

# AtAVi

Accoglienza tramite assistente virtuale

---

UNIVERSITÀ DI PADOVA

## DEFINIZIONE DI PRODOTTO v3.00

~/anSwEr



[answer.group17@gmail.com](mailto:answer.group17@gmail.com)

## Informazioni sul documento

<b>Nome Documento:</b>	<i>DefinizioneDiProdotto_v.3.00</i>
<b>Versione:</b>	3.00
<b>Data di Creazione:</b>	02 Febbraio 2017
<b>Data ultima Modifica:</b>	11 Maggio 2017
<b>Stato:</b>	<b>Approvato</b>
<b>Redazione:</b>	Abdelilah Lahmer, Eugen Saraci.
<b>Verifica:</b>	Matteo Maran, Andrei Tabacariu.
<b>Approvazione:</b>	Leonardo Zoso
<b>Uso:</b>	Esterno
<b>Distribuzione:</b>	<i>Answer</i>
<b>Destinato a:</b>	Prof. Vardanega, Prof. Cardin, <i>Zero12 s.r.l.</i>
<b>Email di riferimento:</b>	<code>answer.group17@gmail.com</code>

## Sommario

Questo documento descrive in dettaglio la struttura delle componenti del prodotto per il progetto AtAVi: Accoglienza tramite Assistente Virtuale.

## Registro delle modifiche

Versione	Data	Autore	Ruolo	Descrizione
3.00	2017-05-12	Leonardo Zoso	<i>Responsabile di progetto</i>	Approvazione del documento. Causa consegna del documento.
2.14	2017-05-11	Andrei Tabacariu	<i>Verificatore</i>	Verifica del documento. Causa modifiche effettuate.
2.13	2017-05-11	Abdelilah Lahmer	<i>Progettista</i>	Aggiunti riferimenti ipertestuali. Causa incremento documento.
2.12	2017-05-11	Eugen Saraci	<i>Progettista</i>	Tolta sezione Send Audio - /interaction. Causa errori riscontrati.
2.11	2017-05-11	Abdelilah Lahmer	<i>Progettista</i>	Tolta sezione Delete Costumers - /costumer/id. Causa errori riscontrati.
2.10	2017-05-11	Eugen Saraci	<i>Progettista</i>	Tolta sezione Update Costumers - /costumer/uuid. Causa errori riscontrati.
2.09	2017-05-11	Abdelilah Lahmer	<i>Progettista</i>	Tolta sezione Get Costumers - /costumer/id. Causa errori riscontrati.

Versione	Data	Autore	Ruolo	Descrizione
2.08	2017-05-11	Abdelilah Lahmer	<i>Progettista</i>	Tolta sezione Add Costumers - /costumer. Causa errori riscontrati.
2.07	2017-05-11	Abdelilah Lahmer	<i>Progettista</i>	Tolta sezione Get Costumers - /costumer. Causa errori riscontrati.
2.06	2017-05-11	Eugen Saraci	<i>Progettista</i>	Tolta sezione Get Employee Rules - /rules/employee/id. Causa errori riscontrati.
2.05	2017-05-11	Eugen Saraci	<i>Progettista</i>	Tolta sezione Get Customer Rules - /rules/customer/id. Causa errori riscontrati.
2.04	2017-05-11	Eugen Saraci	<i>Progettista</i>	Tolta sezione Logout - /auth/logout. Causa errori riscontrati.
2.03	2017-04-26	Matteo Maran	<i>Verificatore</i>	Verifica del documento. Causa modifiche effettuate.
2.02	2017-04-25	Eugen Saraci	<i>Progettista</i>	Modificati diagrammi di attività §8. Causa resoconto consegna.
2.01	2017-04-24	Abdelilah Lahmer	<i>Progettista</i>	Contestualizzati i Design Pattern §A. Causa resoconto consegna.

Versione	Data	Autore	Ruolo	Descrizione
2.00	2017-04-10	Abdelilah Lahmer	<i>Responsabile di progetto</i>	Approvazione del documento. Causa consegna del documento.
1.09	2017-03-30	Eugen Saraci	<i>Verificatore</i>	Verifica del documento. Causa modifiche effettuate.
1.08	2017-03-29	Abdelilah Lahmer	<i>Progettista</i>	Aumentato livello di dettaglio nelle descrizioni delle componenti. Causa resoconto consegna.
1.07	2017-03-28	Alessandro Bari	<i>Progettista</i>	Aumentato livello di dettaglio nelle descrizioni delle componenti. Causa resoconto consegna.
1.06	2017-03-27	Matteo Maran	<i>Progettista</i>	Aumentato livello di dettaglio nelle descrizioni delle componenti. Causa resoconto consegna.
1.05	2017-03-24	Andrei Tabacariu	<i>Progettista</i>	Aumentato livello di dettaglio nelle descrizioni delle componenti. Causa resoconto consegna.

Versione	Data	Autore	Ruolo	Descrizione
1.04	2017-03-22	Leonardo Zoso	<i>Progettista</i>	Spostata sezione API in sezione §6. Causa resoconto consegna.
1.03	2017-03-22	Francesca Del Nin	<i>Progettista</i>	Rivisti diagrammi di attività. Causa resoconto consegna.
1.02	2017-03-21	Abdelilah Lahmer	<i>Progettista</i>	Sistemati tipi di ritorno . Causa resoconto consegna.
1.01	2017-03-20	Alessandro Bari	<i>Progettista</i>	Tolto riferimenti a DynDB. Causa modifiche effettuate.
1.00	2017-03-07	Francesca Del Nin	<i>Responsabile di progetto</i>	Approvazione del documento. Causa consegna del documento.
0.45	2017-03-06	Leonardo Zoso	<i>Verificatore</i>	Verifica del documento. Causa modifiche effettuate.
0.44	2017-03-06	Andrei Tabacariu	<i>Progettista</i>	Stesura sezione §13 Tracciamento: Requisiti-Componenti. Causa incremento del documento.

Versione	Data	Autore	Ruolo	Descrizione
0.43	2017-03-06	Alessandro Bari	<i>Progettista</i>	Stesura sezione §12 Tracciamento: Componenti-Requisiti. Causa incremento del documento.
0.42	2017-03-06	Andrei Tabacariu	<i>Progettista</i>	Stesura sezione §11 Tracciamento: Requisiti-Funzioni. Causa incremento del documento.
0.41	2017-03-06	Alessandro Bari	<i>Progettista</i>	Stesura sezione §10 Tracciamento: Funzioni-Requisiti. Causa incremento del documento.
0.40	2017-03-06	Abdelilah Lahmer	<i>Verificatore</i>	Verifica del documento. Causa modifiche effettuate.
0.39	2017-03-03	Abdelilah Lahmer	<i>Progettista</i>	Stesura sezione §9.6 Sequenza generale senza chiamate ad AVS. Causa incremento del documento.
0.38	2017-03-03	Andrei Tabacariu	<i>Progettista</i>	Stesura sezione §9.4 Sequenza con SNS. Causa incremento del documento.

Versione	Data	Autore	Ruolo	Descrizione
0.37	2017-03-03	Alessandro Bari	<i>Progettista</i>	Stesura sezione §9.3 Sequenza con interazione con il database. Causa incremento del documento.
0.36	2017-03-03	Francesca Del Nin	<i>Progettista</i>	Stesura sezione §9.2 Sequenza con interazione con Slack. Causa incremento del documento.
0.35	2017-03-03	Eugen Saraci	<i>Progettista</i>	Stesura sezione §9.1 Sequenza generale. Causa incremento del documento.
0.34	2017-03-02	Leonardo Zoso	<i>Progettista</i>	Stesura sezione §8 Diagramma di attività. Causa incremento del documento.
0.33	2017-03-02	Francesca Del Nin	<i>Progettista</i>	Stesura sezione §7 API Server. Causa incremento del documento.
0.32	2017-03-01	Alessandro Bari	<i>Verificatore</i>	Verifica del documento. Causa modifiche effettuate.
0.31	2017-02-28	Alessandro Bari	<i>Progettista</i>	Stesura sezione Send Audio - /interaction. Causa incremento del documento.



Versione	Data	Autore	Ruolo	Descrizione
0.30	2017-02-27	Eugen Saraci	<i>Progettista</i>	Stesura sezione Delete Costumers - /costumer/id. Causa incremento del documento.
0.29	2017-02-25	Eugen Saraci	<i>Progettista</i>	Stesura sezione Update Costumers - /costumer/uuid. Causa incremento del documento.
0.28	2017-02-25	Eugen Saraci	<i>Progettista</i>	Stesura sezione Get Costumers - /costumer/id. Causa incremento del documento.
0.27	2017-02-24	Leonardo Zoso	<i>Progettista</i>	Stesura sezione Add Costumers - /costumer. Causa incremento del documento.
0.26	2017-02-24	Leonardo Zoso	<i>Progettista</i>	Stesura sezione Get Costumers - /costumer. Causa incremento del documento.

Versione	Data	Autore	Ruolo	Descrizione
0.25	2017-02-23	Abdelilah Lahmer	<i>Progettista</i>	Stesura sezione §6.2.10 Delete Employee Rules - /rules/employee/id . Causa incremento del documento.
0.24	2017-02-23	Matteo Maran	<i>Progettista</i>	Stesura sezione §6.2.9. Causa incremento del documento.
0.23	2017-02-23	Andrei Tabacariu	<i>Progettista</i>	Stesura sezione Get Employee Rules - /rules/employee/id. Causa incremento del documento.
0.22	2017-02-22	Alessandro Bari	<i>Progettista</i>	Stesura sezione §6.2.8 Add Employee Rule - /rules/employee. Causa incremento del documento.
0.21	2017-02-22	Alessandro Bari	<i>Progettista</i>	Stesura sezione §6.2.7 Get Employee Rules - /rules/employee. Causa incremento del documento.

Versione	Data	Autore	Ruolo	Descrizione
0.20	2017-02-21	Francesca Del Nin	<i>Progettista</i>	Stesura sezione §6.2.6 Delete Customer Rules - /rules/customer/id. Causa incremento del documento.
0.19	2017-02-21	Francesca Del Nin	<i>Progettista</i>	Stesura sezione §6.2.5 Update Customer Rules - /rules/customer/id. Causa incremento del documento.
0.18	2017-02-21	Francesca Del Nin	<i>Progettista</i>	Stesura sezione Get Customer Rules - /rules/customer/id. Causa incremento del documento.
0.17	2017-02-20	Leonardo Zoso	<i>Progettista</i>	Stesura sezione §6.2.4 Add Customer Rule - /rules/customer. Causa incremento del documento.
0.16	2017-02-17	Leonardo Zoso	<i>Progettista</i>	Stesura sezione §6.2.3 Get Customer Rules - /rules/customer. Causa incremento del documento.

Versione	Data	Autore	Ruolo	Descrizione
0.15	2017-02-17	Eugen Saraci	<i>Progettista</i>	Stesura sezione §6.2.2 Change password - /auth/changepsw. Causa incremento del documento.
0.14	2017-02-16	Andrei Tabacariu	<i>Progettista</i>	Stesura sezione Logout - /auth/logout. Causa incremento del documento.
0.13	2017-02-16	Matteo Maran	<i>Progettista</i>	Stesura sezione §6.2.1 Login - /auth. Causa incremento del documento.
0.12	2017-02-15	Eugen Saraci	<i>Progettista</i>	Stesura sezione §6.1 Introduzione. Causa incremento del documento.
0.11	2017-02-14	Abdelilah Lahmer	<i>Progettista</i>	Stesura sezione §4.1 AtAVi::Auth-Back-End. Causa incremento del documento.
0.10	2017-02-13	Abdelilah Lahmer	<i>Verificatore</i>	Verifica del documento. Causa modifiche effettuate.

Versione	Data	Autore	Ruolo	Descrizione
0.09	2017-02-13	Alessandro Bari	<i>Progettista</i>	Stesura sezione §3.8 AtAVi::AWS- Back- End::Skill::Util. Causa incremento del documento.
0.08	2017-02-10	Francesca Del Nin	<i>Progettista</i>	Stesura sezione §3.6 AtAVi::AWS- Back- End::Admin::Rules::Employee. Causa incremento del documento.
0.07	2017-02-09	Francesca Del Nin	<i>Progettista</i>	Stesura sezione §3.5 AtAVi::AWS- Back- End::Admin::Rules::Customer. Causa incremento del documento.
0.06	2017-02-08	Francesca Del Nin	<i>Progettista</i>	Stesura sezione §3.4 AtAVi::AWS- Back- End::Admin::Rules. Causa incremento del documento.
0.05	2017-02-07	Eugen Saraci	<i>Progettista</i>	Stesura sezione §3.3 AtAVi::AWS- Back-End::Auth. Causa incremento del documento.

Versione	Data	Autore	Ruolo	Descrizione
0.04	2017-02-06	Leonardo Zoso	<i>Progettista</i>	Stesura sezione §3.2 AtAVi::AWS-Back-End::Admin. Causa incremento del documento.
0.03	2017-02-06	Eugen Saraci	<i>Progettista</i>	Stesura sezione §3 AtAVi::AWS-Back-End. Causa incremento del documento.
0.02	2017-02-03	Leonardo Zoso	<i>Progettista</i>	Stesura sezione §1 Introduzione. Causa incremento del documento.
0.01	2017-02-03	Leonardo Zoso	<i>Progettista</i>	Creata struttura di base del documento. Causa definizione del documento.

## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Scopo del documento . . . . .	1
1.2	Scopo del prodotto . . . . .	1
1.3	Glossario . . . . .	1
1.4	Riferimenti . . . . .	1
1.4.1	Riferimenti normativi . . . . .	1
1.4.2	Riferimenti informativi . . . . .	1
1.5	Avvertenze . . . . .	2
<b>2</b>	<b>Parametri delle funzioni lambda</b>	<b>3</b>
<b>3</b>	<b>Specifica di AWS-Back-End</b>	<b>4</b>
3.1	AtAVi::AWS-Back-End . . . . .	4
3.1.1	Informazioni generali . . . . .	4
3.2	AtAVi::AWS-Back-End::Admin . . . . .	5
3.2.1	Informazioni generali . . . . .	5
3.3	AtAVi::AWS-Back-End::Admin::Auth . . . . .	6
3.3.1	Informazioni generali . . . . .	6
3.3.2	Funzioni Lambda . . . . .	6
3.3.2.1	auth_changePsw . . . . .	6
3.3.2.2	auth_login . . . . .	7
3.4	AtAVi::AWS-Back-End::Admin::Rules . . . . .	8
3.4.1	Informazioni generali . . . . .	8
3.5	AtAVi::AWS-Back-End::Admin::Rules::Customer . . . . .	9
3.5.1	Informazioni generali . . . . .	9
3.5.2	Funzioni Lambda . . . . .	9
3.5.2.1	rules_getAllCustomerRules . . . . .	9
3.5.2.2	rules_getCustomerRule . . . . .	10
3.5.2.3	rules_addCustomerRule . . . . .	10
3.5.2.4	rules_updateCustomerRule . . . . .	11
3.5.2.5	rules_deleteCustomerRule . . . . .	12
3.6	AtAVi::AWS-Back-End::Admin::Rules::Employee . . . . .	13
3.6.1	Informazioni generali . . . . .	13
3.6.2	Funzioni Lambda . . . . .	13
3.6.2.1	rules_getAllEmployeeRules . . . . .	13
3.6.2.2	rules_getEmployeeRule . . . . .	14
3.6.2.3	rules_addEmployeeRule . . . . .	14
3.6.2.4	rules_updateEmployeeRule . . . . .	15
3.6.2.5	rules_deleteEmployeeRule . . . . .	15

---

3.7	AtAVi::AWS-Back-End::Skill . . . . .	17
3.7.1	Informazioni generali . . . . .	17
3.7.2	Funzioni Lambda . . . . .	17
3.7.2.1	skill_core . . . . .	17
3.8	AtAVi::AWS-Back-End::Skill::Util . . . . .	18
3.8.1	Informazioni generali . . . . .	18
3.8.2	Funzioni Lambda . . . . .	18
3.8.2.1	util_sendSlack . . . . .	18
3.8.2.2	util_addCustomer . . . . .	19
3.8.2.3	util_updateCustomer . . . . .	19
3.8.2.4	util_getCustomer . . . . .	20
3.8.2.5	util_getLastEmployee . . . . .	20
3.8.2.6	util_getSlackChannel . . . . .	21
3.8.2.7	util_getSlackChannelById . . . . .	21
3.8.2.8	util_searchCustomerRule . . . . .	21
3.8.2.9	util_checkConstraints . . . . .	22
<b>4</b>	<b>Specifica di Auth-Back-End</b>	<b>23</b>
4.1	AtAVi::Auth-Back-End . . . . .	23
4.1.1	Informazioni generali . . . . .	23
4.2	Modulo config . . . . .	24
4.3	Modulo lwa . . . . .	24
4.4	Modulo interaction . . . . .	25
<b>5</b>	<b>Specifica di Front-End</b>	<b>26</b>
5.1	AtAVi::Front-End . . . . .	26
5.1.1	Informazioni generali . . . . .	26
5.2	AtAVi::Front-End::AtAViApp . . . . .	27
5.2.1	Informazioni generali . . . . .	27
5.3	AtAVi::Front-End::Controllers . . . . .	28
5.3.1	Informazioni generali . . . . .	28
5.3.2	AtAVi::Front-End::Controllers::AdminLoginController .	29
5.3.3	AtAVi::Front-End::Controllers::ConversationController	30
5.3.4	AtAVi::Front-End::Controllers::AudioController . . . .	31
5.3.5	AtAVi::Front-End::Controllers::AdminController . . . .	32
5.3.6	AtAVi::Front-End::Controllers::ChangePasswordController	35
5.4	AtAVi::Front-End::Views . . . . .	36
5.4.1	Informazioni generali . . . . .	36
5.4.2	AtAVi::Front-End::Views::AdminLoginView . . . . .	36
5.4.3	AtAVi::Front-End::Views::AdminView . . . . .	37
5.4.4	AtAVi::Front-End::Views::ChangePasswordView . . . .	37

---



5.4.5	AtAVi::Front-End::Views::UserView . . . . .	37
<b>6</b>	<b>API Gateway</b>	<b>38</b>
6.1	Introduzione . . . . .	38
6.2	Risorse . . . . .	38
6.2.1	Login - /auth . . . . .	38
6.2.2	Change password - /auth/changePSW . . . . .	40
6.2.3	Get Customer Rules - /rules/customer . . . . .	42
6.2.4	Add Customer Rule - /rules/customer . . . . .	44
6.2.5	Update Customer Rule - /rules/customer/{id} . . . . .	46
6.2.6	Delete Customer Rule - /rules/customer/{id} . . . . .	48
6.2.7	Get Employee Rules - /rules/employee . . . . .	50
6.2.8	Add Employee Rule - /rules/employee . . . . .	52
6.2.9	Update Employee Rule - /rules/employee/{id} . . . . .	54
6.2.10	Delete Employee Rule - /rules/employee/{id} . . . . .	56
6.2.11	Events - /events . . . . .	58
6.2.12	Directives - /directives . . . . .	58
<b>7</b>	<b>API Server</b>	<b>59</b>
7.1	Introduzione . . . . .	59
7.2	Risorse . . . . .	59
7.2.1	Amazon Login - /lwa . . . . .	59
7.2.2	Amazon Logout - /lwa/logout . . . . .	60
7.2.3	Events - /events . . . . .	61
7.2.4	Directives - /directives . . . . .	61
<b>8</b>	<b>Diagramma di attivita</b>	<b>62</b>
<b>9</b>	<b>Diagrammi di sequenza</b>	<b>66</b>
9.1	Sequenza generale . . . . .	66
9.2	Sequenza con interazione con Slack . . . . .	67
9.3	Sequenza con interazione con il database . . . . .	67
9.4	Sequenza con SNS . . . . .	68
9.4.1	Lambda - SNS . . . . .	68
9.4.2	SNS - Lambda . . . . .	68
9.5	Login with Amazon . . . . .	69
9.6	Sequenza generale senza chiamate ad AVS . . . . .	70
9.6.1	Sequenza addCustomerRule . . . . .	70
9.6.2	Sequenza updateCustomerRule . . . . .	71
9.6.3	Sequenza deleteCustomerRule . . . . .	71
9.6.4	Sequenza loginAdmin . . . . .	72

<b>10 Tracciamento: Funzioni-Requisiti</b>	<b>73</b>
<b>11 Tracciamento: Requisiti-Funzioni</b>	<b>79</b>
<b>12 Tracciamento: Componenti-Requisiti</b>	<b>88</b>
<b>13 Tracciamento: Requisiti-Componenti</b>	<b>93</b>
<b>A Design Pattern</b>	<b>98</b>
A.1 Model View Whatever . . . . .	98
A.2 Façade . . . . .	98

## Elenco delle figure

1	AtAVi::AWS-Back-End . . . . .	4
2	AtAVi::AWS-Back-End::Admin . . . . .	5
3	AtAVi::AWS-Back-End::Admin::Auth . . . . .	6
4	AtAVi::AWS-Back-End::Admin::Rules . . . . .	8
5	AtAVi::AWS-Back-End::Admin::Rules::Customer . . . . .	9
6	AtAVi::AWS-Back-End::Admin::Rules::Employee . . . . .	13
7	AtAVi::AWS-Back-End::Skill . . . . .	17
8	AtAVi::AWS-Back-End::Skill::Util . . . . .	18
9	AtAVi::Auth-Back-End . . . . .	23
10	AtAVi::Front-End . . . . .	26
11	AtAVi::Front-End::AtAViApp . . . . .	27
12	AtAVi::Front-End::Controllers . . . . .	28
13	AtAVi::Front-End::Controllers::AdminLoginController . . . . .	29
14	AtAVi::Front-End::Controllers::ConversationController . . . . .	30
15	AtAVi::Front-End::Controllers::AudioController . . . . .	31
16	AtAVi::Front-End::Controllers::AdminController . . . . .	32
17	AtAVi::Front-End::Controllers::ChangePasswordController . . . . .	35
18	AtAVi::Front-End::Views . . . . .	36
19	Diagramma di attivita AtAVi pt 1 . . . . .	62
20	Diagramma di attivita AtAVi pt 2 . . . . .	63
21	Diagramma generale . . . . .	66
22	Diagramma con interazione con Slack . . . . .	67
23	Diagramma con interazione con in database . . . . .	67
24	Diagramma con SNS . . . . .	68
25	Diagramma SNS-lambda . . . . .	68
26	Diagramma lwa . . . . .	69
27	Diagramma generale senza chiamate ad AVS . . . . .	70
28	Diagramma addCustomerRule . . . . .	70
29	Diagramma updateCustomerRule . . . . .	71
30	Diagramma deleteCustomerRule . . . . .	71
31	Diagramma loginAdmin . . . . .	72

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento ha lo scopo di definire in dettaglio la struttura, le relazioni e il funzionamento delle parti del prodotto<sub>G</sub>.

## 1.2 Scopo del prodotto

Lo scopo del prodotto richiesto è implementare un assistente virtuale<sub>G</sub> che svolga la funzione di accoglienza clienti. Il prodotto comprenderà:

- un'interfaccia web<sub>G</sub> di interazione con l'utente utilizzatore del sistema;
- gestione dei servizi *AWS Lambda*<sub>G</sub> per l'interazione con le API<sub>G</sub> dell'assistente virtuale;
- un interfacciamento con *Slack*<sub>G</sub>, in modo da poter notificare gli interessati sull'arrivo del cliente.

## 1.3 Glossario

Allo scopo di rendere più semplice e chiara la comprensione di alcuni termini tecnici, ambigui o abbreviazioni presenti nei documenti, viene reso disponibile *Glossario\_v.4.00* nel quale verranno raccolte le varie spiegazioni. Per evidenziare un termine presente in *Glossario\_v.4.00*, esso verrà marcato con il pedice <sub>G</sub>.

## 1.4 Riferimenti

### 1.4.1 Riferimenti normativi

- *NormeDiProgetto\_v.4.00*;
- *AnalisiDeiRequisiti\_v.4.00*.

### 1.4.2 Riferimenti informativi

- Materiale del corso di Ingegneria del Software:
  - Ingegneria dei Requisiti:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L06.pdf>;

- Diagrammi delle classi:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/E02a.pdf>;
- Diagrammi dei package<sub>c</sub>:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/E02b.pdf>.
- Diagrammi di sequenza:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/E03a.pdf>.
- Design Pattern Strutturali:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/E04.pdf>.
- Design Pattern Creazionali:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/E05.pdf>.
- Design Pattern Comportamentali:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/E06.pdf>.
- Dependency Injection:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/E07.pdf>.
- Model View Controller:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/E08.pdf>.

## 1.5 Avvertenze

A causa delle richieste del capitolato C2 analizzate nel documento *Studio-DiFattibilità.v.1.00*, in buona parte dell'architettura<sub>c</sub> presentata nel seguente documento non è previsto l'utilizzo di classi<sub>c</sub>, gerarchie di classi e pattern collegati. Per tale motivo non vengono sempre rispettate le regole e norme consigliate dal linguaggio UML. Per evitare ambiguità durante la lettura di questo documento, viene descritto in che modo interpretare i diagrammi quando non vengono rispettate le regole di UML. Ciò che viene rappresentato come classe in AtAVi::AWS-Back-End (escluso il package AtAVi::AWS-Back-End::Slack) è da intendere come funzione lambda e come unità indipendente.

## 2 Parametri delle funzioni lambda

Ogni funzione lambda prevede tre parametri di default che non verranno riportati per ogni funzione in quanto si presentano sempre con lo stesso nome e significato; i parametri sono i seguenti:

- **event**: parametro opzionale che permette di passare altri parametri alla funzione;
- **context**: parametro opzionale che permette la gestione delle informazioni a runtime, ad esempio il tempo rimasto prima che la funzione termini per timeout;
- **callback**: parametro opzionale che permette di ritornare informazioni o di eseguire del codice al termine della funzione chiamata.

*NodeJS* permette la gestione delle funzioni di callback, esse prevedono due parametri di default:

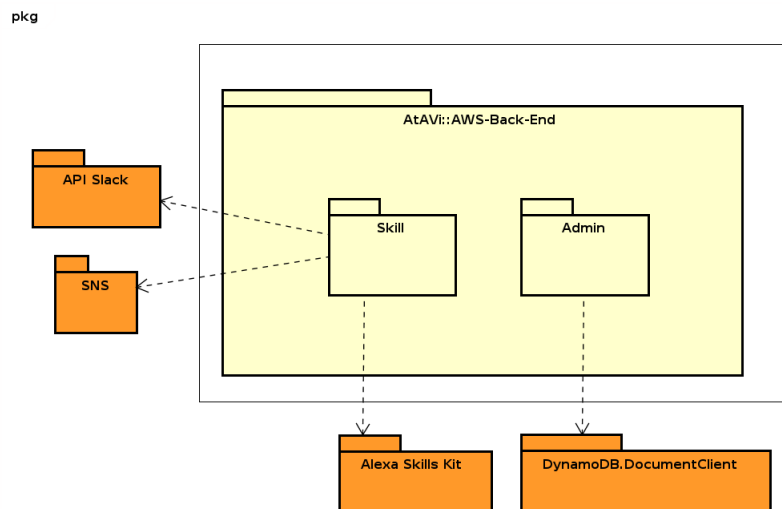
- **error**: permette di indicare un errore, in caso di successo il parametro è *null*;
- **result**: permette di ritornare un valore convertibile in una stringa JSON, in caso di errore questo parametro viene ignorato.

I parametri descritti in input alle funzioni sono passati all'interno dell'oggetto *event*; è possibile accedere a tali parametri tramite la sintassi *event.nomeParametro*.

### 3 Specifica di AWS-Back-End

#### 3.1 AtAVi::AWS-Back-End

##### 3.1.1 Informazioni generali

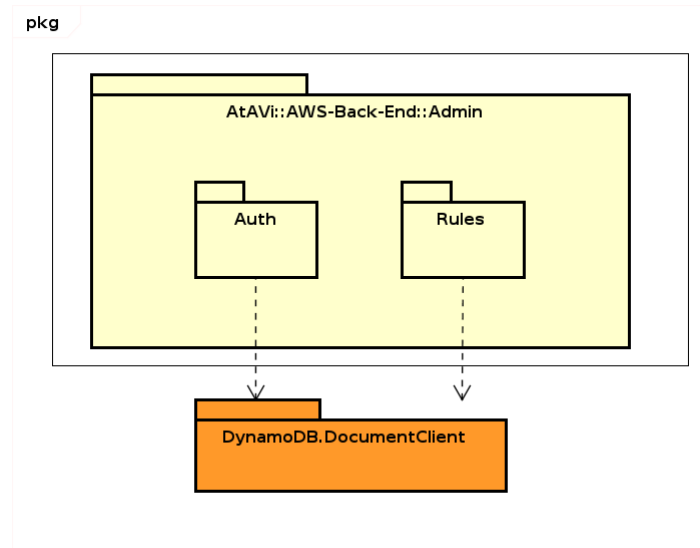


**Figura 1:** AtAVi::AWS-Back-End

- **Descrizione:** package contenente le componenti di AWS-Back-End;
- **Packages contenuti:**
  - Admin: package contenente i vari packages relativi alle funzioni disponibili per l'amministratore;
  - Skill<sub>G</sub>: package contenente il package e le funzioni lambda<sub>G</sub> relativi alla gestione della skill.

