



Extending Production Speech Experience to Voice Search Applications

Dr. Valentine C. Matula

vmatula@avaya.com

+1.614.825.6415

IP Telephony

Contact Centers

Mobility

Services

Delivering High Volume Speech Applications

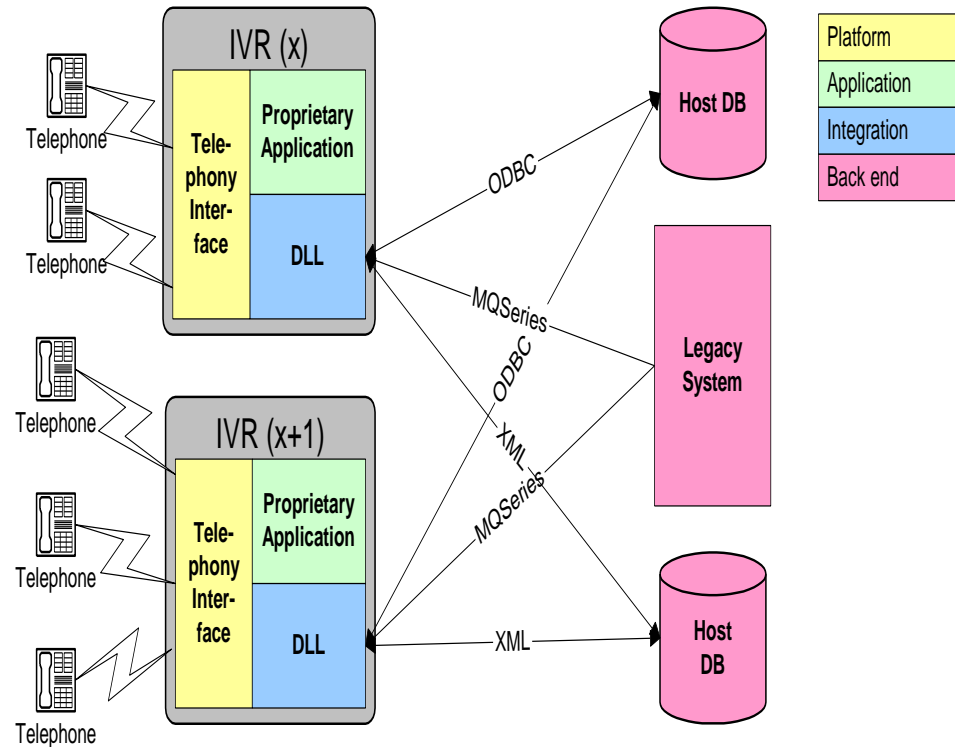
- Learn lessons from basic architectures used for simple speech applications (back when they were hard)
 - An architecture that works
 - Several that don't, and why
 - The five parts of speech application deployment
 - Planning for completely uncontrollable, unexpected success
 - ...and planning for utter disaster
- The flexibility of control
- Instrumentation, Measurements, and Reporting
- Examples from simple speech, complex speech, video deployments...
- Summary and direction

The Evolution of IVR Self-Service

1980's – 1999

- Basic touch tone IVR
- Off loading basic repetitive requests for information
- Isolation from other customer facing applications
- Hard coded backend integration to corporate sources of information
- Proprietary platform and development environments
- IT duplication

