WEB SERVICES FRAMEWORK FOR SPEECHSC

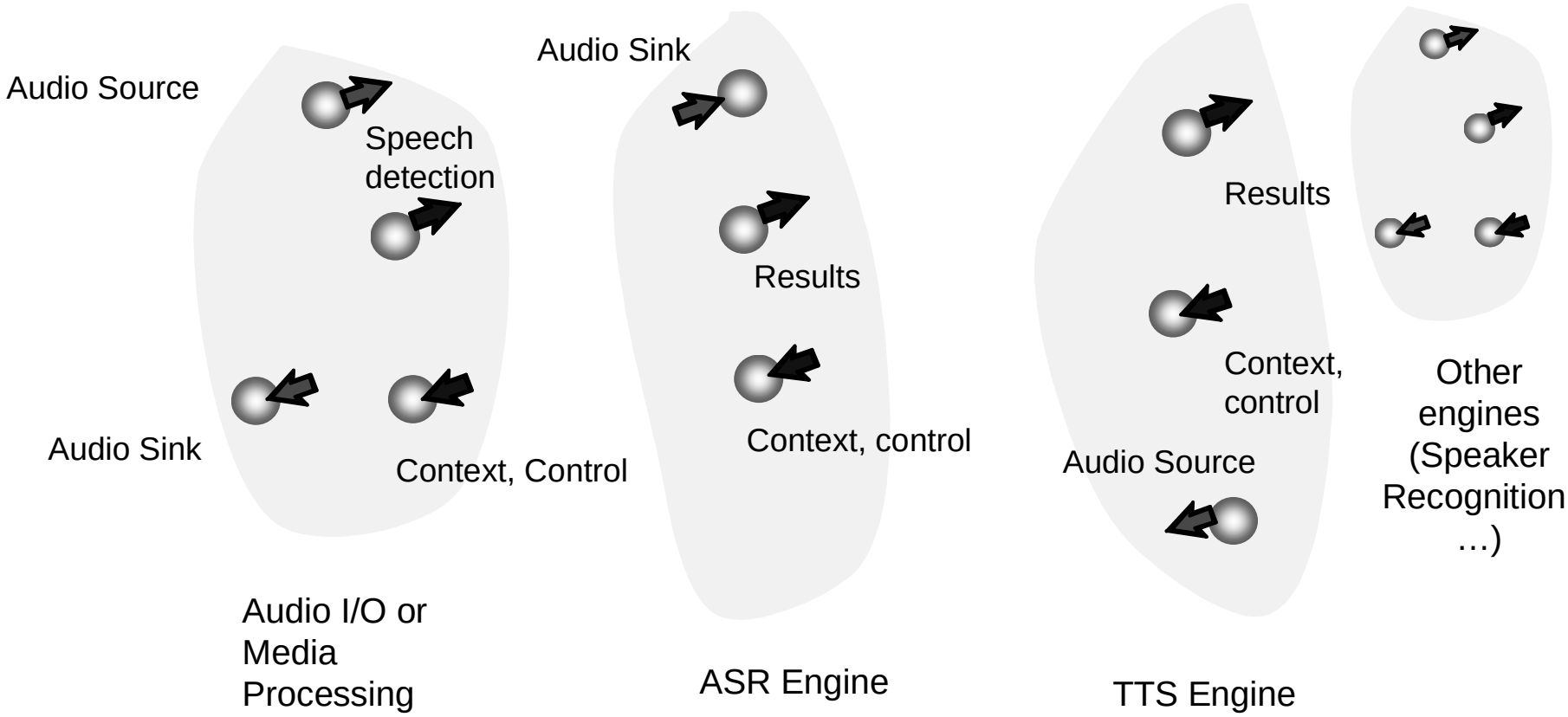
Protocol Evaluation

Stéphane H. Maes, Oracle,
stephane.maes@oracle.com

OVERVIEW: Web Services and SPEECHSC

- Speech Engines and audio sub-systems are considered as web services programmed by SOAP, WSDL, WSFL and discovered via UDDI.
- SOAP is bound to underlying protocol (HTTP, TCP, SIP, BEEP, ...)
- Audio-sub-systems and Speech engines defined by WSDL interfaces
  - Web services programmed with WSDL
  - Web services combined / composed with WSFL
  - Web services discovered by UDDI (or other similar mechanisms)
  - Additional events and messages via SOAP and à la WSXL (coordination among web services)
  - Web services provide advertisement mechanisms
  - Security can be provided by WS-Security