## 3.2 AtAVi::AWS-Back-End::Admin

### 3.2.1 Informazioni generali



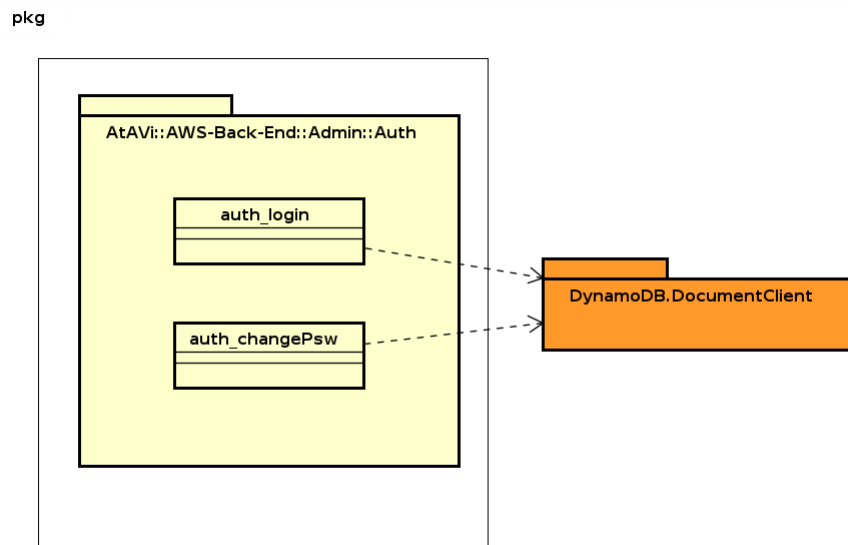
**Figura 2:** AtAVi::AWS-Back-End::Admin

- **Descrizione:** package contenente i vari packages relativi all'amministratore;
- **Packages contenuti:**
  - Auth: package contenente le funzioni lambda che gestiscono l'autenticazione dell'amministratore;
  - Rules: package contenente i vari packages relativi alle regole configurabili dall'amministratore.



### 3.3 AtAVi::AWS-Back-End::Admin::Auth

#### 3.3.1 Informazioni generali



**Figura 3:** AtAVi::AWS-Back-End::Admin::Auth

- **Descrizione:** package contenente le funzioni lambda che gestiscono l'autenticazione dell'amministratore;

#### 3.3.2 Funzioni Lambda

##### 3.3.2.1 auth\_changePsw

- **Descrizione:** funzione che permette il cambio della password del pannello di amministrazione;
- **Input:**
  - `auth` - token che permette di effettuare operazioni di amministrazione;
  - `old_password` - vecchia password;
  - `new_password` - nuova password;
  - `repeat_password` - nuova password.

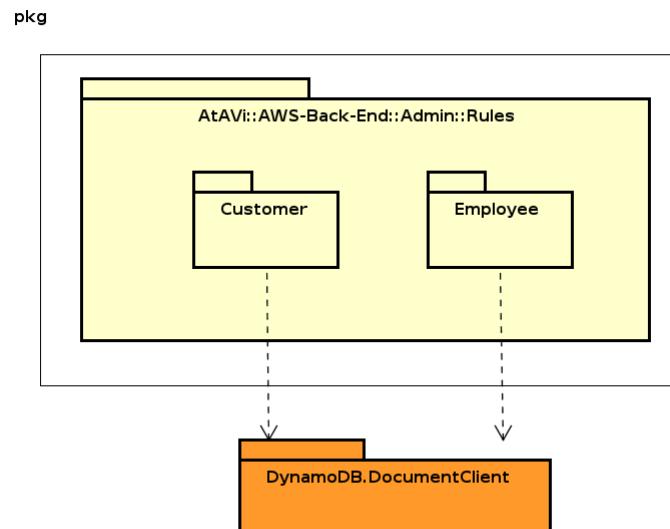
- **Output:**
  - messaggio di successo.
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, dove si trovano le credenziali di accesso per l'amministratore.

### 3.3.2.2 auth\_login

- **Descrizione:** funzione che permette la disconnessione dal pannello di controllo dell'amministratore;
- **Input:**
  - password - password dell'amministratore.
- **Output:**
  - token - token che permette di effettuare operazioni di amministrazione.
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, dove si trovano le credenziali di accesso per l'amministratore.

### 3.4 AtAVi::AWS-Back-End::Admin::Rules

#### 3.4.1 Informazioni generali



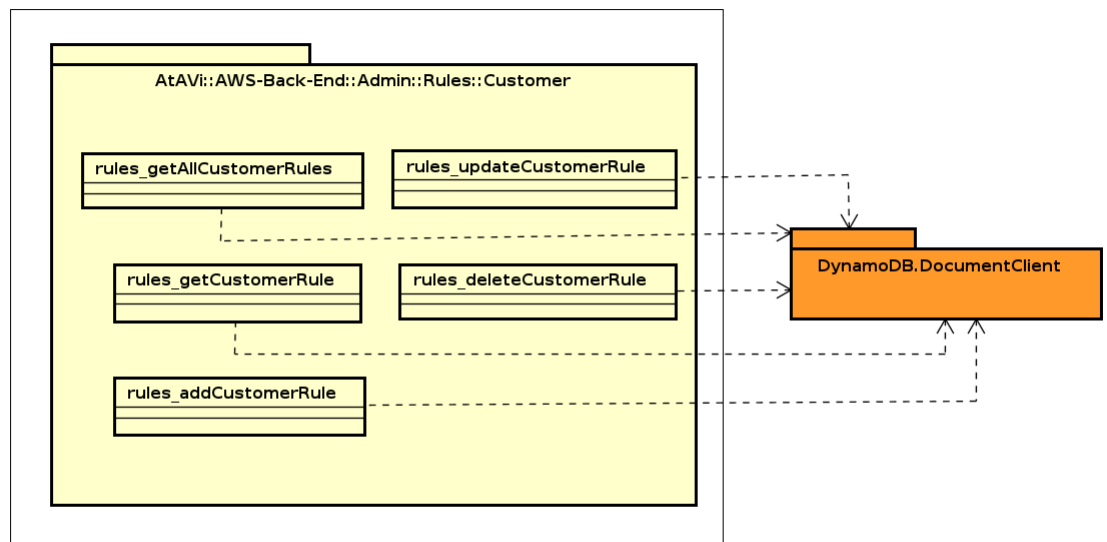
**Figura 4:** AtAVi::AWS-Back-End::Admin::Rules

- **Descrizione:** package contenente i vari packages relativi alle regole configurabili da parte dell'amministratore;
- **Packages contenuti:**
  - Customer: package contenente le funzioni lambda che gestiscono le regole riguardanti i clienti;
  - Employee: package contenente le funzioni lambda che gestiscono le regole riguardanti gli impiegati.

### 3.5 AtAVi::AWS-Back-End::Admin::Rules::Customer

#### 3.5.1 Informazioni generali

pkg



**Figura 5:** AtAVi::AWS-Back-End::Admin::Rules::Customer

- **Descrizione:** package contenente le funzioni lambda che configurano le regole riguardanti i clienti;

#### 3.5.2 Funzioni Lambda

##### 3.5.2.1 rules\_getAllCustomerRules

- **Descrizione:** funzione che ritorna un array contenente tutte le regole dei clienti configurate;
- **Input:**
  - `auth` - token che permette di effettuare operazioni di amministrazione.

- **Output:**
  - array contenente tutte le regole.
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale verrà fatta la richiesta di tutte le regole relative ai clienti.

### 3.5.2.2 rules\_getCustomerRule

- **Descrizione:** funzione che ritorna un oggetto contenente i dati della regola richiesta;
- **Input:**
  - id - uuid della regola.
- **Output:**
  - ritorna un oggetto contenente tutti i campi di una regola.
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale verrà fatta la richiesta dei dati di una specifica regola relativa ai clienti.

### 3.5.2.3 rules\_addCustomerRule

- **Descrizione:** funzione che aggiunge al database, relativo alle regole dei clienti, una nuova regola caratterizzata da un uuid<sub>c</sub>. Saranno inoltre inseriti nome, cognome, azienda relativa al cliente e l'uuid della regola relativa all'impiegato. Questa funzione infine ritorna un messaggio di successo o di errore;
- **Input:**
  - auth - token che permette di effettuare operazioni di amministrazione;
  - first\_name - nome del cliente;
  - last\_name - cognome del cliente;
  - company - azienda del cliente;

- employee\_id - uuid della regola dell'impiegato associato;
- **Output:**
  - id - uuid della regola appena aggiunta;
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale verrà aggiunta una nuova regola relativa ai clienti nel database.

#### 3.5.2.4 rules\_updateCustomerRule

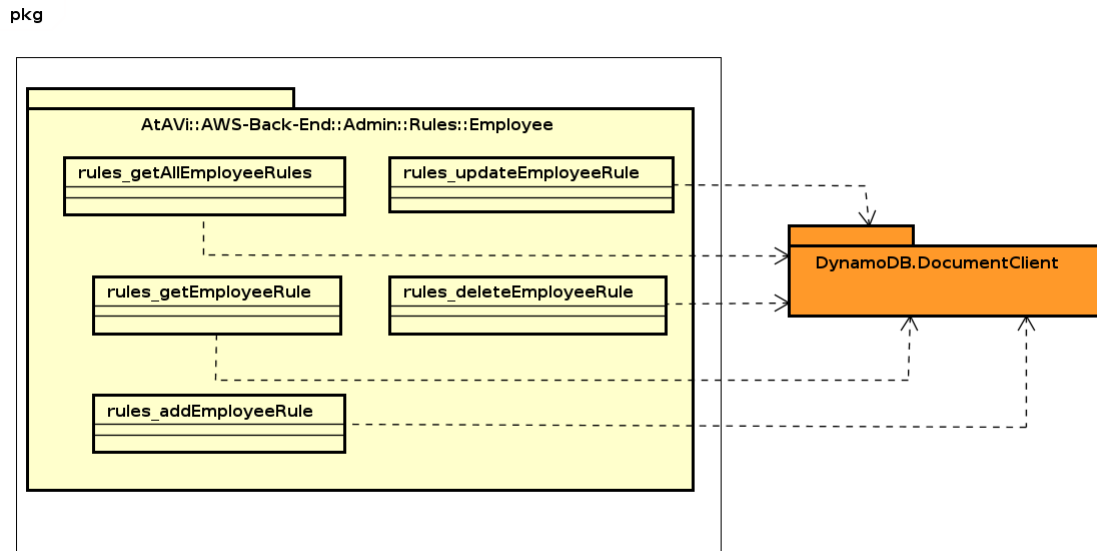
- **Descrizione:** funzione che modifica una regola già esistente, relativa ad un cliente o a clienti della stessa azienda, con i dati forniti dall'amministratore e successivamente ritorna un messaggio di successo o di errore;
- **Input:**
  - auth - token che permette di effettuare operazioni di amministrazione;
  - id - uuid della regola del cliente;
  - first\_name - nome del cliente;
  - last\_name - cognome del cliente;
  - company - azienda del cliente;
  - employee\_id - uuid della regola dell'impiegato associato.
- **Output:**
  - message - messaggio di successo;
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale verrà aggiunta una nuova regola relativa ai clienti nel database.

#### 3.5.2.5 rules\_deleteCustomerRule

- **Descrizione:** funzione che elimina dal database una regola, relativa ad un cliente o ai clienti della stessa azienda, ritornando un messaggio di successo o di errore;
- **Input:**
  - auth - token che permette di effettuare operazioni di amministrazione.;
  - id - uuid della regola;
- **Output:**
  - message - messaggio di successo.
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale verrà eliminata una regola relativa ai clienti nel database.

## 3.6 AtAVi::AWS-Back-End::Admin::Rules::Employee

### 3.6.1 Informazioni generali



**Figura 6:** AtAVi::AWS-Back-End::Admin::Rules::Employee

- **Descrizione:** package contenente le funzioni lambda relative alla configurazione delle regole riguardanti gli impiegati;

### 3.6.2 Funzioni Lambda

#### 3.6.2.1 rules\_getAllEmployeeRules

- **Descrizione:** funzione che ritorna un array contenente tutte le regole configurate relative agli impiegati;
- **Input:**
  - auth - token che permette di effettuare operazioni di amministrazione;
- **Output:**
  - array contenente tutte le regole degli impiegati;



- **Interazione con altri componenti:**

- DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale verrà fatta la richiesta di tutte le regole relative agli impiegati.

### 3.6.2.2 rules\_getEmployeeRule

- **Descrizione:** funzione che ritorna un oggetto contenente i dati della regola richiesta relativa agli impiegati;
- **Input:**
  - id - uuid della regola.
- **Output:**
  - ritorna un oggetto contenente tutti i campi di una regola.
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale verrà fatta la richiesta dei dati di una specifica regola relativa ai clienti.

### 3.6.2.3 rules\_addEmployeeRule

- **Descrizione:** funzione che aggiunge al database relativo alle regole degli impiegati una nuova regola, che sarà caratterizzata da un uuid, inserendo il nome completo dell'impiegato e il nome del canale *Slack*, ritornando un messaggio di successo o di errore;
- **Input:**
  - auth - token che permette di effettuare operazioni di amministrazione;
  - fullname - nome e cognome dell'impiegato;
  - slack - canale slack.
- **Output:**
  - id - uuid della regola appena aggiunta;
- **Interazione con altri componenti:**

- DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale verrà aggiunta nel database una nuova regola relativa agli impiegati.

#### 3.6.2.4 rules\_updateEmployeeRule

- **Descrizione:** funzione che modifica una regola, relativa ad un impiegato, già esistente con i dati forniti dall'amministratore, ritornando un messaggio di successo o di errore;
- **Input:**
  - auth - token che permette di effettuare operazioni di amministrazione;
  - id - uuid della regola;
  - fullname - nome e cognome dell'impiegato;
  - slack - canale slack.
- **Output:**
  - message - messaggio di successo;
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale verrà modificata nel database una nuova regola relativa agli impiegati.

