurate with the security policies. IaaS cloud facilities would imply restrictions on the choice of hardware, hypervisor, and operating system platforms. If a viable option is determined, a select set of projects will be deployed within the IaaS facility (standard service offerings only) along with identifying and mitigating any operational issues discovered. The following specific tasks are necessary:

- Develop an initial list of vendors and their facilities based on broad market research
- Develop a final list of small number of vendors based on checklist criterion
- Develop a deployment architecture for the joint facility
- Define outsourced call center performance metrics
- Conduct acquisition of suitable contracts, etc.
- Develop project plan that addresses:
  - Define and negotiate SLAs
  - Define security requirements
  - Define SOPs
  - Conduct Certification & Accreditation activities and ensure compliance with enterprise security standards, policy, and requirements
  - Conduct a post-implementation review
- Determine the LAN/WAN bandwidth requirements
- Implement the plan and ensure that a presence is established that meets a subset of the requirements of test environments in a vendor call center facility

Implication: Direct impact of testing infrastructure call center performance and service delivery

Dependency: None

Organization: Infrastructure Management

I.6 Change and Configuration Management Tasks

Technical Area: CM-001

Activity: Develop and implement configuration management (CM) policy and procedures

Description: Configuration Management provides a logical model of the infrastructure or a service by identifying, controlling, maintaining, and verifying the versions of Configuration Items (CI) in existence. The goals of Configuration Management are to:

- Account for all the IT assets and configurations within the scope of the proof of concept
- Provide accurate information on configurations and their documentation to support all the other Service Management processes

- Provide a sound basis for Change Management and Release Management
- Verify the configuration records against the infrastructure and resolve any exceptions

Implication: Configuration Management processes and procedures are a best practice and help organizations effectively control their IT infrastructure and services.

Dependency: None

Organization: Configuration Management

Technical Area: CM-002

Activity: Develop and implement a change management policy and procedures

Description: A Change Management process, including the procedures, tools, and dependencies, should be developed and implemented. Change Management, which should be planned concurrently with Configuration Management, is responsible for managing changes within the environment involving:

- Hardware
- Communications equipment and software
- System software
- All documentation and procedures associated with operating, supporting, and maintaining the proof of concept

Implication: Change Management processes and procedures are a best practice and help organizations effectively control their IT infrastructure and services in tandem with Configuration Management.

Dependency: None

Organization: Change Management

I.7 Testing Facility Service Desk Tasks

Technical Area: TFSD-001

Activity: Determine Service Desk model, whether local, central, or virtual, and the organizational structure, i.e., location(s), number of customers to be supported, working or support hours, etc.

Description: Consider three types of structures for optimum usage:

- Local Service Desk
- Central Service Desk
- Virtual Service Desk

Traditionally, organizations have created local support desks to meet local needs. This is practicable until the organization has multiple locations requiring support services. For the central service desk, multiple locations are supported and all service requests are logged to a central physical location. Finally, if there are no preferences or compelling needs for physical location of the Service Desk and the associated services, then a virtual service desk model can be situated and accessed from anywhere.

Implication: A Service Desk offers the following benefits to the organization:

- Provides a strategic function to identify and lower the cost of ownership for supporting the computing and support infrastructure
- Supports the integration and management of change across distributed business, technology, and process boundaries
- Reduces costs by the efficient use of resource and technology
- Supports the optimization of investments and the management of the businesses support services
- Helps to ensure long-term customer retention and satisfaction
- Assists in the identification of business opportunities

Dependency: None

Organization: Testing Facility Service Desk

Technical Area: TFSD-002

Activity: Develop Service Desk standard operating procedures

Description: Standard Operating Procedures for the Service Desk needed to be developed. Specifically, SOPs should address the following:

- Common structured interrogation technique for the support organization. A common structured dialogue to managing customer requests is essential, irrespective of who responds to the customer.
- Escalation and prioritization criteria for Service Requests
- Customer details and identification, i.e., correct and unique customer identification is essential to ensure that customer details, existing requests, and management information are uniquely and easily referenced
- Management of the customer database
- Reference materials used by customers and support staff are well maintained, kept up-to-date, and regularly reviewed, including:
  - Training manuals
  - Lists of Known Errors and solutions
  - Product and application documentation
  - Hardware documentation

  - Knowledge bases
  - Command procedures, scripts, and programs

Implication: Standard SOPs are essential for the efficient, high-quality support of the infrastructure and customers. When disparate and distributed architectures are combined, often in a piecemeal approach, the management and support of such an environment becomes very expensive, time consuming, and frequently an exercise in futility.

