Implementare bot e skill Alexa con Azure Cognitive Services

Andrea Saltarello – Presidente UGIdotNET
https://github.com/andysal

Twitter: @andysal74

Enos Recanati – Senior Software Dev @ Managed Designs
https://github.com/enosrecanati

Twitter: @enosrecanati

Ciao!

Benvenut* al primo *#Aperitech* organizzato da **UGIdotNET** grazie al supporto di **Codemotion** e **LVenture**

A long time ago in a galaxy far, far away....

20 luglio 2001: first men on the Moon

- Pierre Greborio
- Andrea Saltarello
- Gilberto Zampatti
- Roberto Brunetti
- Aldo Prinzi
- Franco Perduca
- Andrea Gorgaini
- Davide Ronchi
- David Papini
- Francesco Meani
- Gianluca Hotz
- Marco Barzaghi
- Fabrizio Saro
- Lorenzo Leonello
- Enzo Re
- Alessandro Ghizzardi
- Walter Felician

User Group Italiano .NET

http://www.ugidotnet.org

ARTICOLO 1

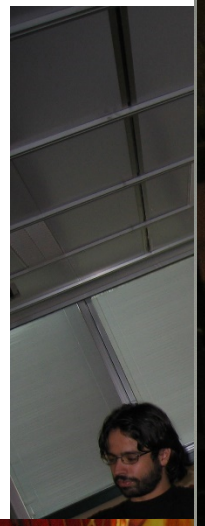
E’ costituita una associazione culturale indipendente con la seguente denominazione «User Group Italiano .NET (UGIdotNET)»

ARTICOLO 3

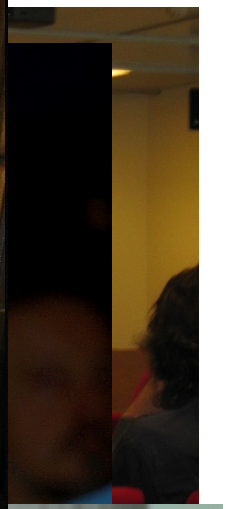
L’associazione non ha alcun fine di lucro

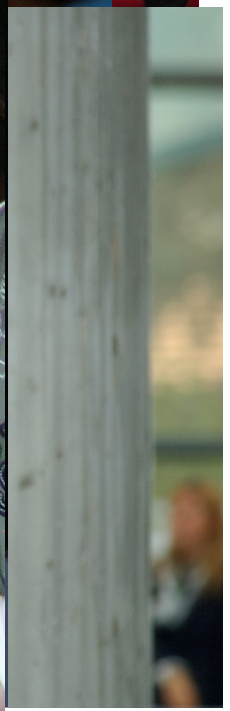
UGIdotNET by the numbers

- **17021** iscritti
- **1212** pubblicazioni, realizzate da **94** autori
- **57** giornate di seminari gratuiti
- **309** sessioni tecniche di approfondimento
- Insieme agli amici di **ASPItalia**, 11 edizioni di CommunityDays.it
- Prossimi appuntamenti:
  - Evento «full day» il 9 aprile presso Microsoft House @ Milano
  - Aperitech serale a maggio non appena estorco una data ad Enzo ☺
  - Evento «full day» il 22 maggio presso Microsoft House @ Milano




