



AtAVi

---

# Analisi SDK dei principali A.V. v1.0.0

---

## Sommario

Documento contenente l'analisi degli *SDK<sub>g</sub>* dei principali assistenti virtuali relativo al *prodotto<sub>g</sub>* AtAVi determinato dal gruppo Co.Code nel corso della realizzazione del *progetto<sub>g</sub>* AtAVi.

|                          |   |
|--------------------------|---|
| <b>Versione</b>          | 1.0.0   |
| <b>Data di redazione</b> | 2017-01-06  |
| <b>Redazione</b>         | Luca Bertolini<br>Simeone Pizzi<br>Nicola Tintorri        |
| <b>Verifica</b>          | Mattia Bottaro  |
| <b>Approvazione</b>      | Pier Paolo Tricomi  |
| <b>Uso</b>               | Esterno   |
| <b>Distribuzione</b>     | prof. Tullio Vardanega<br>prof. Riccardo Cardin<br>Zero12 |

## Diario delle modifiche

| Versione | Riepilogo   | Autore             | Ruolo                 | Data       |
|----------|---|--------------------|-----------------------|------------|
| 1.0.0    | Approvazione del documento                                  | Pier Paolo Tricomi | <i>Amministratore</i> | 2017-01-06 |
| 0.3.1    | Correzione errori segnalati                                 | Nicola Tintorri    | <i>Analista</i>       | 2017-01-06 |
| 0.3.0    | Verifica  | Mattia Bottaro     | <i>Verificatore</i>   | 2017-01-06 |
| 0.2.3    | Stesura sezione Conclusioni                                 | Nicola Tintorri    | <i>Analista</i>       | 2017-01-05 |
| 0.2.2    | Stesura sezione Confronto tra i diversi assistenti virtuali | Luca Bertolini     | <i>Responsabile</i>   | 2017-01-05 |
| 0.2.1    | Correzione errori segnalati                                 | Simeone Pizzi      | <i>Analista</i>       | 2017-01-05 |
| 0.2.0    | Verifica  | Mattia Bottaro     | <i>Verificatore</i>   | 2017-01-04 |
| 0.1.2    | Stesura sezione Siri  | Nicola Tintorri    | <i>Analista</i>       | 2017-01-04 |
| 0.1.1    | Correzione errori segnalati e stesura sezione Alexa         | Simeone Pizzi      | <i>Analista</i>       | 2017-01-04 |
| 0.1.0    | Verifica  | Mattia Bottaro     | <i>Verificatore</i>   | 2017-01-03 |
| 0.0.2    | Stesura sezioni wit.ai e API.AI                             | Luca Bertolini     | <i>Responsabile</i>   | 2017-01-03 |
| 0.0.1    | Stesura struttura documento                                 | Luca Bertolini     | <i>Responsabile</i>   | 2017-01-02 |

## Indice

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduzione</b>                                | <b>4</b>  |
| 1.1      | Scopo del documento . . . . .                      | 4         |
| 1.2      | Scopo del <i>prodotto</i> <sub>g</sub> . . . . .   | 4         |
| 1.3      | Glossario . . . . .                                | 4         |
| 1.4      | Riferimenti . . . . .                              | 4         |
| 1.4.1    | Normativi . . . . .                                | 4         |
| 1.4.2    | Informativi . . . . .                              | 4         |
| <b>2</b> | <b>wit.ai</b>                                      | <b>5</b>  |
| 2.1      | Descrizione . . . . .                              | 5         |
| 2.2      | Caratteristiche . . . . .                          | 5         |
| <b>3</b> | <b>API.AI</b>                                      | <b>6</b>  |
| 3.1      | Descrizione . . . . .                              | 6         |
| 3.2      | Caratteristiche . . . . .                          | 6         |
| <b>4</b> | <b>Alexa</b>                                       | <b>7</b>  |
| 4.1      | Descrizione . . . . .                              | 7         |
| 4.2      | Caratteristiche . . . . .                          | 7         |
| <b>5</b> | <b>Siri</b>  | <b>8</b>  |
| 5.1      | Descrizione . . . . .                              | 8         |
| 5.2      | Caratteristiche . . . . .                          | 8         |
| <b>6</b> | <b>Confronto tra i diversi assistenti virtuali</b> | <b>9</b>  |
| <b>7</b> | <b>Conclusioni</b>                                 | <b>10</b> |