#### 3.6.2.5 rules\_deleteEmployeeRule

- **Descrizione:** funzione che elimina dal database una regola relativa ad un impiegato, eliminando anche le corrispondenti regole associate tra le regole relative ai clienti, ritornando un messaggio di successo o di errore;
- **Input:**
  - auth - token che permette di effettuare operazioni di amministrazione;
  - id - uuid della regola;
- **Output:**

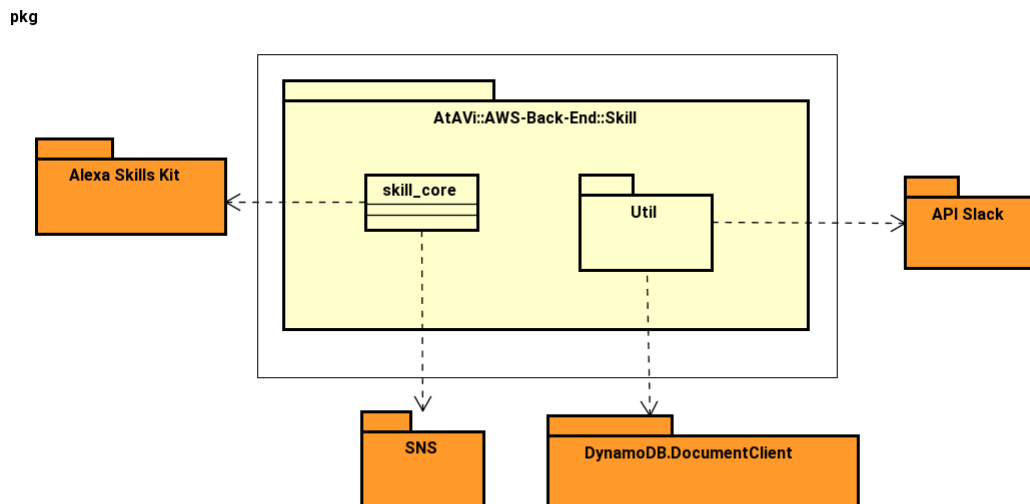
- message - messaggio di successo;

- **Interazione con altri componenti:**

- DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale verrà eliminata una regola relativa un impiegato nel database.

## 3.7 AtAVi::AWS-Back-End::Skill

### 3.7.1 Informazioni generali



**Figura 7:** AtAVi::AWS-Back-End::Skill

- **Descrizione:** package contenente il package Util e le funzioni lambda per l'attivazione e disattivazione di *AtAVi*;
- **Package contenuto:**
  - Util: package contenente le funzioni utili alla skill;

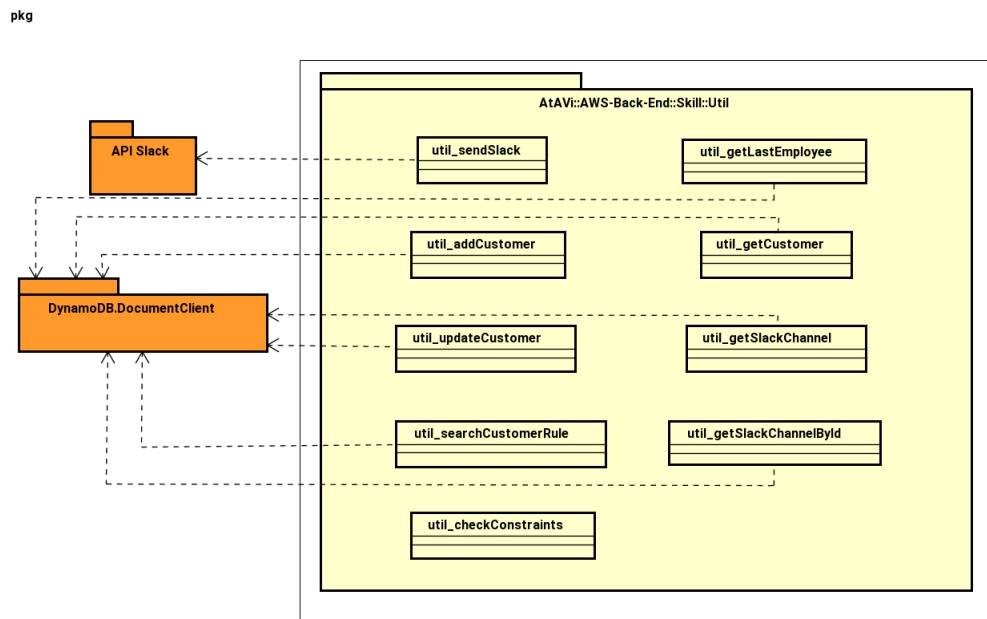
### 3.7.2 Funzioni Lambda

#### 3.7.2.1 skill\_core

- **Descrizione:** funzione che gestisce gli intent della skill;
- **Interazione con altri componenti:**
  - SNS: servizio di notifiche di Amazon Web Services;
  - Alexa Skills Kit: API di Amazon che permette di accedere alla skill.

### 3.8 AtAVi::AWS-Back-End::Skill::Util

#### 3.8.1 Informazioni generali



**Figura 8:** AtAVi::AWS-Back-End::Skill::Util

- **Descrizione:** package contenente le funzioni lambda utili alla skill;

#### 3.8.2 Funzioni Lambda

##### 3.8.2.1 util\_sendSlack

- **Descrizione:** funzione che invia un messaggio *Slack* in un determinato canale;
- **Input:**
  - text - contenuto del messaggio;
  - channel - canale slack.
- **Output:**
  - message - messaggio di successo

**Interazione con altri componenti:**

- API Slack: componente per l'interfacciamento con *Slack*.

**3.8.2.2 util\_addCustomer**

- **Descrizione:** funzione che aggiunge un nuovo cliente al database, inserendo il nome, il cognome, l'azienda e imposta l'ultima visita al giorno corrente;
- **Input:**
  - first\_name - nome del cliente;
  - last\_name - cognome del cliente;
  - company - azienda del cliente;
  - last\_employee - l'ultimo impiegato che l'utente ha incontrato.
- **Output:**
  - id - uuid del cliente appena aggiunto.
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale viene aggiunto un cliente al database.

**3.8.2.3 util\_updateCustomer**

- **Descrizione:** funzione che aggiorna l'ultimo impiegato incontrato e la data dell'ultima visita di un dato cliente;
- **Input:**
  - id - uuid del cliente;
  - last\_employee - l'ultimo dipendente incontrato dal cliente.
- **Output:**
  - message - messaggio di successo.
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale vengono aggiornate le informazioni sul cliente nel database.

#### 3.8.2.4 util\_getCustomer

- **Descrizione:** funzione che ritorna un oggetto contenente i dati relativi ad un determinato cliente;
- **Input:**
  - first\_name - nome del cliente;
  - last\_name - cognome del cliente;
  - company - azienda del cliente.
- **Output:**
  - un oggetto contenente tutti i dati di un cliente con il nome, cognome e azienda forniti in input.
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale viene interrogato il database.

#### 3.8.2.5 util\_getLastEmployee

- **Descrizione:** funzione che ritorna il nome e cognome dell'ultimo impiegato incontrato da un dato cliente;
- **Input:**
  - first\_name - nome del cliente;
  - last\_name - cognome del cliente.
- **Output:**
  - fullname - nome e cognome dell'impiegato.
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale viene interrogato il database.

#### 3.8.2.6 util\_getSlackChannel

- **Descrizione:** funzione che ritorna un oggetto contenente i dati di una determinata regola partendo dal nome dell'impiegato;
- **Input:**
  - fullname - nome dell'impiegato.
- **Output:**
  - oggetto contenente tutti i dati di una regola;
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale viene interrogato il database.

#### 3.8.2.7 util\_getSlackChannelById

- **Descrizione:** funzione che ritorna il canale slack dato l'id della regola;
- **Input:**
  - id - uuid della regola.
- **Output:**
  - fullname - nome e cognome dell'impiegato;
  - slack - canale slack.
- **Interazione con altri componenti:**
  - DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale vengono aggiornate le informazioni sul cliente nel database.

#### 3.8.2.8 util\_searchCustomerRule

- **Descrizione:** funzione che ritorna un oggetto contenente i dati di una determinata regola partendo dal nome dell'impiegato o dall'azienda;
- **Input:**
  - first\_name - nome del cliente;



- last\_name - cognome del cliente;
- company - azienda del cliente.

- **Output:**

- oggetto contenente tutti i dati di una regola;

- **Interazione con altri componenti:**

- DynamoDB.DocumentClient: modulo relativo al database DynamoDB, tramite il quale viene interrogato il database.

### 3.8.2.9 util\_checkConstraints

- **Descrizione:** funzione che controlla se esiste una regola associata ad un cliente, se non esiste controlla se esiste last employee;

- **Input:**

- first\_name - nome dell'interlocutore;
  - last\_name - cognome dell'interlocutore;
  - company - nominativo dell'azienda dell'interlocutore.

- **Output:**

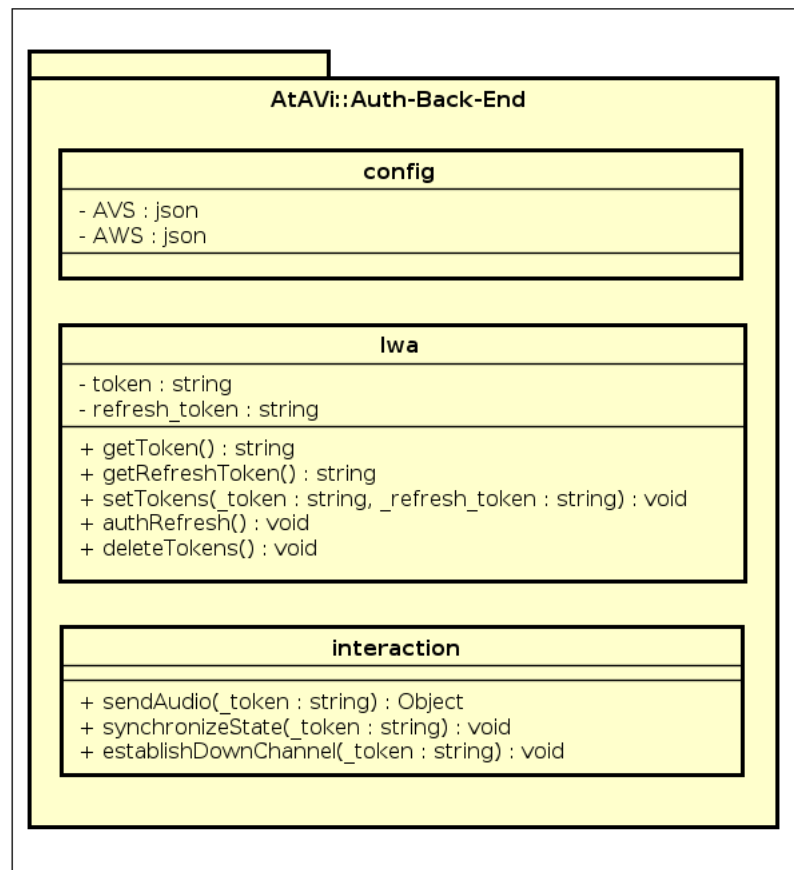
- dest: fullname del destinatario oppure undefined;
  - channel: canale a cui inviare il messaggio oppure undefined.

## 4 Specifica di Auth-Back-End

### 4.1 AtAVi::Auth-Back-End

#### 4.1.1 Informazioni generali

pkg



**Figura 9:** AtAVi::Auth-Back-End

- **Descrizione:** package contenente i moduli relativo al server Node.js<sub>G</sub> utilizzato per la gestione dell'autenticazione con Amazon<sub>G</sub> e l'invio di audio ad API-Gateway<sub>G</sub>.

- **Moduli contenuti:**
  - config: modulo contenente i dati necessari per l'autenticazione con Amazon e l'interazione con API gateway;
  - lwa: modulo che contiene e gestisce token<sub>g</sub> e refresh\_token;
  - interaction: modulo che gestisce le funzioni per l'interazione con API-Gateway.

## 4.2 Modulo config

- **Descrizione:** modulo che gestisce i dati per l'autenticazione con Amazon e l'interazione con API Gateway;
- **Campi dati:**
  - AWS: JSON contenente productID, clientID, clientSecret e altri dati necessari all'autenticazione con Amazon;
  - AVS: JSON contenente i dati necessari a stabilire una connessione con API gateway.

## 4.3 Modulo lwa

- **Descrizione:** modulo che contiene e gestisce token e refresh\_token;
- **Campi dati:**
  - token: stringa che permette l'accesso e l'utilizzo della skill e di Alexa Voice Service;
  - refresh\_token: stringa che permette di richiedere un nuovo token alla scadenza di questo.
- **Metodi:**
  - setTokens(\_token, refresh\_token): funzione che aggiorna i valori dei token;
  - deleteTokens(): funzione che cancella i valori salvati nei token;
  - getToken(): ritorna il token;
  - getRefreshToken(): ritorna il refresh\_token;
  - authRefresh(): funzione che si occupa di richiedere nuovi token e refresh\_token.

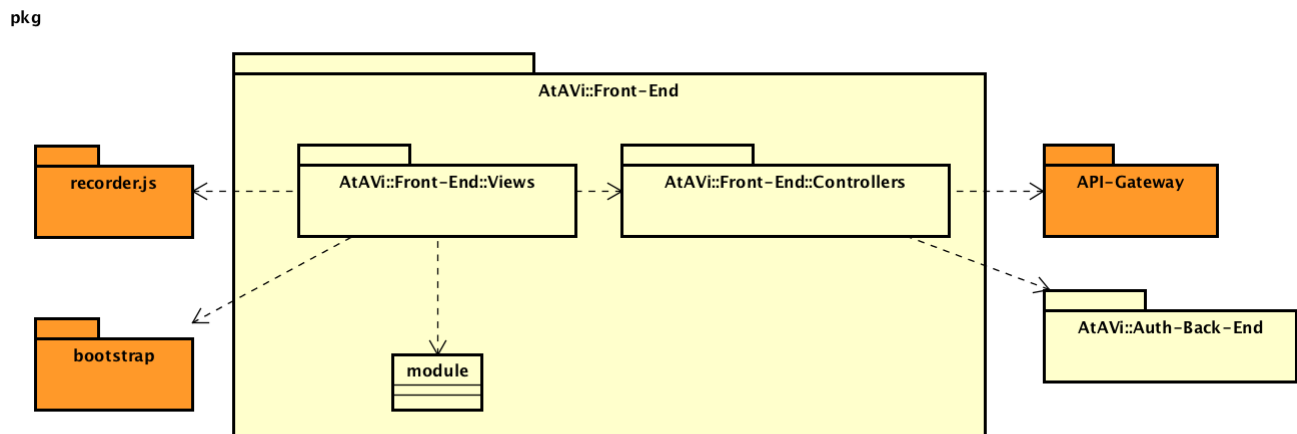
## 4.4 Modulo interaction

- **Descrizione:** modulo che gestisce le funzioni per l'interazione con API-Gateway;
- **Metodi:**
  - `sendAudio(_token)`: funzione che permette di inviare audio ad API gateway;
  - `synchronizeState(_token)`: funzione che permette di stabilire la connessione ad API gateway;
  - `establishDownChannel(_token)`: funzione che permette di mantenere la connessione con API gateway.

## 5 Specifica di Front-End

### 5.1 AtAVi::Front-End

#### 5.1.1 Informazioni generali



**Figura 10:** AtAVi::Front-End

- **Descrizione:** package contenente le componenti del Front End<sub>G</sub> di Atavi;
- **Packages contenuti:**
  - Controllers: package contenente i controllers del front-end dell'applicazione;
  - Views: package contenente le views front-end dell'applicazione;
  - AtAViApp: modulo di angular dell'applicazione.