THIS JULY





MAY THE SOURCE BE WITH YOU

Implementare bot e skill Alexa con Azure Cognitive Services

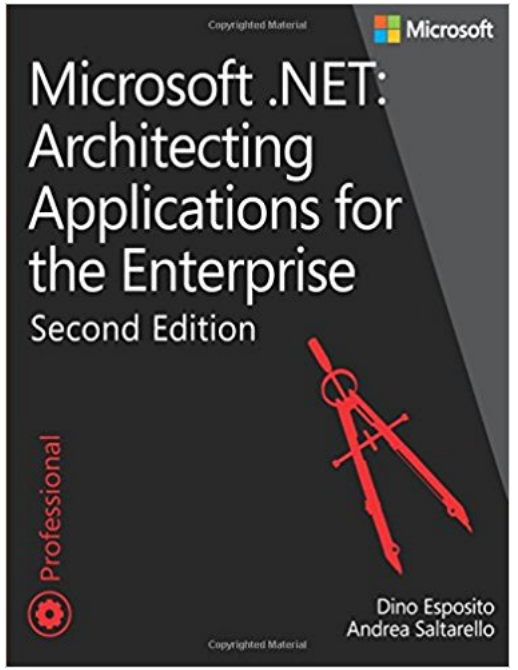
Andrea Saltarello – Presidente UGIdotNET
https://github.com/andysal

Twitter: @andysal74

Enos Recanati – Senior Software Dev @ Managed Designs
https://github.com/enosrecanati

Twitter: @enosrecanati

Un po’ di contesto



- Dal 2013 sviluppo **Merp**, un ERP open source (licenza AGPL3): nato quale «companion project» per la seconda edizione del mio libro, oggi è il *barebone* che utilizziamo in azienda
- Dal 2015 Enos ha iniziato a collaborare allo sviluppo di **Merp**
- Il 30 marzo 2016, durante la conferenza Build 2016, Microsoft presentò il proprio Bot Framework basato sul servizio cognitivo chiamato **LUIS**

Luis, chi era costui?

Per fare un ~~albero~~bot, ci vuole…

	Azure	Alexa
Client	Canali	Device/App
Endpoint	Bot Framework	Lambda
Servizio cognitivo	LUIS	NLP
Storage	Blob/CosmosDB	[DynamoDB]

Bot: quale gusto?

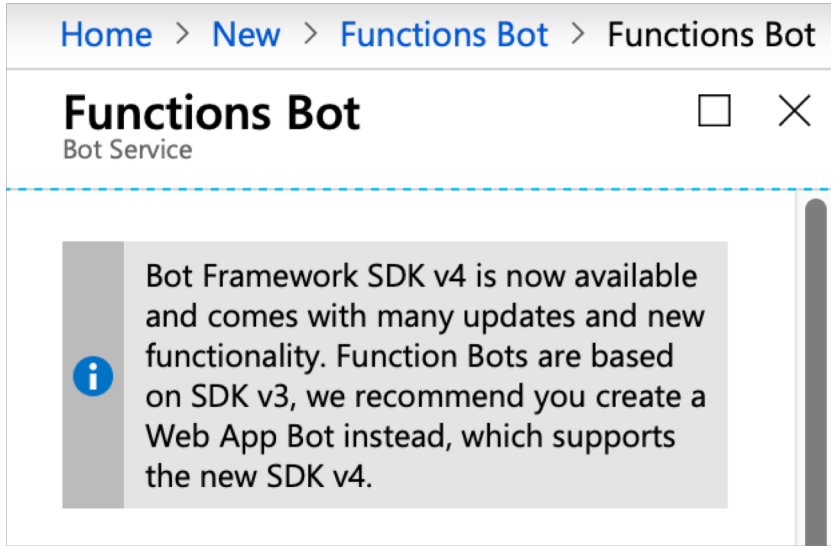
Tutto inizia creando una *bot app* mediante la dashboard di Azure, scegliendo:

- App type
  - **Azure Function**
  - **Web App**
- Pricing tier
  - **F0** (messaggi channel illimitati, 10K msg/mese su direct line. No SLA)
  - **S1** (messaggi channel illimitati, 0,422 EUR ogni 1K msg su direct line. SLA 99,9%)

Azure Functions

Una **Functions Bot** permette di scegliere:

- Esclusivamente lo SDK v3
- Linguaggio: C#/NET46, JS/Node
- Stile: Basic, echo, LUIS
- Hosting plan: App Service plan, Consumption



