VR Architecture Recommendations & E2E User Experience Goals

George Joseph, VP Engineering

gjoseph@idwtechnologies.com

Recommended Core Design Goals for AGL

Architecture and associated implementation need to be reusable and leverageable. Implementation shall not put burden on end-user to determine which speech engine is best.

Needs to be pre-commercial grade and not demo quality. All hw/sw hooks should be implemented, not subset.

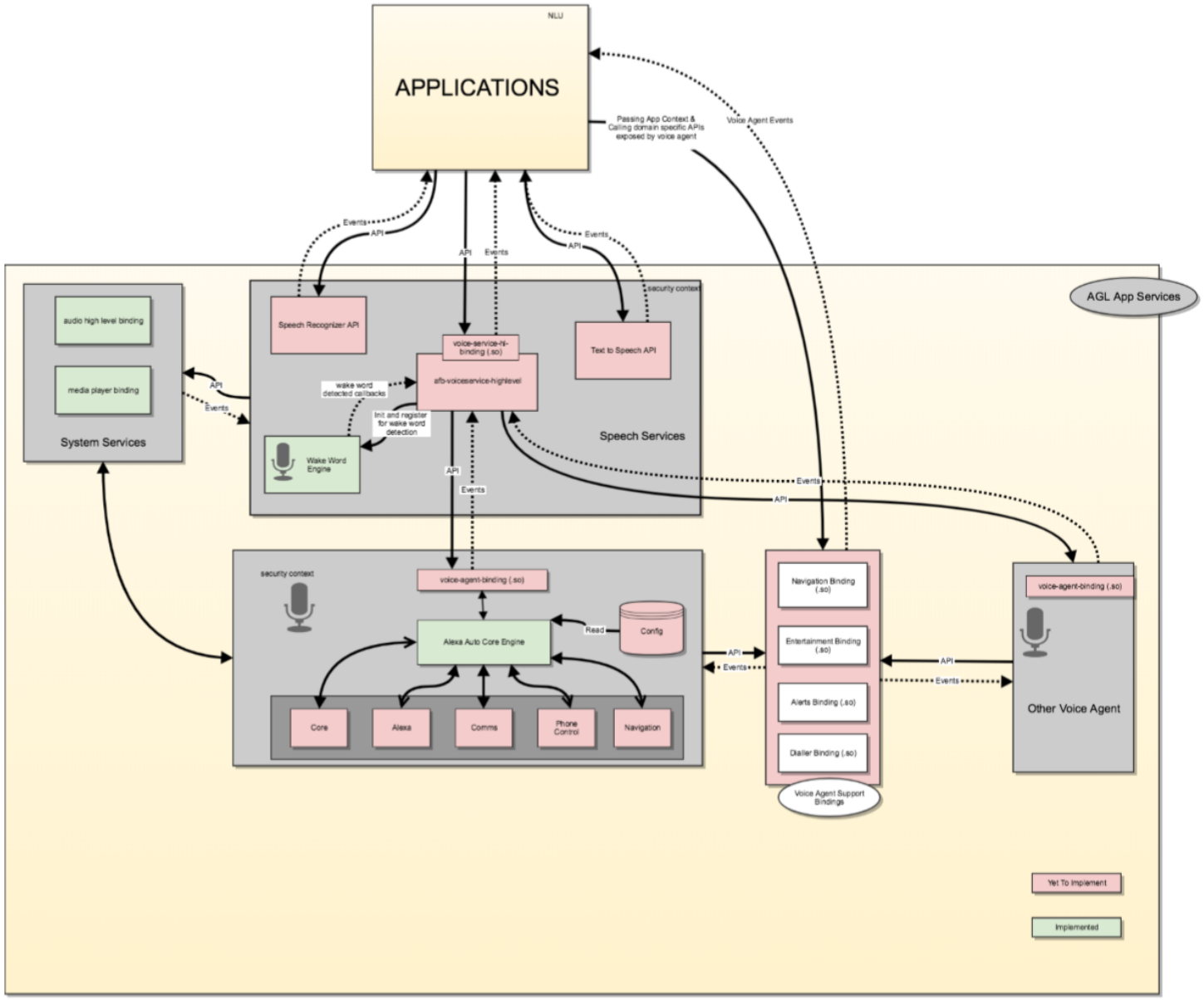
Design for minimal latency and best user experience that would be pre-commercial grade. Importance of building architecture with KPI goals now.

Design choices should be driven to enable best customer experience as opposed to integration of voice engine choice.

3-5 years from now, ASR will be hardware accelerated. Architecture design needs to enable this and future roadmap optimizations.

Ex: DSP hooks enabled for "Ok, Google" on Snapdragon 8974.

Proposed VR Framework Design Architecture



Design objectives

- Modular is needed but **not at the cost** of higher latency and user experience.
- **KPIs:** should be focused on vehicle functions & what use cases can tolerate: Nav, Radio, Phone, Music, HVAC, Alerts, POI (local cache), advanced functions (Cruise Control) first and must work well. Must be fast, less than .5 seconds
  - Context search on the internet and arbitration will be slower due to current internet connectivity latency and search. Actions returned to the vehicle need to be processed within .5 seconds.
  - Waiting for 50 to 100ms for voice front to hand off to ASR engine is not going to work.
- Setup needs to be **dead simple**, training must be ongoing or augmented via the cloud. Ability to **train models** in cloud and push to embedded devices give birth to flexible and organic implementations. **Learning frameworks** such as machine learning, AI, contextual reasoning, and personalization need to supported openly by speech engine providers.
- CONCERN: Providing any notification to start and end speech takes us farther away from **truly enabling natural speech**. Any hard coded event expectation needs to handled within the ASR/AI framework. For example, conversational speech, why does the voice binder need to provide end of speech notification? onEndOfSpeechDetected, ASR should handle this?
- In non-automotive use cases
  - **zones** function differently and therefore VR requirements need to consider **separate paths** in architecture.
  - need designs with ‘support button’ and ‘always listening ASR’. How does this change the framework?
- Support for combo actions and advanced dialogues need to be supported, “Jarvis, tune to 98.5 and set my cruise at 85 miles per hour.”