Dependency: None

Organization: Testing Facility Service Desk

Technical Area: TFSD-003

Activity: Design and implement Testing Facility Service Desk infrastructure

Description: The Testing Facility Service Desk technical solution must be designed and developed. A number of technologies are available to assist the Service Desk, each with its advantages and drawbacks. It is important to ensure that the blend of technology, process, and Service Desk staff will meet the needs of the business and the service offerings provided. Technology should complement and enhance service, not replace it. Technology must support business processes, and be adaptable to current and future demands. It is also important to understand that automation carries an increased need for discipline and accountability. Service Desk technologies should therefore include:

- Integrated Service Management and Operations Management systems
- Advanced telephone systems [e.g., auto-routing, hunt groups, Computer Telephony Integration (CTI), Voice Over Internet Protocol (VOIP)]
- Interactive Voice Response systems
- Knowledge, search and diagnostic tools
- Automated operations and Network Management tools.

Implication: Many support functions start as paper-based systems, with individuals recording and updating details and solutions. However well-defined the processes, procedures, and documentation are, it is not possible to do more than log incidents and track them until they are completed. Computerized Service Desk tools are essential for the modern support operation. Electronic management facilitates improved efficiency, accuracy, and fast access to past solutions, known errors, call histories, and management information.

Dependency: None

Organization: Testing Facility Service Desk

Technical Area: TFSD-004

Activity: Develop and publish a Service Catalog

Description: A formal Service Catalog should be developed. The Service Catalog defines the services the organization provides to the customer along with agreed-upon expectations.

Implication: The introduction of the Service Catalog should increase customer awareness and to contribute to demand of service offerings.

Dependency: None

Organization: Testing Facility Service Desk

For Pubic Release – In Accordance with Data Rights

Appendix J. Sample Test Matrix

MITRE Site

Participants:

Connection: 3 Outernet drops

Devices: 10 ZVRS devices, 1 laptop with webcam capabilities

FCC Site

Participants:

Connection: DSL line

Devices: 2 laptops with Bria softphone (emulating 2 customer service representatives (CSRs), laptop with call center monitoring panel, laptop with Asterisk CLI console, router, laptop to display the webcam stream from test lab. Note: total of 5 laptops

Demo Scenario

- FCC Site
  - On CSR's Laptop 1 (CSR 1):
    - CSR logs in to the Complaints queue as an available representative by dialing 10
    - CSR logs in to the General Questions queue as an available representative by dialing 20
  - On CSR's Laptop 1 (CSR 2):
    - CSR logs in to the Complaints queue as an available representative by dialing 10
  - Call Center Monitoring panel:
    - Displays CSR 1 available in both Complaints queue and General Questions queue
    - Displays CSR 2 available in Complaints queue only
    - MITRE site
  - ZVRS devices used to place all calls (703 570 5868)
    - ZVRS device 1 dialed, option 4 selected
    - ZVRS device 2 dialed, option 4 selected
    - ZVRS device 3 dialed, option 5 selected
    - ZVRS device 4 dialed, option 4 selected
    - ZVRS device 5 dialed, option 4 selected
    - ZVRS device 6 dialed, option 4 selected
    - ZVRS device 7 dialed, option 5 selected
    - ZVRS device 8 dialed, option 4 selected
    - ZVRS device 9 dialed, option 5 selected