Web App

Una **Web App Bot** permette di utilizzare arbitrariamente le versioni 3 e 4 dello SDK. Entrambe supportano **Node.js** mentre, relativamente, a .NET:

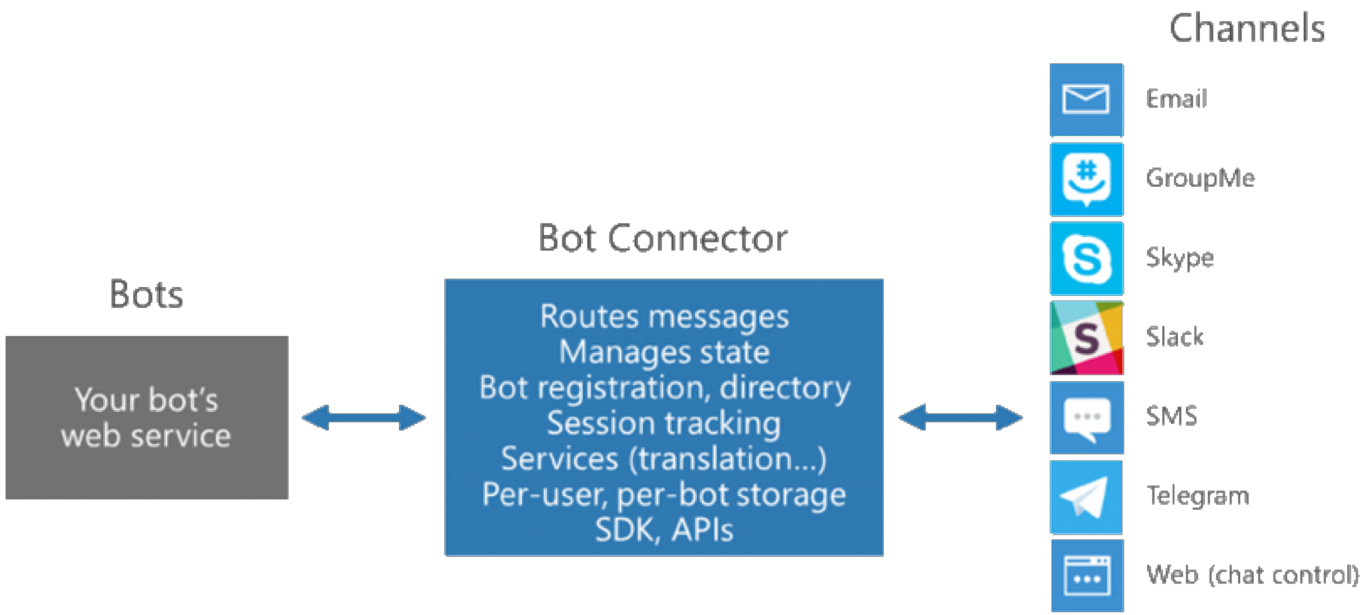
- La **v3** utilizza **ASP .NET WebAPI 2**
- La **v4** utilizza **ASP .NET Core**