CONCEPTUAL VIEW



- WSDL's component defined by a set of typed ports:
  - Sink
  - Source
  - Context

EVALUATION CRITERIA

- **The evaluation follows the methodology described in section 3 of http://www.ietf.org/internet-drafts/draft-ietf-speechsc-protocol-eval-01.txt.**
- **Caveats:**
  - The web service framework is generic and extensible
  - There is no syntax and semantics associated to the control of speech engines.
    - Such syntax and semantics can be easily specified following the web service framework (could be inspired from MRCP or other Speech API)
    - The framework remains extensible
    - This practice is integral part of the Web Service framework and does not require any modification.
  - The framework can be bound to numerous transport protocols
  - Additional features are available today through tools and middleware offering rather than standard specifications.
    - This is considered to demonstrate that such capabilities are supported by the web service framework.
  - The evaluation assumes that these inherent characteristics of the web service framework are exploited:
    - If no change is required and only syntax and semantics must be defined, the framework is considered to support the requirements (total compliance: **T**).

OVERVIEW OF ANALYSIS

- See Section 8 in http://www.ietf.org/internet-drafts/draft-ietf-speechsc-protocol-eval-01.txt.
- A web services framework that implements SPEECHSC would satisfy all the requirements identified in:
  - http://www.ietf.org/internet-drafts/draft-ietf-speechsc-reqts-02.txt (mostly with T marks or P+)
  - The use cases intermediate work documents (e.g. draft-maes-speechsc-use-case-00.txt) that were considered to motivate these requirements.

  With the caveats identified previously
- Finalization of a web service-based specification for SPEECHSC essentially involves (first version):
  - Integration of the web service framework for SPEECHSC within the IETF stack with bindings to associated streamed media exchanges.
  - Specification of the SPEECHSC syntax and semantics (e.g. MRCP syntax) or other Speech API syntax
  - Optional possibly selection of the recommended underlying transport protocols.
    - This may include defining new bindings for SOAP and optimizations.
- Future versions could involve richer specifications

NOTABLE POINTS

- Some excerpt from the evaluation of web service framework
- For a complete analysis, see section 8 in http://www.ietf.org/internet-drafts/draft-ietf-speechsc-protocol-eval-01.txt

EVALUATION DETAILS

- **Protocol efficiency**
- **P+ to P:**
  - Web services are by definition more verbose protocols. Hence, at this stage this does not qualify for a T mark.
  - However work is in progress (e.g. OMA, JCP) to optimize the exchanges to handle:
    - Client with limited resources
    - Constrained bandwidth
  - These rely on protocol compression and optimization (e.g. JSR 172, XML RPC), caching and gateways.
  - As such the protocols qualify as **P+**.
  - In addition, based on the qualification of efficiency provided in the requirement document, the web service framework proposed for SPEECHSC relies on known efficient techniques:
    - Asynchronous pre-programming of the engines as web services to reduce exchanges and avoid racing conditions
    - Possibility to piggy back on response message if transported on optimized protocols like SIP or BEEP.
    - state caching in the engines that are considered as stand-alone, pre-packaged and pre-programmed engines.
    - etc…

EVALUATION DETAILS

Analysis of Duplexing and Parallel Operation Requirements

- **T:**
  - Web services allow control (interface) and composition of web services at will (e.g. WSFL). Also, it does not pre-supposes how many ports or streams are associated to the engine. Different inbound and outbound can be used at will; in full duplex or even between engines as supported by WSFL and WSXL.
- **Full Duplex operation**
- **T:**
  - See above
- **Multiple services in parallel**
- **T:**
  - See above and combination of services below
- **Combination of services**
- **T:**
  - Web services allow control (interface) and composition of web services at will (e.g. WSFL) into complex parallel, serial or coordinated combinations as supported by WSFL and WSXL.