Elenco delle tabelle

1      Confronto delle caratteristiche tra i diversi assistenti virtuali . . . . . 9

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento riporta un'analisi eseguita dal gruppo Co.Code sui principali  $SDK_g$  di assistenti virtuali presenti sul mercato. L'analisi in questione ha lo scopo di mettere in mostra i punti di forza e le debolezze degli  $SDK_g$  presi in considerazione.

## 1.2 Scopo del *prodotto*<sub>g</sub>

Si vuole creare un'applicazione web che permetta ad un ospite, in visita all'ufficio di Zero12, di interrogare un assistente virtuale per annunciare la propria presenza, avvisare l'interessato del suo arrivo sul sistema di comunicazione aziendale (Slack) e nel frattempo essere intrattenuto con varie attività.

## 1.3 Glossario

Allo scopo di evitare ogni ambiguità nel linguaggio e rendere più semplice e chiara la comprensione dei documenti, viene allegato il "*Glossario v1.0.0*". Le parole in esso contenute sono scritte in corsivo e marcate con una 'g' a pedice (p.es. *Parola*<sub>g</sub>).

## 1.4 Riferimenti

### 1.4.1 Normativi

- *Capitolato*<sub>g</sub> d'appalto C2 - AtAVi: Accoglienza tramite Assistente Virtuale  
<http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C2.pdf>;
- "*Norme di Progetto v1.0.0*".

### 1.4.2 Informativi

- wit.ai:  
<https://wit.ai>
- API.AI:  
<https://api.ai>
- Alexa:  
<https://developer.amazon.com/alexa-skills-kit>  
<https://developer.amazon.com/alexa-voice-service>
- Siri:  
<https://developer.apple.com/library/content/documentation/Intents/Conceptual/SiriIntegrationGuide/index.html>

## 2 wit.ai

### 2.1 Descrizione

wit.ai è una società nata nell'ottobre del 2013 e acquisita da Facebook Inc. nel 2015.

L'obiettivo di wit.ai è quello di semplificare la creazione di applicazioni che prevedono interazioni testuali o vocali; per farlo viene messa a disposizione degli sviluppatori una piattaforma di linguaggio naturale aperta ed estensibile che ha la peculiarità di apprendere tramite ogni interazione eseguita.



### 2.2 Caratteristiche

wit.ai mette a disposizione un  $SDK_g$  gratuito ed *open source*<sub>g</sub> per il riconoscimento del linguaggio naturale. Questa piattaforma è caratterizzata dall'utilizzo di *Context*, *Intent* ed *Entity* che sono dei costrutti messi a disposizione per tradurre le richieste vocali dell'utente in dati processabili. In particolare il *Context* si utilizza per monitorare lo stato della conversazione tra l'utente e wit.ai. La piattaforma non del tutto stabile (*crash*<sub>g</sub>, *freeze*<sub>g</sub> temporanei, ecc..) e l'assenza di moduli di integrazione diretti con  $API_g$  esterne rendono lo sviluppo su wit.ai non ottimale.

In questa sezione si vogliono elencare le principali caratteristiche del  $SDK_g$  in questione:

- ambiente *open source*<sub>g</sub> e totalmente gratuito per progetti sia pubblici che privati;
- programmatori agevolati a sviluppare grazie ai costrutti *Intent* ed *Entity*;
- compatibilità con molti sistemi operativi, tra cui: *iOS*<sub>g</sub>, *Android*<sub>g</sub>, *Windows Phone*<sub>g</sub>, *Raspberry Pi*<sub>g</sub>, *Python*<sub>g</sub> e *C++*<sub>g</sub>;
- **Context**: è un "object" che lo sviluppatore utilizzerà per monitorare lo stato della conversazione tra l'utente e Wit.ai. La sua principale funzione è quella di trasformare il linguaggio naturale in dati processabili da Wit;

## 3 API.AI

### 3.1 Descrizione

API.AI è una società nata nell'ottobre del 2010 e acquisita da Google Inc. nel 2016. API.AI è una piattaforma di conversazione che permette interazioni sofisticate con il linguaggio naturale.