## 5.2 AtAVi::Front-End::AtAViApp

pkg

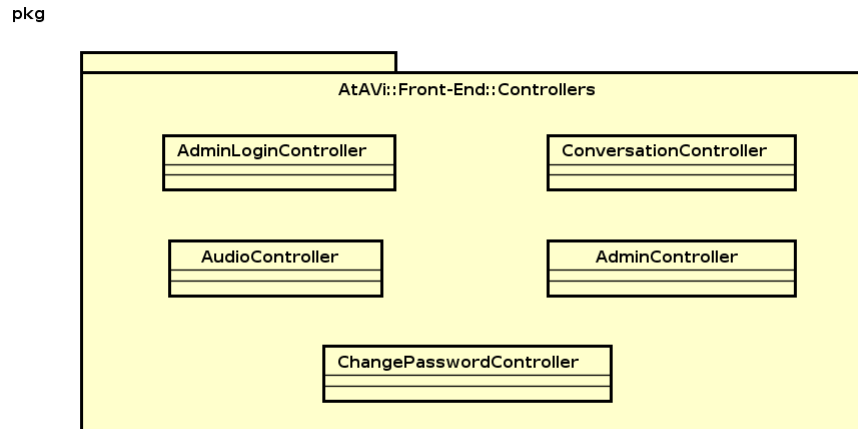
AtAViApp
<ul style="list-style-type: none"><li>- \$rootScope : \$rootScope</li><li>- url : string</li></ul>

**Figura 11:** AtAVi::Front-End::AtAViApp

### 5.2.1 Informazioni generali

- **Descrizione:** contiene il modulo della nostra applicazione AtAviApp;
- **Attributi:**
  - **\$rootScope : \$rootScope:** Campo dati contenente il riferimento all'oggetto globale \$rootScope creato da Angular;
  - **url : string:** variabile di tipo stringa che contiene l'indirizzo url per accedere alle API di AWS.
- **Interazione con altri componenti:**
  - **Front-End::Views:** package che contiene le views dell'applicazione;
  - **Controllers:** package che contiene i controllers dell'applicazione.

## 5.3 AtAVi::Front-End::Controllers



**Figura 12:** AtAVi::Front-End::Controllers

### 5.3.1 Informazioni generali

- **Descrizione:** package che contiene i controller individuati per la parte front-end dell'applicazione;
- **Interazione con altri componenti:**
  - **API-Gateway:** servizio di Amazon che permette la gestione delle API a cui verranno fatte le richieste;
  - **Auth-Back-End:** server in node.JS dell'applicazione, che espone un'API;
  - **AtAViApp:** modulo dell'applicazione Angular.

### 5.3.2 AtAVi::Front-End::Controllers::AdminLoginController

pkg

AdminLoginController
<ul style="list-style-type: none"><li>- \$rootScope : \$rootScope</li><li>- \$scope : \$scope</li><li>- \$http : \$http</li><li>- localStorage : localStorage</li></ul>
<ul style="list-style-type: none"><li>+ login() : void</li></ul>

**Figura 13:** AtAVi::Front-End::Controllers::AdminLoginController

- **Descrizione:** questa classe permette di effettuare il login al nostro sistema come amministratore;
- **Attributi:**
  - **\$rootScope : \$rootScope:** Campo dati contenente il riferimento all'oggetto globale \$rootScope creato da Angular;
  - **\$scope : \$scope:** campo dati contenente un riferimento all'oggetto \$scope creato da Angular. Viene utilizzato come mezzo di comunicazione tra il controller e la view. Contiene gli oggetti che definiscono il viewmodel e il model dell'applicazione;
  - **\$http : \$http:** campo dati che contiene un riferimento al servizio \$http che permette la comunicazione con il protocollo HTTP;
  - **localStorage : localStorage:** campo dati che permette l'accesso ai dati salvati nel localStorage.
- **Metodi:**
  - **login():** la funzione effettua una chiamata POST alle API di AWS, le quali ritornano il token JWT in caso di successo. Tale token verrà utilizzato per accedere alle funzioni di amministrazione.



### 5.3.3 AtAVi::Front-End::Controllers::ConversationController

pkg

ConversationController
- conversation : Array[Object] - \$scope : \$scope - nextAlexa : String - nextUtente : String
+ submitAlexa() : void + submitUtente() : void

**Figura 14:** AtAVi::Front-End::Controllers::ConversationController

- **Descrizione:** questa classe permette di gestire la conversazione con Alexa che viene stampata a video;
- **Relazione con altre classi:**
- **Attributi:**
  - **conversation : Array[Object]:** campo dati che contiene la conversazione tra Alexa e l'utente. Gli oggetti presenti nell'array sono nel seguente formato:
 

```
{
  "sender" : [string],
  "text" : [string]
}
```
  - **nextAlexa : string:** rappresenta la frase corrente di Alexa;
  - **nexttente : string:** rappresenta la frase corrente dell'utente.
- **Metodi:**
  - **submitAlexa():** funzione che aggiunge la frase corrente di Alexa alla conversazione;
  - **submitUtente():** funzione che aggiunge la frase corrente dell'utente alla conversazione.

### 5.3.4 AtAVi::Front-End::Controllers::AudioController

pkg

AudioController
- recording : boolean - \$scope : \$scope - \$http : \$http
+ startRecord() : void + stopRecord() : void + changeStatus() : void

**Figura 15:** AtAVi::Front-End::Controllers::AudioController

- **Descrizione:** questa classe permette di registrare l'audio in input e riprodurre la risposta di Alexa<sub>G</sub>;
- **Utilizzo:** fornisce la funzionalità di registrazione dell'input dell'utente e invia l'audio al server; è inoltre in grado di ricevere la risposta e riprodurla;
- Relazione con altre classi:
  - **ConversationController.**
- **Attributi:**
  - **recording : boolean:** variabile che rappresenta lo stato della registrazione;
  - **\$scope : \$scope:** campo dati contenente un riferimento all'oggetto \$scope creato da Angular. Viene utilizzato come mezzo di comunicazione tra il controller e la view. Contiene gli oggetti che definiscono il viewmodel e il model dell'applicazione;
  - **\$http : \$http:** campo dati che contiene un riferimento al servizio \$http che permette la comunicazione con il protocollo HTTP.
- **Metodi:**
  - **startRecord():** funzione che inizia la registrazione dell'audio;
  - **stopRecord():** funzione che termina la registrazione dell'audio;
  - **changeStatus():** funzione che chiama o startRecord o stopRecord cambiando lo stato della variabile recording.

### 5.3.5 AtAVi::Front-End::Controllers::AdminController

pkg

AdminController
- \$rootScope : \$rootScope - \$scope : \$scope - \$http : \$http - \$index : int - localStorage : localStorage - newName : string - newFname : string - newCompany : string - rule : string - newEmplS : string - newChannel : string - pulsantiS : Array[boolean] - pulsantiE : Array[boolean] - slack : Array[Object] - employee : Array[Object]
+ newSlack() : void + newEmployee() : void + removeSlack() : void + removeEmployee() : void + modifyS() : void + modifyE() : void + saveS() : void + saveE() : void + showSaveEmpl() : boolean + showSaveSlack() : boolean + deleteSlack() : void + deleteEmpl() : void

**Figura 16:** AtAVi::Front-End::Controllers::AdminController

- **Descrizione:** questa classe consente di gestire la creazione o la modifica di una regola riguardante l'associazione tra impiegati e clienti dell'azienda;
- **Attributi:**
  - **\$rootScope : \$rootScope:** Campo dati contenente il riferimento all'oggetto globale \$rootScope creato da Angular;
  - **\$scope : \$scope:** campo dati contenente un riferimento all'oggetto \$scope creato da Angular. Viene utilizzato come mezzo di comunicazione tra il controller e la view. Contiene gli oggetti che definiscono il viewmodel e il model dell'applicazione;
  - **\$http : \$http:** campo dati che contiene un riferimento al servizio \$http che permette la comunicazione con il protocollo HTTP;
  - **\$index : \$index:** variabile che contiene l'indice della riga dell'elemento selezionato nelle tabelle contenenti le regole;

- **localStorage : localStorage** : campo dati che permette l'accesso ai dati salvati nel localStorage;
- **newLname : string**: contiene il cognome di un nuovo cliente;
- **newFname : string**: contiene il cognome di un nuovo cliente;
- **newCompany : string**: contiene l'azienda per cui lavora un nuovo cliente;
- **rule : string**: contiene la regola associata ad un nuovo cliente;
- **newEmplS : string**: contiene il nome completo di un nuovo impiegato dell'azienda;
- **newChannel : string**: contiene un nuovo canale slack da associare ad un impiegato dell'azienda;
- **pulsantiS : Array[boolean]**: contiene un array che, per ogni riga della tabella di associazione Impiegati-Slack, tiene conto se è in corso una modifica oppure no;
- **pulsantiE : Array[boolean]**: contiene un array che, per ogni riga della tabella di associazione Cliente-Slack, tiene conto se è in corso una modifica oppure no;
- **slack : Array[Object]**: contiene un array di JSON che contiene le regole di associazione tra impiegati e canali slack salvate nel database. L'oggetto utilizzato nel seguente è formato:

---

```
{  
  "id" : [string],  
  "fullname" : [string],  
  "rule" : [string]  
}
```

---

- **employee : Array[Object]**: contiene un array di JSON che contiene le regole di associazione tra clienti e impiegati salvate nel database. L'oggetto utilizzato è nel seguente formato:

---

```
{  
  "id" : [string],  
  "first_name" : [string],  
  "last_name" : [string],  
  "company" : [string]  
}
```

---

- Metodi:

- **newSlack()**: metodo che consente di creare una regola che associa un impiegato ad un canale Slack;
- **newEmployee()**: metodo che consente di creare una regola che associa un cliente ad un impiegato;
- **removeSlack(index)**: metodo che consente di rimuovere una regola che associa un impiegato ad un canale Slack;
- **removeEmployee(index)**: metodo che consente di rimuovere una regola che associa un cliente ad un impiegato;
- **modifyS(index)**: metodo che modifica il valore dell'array pulsantiS quando viene richiesta la modifica di un campo;
- **modifyE(index)**: metodo che modifica il valore dell'array pulsantiE quando viene richiesta la modifica di un campo;
- **saveS(index)**: metodo che effettua una chiamata alle nostre API, consentendo di modificare una regola di associazione tra impiegati e canali Slack secondo quanto richiesto dall'amministratore;
- **saveE(index)**: metodo che effettua una chiamata alle nostre API, consentendo di modificare una regola di associazione tra clienti e impiegati secondo quanto richiesto dall'amministratore;
- **showSaveEmpl(index)**: metodo che ritorna il valore dell'elemento in posizione index dell'array pulsantiE;
- **ShowSaveSlack(index)**: metodo che ritorna il valore dell'elemento in posizione index dell'array pulsantiS;
- **deleteSlack(index)**: metodo che rimuove la regola della riga - index di associazione tra impiegati e canali slack;
- **deleteEmpl(index)**: metodo che rimuove la regola della riga index di associazione tra clienti e impiegati.

### 5.3.6 AtAVi::Front-End::Controllers::ChangePasswordController

pkg

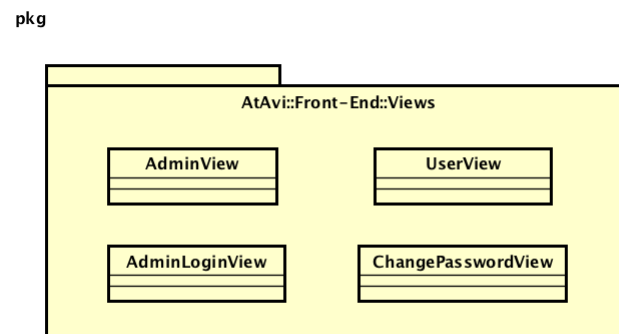
ChangePasswordController
<ul style="list-style-type: none"><li>- \$rootScope : \$rootScope</li><li>- \$scope : \$scope</li><li>- \$http : \$http</li><li>- localStorage : localStorage</li></ul>
<ul style="list-style-type: none"><li>+ change() : void</li></ul>

**Figura 17:** AtAVi::Front-End::Controllers::ChangePasswordController

- **Descrizione:** questa classe permette di cambiare la password di amministrazione del nostro sistema;
- **Interazione con altri componenti:**
  - Front-End::Views::ChangePasswordView;
- **Attributi:**
  - **\$rootScope : \$rootScope:** Campo dati contenente il riferimento all'oggetto globale \$rootScope creato da Angular;
  - **\$scope : \$scope:** campo dati contenente un riferimento all'oggetto \$scope creato da Angular. Viene utilizzato come mezzo di comunicazione tra il controller e la view. Contiene gli oggetti che definiscono il viewmodel e il model dell'applicazione;
  - **\$http : \$http:** campo dati che contiene un riferimento al servizio \$http che permette la comunicazione con il protocollo HTTP;
  - **localStorage : localStorage:** campo dati che permette l'accesso ai dati salvati nel localStorage.
- **Metodi:**
  - **change():** la funzione effettua una chiamata POST alle API di AWS, passando i valori prelevati dalla form della view che sono: vecchia password, nuova password e conferma password. Se i valori inseriti sono corretti questa chiamata esegue il cambio

della password di amministrazione. Infine questa funzione mostra un messaggio di errore o di successo a seconda del risultato dell'operazione richiesta.

## 5.4 AtAVi::Front-End::Views



**Figura 18:** AtAVi::Front-End::Views

### 5.4.1 Informazioni generali

- **Descrizione:** package che contiene le views individuati per la parte front-end dell'applicazione;
- **Interazione con altri componenti:**
  - Front-End::Controllers.

### 5.4.2 AtAVi::Front-End::Views::AdminLoginView

- **Descrizione:** contenente le form necessarie per effettuare il login di amministrazione;
- **Relazione con altre classi:**
  - AdminLoginController.

#### 5.4.3 AtAVi::Front-End::Views::AdminView

- **Descrizione:** contiene le form e gli input necessari per gestire le regole;
- **Relazione con altre classi:**
  - AdminController.

#### 5.4.4 AtAVi::Front-End::Views::ChangePasswordView

- **Descrizione:** contiene le form necessarie per effettuare il cambio della password di amministrazione;
- **Relazione con altre classi:**
  - ChangePasswordController.

#### 5.4.5 AtAVi::Front-End::Views::UserView

- **Descrizione:** contenente di visualizzare la conversazione con Alexa e di interagire con il sistema mediante il pulsante di attivazione;
- **Relazione con altre classi:**
  - AudioController;
  - ConversationController.