EVALUATION DETAILS

Analysis of additional considerations (non-normative)

- **P+ to T:**
  - The framework supports:
    - Use of SDP to describe sessions and streams for the streamed channels
    - Time stamps could be transmitted as part of the control messages at the web service level or in band (e.g. with dynamic payload switch or within the payload).
  - The framework is compatible with any encoding scheme. This is illustrated by the work on SRF (Speech Recognition Framework) driven at 3GPP that supports conventional and DSR optimized codecs and possible exchange of speech meta-information (e.g. data that may be required to facilitate and enhance the server-side processing of the input speech and facilitate the dialog management in an automated voice service. These may include keypad events over-riding spoken input, notification that the UE is in hands-free mode, client-side collected information (speech/no-speech, barge-in), etc….).
  - - SOAP over SIP or BEEP to support the framework can also support VCR controls.
  - real-time messaging between engine and control is supported within the framework (e.g. via SOAP or XML events). The framework also support exchange between engines (same process; see also WSXL).
  - Although non-normative, the web service framework described in section 1 probably deserves marks of P+ to T.

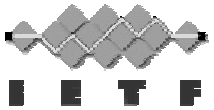
EVALUATION DETAILS

Analysis of Security considerations

- **P+ to T:**
  - Web services are evolving to provide security, authentication, encryption, trust management and privacy. This is now an OASIS activity: WS-Security.
  - This framework would enable SPEECHSC to employ the security mechanism provided by WS-Security for the remote control aspects. Exchanged media can rely on security mechanism at the transport / streaming level.
  - The web service framework probably deserves marks of **P+** to **T**.

ADDITIONAL CONSIDERATIONS

- Fits web service evolution:
  - Can reuse web services tools and middleware to deploy
  - Can reuse web service standard framework for specification
- Standard-based:
  - Specifications exist or are developed, tested and getting widely supported.
- Robust, modular, scalable and distributable
- Ease of integration:
  - Independent of connectivity and gateway vendor
  - Integration of different engines
  - Independent of the application platform:
- Remove complexities:
  - no engine step by step hand holding - engine performs these tasks on its own, racing conditions, separate audio exchange from controls
- Design to be extensible, discoverable and composed:
  - no limitations as previous APIs approaches.
- Can reuse syntax and semantics from MRCP and other speech APIs



ORACLE

DETAILED ANALYSIS

Background material

EVALUATION DETAILS

Analysis of General Requirements

- **Reuse Existing Protocols**
- **T:**
  - Web services are is a class of protocols (framework) widely studied and developed across numerous standard bodies like W3C, OASIS, WS-I, Liberty, Parlay and adapted to numerous deployment environments issues at IETF, OMA, 3GPP, 3GPP2, JCP, etc…
- **Maintain Existing Protocol Integrity**
- **T:**
  - Web services is an XML-based framework that is by definition extensible to support appropriate syntax and semantics.
  - Web services are bound on underlying transport protocols. Numerous such binding have been specified. Others are in development. By handling at SPEECHSC at the level of the Web services framework, the integrity is maintained for:
    - underlying transport protocols (to which the web service are bound (e.g. SOAP)
    - web service framework
  - This does not prevent introducing bindings to new protocols if needed. For example, binding to SIP or BEEP could be advantageous for mobile deployments.

EVALUATION DETAILS

- **Avoid Duplicating Existing Protocols**
- **T:**
  - By definition, the web service framework can be specified to remote control any web service. Specified syntax can be limited to avoid duplicating remote control functionalities offered by other protocols.
  - At the same time, the extensibility inherent to the framework guarantees that it is possible to specify (standard) or define (application specific) remote control for other entities beyond the current scope of SPEECHSC.
  - In that context and in view of unifying the remote control framework exposed to an application developer or a system integrator, it may be of interest to provide remote control syntax for special entities like prompt player etc…

EVALUATION DETAILS

- **Protocol efficiency**
- **P+ to P:**
  - Web services are by definition more verbose protocols. Hence, at this stage this does not qualify for a T mark.
  - However work is in progress (e.g. OMA, JCP) to optimize the exchanges to handle:
    - Client with limited resources
    - Constrained bandwidth
  - These rely on protocol compression and optimization (e.g. JSR 172, XML RPC), caching and gateways.
  - As such the protocols qualify as **P+**.
  - In addition, based on the qualification of efficiency provided in the requirement document, the web service framework proposed for SPEECHSC relies on known efficient techniques:
    - Asynchronous pre-programming of the engines as web services to reduce exchanges and avoid racing conditions
    - Possibility to piggy back on response message if transported on optimized protocols like SIP or BEEP.
    - state caching in the engines that are considered as stand-alone, pre-packaged and pre-programmed engines.
    - etc…