Al netto dei template utilizzabili dal portale, qui ci sono tutti i gusti

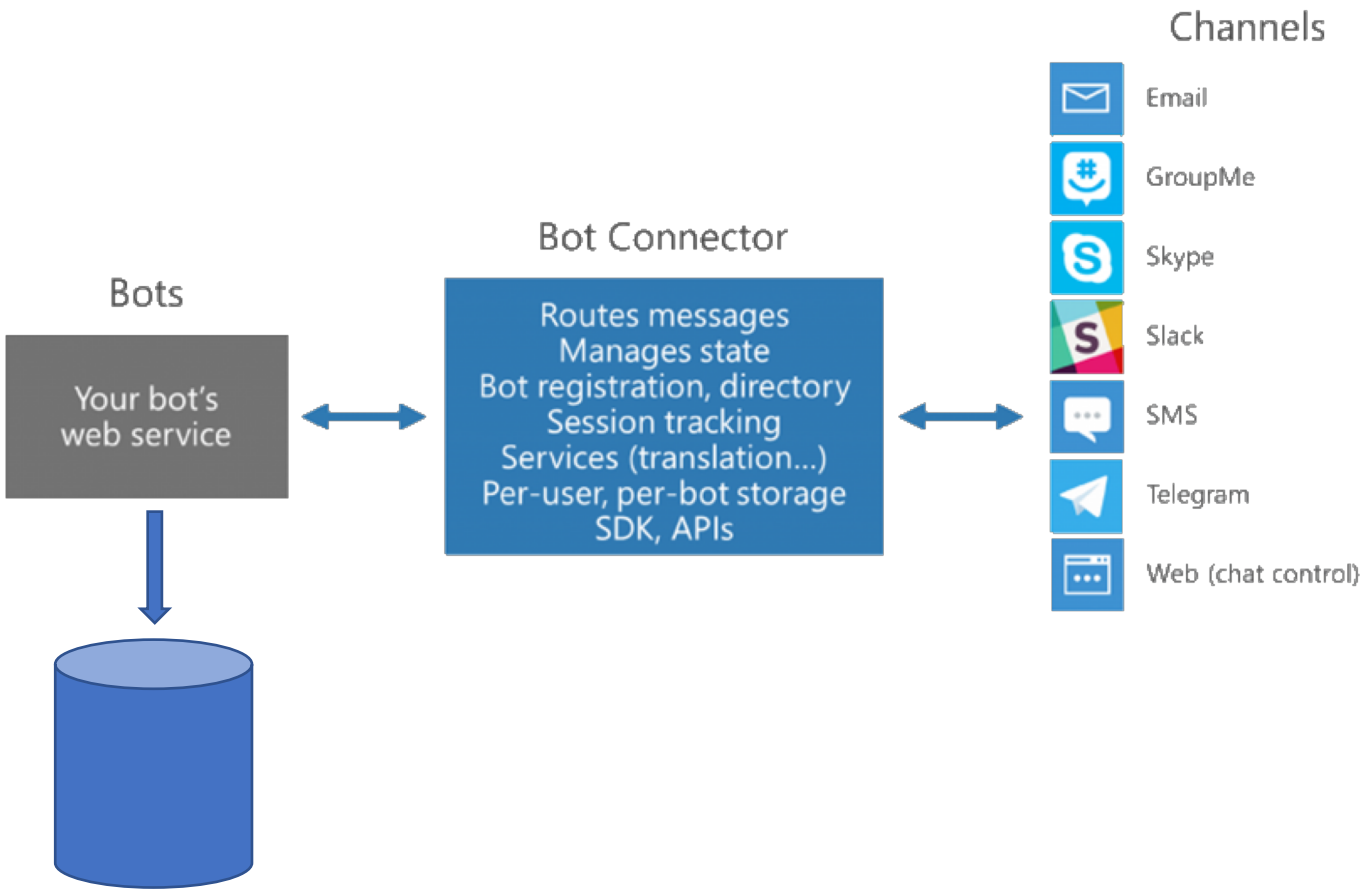
Demo

Creare un bot

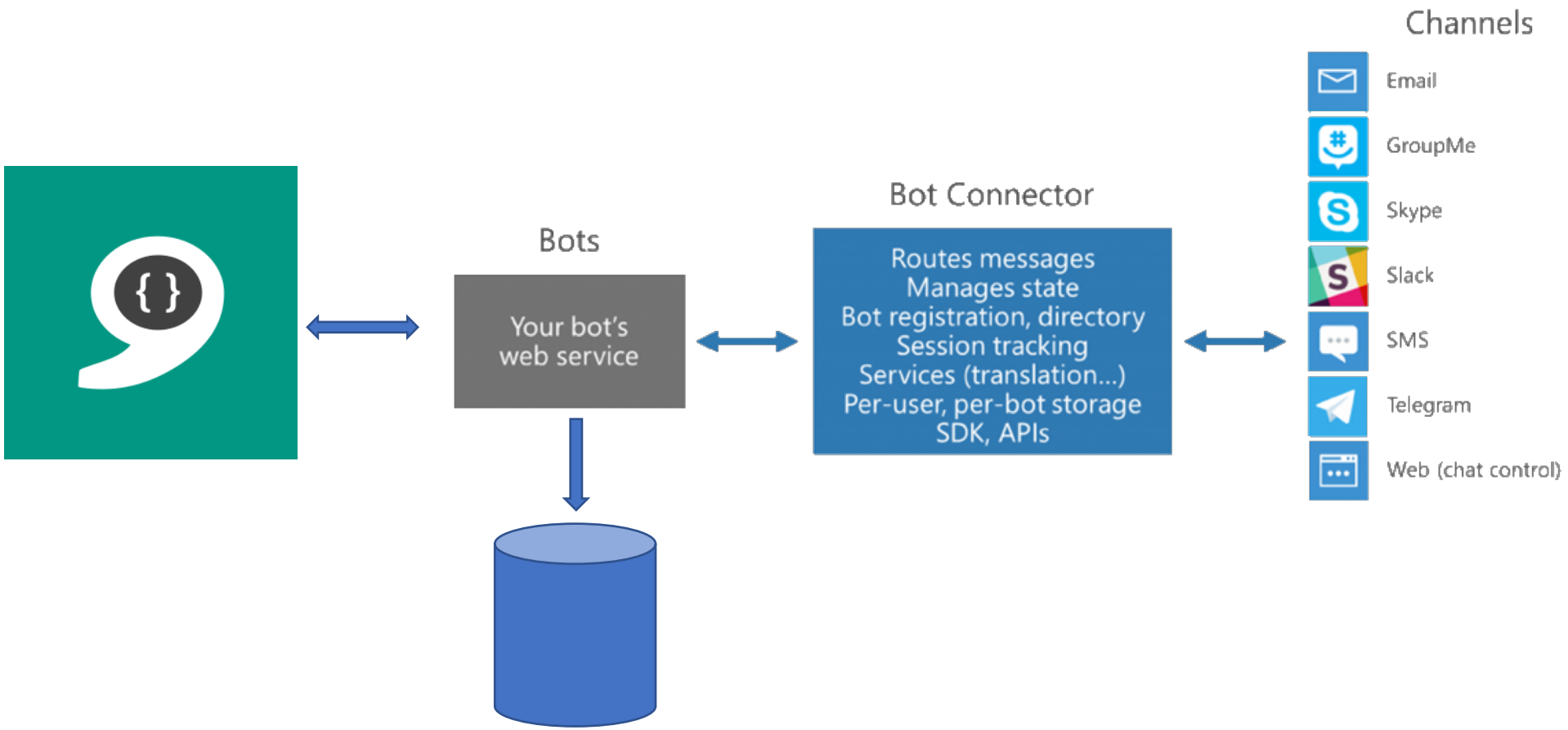
Bot Framework a colpo d'occhio



Enter LUIS


Channels
Email
GroupMe
Skype
Slack
SMS
Telegram
Web (chat control)
Bot Connector
Routes messages
Manages state
Bot registration, directory
Session tracking
Services (translation…)
Per-user, per-bot storage
SDK, APIs
Bots
Your bot's
web service

Enter LUIS


Channels
Email
GroupMe
Skype
Slack
SMS
Telegram
Web (chat control)
Bot Connector
Routes messages
Manages state
Bot registration, directory
Session tracking
Services (translation…)
Per-user, per-bot storage
SDK, APIs
Bots
Your bot's
web service

Demo

Martin @ Merp

LUIS 1-2-3

LUIS riconosce delle frasi testuali (*utterance*) associa[te/ndole] ad un catalogo di intenti (*intent*) relativi ad un set di *entità,* restituendoci:

- L'intento corrispondente alla utterance
- Le entità, ed il loro valore, riscontrate nella utterance

Sarà compito del nostro codice utilizzare il risultato del parsing per eseguire una azione.