## 6 API Gateway

### 6.1 Introduzione

API Gateway è un servizio offerto da Amazon che risulta fondamentale nell'architettura adottata dal gruppo per questo capitolato. API Gateway espone un'interfaccia RESTful<sub>g</sub> che verrà utilizzata principalmente come trigger per attivare le funzioni lambda presenti in AtAVi::AWS-Back-End.

### 6.2 Risorse

#### 6.2.1 Login - /auth

- **Description:** permette il login al pannello di amministrazione. In caso di successo viene ritornato un token che permette di compiere operazioni nell'area di amministrazione, in caso di errore viene ritornato un messaggio d'errore;
- **Resource:** /auth;
- **Authorization:** nessuna;
- **Method:** POST;
- **Request BODY:**

---

```
{
  "password": [string] //required
}
```

---

- **Responses:**

---

```
// HTTP Status Code 200
// OK
```

```
{
  "token": [string]
}
```

---

---

```
// HTTP Status Code 400
// Bad Request
```

```
{
  "message": [string]
```

---

```
}
```

---

```
// HTTP Status Code 500  
// Internal Server Error
```

---

```
{  
  "message": [string]  
}
```

---

### 6.2.2 Change password - /auth/changePSW

- **Description:** permette il cambio password dell'amministratore;
- **Resource:** /auth/changePSW;
- **Authorization:** token;
- **Method:** POST;
- **Request BODY:**

---

```
{  
  "old_password": [string] //required  
  "new_password": [string] //required  
  "repeat_password": [string] //required  
}
```

---

- **Responses:**

---

```
// HTTP Status Code 200  
// OK
```

```
{  
  //empty  
}
```

---

---

```
// HTTP Status Code 400  
// Bad Request
```

```
{  
  "message": [string]  
}
```

---

---

```
// HTTP Status Code 401  
// Not Authorized
```

```
{  
  "message": [string]  
}
```

---

---

```
// HTTP Status Code 500  
// Internal Server Error
```

---

```
{  
  "message": [string]  
}
```

---

### 6.2.3 Get Customer Rules - /rules/customer

- **Description:** permette di ottenere un array contenente tutte le regole dei clienti presenti nel database;
- **Resource:** /rules/customer;
- **Authorization:** token;
- **Method:** GET<sub>G</sub>;
- **Request URL:**

---

```
{  
    // empty  
}
```

---

- **Responses:**

---

```
// HTTP Status Code 200  
// OK  
  
{  
  [  
    {  
      "id": [string],  
      "first_name": [string],  
      "last_name": [string],  
      "company": [string],  
      "employee_id": [string]  
    }  
  ]  
}
```

---

---

```
// HTTP Status Code 401  
// Not Authorized  
  
{  
  "message": [string]  
}
```

---

---

```
// HTTP Status Code 500
```

---

```
// Internal Server Error

{
  "message": [string]
}
```

---

#### 6.2.4 Add Customer Rule - /rules/customer

- **Description:** permette di aggiungere una regola ad un cliente;
- **Resource:** /rules/customer;
- **Authorization:** token;
- **Method:** POST;
- **Request BODY:**

---

```
{  
  "first_name": [string], // required  
  "last_name": [string], // required  
  "company": [string], // required  
  "employee_id": [string] // required  
}
```

---

- **Responses:**

---

```
// HTTP Status Code 200  
// OK
```

```
{  
  "id": [string]  
}
```

---

---

```
// HTTP Status Code 400  
// Bad Request
```

```
{  
  "message": [string]  
}
```

---

---

```
// HTTP Status Code 401  
// Not Authorized
```

```
{  
  "message": [string]  
}
```

---

---

```
// HTTP Status Code 500
// Internal Server Error

{
  "message": [string]
}
```

---



### 6.2.5 Update Customer Rule - /rules/customer/{id}

- **Description:** permette di aggiornare la regola di un cliente;
- **Resource:** /rules/customer/{id};
- **Authorization:** token;
- **Method:** PUT<sub>G</sub>;
- **Request BODY:**

---

```
{
  "first_name": [string], // required
  "last_name": [string], // required
  "company": [string], // required
  "employee_id": [string] // required
}
```

---

- **Responses:**

---

```
// HTTP Status Code 200
// OK
```

```
{
  //empty
}
```

---

---

```
// HTTP Status Code 400
// Bad Request
```

```
{
  "message": [string]
}
```

---

---

```
// HTTP Status Code 401
// Not Authorized
```

```
{
  "message": [string]
}
```

---

---

```
// HTTP Status Code 500
// Internal Server Error

{
  "message": [string]
}
```

---

### 6.2.6 Delete Customer Rule - /rules/customer/{id}

- **Description:** permette di eliminare la regola di un cliente;
- **Resource:** /rules/customer/{id};
- **Authorization:** token;
- **Method:** DELETE<sub>G</sub>;
- **Request BODY:**

---

```
{  
  // empty  
}
```

---

- **Responses:**

---

```
// HTTP Status Code 200  
// OK
```

```
{  
  //empty  
}
```

---

---

```
// HTTP Status Code 204  
// Empty Response
```

```
{  
  //empty  
}
```

---

---

```
// HTTP Status Code 400  
// Bad Request
```

```
{  
  "message": [string]  
}
```

---

---

```
// HTTP Status Code 401  
// Not Authorized
```

```
{
```

---

```
    "message": [string]
  }
```

---

```
// HTTP Status Code 500
// Internal Server Error
```

```
{
  "message": [string]
}
```

---

### 6.2.7 Get Employee Rules - /rules/employee

- **Description:** permette di ottenere un array contenente tutte le regole dei dipendenti presenti nel database;
- **Resource:** /rules/employee;
- **Authorization:** token;
- **Method:** GET;
- **Request URL:**

---

```
{  
    // empty  
}
```

---

- **Responses:**

---

```
// HTTP Status Code 200  
// OK  
  
{  
    [  
        {  
            "id": [string],  
            "fullname": [string],  
            "slack": [string]  
        }  
    ]  
}
```

---

---

```
// HTTP Status Code 401  
// Not Authorized  
  
{  
    "message": [string]  
}
```

---

---

```
// HTTP Status Code 500  
// Internal Server Error
```

---

```
{  
  "message": [string]  
}
```

---

### 6.2.8 Add Employee Rule - /rules/employee

- **Description:** permette di aggiungere una regola ad un dipendente;
- **Resource:** /rules/customer;
- **Authorization:** token;
- **Method:** POST;
- **Request BODY:**

---

```
{
  "fullname": [string], // required
  "slack": [string], // required
}
```

---

- **Responses:**

---

```
// HTTP Status Code 200
// OK
```

```
{
  "id": [string]
}
```

---

---

```
// HTTP Status Code 400
// Bad Request
```

```
{
  "message": [string]
}
```

---

---

```
// HTTP Status Code 401
// Not Authorized
```

```
{
  "message": [string]
}
```

---

---

```
// HTTP Status Code 500
// Internal Server Error
```

---

```
{  
  "message": [string]  
}
```

---



### 6.2.9 Update Employee Rule - /rules/employee/{id}

- **Description:** permette di aggiornare la regola di un dipendente;
- **Resource:** /rules/employee/{id};
- **Authorization:** token;
- **Method:** PUT;
- **Request BODY:**

---

```
{
  "fullname": [string], // required
  "slack": [string] // required
}
```

---

- **Responses:**

---

```
// HTTP Status Code 200
// OK
```

```
{
  //empty"
}
```

---

---

```
// HTTP Status Code 400
// Bad Request
```

```
{
  "message": [string]
}
```

---

---

```
// HTTP Status Code 401
// Not Authorized
```

```
{
  "message": [string]
}
```

---

---

```
// HTTP Status Code 500
// Internal Server Error
```

---

```
{  
  "message": [string]  
}
```

---

#### 6.2.10 Delete Employee Rule - /rules/employee/{id}

- **Description:** permette di eliminare la regola di un dipendente;
- **Resource:** /rules/employee/{id};
- **Authorization:** token;
- **Method:** DELETE;
- **Request BODY:**

---

```
{  
  // empty  
}
```

---

- **Responses:**

---

```
// HTTP Status Code 200  
// OK
```

```
{  
  //empty  
}
```

---

---

```
// HTTP Status Code 204  
// Empty Response
```

```
{  
  //empty  
}
```

---

---

```
// HTTP Status Code 400  
// Bad Request
```

```
{  
  "message": [string]  
}
```

---

---

```
// HTTP Status Code 401  
// Not Authorized
```

```
{
```

---

```
    "message": [string]
  }
```

---

```
// HTTP Status Code 500
// Internal Server Error
```

```
{
  "message": [string]
}
```

---

#### 6.2.11 Events - /events

- **Description:** questa risorsa funge da proxy, qualsiasi richiesta inviata a questo indirizzo verrà inoltrata a <https://avs-alexa-na.amazon.com/events>.
- **Resource:** /events;
- **Authorization:** Nessuna;
- **Method:** POST;
- **Request BODY:** fare riferimento a <https://developer.amazon.com/public/solutions/alexa/alexa-voice-service/content/avs-api-overview>

#### 6.2.12 Directives - /directives

- **Description:** questa risorsa funge da proxy, qualsiasi richiesta inviata a questo indirizzo verrà inoltrata a <https://avs-alexa-na.amazon.com/directives>.
- **Resource:** /directives;
- **Authorization:** Nessuna;
- **Method:** POST;
- **Request BODY:** fare riferimento a <https://developer.amazon.com/public/solutions/alexa/alexa-voice-service/content/avs-api-overview>

## 7 API Server

### 7.1 Introduzione

AtAVi::Auth-Back-End rappresenta un server in Node.JS il cui scopo è quello di fornire un' interfaccia RESTful ad AtAVi::Front-End facendo da tramite fra il Front-End, API Gateway e il servizio LWA. L'utilizzo di questo server nasce dall'esigenza di dover mantenere privati alcuni parametri necessari per la comunicazione con Alexa Voice Service e LWA, se AtAVi::Auth-Back-End non ci fosse, i dati privati andrebbero salvati nei file javascript del Front-End risultando facilmente accessibili a chiunque.

### 7.2 Risorse

#### 7.2.1 Amazon Login - /lwa

- **Description:** permette il login con Amazon. In caso di successo viene visualizzata la schermata di Login with Amazon. In caso d'errore viene ritornato un messaggio d'errore;
- **Resource:** /lwa;
- **Authorization:** nessuna;
- **Method:** POST;
- **Request BODY:**

---

```
{  
  // empty  
}
```

---

- **Responses:**

---

```
// HTTP Status Code 200  
// OK  
  
{  
  "success_message": [string]  
}
```

---

### 7.2.2 Amazon Logout - /lwa/logout

- **Description:** permette il logout da Amazon. In caso di successo viene visualizzata la schermata principale di *AtAVi*. In caso d'errore viene ritornato messaggio d'errore;
- **Resource:** /lwa/logout;
- **Authorization:** auth\_token;
- **Method:** POST;
- **Request BODY:**

---

```
{  
    // empty  
}
```

---

- **Responses:**

---

```
// HTTP Status Code 200  
// OK  
  
{  
    "success_message": [string]  
}
```

---

---

```
// HTTP Status Code 401  
// Not Authorized  
  
{  
    "message": [string]  
}
```

---

### 7.2.3 Events - /events

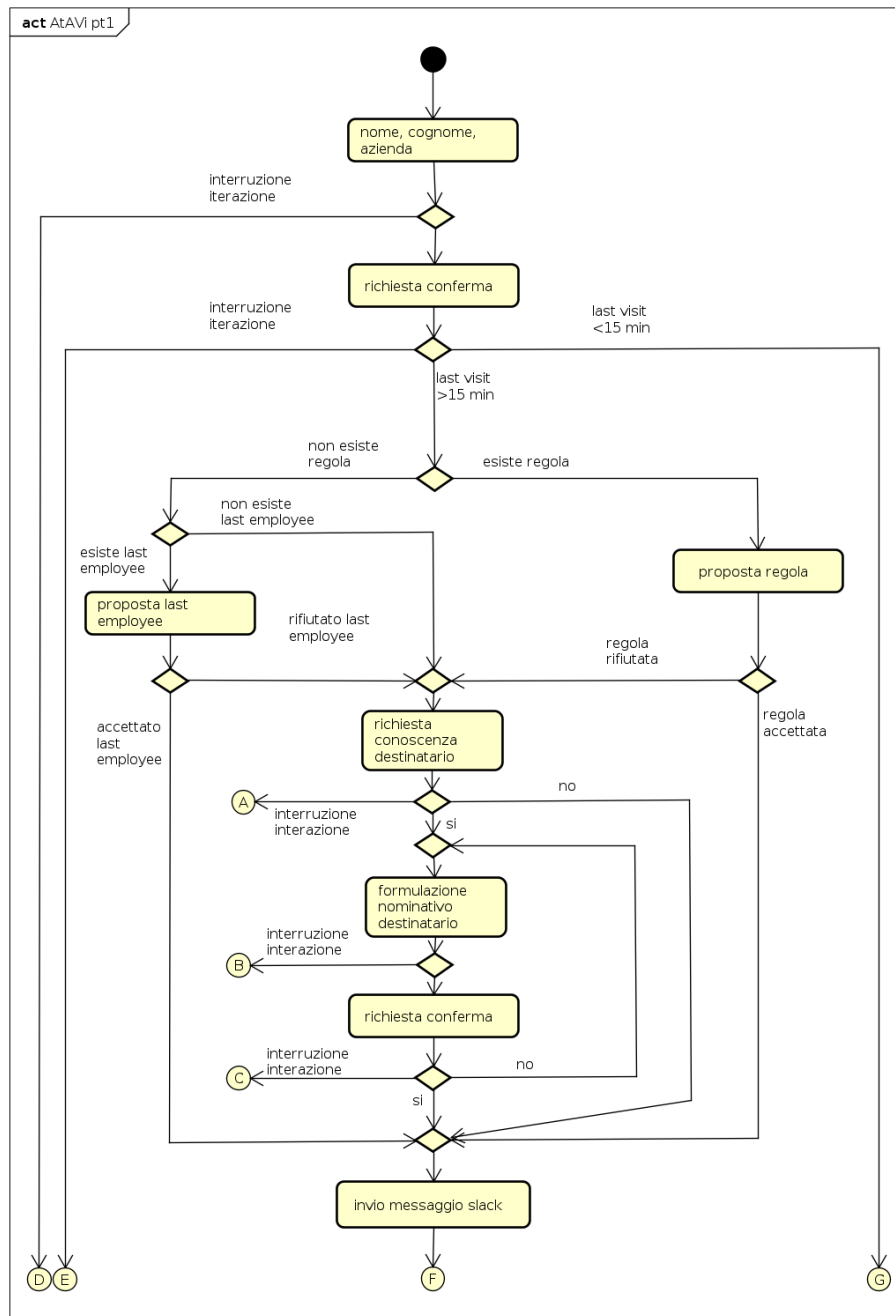
- **Description:** questa risorsa funge da proxy, qualsiasi richiesta inviata a questo indirizzo verrà inoltrata a <https://avs-alexa-na.amazon.com/events>.
- **Resource:** /events;
- **Authorization:** Nessuna;
- **Method:** POST;
- **Request BODY:** fare riferimento a <https://developer.amazon.com/public/solutions/alexa/alexa-voice-service/content/avs-api-overview>