EVALUATION DETAILS

- **Explicit invocation of services**
- **T:**
  - Web services are typically used in a client-server environment. Solutions exist for peer to peer (service to service) etc…
  - Web services have been designed to support clients and servers at least one of which is operating directly on behalf of the user requesting the service.
  - In addition, work on-going at OMA and JCP addresses some of these issues in mobile environment with the introduction of possible web service gateways.
- **Server Location and Load Balancing**
- **T:**
  - Web services are widely developed for e-business applications. Numerous tools and mechanisms have been provided for service discovery ad advertisement. In addition, numerous offerings provide routing and load balancing capabilities as part of the web application server used to deploy the web service.
  - Note that web services do not specify server location or load balancing; but they are deployed on systems that provide such functionalities. As web services are expected to be widely used in the future and central to most e-business offerings, it is to expect that such tools will become even more pervasive and efficient.

EVALUATION DETAILS

- **Simultaneous services**
- **T:**
  - Web services allow control (interface) and composition of web services at will (e.g. WSFL).
  - See also section on combination of services
- **Multiple media sessions**
- **T:**
  - The framework does not pre-supposes how many ports or streams are associated to the engine. Different inbound and outbound can be used at will.

Analysis of TTS requirements

- **Requesting Text Playback**
- **T:** (supported – syntax to be defined; which is consistent with the web service framework)
  - TTS engines can be pre-programmed as web services to perform TTS on incoming text. This is simply a matter of agreeing on the control syntax to do so. The text to play back can be part of the control instructions transmitted in SOAP to the TTS engine.

EVALUATION DETAILS

- **Text Formats**
- **T:**
  - Exchanged format for text can be any MIME type; including plain text.
- **SSML**
- **T:**
  - Exchanged format for text can be any MIME type; including XML and hence SSML.
- **Text in Control Channel**
- **T:**
  - Exchanged format for text can be any MIME type that can include text or XML. The XML can include address information (URI).
- **Document Type Indication**
- **T:**
  - SOAP and the web service framework built on SOAP rely on XML and MIME type to identify media types. This is at the core of data exchange in SOAP.
- **Control Channel**
- **T:**
  - SOAP and WSDL support the remote control of the web services (engines or media processing entity).

EVALUATION DETAILS

- **Playback Controls**
- **T:** (supported – syntax to be defined; which is consistent with the web service framework)
  - This is simply a matter of agreeing on the control syntax to do so as part of the control instructions transmitted in SOAP to the TTS engine.
- **Session Parameters**
- **T:**
  - Session parameters are presumably content delivered as part of the control instructions transmitted in SOAP to the TTS engine.
- **Speech Markers**
- **T:**
  - Speech markers are presumably content delivered as part of the control instructions transmitted in SOAP to the TTS engine. See also SSML.

EVALUATION DETAILS

Analysis of ASR requirements

- **Requesting Automatic Speech Recognition**
- **T:** (supported – syntax to be defined; which is consistent with the web service framework)
  - ASR engines can be pre-programmed as web services to perform speech recognition on incoming audio. This is simply a matter of agreeing on the control syntax to do so. The instructions and parameters (including data files like grammars etc…) can be part of the control instructions transmitted in SOAP to the ASR engine.
  - Results can be part of the web service messaging as supported by the web service framework.
- **XML**
- **T:**
  - Exchanged format for message can be any MIME type; including XML and hence XML for controlling the ASR.
- **Grammar Specification**
- **T:**
  - Grammar specification can be part of the messages to control the ASR. This includes any MIME type; including XML for passing grammars by values, other MIME format including binary and URI for passing grammars by reference.

EVALUATION DETAILS

- **Explicit Indication of Grammar Format**
- **T:**
  - SOAP and the web service framework built on SOAP rely on XML and MIME type to identify media types. This is at the core of data exchange in SOAP.
- **Grammar sharing**
- **T:**
  - The framework described in section 1 supports pre-programming of the engines per utterance, per session or in an unlimited manner. This way grammar sharing can easily be achieved and controlled by an external controller, application etc…
- **Session Parameters**
- **T:**
  - Session parameters are presumably content delivered as part of the control instructions transmitted in SOAP to the ASR engine.