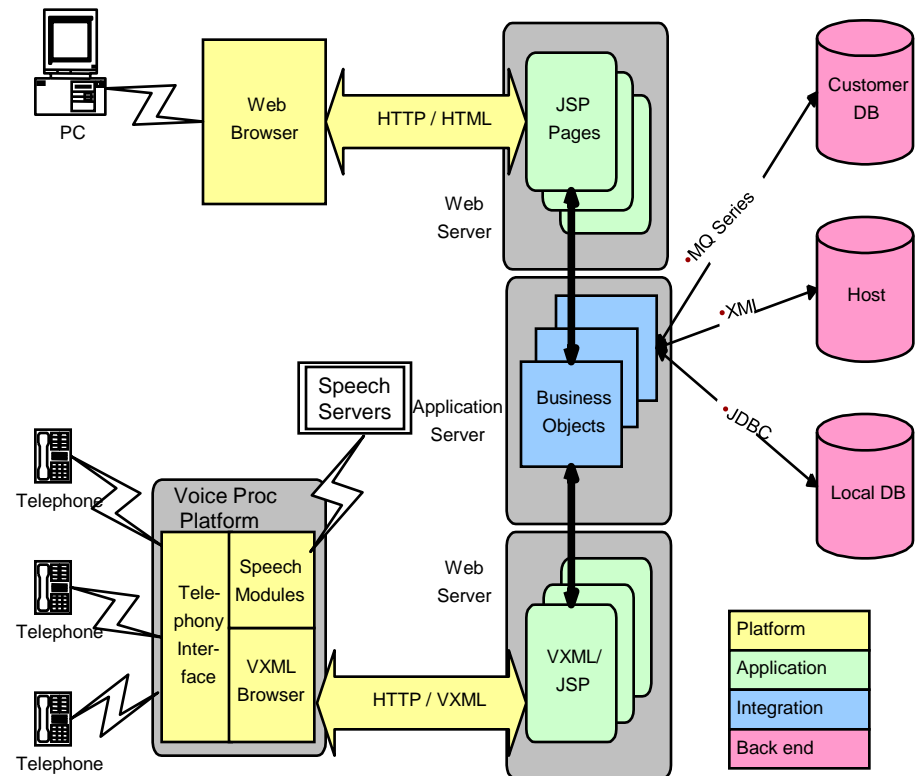
Legacy IVR Environment



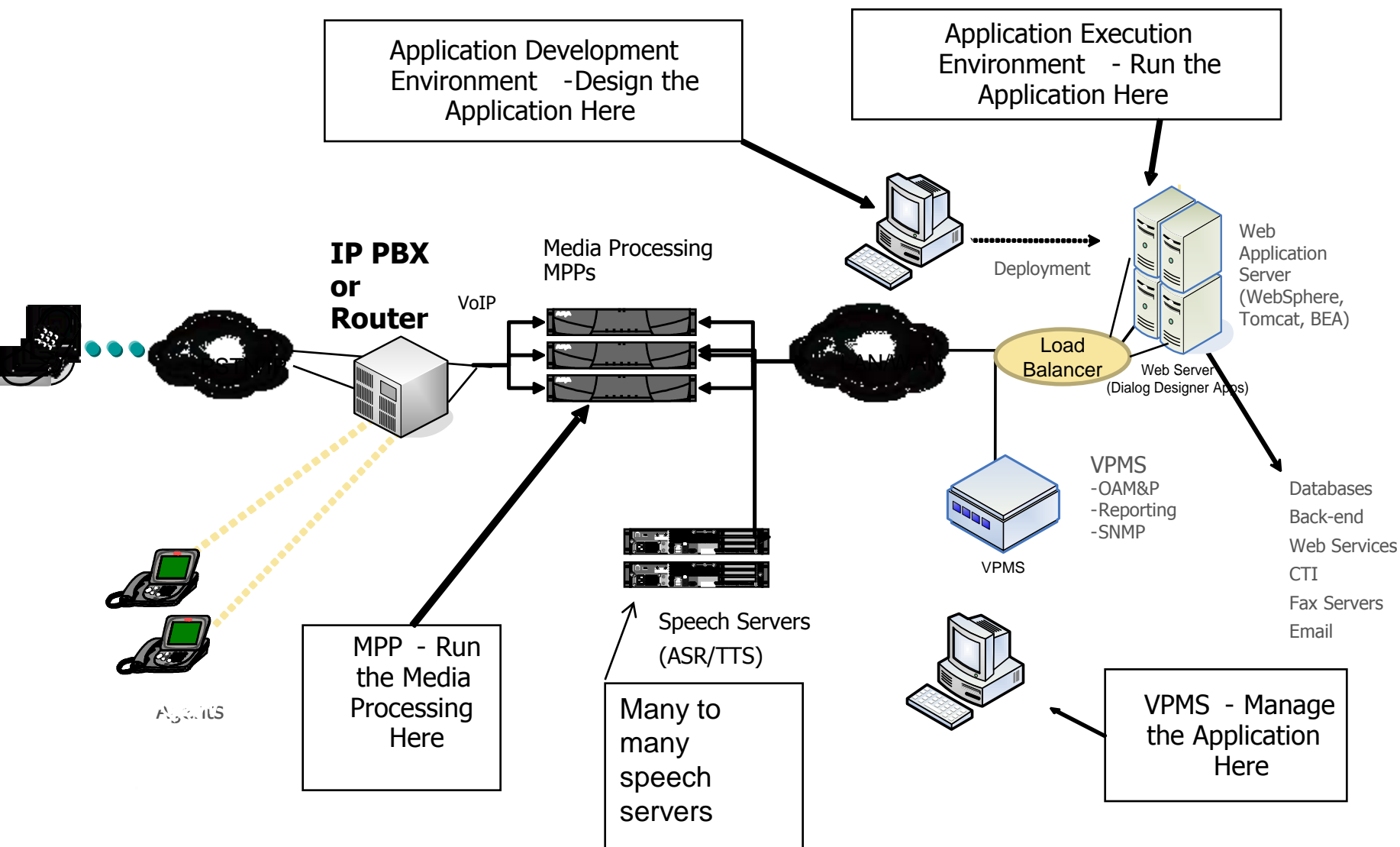
The Evolution of IVR Self-Service

● 2000 – Present

- Speech is a preferred user interface
- Web world has created self service tools, standards and infrastructure
- Web Services and Voice XML allow speech applications to leverage web investments
- Enable reuse of:
 - Infrastructure
 - Applications
 - Integrations
 - Programmers
- Capabilities such as web services, CTI, and CCXML editing in application development environment (ADE)



Voice Processing Architecture – Physical View



The Key Components of Voice Processing

A modern voice processing system integrates into a web-based IT infrastructure. It is a “software-only” system, with the following key components:

- **Voice Traffic Load / Reliability Balancer**
- **Media Processing Platform (MPP)**
 - VoiceXML Browser and media handling
 - Scales based on # of ports
- **Speech Servers**
 - From several vendors for capability, language, and price choices
- **Application Execution Environment (AEE)**
 - Web Application Server for hosting applications
 - Code to generate VoiceXML dynamically to send to MPPs
 - HTML Traffic Load Balancer
- **Voice Processing Management System (VPMS)**
 - Single point of administration, management and reporting
- **Application Development Environment (ADE)**

Desirable Attributes

● Standards Compliance