  - ZVRS device 10 dialed, option 4 selected
  - FCC site
- CSR 1 answers next incoming call, connected to device 1
- CSR 2 answers next incoming call, connected to device 3
- Devices 2, 4, 5, 6, 8 and 10 can be observed waiting in Complaints queue
- Devices 7 and 9 can be observed waiting in General Questions queue
  - MITRE site
- Device 1 is disconnected from CSR 1
- Device 3 is disconnected from CSR 2
  - FCC Site
- CSR 1 answers next incoming call, connected to device 2
- CSR 2 answers next incoming call, connected to device 7
- Devices
- Devices 4, 5, 6, 8 and 10 can be observed waiting in Complaints queue
- Device 9 can be observed waiting in General Questions queue
  - MITRE site
- Device 1 is disconnected from CSR 2
- Device 3 is disconnected from CSR 7
  - FCC Site
- CSR 1 answers next incoming call, connected to device 4
- CSR 2 answers next incoming call, connected to device 9
- Devices 5, 6, 8 and 10 can be observed waiting in Complaints queue
- 0 devices can be observed waiting in General Questions queue

Table 9 is an example of how to add a traceable set of test scenarios and track the Pass/Fail status.

Table 9. Sample Test Matrix

#	Dependency	Overview	Originating Device	Connection Type	Destination Device	Test Case	Expected Result	Actual Result	Pass/Fail
1		No Agent in queue	ZVRS (Z20)	Wired Outernet	AWS ACD	Dia (703) 570 5868 Wait for the video to finish playing Select Option 4	Video is played notifying the user that no agents available Video is played "Goodbye" Call disconnected	Video is played notifying the user that no ASL agents available Video is played "Goodbye" Call disconnected	Pass
2		CSR1 logs in to the General Questtons queue	PC (Windows)	Wired Outernet	AWS ACD	Dial *10	ASL agent is logged in to the Complaints queue Video is played notifying ASL agent that he/she has been logged in to the queue	ASL agent is logged in to the Complaints queue Video is played notifying ASL agent that he/she has been logged in to the queue	Pass
3		CSR2 logs in to the Complaint queue	PC (Windows)	Wired Outernet	AWS ACD	Dial *20	ASL agent is logged in to the Complaints queue Video is played notifying ASL agent that he/she has been logged in to the queue	ASL agent is logged in to the Complaints queue Video is played notifying ASL agent that he/she has been logged in to the queue	Pass
4	1	Device 1 Call	ZVRS (Z20)	Wired Outernet	CSR1 device	Dial (703) 570 5868 Wait for the video to finish playing Select Option 4	Video with number 4 is played back Video notifying the user that they have selected General Questions queue is played. User is placed in the queue to wait for the next available CSR	Video with number 4 is played back Video notifying the user that they have selected General Questions queue is played. User is placed in the queue to wait for the next available CSR	

#	Dependency	Overview	Originating Device	Connection Type	Destination Device	Test Case	Expected Result	Actual Result	Pass/Fail
5	2	Device 2 Call	ZVRS (Z20)	Wired Outernet	AWS ACD	Dial (703) 570 5868 Wait for the video to finish playing Select Option 5	Video with number 5 is played back Video notifying the user that they have selected Complaints+AA2 queue is played. User is placed in the queue to wait for the next available CSR	Video with number 5 is played back Video notifying the user that they have selected Complaints+ A2 queue is played. User is placed in the queue to wait for the next available CSR	Pass
6	3	Device 3 Call	ZVRS (Z20)	Wired Outernet	AWS ACD	Dial (703) 570 5868 Wait for the video to finish playing Select Option 4	Video with number 4 is played back Video notifying the user that they have selected General Questions queue is played. User 1 is placed in the queue to wait for the next available representative	Video with number 4 is played back Video notifying the user that they have selected General Questions queue is played. User 1 is placed in the queue to wait for the next available representative	Pass
7	4	Device 4 Call	ZVRS (Z20)	Wired Outernet	AWS ACD	Dial (703) 570 5868 Wait for the video to finish playing Select Option 5	Video with number 5 is played back Video notifying the user that they have selected Complaints queue is played. User 1 is placed in the queue to wait for the next available representative	Video with number 5 is played back Video notifying the user that they have selected Complaints queue is played. User 1 is placed in the queue to wait for the next available representative	Pass