EVALUATION DETAILS

- **Input Capture**
- **T:** (supported – syntax to be defined; which is consistent with the web service framework)
  - ASR engines can be pre-programmed as web services to perform speech recognition on incoming audio. This is simply a matter of agreeing on the control syntax to do so. The instructions and parameters (including data files like grammars etc…) can be part of the control instructions transmitted in SOAP to the ASR engine. This cab include the syntax and instructions to capture the audio.

Analysis of Speaker Identification and Verification Requirements

- **Requesting SI/SV**
- **T:** (supported – syntax to be defined; which is consistent with the web service framework)
  - SI or SV engines can be pre-programmed as web services to perform speaker recognition on incoming audio. This is simply a matter of agreeing on the control syntax to do so. The instructions and parameters (including data files like voice prints, etc…) can be part of the control instructions transmitted in SOAP to the SI or SV engine.
  - Results can be part of the web service messaging as supported by the web service framework.

EVALUATION DETAILS

- **Identifiers for SI/SV**
- **T:**
  - This can be part of the control message.
- **State for multiple utterances**
- **T:**
  - This can be achieved by appropriately programming the SI or SV engine across multiple utterances. This is simply a matter of agreeing on the control syntax to do so. The framework supports spanning multiple utterances.
- **Input Capture**
- **T:** (supported – syntax to be defined; which is consistent with the web service framework)
  - SI or SV engines can be pre-programmed as web services to perform speaker recognition on incoming audio. This is simply a matter of agreeing on the control syntax to do so. The instructions and parameters (including data files like grammars etc…) can be part of the control instructions transmitted in SOAP to the ASR engine. This can include the syntax and instructions to capture the audio.
- **2.4.5 SI/SV functional extensibility**
- **T:**
  - By definition a web service framework and XML are extensible to new functionality and describe how extensibility is achieved.

EVALUATION DETAILS

Analysis of Duplexing and Parallel Operation Requirements

- **T:**
  - Web services allow control (interface) and composition of web services at will (e.g. WSFL). Also, it does not pre-supposes how many ports or streams are associated to the engine. Different inbound and outbound can be used at will; in full duplex or even between engines as supported by WSFL and WSXL.
- **Full Duplex operation**
- **T:**
  - See above
- **Multiple services in parallel**
- **T:**
  - See above and combination of services below
- **Combination of services**
- **T:**
  - Web services allow control (interface) and composition of web services at will (e.g. WSFL) into complex parallel, serial or coordinated combinations as supported by WSFL and WSXL.

EVALUATION DETAILS

Analysis of additional considerations (non-normative)

- **P+ to T:**
  - The framework supports:
    - Use of SDP to describe sessions and streams for the streamed channels
    - Time stamps could be transmitted as part of the control messages at the web service level or in band (e.g. with dynamic payload switch or within the payload).
  - The framework is compatible with any encoding scheme. This is illustrated by the work on SRF (Speech Recognition Framework) driven at 3GPP that supports conventional and DSR optimized codecs and possible exchange of speech meta-information (e.g. data that may be required to facilitate and enhance the server-side processing of the input speech and facilitate the dialog management in an automated voice service. These may include keypad events over-riding spoken input, notification that the UE is in hands-free mode, client-side collected information (speech/no-speech, barge-in), etc….).
  - - SOAP over SIP or BEEP to support the framework can also support VCR controls.
  - real-time messaging between engine and control is supported within the framework (e.g. via SOAP or XML events). The framework also support exchange between engines (same process; see also WSXL).
  - Although non-normative, the web service framework described in section 1 probably deserves marks of P+ to T.

EVALUATION DETAILS

Analysis of Security considerations

- **P+ to T:**
  - Web services are evolving to provide security, authentication, encryption, trust management and privacy. This is now an OASIS activity: WS-Security.
  - This framework would enable SPEECHSC to employ the security mechanism provided by WS-Security for the remote control aspects. Exchanged media can rely on security mechanism at the transport / streaming level.
  - The web service framework probably deserves marks of **P+** to **T**.