Use case Driven Requirements

Expect the user to initiate VR command at any time. i.e. When on a call.	• Drives requirement for software input mixer (multiple sources)
End to End ASR response time	• Core vehicle functions, less 0.5 seconds • Context Search, response in 1.0 second
Natural language conversation & dialogue	• Do not want consumer to determine arbitration and underlying technology. • Alexa, Nuance, Jarvis
Start now with wake up word to start but forecast what technology is going to look like in 3 years	• Continuous listen should not be for key words but specific intents
Need online / offline ASR	• Many times, there are internet connectivity gaps

VR needs to co-exist with Apple CarPlay / Android Auto VR

Siri / Google -> These keys words must be passed along when called out explicitly.

Embedded -> seamless experience with minimized latency including when using the cloud.

Context beyond just position but vehicles around you.

KPI Requirements for Hands-Free subsystems

System may interface to a wired headset/microphone or wireless Bluetooth or proprietary.

Audio preprocessing of input voice signal (from the microphone) and necessary KPI for background noise suppression shall be at least 15 dB.

Echo cancellation to suppress background echo in high noise environment by at least 35 dB.

Provide user with a “please wait” prompt when any execution of a command takes longer than 0.5 seconds.

Text to Speech

Uniform voice across all systems

Implies hook to feed infotainment system's Text to Speech engine needs

Configurable for multiple languages

Selectable personalities

Framework for changing voice depending if your vehicle is friend or slave ☺

Hardware Hooks for Assisted

Looking for hardware assisted filter prior to all the fancy ASR software algorithms

Update of intents and training over the network over time

Concerns about duplicate frames, can we have a shared memory buffer for frames with multiple ASR agents and continuous processing

Future Requirements

ANALYTICS FOR FRAMEWORK TO VR AVAILABILITY AND ACCURACY

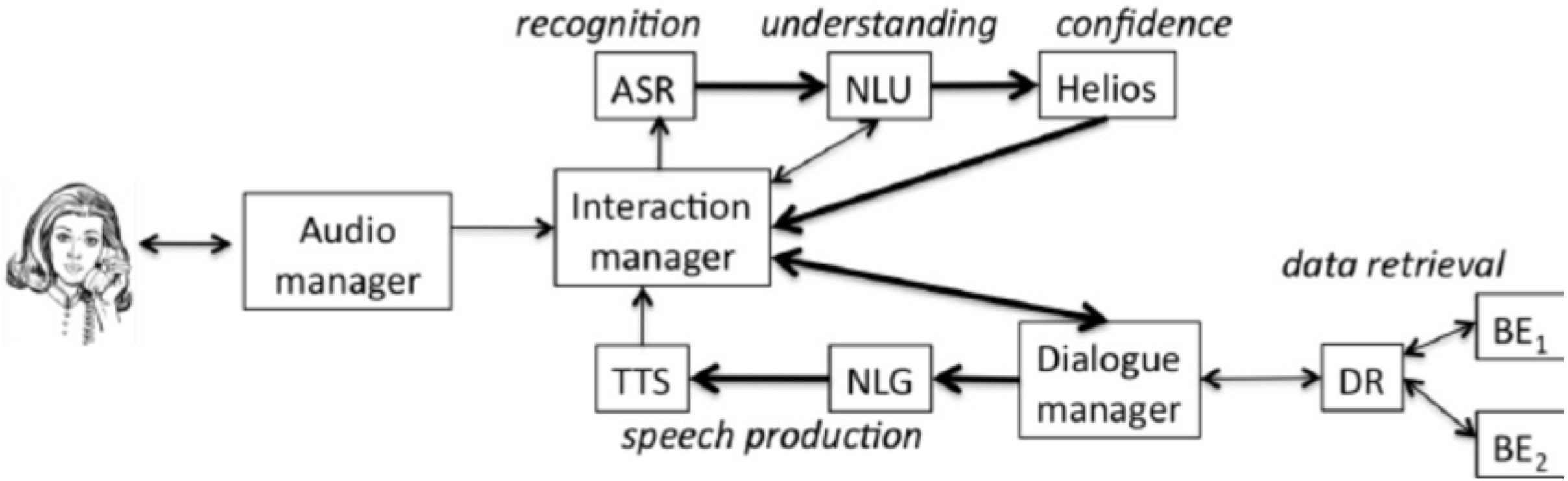
TIGHT INTEGRATION WITH NON SPEECH MODALITIES

HOOKS FOR ADVANCED VEHICLE FUNCTIONS AND SENSORS SUCH “DEFLATE MY TIRES TO X PSI” OR SUPER INFLATE MY TIRES (OFF ROAD USE-CASES)

ARCHITECTURE AND DESIGN BUILT WITH DATA PRIVACY IN MIND

GPU SUPPORT / OFF LOAD FOR DEEP LEARNING

Evolution to conversation with your vehicle



Data Mining to Support Human-Machine Dialogue for Autonomous Agents - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/A-traditional-pipeline-architecture-for-spoken-dialogue-systems-Heavy-arrows-emphasize_fig1_221435608 [accessed 7 Sep, 2018]

Questions?

Thank you

George Joseph

VP Engineering

gjoseph@idwtechnologies.com

IDW Technologies LLC