Ad esempio: «la fattura 1/2018 è stata pagata?» potrebbe essere una *utterance* riferita alla istanza della entità **fattura** avente codice 1/2018

Demo

LUIS

LUIS: quanto costa?

Il pricing di LUIS si declina su 3 livelli:

- **Free**: 10000 transazioni testuali/mese (5 call/sec)
- **Text**: €1,265 EUR ogni 1000 transazioni testuali (50 call/sec)

Dove:

- Transazione testuale: max 500 caratteri

Bot Framework: Canali e direct line

Gli utenti interagiscono con un bot mediante:

- **Canali**: sono gli user agent direttamente supportati
- **Direct line**: sono gli user agent custom

Il modo più semplice di iniziare è la direct line «web chat», utilizzabile includendo nel markup:

- Il codice disponibile nella blade dell'app (no personalizzazione)
- Il codice (personalizzabile) pubblicato qui

Demo

Web chat

Supporto vocale @ web chat

Il componente web chat:

1. Registra la utterance tramite input audio
2. Converte la registrazione in testo mediante il browser o gli **Speech Services**
3. Invia il testo a **LUIS**, che invia la risposta
4. Pronuncia la risposta usando il browser o gli **Speech Services**

Sel custa? Se uso gli Speech Services:

- Free: 1 req, 5 ore/mese S2T, 5M char/mese T2S
- Standard: 20 req, € 0,844/hour S2T, € 3,374 1M chars T2S

Demo

Voce @ Web chat

Canali: attivazione

Supporto out of the box per:

Cortana	Email	Facebook	GroupMe
Kik	Microsoft Teams	Skype	Skype 4 Business
Slack	Telegram	Twilio	(Web chat)

Canali: il caso Facebook Messenger

La ricetta è:

1. Creare pagina
2. Creare app
3. Creare channel specificando: **page id**, **app id** e **secret**
4. In FB app, consentire accesso API in impostazioni avanzate
5. In Dashboard, configurare Messenger:
    1. scegliendo la pagina e recuperando il token utile per completare punto 3
    2. Impostando callback URL e Verify token (recuperati al punto 3)
    3. Selezionando: **messages**, **messages_postbacks**, **messages_optins**, **messages_deliveries**
6. In webhooks, selezionare pagina

Ripetere *ad libitum* per tutti i canali desiderati

Demo

Facebook MEssenger

Canali vs. autenticazione

TL;DR; L’autenticazione non è gratis

I canali non ci forniscono tra i claim uno userid riconciliabile a quelli che abbiamo nel profilo Identity quindi l’autenticazione della chat viene effettuata:

1. Configurando almeno un authentication provider per il bot nella dashboard
2. Inviando all’utente una *adaptive card* per il login contenente lo url della pagina di login di Merp
3. Producendo un token da inserire nella chat

D’ora in poi la conversazione è autenticata

Demo

Autenticazione

Stato di una conversazione

Nel mondo Bot Framework, le conversazioni hanno uno stato che può essere memorizzato:

- In memoria
- Nell'**Azure Table Storage**
- In **CosmosDb**

Lo stato permette di trasformare una raffica di richieste/risposte in una conversazione. Ad esempio, può servire per «ricordare» quale sia la subscription attiva