- VoiceXML 2.0/2.1, CCXML, MRCP, H.323 & SIP
- Software only Solution on Commodity servers,
- Service Oriented Architecture

● Highly Scalable Modular Architecture

- Dynamic Licensing, Dynamic Registration of Endpoints
- Scale from 1 to thousands of ports.
- Modular “Appliance like” media processing platform on commodity hardware
- Architecture options supporting separate speech processors for heavy ASR
- Single point for administration, management and reporting.

● Reliable Architecture

- Voice traffic load balancing
- Web traffic load balancing

● Common Components

- Development Tools, Licensing, Management, Disaster Recovery Mechanism
- Web Services for CTI, management, centralized reporting, & IT interaction

How to Help Get Back Your Future Weekends:

Failure Case Testing

- Pull a plug, any plug...
 - Power
 - Network connectivity
 - System / license server path
 - Any CPU process: Kill it, what happens?
- Post power-fail restart
 - All processors restart all at one time, unsequenced
 - All calls arrive nearly simultaneously and start “in parallel”
 - Extreme synchronicity stresses all systems
- License expiration tests
- Extreme traffic tests
 - Long call hold time, long prompts (8 minutes?), long ASR open mike
 - Short calls (e.g., 6 seconds, hang up, do it again...)
 - Twice or thrice the call volume—what happens when traffic is removed?
 - FAX calls (does the recognizer peg the CPU?)