### 7.2.4 Directives - /directives

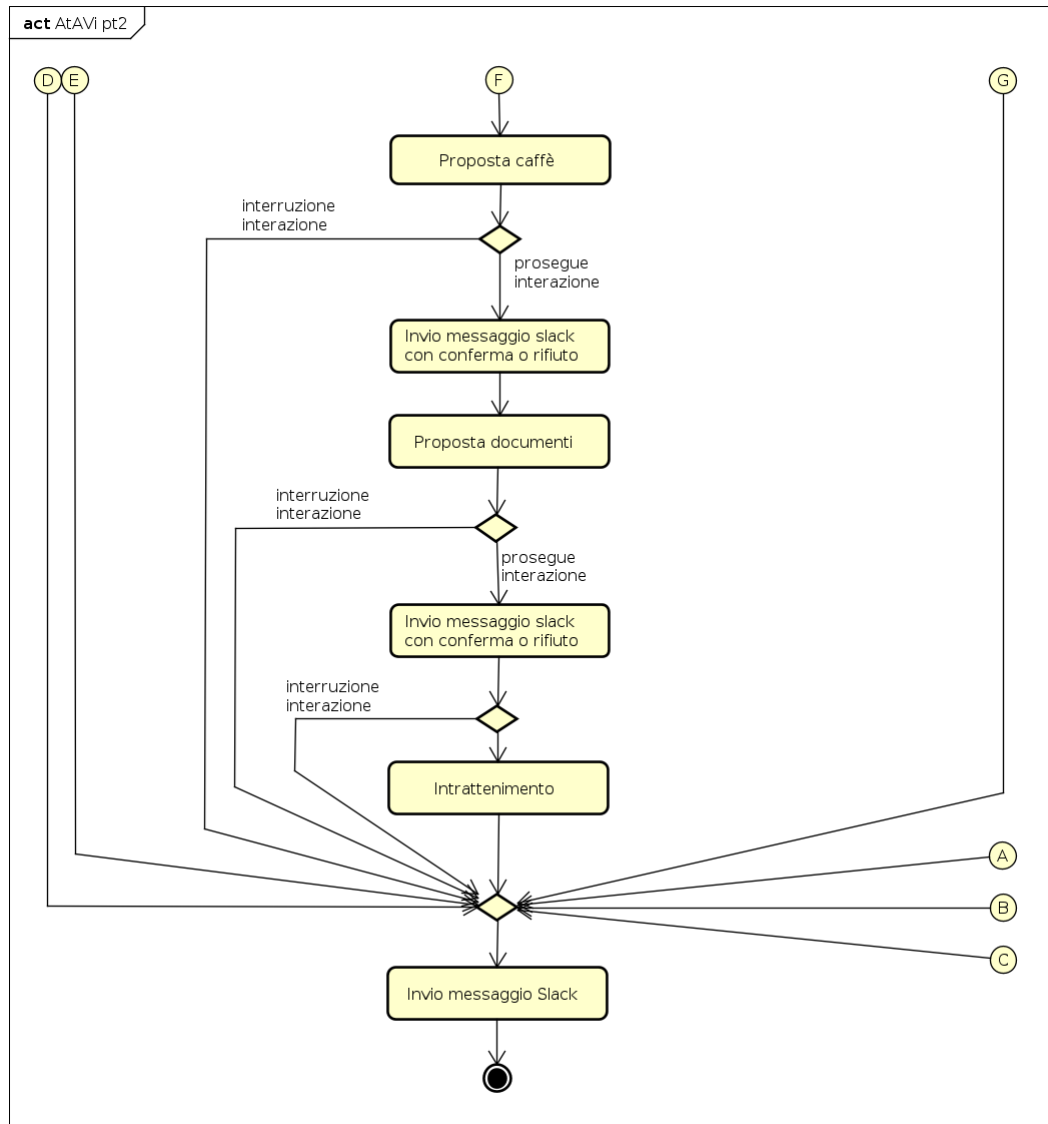
- **Description:** questa risorsa funge da proxy, qualsiasi richiesta inviata a questo indirizzo verrà inoltrata a <https://avs-alexa-na.amazon.com/directives>.
- **Resource:** /directives;
- **Authorization:** Nessuna;
- **Method:** POST;
- **Request BODY:** fare riferimento a <https://developer.amazon.com/public/solutions/alexa/alexa-voice-service/content/avs-api-overview>



## 8 Diagramma di attivita



**Figura 19:** Diagramma di attivita AtAVi pt 1



**Figura 20:** Diagramma di attività AtAVi pt 2

Le attività rappresentate nel diagramma prevedono delle sottoattività interne ad *AtAVi* quali ad esempio l'invio dell'audio dal front-end ad AVS o la chiamata di funzioni interne; nel diagramma è rappresentato il flusso principale di interazione tra *AtAVi* e l'interlocutore. All'inizio dell'interazione tra *AtAVi* e l'interlocutore viene richiesto di fornire nome, cognome e azienda, viene poi richiesta una conferma dei dati. Viene controllato l'orario dell'ultima visita, se l'intervallo è minore di 15 minuti viene mandato direttamente un messaggio *Slack* per avvisare che l'interlocutore sta ancora aspettando di essere accolto. Se invece l'intervallo è maggiore di 15 minuti viene consultato il database:

- se esiste una regola che associa l'interlocutore o l'azienda a uno dei dipendenti allora esso viene proposto all'interlocutore come destinatario;
- se non esiste una regola per l'interlocutore o l'azienda, si verifica se in precedenza è già stata registrata una visita;
- l'interlocutore può interrompere l'interazione.

Se esiste una regola e il destinatario proposto all'interlocutore viene rifiutato, allora si procede a verificare se in precedenza è già stata registrata una visita, in ognuno dei casi in cui viene controllata la presenza di una visita passata:

- se esiste una visita registrata, il destinatario di quest'ultima viene proposto all'interlocutore;
- se non esiste una visita registrata si procede chiedendo all'interlocutore di indicare un destinatario;
- l'interlocutore può interrompere l'interazione.

Nel caso in cui l'interlocutore rifiuti il destinatario proposto in base all'ultima visita, si richiede di indicare un destinatario, successivamente viene chiesta una conferma, se viene rifiutata si ripete l'operazione. L'interlocutore può alternativamente interrompere l'interazione.

Quando l'interlocutore accetta uno dei destinatari proposti o fornisce e conferma un destinatario ad *AtAVi*, viene inviato un messaggio *Slack* che avvisa dell'arrivo dell'interlocutore in azienda. Successivamente *AtAVi* chiede all'interlocutore se gradisce un caffè, l'interlocutore può accettare o rifiutare, alternativamente l'interlocutore può interrompere l'interazione. In caso di risposta positiva viene inviato un messaggio tramite *Slack*.

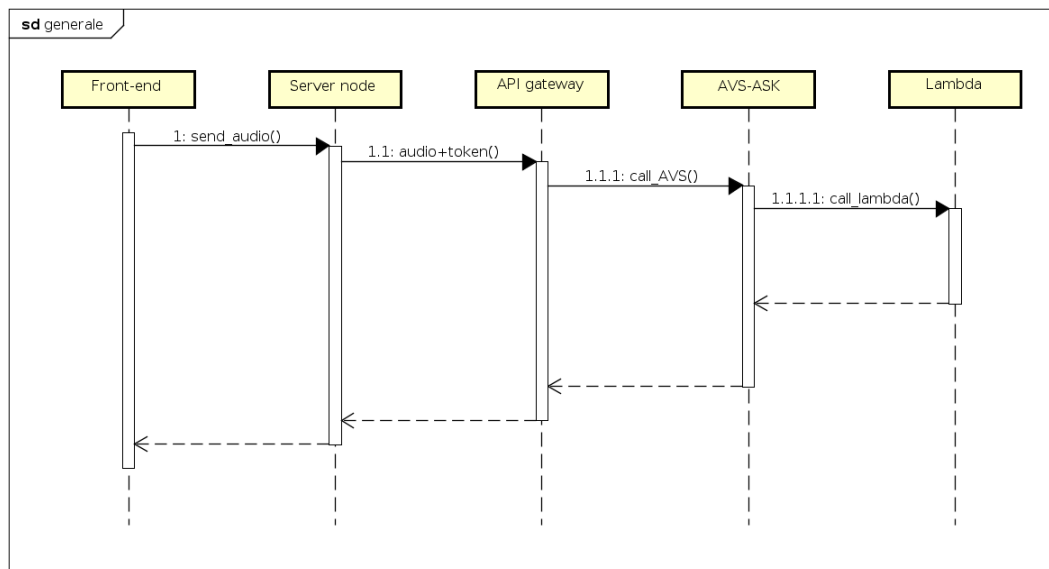
*AtAVi* chiede all'interlocutore se gradisce ricevere dei documenti e l'interlocutore può accettare, rispondere che non sono necessari o interrompere l'interazione, in caso di risposta positiva viene inviato un messaggio tramite *Slack*.

*AtAVi* prosegue con l'intrattenimento dell'interlocutore oppure l'interazione viene interrotta. In tutti i casi in cui l'interazione è interrotta, oppure alla fine dell'intrattenimento, viene inviato un messaggio tramite *Slack*.

## 9 Diagrammi di sequenza

Per rappresentare i diagrammi di sequenza abbiamo considerato come entità i principali componenti del prodotto *AtAVi* e per ognuno di essi abbiamo indicato il periodo di attivazione rispetto al flusso dei casi d'uso.

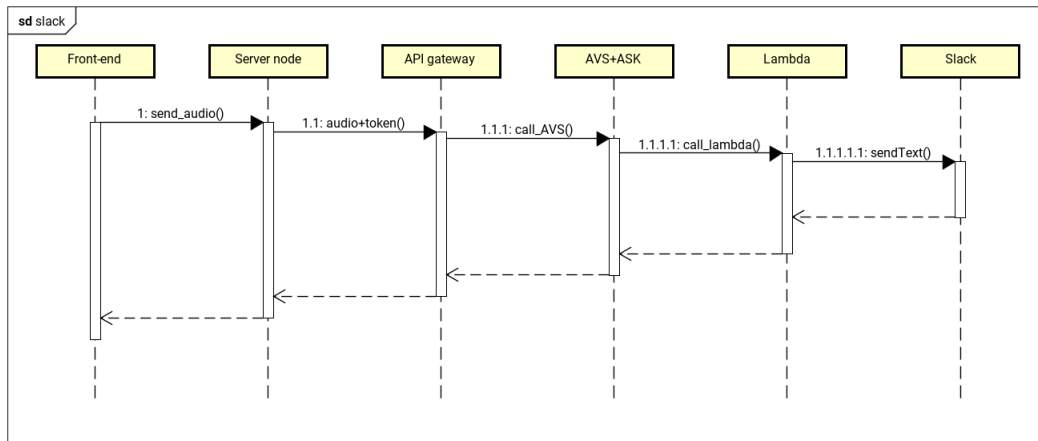
### 9.1 Sequenza generale



**Figura 21:** Diagramma generale

L'interazione base tra le entità consiste nell'invio dell'audio registrato dal front-end al server, quest'ultimo permette di aggiungere un token per l'utilizzo di *Alexa*. API gateway riceve i dati e passa le informazioni ad AVS che traduce l'audio in dati e riconosce la skill. Infine viene chiamata la funzione lambda corretta che alla sua terminazione avvisa AVS. AVS traduce la risposta in audio e la invia ad API gateway che ritorna l'audio al server che a sua volta lo invia al front-end.

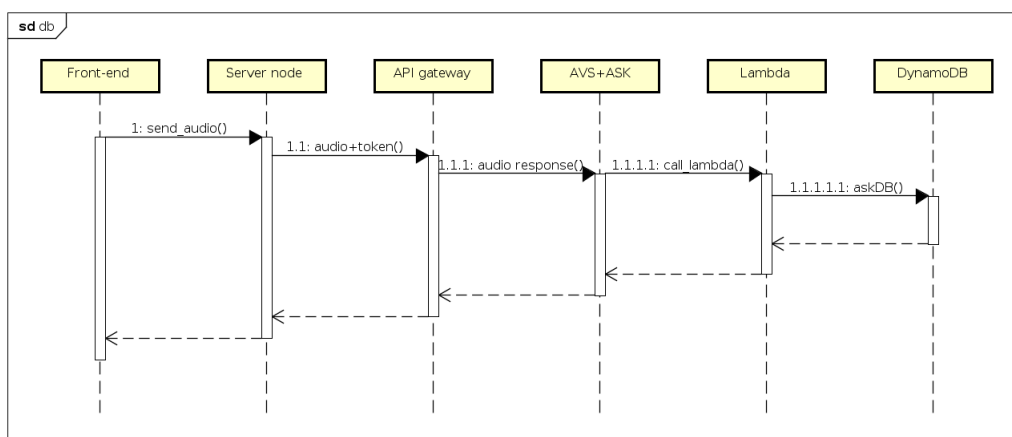
## 9.2 Sequenza con interazione con Slack



**Figura 22:** Diagramma con interazione con Slack

Nel caso in cui l'interazione preveda l'invio di un messaggio tramite *Slack*, allo schema generale si aggiunge l'invio del testo da parte di una lambda function al modulo *Slack*.

## 9.3 Sequenza con interazione con il database

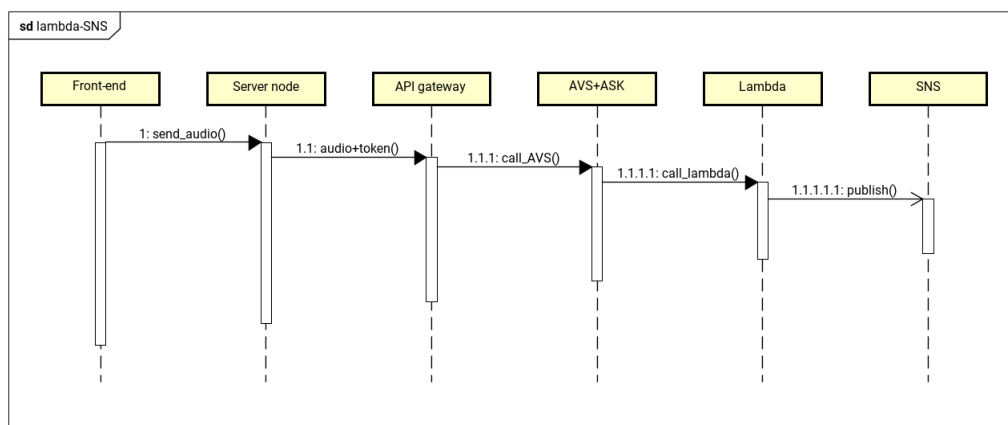


**Figura 23:** Diagramma con interazione con in database

Se l'interazione prevede una richiesta al database, allo schema generale si aggiunge una richiesta a *DynamoDB*.