#	Dependency	Overview	Originating Device	Connection Type	Destination Device	Test Case	Expected Result	Actual Result	Pass/Fail
8	4	Device 5 Call	ZVRS (Z20)	Wired Outernet	AWS ACD	Dial (703) 570 5868 Wait for the video to finish playing Select Option 5	Video with number 5 is played back Video notifying the user that they have selected Complaints queue is played. User 1 is placed in the queue to wait for the next available representative	Video with number 5 is played back Video notifying the user that they have selected Complaints queue is played. User 1 is placed in the queue to wait for the next available representative	Pass
9	3	Device 6 Call	ZVRS (Z20)	Wired Outernet	AWS ACD	Dial (703) 570 5868 Wait for the video to finish playing Select Option 4	Video with number 4 is played back Video notifying the user that they have selected General Questions queue is played. User 1 is placed in the queue to wait for the next available representative	Video with number 4 is played back Video notifying the user that they have selected General Questions queue is played. User 1 is placed in the queue to wait for the next available representative	Pass
10	5	Device 1 ends the call	ZVRS (Z20)	Wired Outernet	AWS ACD	Device 1 ends the call	Device 3 is connected to a representative in General Questions Queue	Device 3 is connected to a representative in General Questions Queue	Pass

#	Dependency	Overview	Originating Device	Connection Type	Destination Device	Test Case	Expected Result	Actual Result	Pass/Fail
11		Device 7 Call	ZVRS (Z20)	Wired Outernet	AWS ACD	Dial (703) 570 5868 Wait for the video to finish playing Select Option 4	Video with number 5 is played back Video notifying the user that they have selected General Questions queue is played. User 1 is connected to the next available CSR	Video with number 5 is played back Video notifying the user that they have selected General Questions queue is played. User 1 is connected to the next available CSR	Pass
12	6	Device 2 ends the call	ZVRS (Z20)	Wired Outernet	AWS ACD	Device 2 ends the call	Device 4 is connected to a representative in General Questions Queue	Device 4 is connected to a representative in General Questions Queue	Pass
13		Device 8 Call	ZVRS (Z20)	Wired Outernet	AWS ACD	Dial (703) 570 5868 Wait for the video to finish playing Select Option 4	Video with number 5 is played back Video notifying the user that they have selected General Questions queue is played. User 1 is connected to the next available CSR	Video with number 5 is played back Video notifying the user that they have selected General Questions queue is played. User 1 is connected to the next available CSR	

#	Dependency	Overview	Originating Device	Connection Type	Destination Device	Test Case	Expected Result	Actual Result	Pass/Fail
14		Device 9 Call	ZVRS (Z20)	Wired Outernet	AWS ACD	Dial (703) 570 5868 Wait for the video to finish playing Select Option 4	Video with number 5 is played back Video notifying the user that they have selected General Questions queue is played. User 1 is connected to the next available CSR	Video with number 5 is played back Video notifying the user that they have selected General Questions queue is played. User 1 is connected to the next available CSR	
15		Device 10 Call	ZVRS (Z20)	Wired Outernet	AWS ACD	Dial (703) 570 5868 Wait for the video to finish playing Select Option 5	Video with number 5 is played back Video notifying the user that they have selected General Questions queue is played. User 1 is connected to the next available CSR	Video with number 5 is played back Video notifying the user that they have selected General Questions queue is played. User 1 is connected to the next available CS	

Acronyms

ACD	Automatic Call Distribution
AMI	Asterisk Manager Interface
ASL	American Sign Language
API	Application Programming Interface
BIA	Business Impact Analysis
BCM	Business Continuity Management
BOM	Bill of Materials
BRD	Business Requirements Document
BRI	Basic Rate Interface
C	Connectivity
CAMH	CMS Alliance to Modernize Healthcare
CAB	Change Advisory Board
CCB	Change Control Board
CCDP	Call Center Deployment Plan
CDR	Call Detail Record
CLI	Command Line Interface
CM	Configuration Management
CMS	Centers for Medicare & Medicaid Services
COOP	Continuity of Operations Plan
COTS	Commercial Off-the-Shelf
CPU	Central Processing Unit
CR	Change Request
CSR	Customer Service Representative
CTI	Computer Telephony Integration
DASD	Direct Access Storage Drive
DID	Direct Inward Dialing
DNS	Domain Name Service
DR	Disaster Recovery