SOA: What is this?



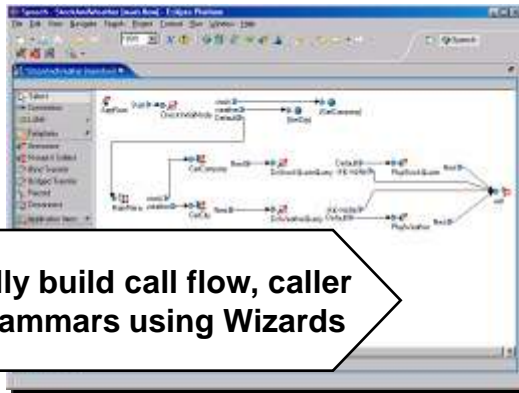
- **Service** = a function performed by a software object
- **Web service** = software function with an XML interface transported by SOAP
- **SOA** (service oriented architecture) = creation of software by publishing and consuming services
- “*An old idea whose time has come*” = similar to earlier object oriented efforts like CORBA; use of web standards and the penetration of the web have “turned the corner” on acceptance

Where A Web Services Architecture Can Help

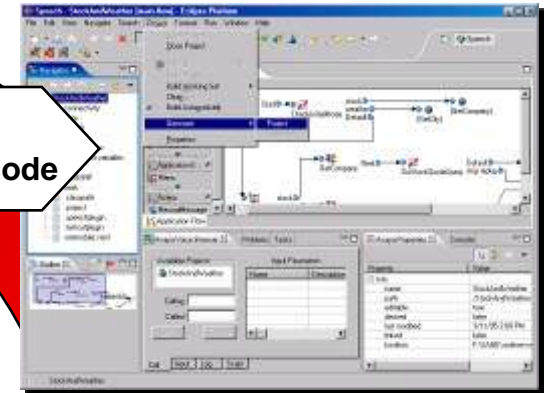
- Example: Published web services interface to invoke CCXML
 - To launch and /or control new calls via CCXML scripting
- Most enterprise systems expose Web Services for use by other applications (include voice search applications).
- Using IT Web Services from an webserver-based voice search application can remove the need (temptation?) for proprietary tags or extensions to VoiceXML

Design, Develop, Simulate, Deploy, Manage

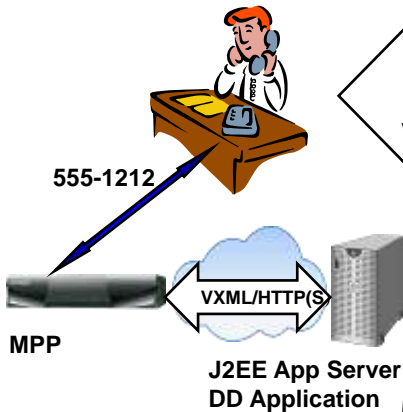
1. Graphically build call flow, caller prompts, grammars using Wizards



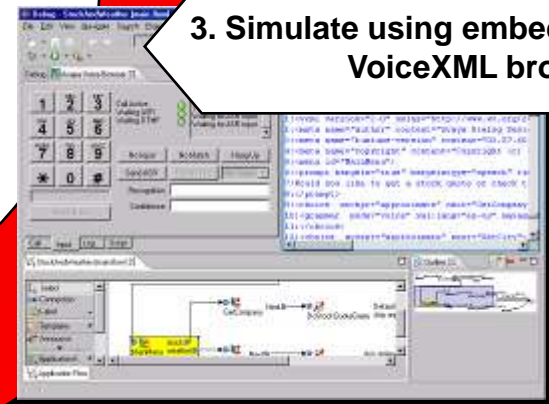
2. Generate application code



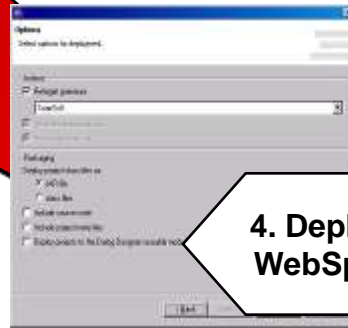
5. Servlet generates VoiceXML, which is processed on Voice Processing Platform (MPP)