### 3.2 Caratteristiche

API.AI fornisce *SDK<sub>g</sub>* per i principali linguaggi di programmazione tra i quali *C++<sub>g</sub>*, *C#<sub>g</sub>*, *Java<sub>g</sub>*, *Node.js<sub>g</sub>*, *JavaScript<sub>g</sub>* e *Python<sub>g</sub>*. Inoltre può essere integrato con *Amazon Echo<sub>g</sub>* e *Microsoft Cortana<sub>g</sub>*.

Le applicazioni sviluppate su questa piattaforma sono costituite da *Agent*, i quali si occupano di trasformare il linguaggio naturale in dati processabili. Tali *Agent* sono a loro volta costituiti da *Intent*, che hanno il compito di associare la richiesta dell'utente ad una determinata azione del software, ed *Entity*, che sono strumenti per estrarre dal linguaggio naturale i parametri attesi.

## 4 Alexa

### 4.1 Descrizione

Alexa è stata annunciata per la prima volta nel novembre del 2014 come assistente virtuale dello *smart speaker Amazon Echo<sub>g</sub>*, sviluppato da Amazon.com. Le funzionalità di Alexa sono ispirate dal computer di bordo delle navicelle interstellari presenti nelle serie TV sci-fi “Star Trek”.



### 4.2 Caratteristiche

Amazon.com mette a disposizione gratuitamente Alexa Voice Service (AVS), un servizio che permette di integrare, in qualsiasi dispositivo e applicazione con accesso al web, l'assistente virtuale Alexa. La compagnia inoltre fornisce l'*Alexa Skills Kit (ASK)*, un *SDK<sub>g</sub>* che permette di estendere le funzionalità dell'assistente virtuale. Per creare una *Custom Skill* ASK richiede:

- **Insieme di intenti:** rappresentano le azioni che l'utente può compiere con quella determinata *Skill*;
- **Insieme di affermazioni d'esempio:** specificano le parole e le frasi che l'utente può pronunciare per invocare gli intenti;
- **Nome della *Skill*:** identifica la *Skill* e viene pronunciato dall'utente per interagire con essa;
- **Servizio *cloud<sub>g</sub>*:** accetta gli intenti come richieste strutturate ed agisce in base ad essi;
- **Configurazione:** combina gli elementi precedentemente descritti in modo che Alexa possa indirizzare le richieste al servizio creato per la *Skill*.

Amazon.com suggerisce di usare *AWS Lambda<sub>g</sub>* per la creazione di servizi *cloud<sub>g</sub>*.



## 5 Siri

### 5.1 Descrizione

Siri nasce come applicazione indipendente per  $iOS_g$  resa disponibile tramite il canale commerciale App Store, per poi essere acquisita da  $Apple_g$  Inc. nel 2011. È presente nei dispositivi  $iOS_g$ ,  $macOS_g$ ,  $watchOS_g$  e  $tvOS_g$  come assistente virtuale.



### 5.2 Caratteristiche

Essendo Siri parte integrante di  $iOS_g$ , il suo utilizzo è vincolato alla creazione di un'applicazione per tale sistema operativo. Inoltre questa applicazione deve appartenere ad uno degli ambiti prestabiliti da SiriKit i quali sono:

- chiamate  $VoIP_g$ ;
- messaggistica;
- pagamenti;
- foto;
- allenamento;
- trasporti;
- prenotazioni ristoranti;
- $CarPlay_g$ .