DRBC	Disaster Recovery and Business Continuity
DTMF	Dual Tone Multi-Frequency
DVC	Direct Video Connectivity
E2E	End to End
ESB	Enterprise Service Bus
FCC	Federal Communications Commission
FFRDC	Federally Funded Research and Development Center
FQDN	Fully Qualified Domain Name
GB	Gigabyte
HHS	Department of Health and Human Services
HTTP	Hypertext Transport Protocol
HTTPS	Hypertext Transport Protocol Secure
IaaS	Infrastructure as a Service
ICD	Interface Control Document
ICE	Interactive Connectivity Establishment
IDE	Integrated Development Environment
IETF	Internet Engineering Task Force
IO	Input Output
IOC	Initial Operating Capability
IP	Internet Protocol
IPPD	Integrated Process and Process Development
IPT	Integrated Product Team
IT	Information Technology
ITSCM	IT Service Continuity Management
IVR	Interactive Voice Response
IV&V	Independent Verification and Validation
JDBC	Java Database Connectivity
JSP	Java Server Pages
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol

MAN	Metropolitan Area Network
MOU	Memorandum of Understanding
NAT	Network Address Translation
NOC	Network Operations Center
NVA	Non-Value Added
O&M	Operations and Maintenance
ODBC	Open Database Connectivity
ORM	Operational Requirements Matrix
ORR	Operational Readiness Review
OS	Operating System
PBX	Private Branch Exchange
PCMA	Pulse Code Modulation alaw
PMO	Program Management Organization
POC	Proof of Concept
PRI	Primary Rate Interface
PSTN	Public Switched Telephone Network
RD	Requirements and Design
RDBMS	Relational Database Management System
RFC	Request for Comments
RFP	Request for Proposal
ROI	Return on Investment
RTP	Real-Time Protocol
RTT	Real Time Text
S	Security
SAN	Storage Area Network
SCC	Service Call Center
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SME	Subject Matter Expert
SOP	Standard Operating Procedure

SOW	Statement of Work
SSN	Social Security Number
STUN	Session Traversal Utilities for NAT
TB	Terabyte
TDS	Testing Data Strategy
TFSD	Testing Facility Service Desk
TRS	Telecommunications Relay Service
TTD	Tele Typing Device
TTY	Teletypewriter
TURN	Traversal Using Relays around NAT
ULAW	μ-law Algorithm
URI	Universal Resource I
USB	Universal Serial Bus
VM	Virtual Machine
VMC	Virtual Machine Configuration
VMCG	Virtual Machine Capability Growth
VOIP	Voice Over Internet Protocol
VRS	Video Relay Service
WAN	Wide Area Network
XMPP	Extensible Messaging and Presence Protocol

List of References

1. Asterisk™: The Definitive Guide, Fourth Edition, by Russell Bryant, Leif Madsen, and Jim Van Meggelen, Copyright 2013
2. The Asterisk Cookbook, by Russell Bryant, Leif Madsen, Copyright 2013
3. The Asterisk Guru, http://www.asteriskguru.com/
4. Asterisk, The Definitive Guide, http://asteriskdocs.org/en/3rd_Edition/asterisk-book-html-chunk/index.html
5. Asterisk Quick Start Guide: http://www.asterisk.org/sites/asterisk/files/mce_files/documents/asterisk_quick_start_guide.pdf
6. Asterisk Step-by-step Installation, http://www.voip-info.org/wiki/view/Asterisk+Step-by-step+Installation
7. Asterisk, http://www.asterisk.org/
8. iTRS ENUM Database, https://www.neustar.biz/services/network-routing-and-addressing/itrs-enum-database
9. Asterisk Project Documentation, https://wiki.asterisk.org/wiki/display/AST/Include+Statements
10. Current Asterisk News, http://www.venturevoip.com/news.php
11. Appendix A - Internet Engineering Task Force (IETF) Request for Comments (RFC)