3. Simulate using embedded VoiceXML browser

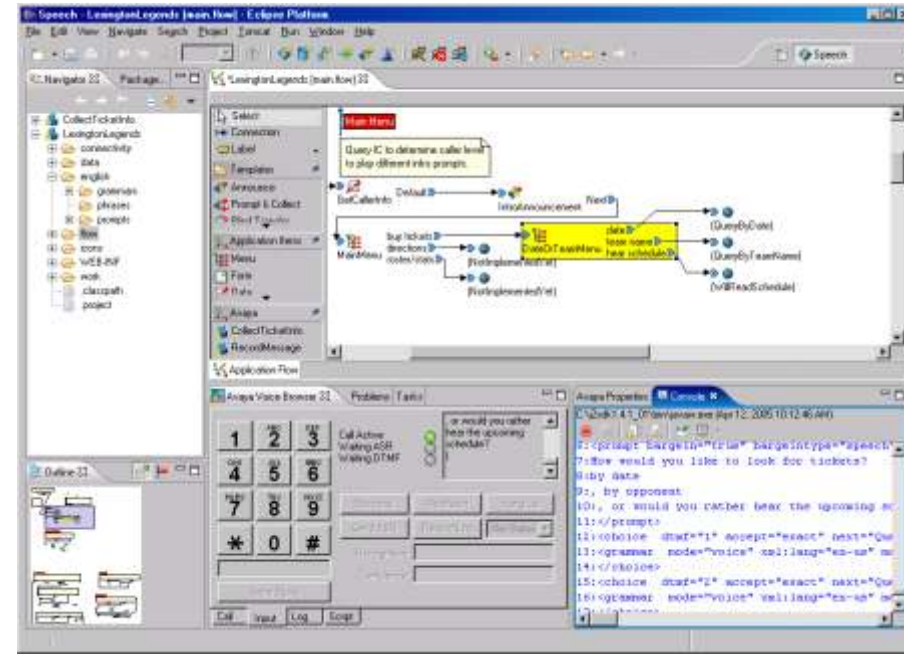


4. Deploy Java servlet to J2EE WebSphere, Weblogic, Tomcat



Desirable Application Development Environment Attributes

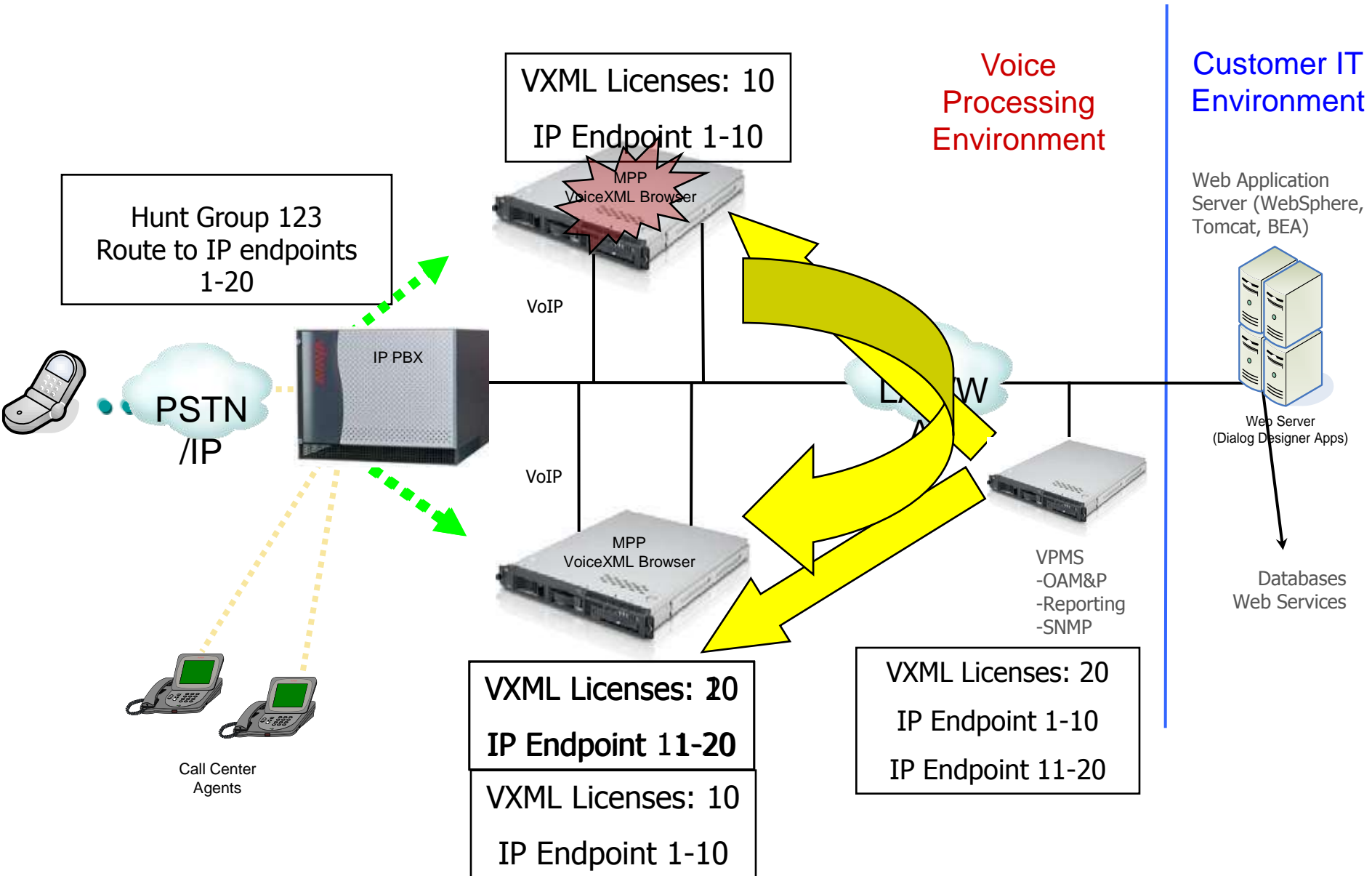
- Speech application development, testing, deployment
- Live Call-flow Highlighting
- Debugging and Scripting
- Desktop or dedicated lab unit VoiceXML browser support
- Prompt & Grammar Management
- Web Services via WSDL integration wizard
- Option for “breadcrumb” data reporting



File Name:	reversetelephonelookup	<input checked="" type="checkbox"/> Open file for editing
WSDL URL:	http://sage-linux.sv.avaya.com:8080/axis/services/LookUpF	
Operation:	[Dropdown menu]	
Package for types:	lookup(string) - LookUpResult connectivity.ws.beans	
<input type="button" value="OK"/> <input type="button" value="Cancel"/>		



Dynamic Licensing



Delivering High Volume Speech Applications

- Modularity so that you can start small, and then grow the components that need the additional resources
- Test the production aspects, not just the interface and flow
- “Software-only” means you can buy the hottest processors of the day
- The three-state application environment separates voice processing from data crunching—and web services enables the separation
- Avoiding proprietary tags, links, etc. for short term savings gives you long term opportunities
- Disaster recovery and flexible deployment saves weekends
- Don't forget Instrumentation, Measurements, and Reporting