Billing/Pricing model

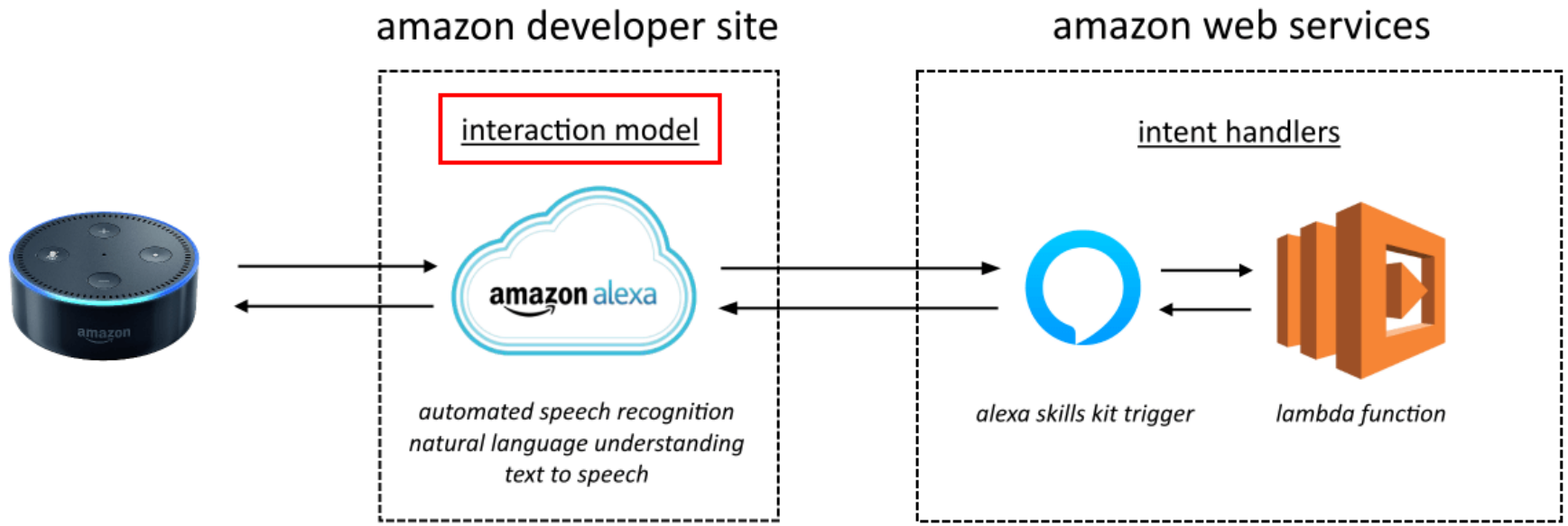
- Bot Framework:
  - Web app: 0 - 0,422 EUR per 1000 messaggi (S1)
  - Compute
- LUIS: 1,265 EUR/m (S1)
- [Speech to Text: 3,374 EUR per 1000 utterance]
- [Text to Speech: 3,374 EUR per 1000 messaggi]
- CosmosDB: 18 EUR/m (400 RU, 1Gb)

Ogni tx:

- vocale costa 0,8 cent (o 0,026 EUR includendo CosmosDB)
- Testuale costa 0,013 cent (0,02 EUR includendo CosmosDB)


alexa

Alexa a colpo d’occhio



Alexa 101

Molto simile allo stack Azure:

- Il modello di training è basato sulla triade *Intent*/*Utterance*/*Slot* (omologhi delle *entity*)
- L'intent handler è un arbitrario endpoint http che riceve in POST i dati della richiesta in json (per default è una lambda AWS, quindi si può usare .NET Core)
- Lo stato della conversazione è persistito in DynamoDB

Alexa vs. LUIS: le differenze

- I nomi degli intenti non accettano spazi
- Utterance:
  - non possono contenere punteggiatura (es: "?")
  - i numeri devono essere scritti sotto forma di parole separate da spazi (es: 2010 -> due mila dieci)
  - le lettere scritte in maiuscolo vengono lette in modo espanso (es: EL -> e elle)
  - gli slot sono «privati» ai singoli intent

Demo

Martin @ Alexa

Grazie! Domande?