L'assistente virtuale di casa  $Apple_g$  processa per intero ogni frase dell'utente e ne ricava le intenzioni. Tali intenzioni, al fine di essere rappresentate in dati processabili, sono tradotte in *Intents*. Una volta acquisiti i dati e aver creato l'*Intent*, nel caso questi non risultino validi, Siri riformula le domande per gli attributi interessati.

## 6 Confronto tra i diversi assistenti virtuali

| <b>A.V.</b><br><b>Caratteristica</b>            | <b>wit.ai</b>   | <b>API.AI</b>  | <b>Siri</b>            | <b>Alexa</b>  |
|---|---|--|------------------------|---|
| Risveglio vocale                                | ✗   | ✗  | obbligatorio           | ✓   |
| Creazione <i>Intent</i> con NLU                 | ✓   | ✓  | ✓                      | ✓   |
| <i>Intents</i> personalizzabili                 | ✓   | ✓  | ✗                      | ✓   |
| Entità predefinite                              | ✓   | ✓  | ✓                      | ✓   |
| Ambiti predefiniti conversazione                | forniti dalla community   | 35+  | 8                      | ✗   |
| Costi   | gratis  | gratis senza ambiti predefiniti  | gratis                 | gratis fino a 1.000.000 di chiamate (per <i>AWS Lambda<sub>g</sub></i> )      |
| Piattaforme e linguaggi di prog. supportati     | <i>Node.js<sub>g</sub></i> ;<br><i>Python<sub>g</sub></i> ;<br><i>Ruby<sub>g</sub></i> ;<br><i>HTTP<sub>g</sub></i> | <i>Android<sub>g</sub></i> ; <i>iOS<sub>g</sub></i> ;<br><i>Apple Watch<sub>g</sub></i> ;<br><i>Node.js<sub>g</sub></i> ;<br><i>Cordova<sub>g</sub></i> ; <i>Unity<sub>g</sub></i> ;<br><i>C#<sub>g</sub></i> ; <i>Xamarin<sub>g</sub></i> ;<br><i>Windows Phone<sub>g</sub></i> ;<br><i>Python<sub>g</sub></i> ;<br><i>JavaScript<sub>g</sub></i> ; <i>PHP<sub>g</sub></i> ;<br><i>Botkit<sub>g</sub></i> ; <i>C++<sub>g</sub></i> ;<br><i>Mac OS X<sub>g</sub></i> ;<br><i>HTML5<sub>g</sub></i> ; <i>Ruby<sub>g</sub></i> | <i>iOS<sub>g</sub></i> | qualsiasi linguaggio permetta di accettare richieste <i>HTTPS<sub>g</sub></i> |
| Documentazione <i>API<sub>g</sub></i> esaustiva | ✓   | ✓  | ✓                      | ✓   |
| Gui modificazione <i>Intents</i>                | ✓   | ✓  | ✗                      | ✓   |
| Autenticazione                                  | <i>OAuth2<sub>g</sub></i>   | <i>double token<sub>g</sub></i>  | ✗                      | LWA( <i>Login with Amazon</i> )   |
| Numero di lingue supportate                     | 15  | 50   | 21                     | 2   |

**Tabella 1:** Confronto delle caratteristiche tra i diversi assistenti virtuali

## 7 Conclusioni

Dall’attenta analisi del gruppo Co.Code è emerso che *SiriKit* non è adeguato a soddisfare le esigenze del *progetto<sub>g</sub>* AtAVi in quanto gli ambiti forniti sono limitati, le interazioni con l’utente non sono personalizzabili ed il supporto a piattaforme diverse da *iOS<sub>g</sub>* è assente.

I restanti *SDK<sub>g</sub>* non presentano tali limitazioni e quindi potrebbero essere tutti potenzialmente sfruttati. La scelta del gruppo è ricaduta su API.AI per la maggiore stabilità rispetto a wit.ai, per il maggior numero di lingue disponibili in vista di un supporto futuro ad interazioni con utenti di diversa nazionalità, per la maggiore maturità della piattaforma e per la possibilità di esportare gli *Agents* in formati compatibili con i principali assistenti virtuali presenti sul mercato, tra cui Cortana ed Alexa.