## 9.4 Sequenza con SNS

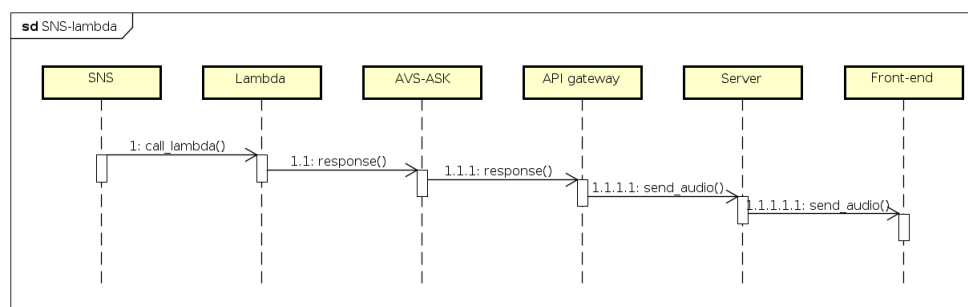
### 9.4.1 Lambda - SNS



**Figura 24:** Diagramma con SNS

Il servizio SNS permette a una funzione di invocare indirettamente un'altra funzione lambda senza dover aspettare una risposta, è infatti SNS che si occupa di invocare quest'ultima.

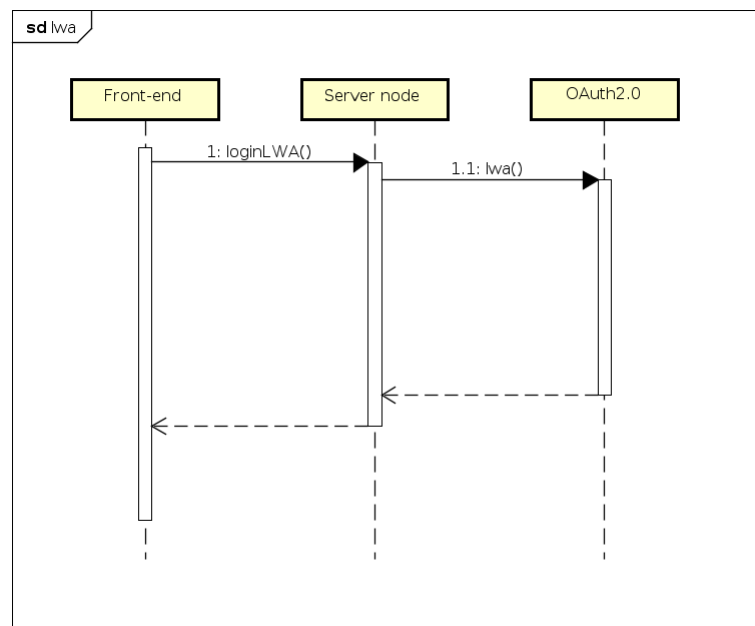
### 9.4.2 SNS - Lambda



**Figura 25:** Diagramma SNS-lambda

Quando SNS riceve una richiesta, la funzione lambda in ascolto si risveglia.

## 9.5 Login with Amazon

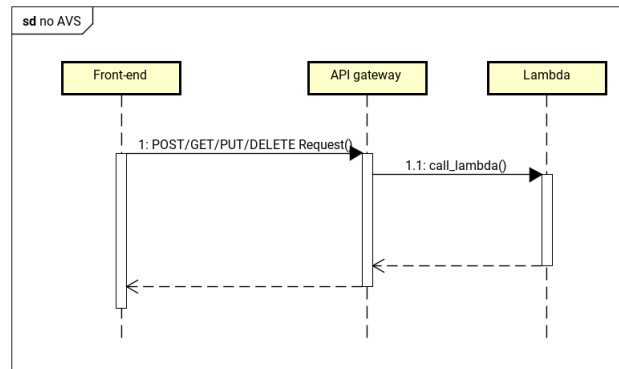


**Figura 26:** Diagramma lwa

Per effettuare il login tramite *Amazon* viene chiamata la funzione `loginLWA()` nel server, il server si occupa di interagire con *OAuth2.0* con il quale comunica dati sensibili necessari all'autenticazione.



## 9.6 Sequenza generale senza chiamate ad AVS

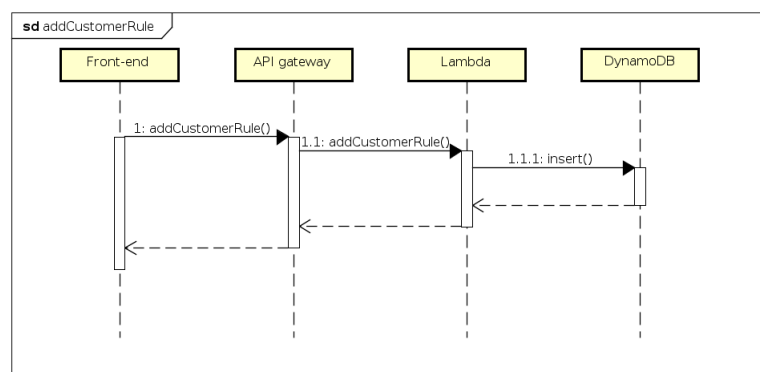


**Figura 27:** Diagramma generale senza chiamate ad AVS

Se per l'azione da svolgere non è necessario il passaggio per AVS, è possibile evitare la chiamata al server Node.JS comunicando direttamente con API Gateway il quale si occuperà di chiamare una lambda function. Anche in questo caso sono possibili vari scenari come quelli proposti sopra (messaggio su *Slack*, notifica su SNS, interazione col Database), ma la sequenza rimane immutata con l'unica differenza di non dover più passare per AVS e il server Node.

Riportiamo in seguito alcuni esempi specifici.

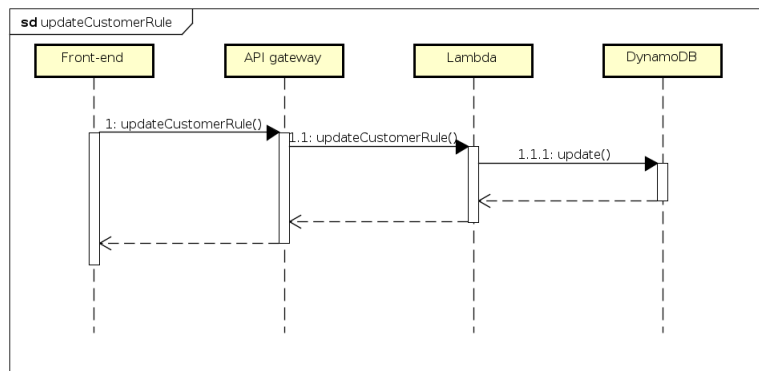
### 9.6.1 Sequenza addCustomerRule



**Figura 28:** Diagramma addCustomerRule

Quando l'amministratore vuole aggiungere una regola tramite la funzione `addCustomerRule()`, viene fatta una chiamata POST a API Gateway che chiama la funzione lambda corretta, la quale interagisce con il database.

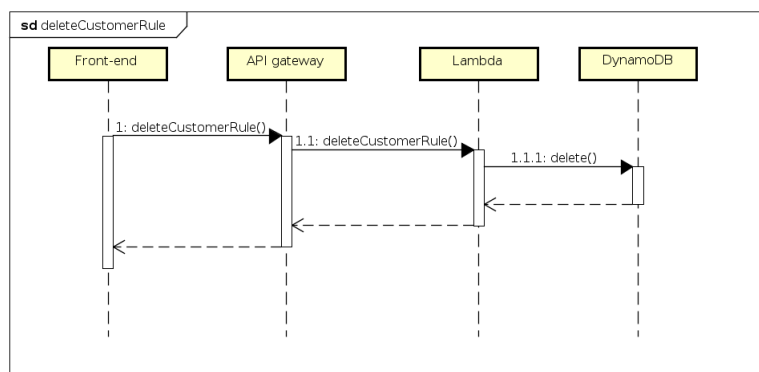
### 9.6.2 Sequenza `updateCustomerRule`



**Figura 29:** Diagramma `updateCustomerRule`

Quando l'amministratore vuole cambiare una regola, viene chiamata la funzione `updateCustomerRule()` che fa una chiamata PUT a API Gateway che a sua volta chiama la funzione lambda corretta, la quale interagisce con il database.

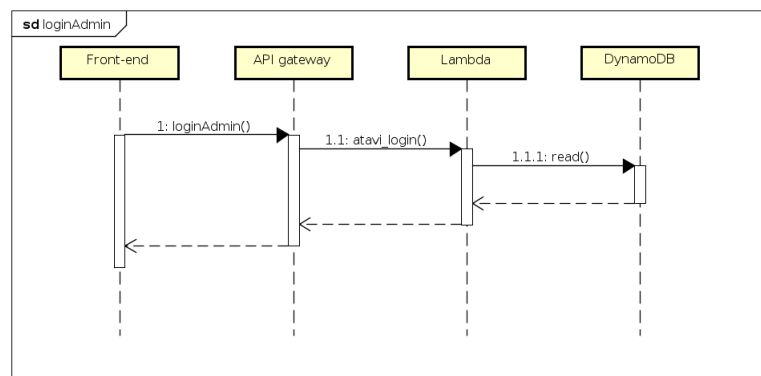
### 9.6.3 Sequenza `deleteCustomerRule`



**Figura 30:** Diagramma `deleteCustomerRule`

Quando l'amministratore vuole eliminare una regola, viene chiamata la funzione `deleteCustomerRule()` che fa una chiamata DELETE a API Gateway che a sua volta chiama la funzione lambda corretta, la quale interagisce con il database.

#### 9.6.4 Sequenza loginAdmin



**Figura 31:** Diagramma loginAdmin

Per effettuare il login al pannello amministratore l'utente inserisce la password e preme il tasto che chiama `loginAdmin()` che fa una richiesta POST a API Gateway che chiama la funzione lambda `atavi_login()` che a sua volta interagisce con il database.

## 10 Tracciamento: Funzioni-Requisiti

Funzione	Id Requisito
AtAVi::AWS-Back-End::Admin::auth::auth_login()	RFObb8 RFObb8.1 RFObb8.1.1 RFObb9
AtAVi::AWS-Back-End::Admin::auth::auth_changePsw()	RFObb9.3 RFObb9.3.1 RFObb9.3.2 RFObb9.3.3 RFObb9.3.4
AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_getCustomerRule()	RFObb9.1
AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_addCustomerRule()	RFObb9.1 RFObb9.1.1 RFObb9.1.1.1 RFObb9.1.1.1.1 RFObb9.1.1.1.1.1 RFObb9.1.1.1.1.2 RFObb9.1.1.1.1.3
AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_updateCustomerRule()	RFObb9.1 RFObb9.1.2 RFObb9.1.2.2 RFObb9.1.2.2.1
AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_deleteCustomerRule()	RFObb9.1.3
AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_getAllEmployeeRules()	RFObb9.1

Funzione	Id Requisito
AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_getEmployeeRule()	RFObb9.1
AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_addEmployeeRule()	RFObb9.1 RFObb9.1.1 RFObb9.1.1.2 RFObb9.1.1.2.1 RFObb9.1.1.2.1.1 RFObb9.1.1.2.1.2 RFObb9.1.1.3
AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_updateEmployeeRule()	RFObb9.1.2 RFObb9.1.2.1 RFObb9.1.2.1.1 RFObb9.1.2.1.2
AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_deleteEmployeeRule()	RFObb9.1.3
AtAVi::AWS-Back-End::Skill::atavi_core()	RFObb3 RFObb3.1 RFDes3.2 RFObb4 RFObb4.2 RFDes4.3 RFObb4.4
AtAVi::AWS-Back-End::Skill::Util::util_sendSlack()	RFObb4.1 RFObb6.1 RFObb6.2 RFObb6.3.3.2.1.1
AtAVi::AWS-Back-End::Skill::Util::util_addCustomer()	RFObb2 RFObb2.2 RFObb2.2.1

Funzione	Id Requisito
	RFObb2.2.2 RFObb2.2.3 RFObb2.2.4 RFObb5 RFObb5.1 RFObb5.2
AtAVi::AWS-Back-End::Skill::Util::util_updateCustomer()	RFObb5.3 RFObb5.4 RFObb5.4.1 RFObb5.4.2 RFObb10
AtAVi::AWS-Back-End::Skill::Util::util_searchCustomerRule() ( )	RFObb7 RFObb7.3
AtAVi::AWS-Back-End::Skill::Util::util_getCustomer()	RFObb2.1
AtAVi::AWS-Back-End::Skill::Util::util_getSlackChannel()	RFObb6.3.3.2.1.1
AtAVi::AWS-Back-End::Skill::Util::util_getLastEmployee()	RFObb7.3
AtAVi::Auth-Back-End::lwa::setTokens()	RFObb12 RFObb12.4
AtAVi::Auth-Back-End::lwa::deleteTokens()	RFObb12 RFObb12.4
AtAVi::Auth-Back-End::lwa::getToken()	RFObb12 RFObb12.4
AtAVi::Auth-Back-End::lwa::getRefreshToken()	RFObb12 RFObb12.4
AtAVi::Auth-Back-End::lwa::authRefresh()	RFObb12 RFObb12.4
AtAVi::Auth-Back-End::interaction::sendAudio()	RFObb12 RFObb12.4
AtAVi::Auth-Back-End::interaction::synchronizeState()	RFObb12

Funzione	Id Requisito
	RFObb12.4
AtAVi::Auth-Back-End::interaction::establishDownChannel( )	RFObb12 RFObb12.4
AtAVi::Front-End::Controllers::AdminLoginController::login Admin()	RFObb8.1 RFObb8.1.1
AtAVi::Front-End::Controllers::ConversationController::updateConversation()	RFObb1 RFObb5 RFObb5.4 RFObb6.3 RFObb6.3.3 RFObb7 RFObb10 RFObb10.1 RFObb10.2 RFDes10.2.1.1 RFObb11 RFObb11.1
AtAVi::Front-End::Controllers::AudioController::playAudio( )	RFObb1 RFObb5 RFObb5.4 RFObb6.3 RFObb6.3.3 RFObb7 RFObb10 RFObb10.1 RFObb10.2 RFDes10.2.1.1

Funzione	Id Requisito
	RFDes11 RFDes11.1
AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()	RFDes3.2  RFObb5.1 RFObb5.2 RFObb5.3 RFObb5.4.1 RFObb5.4.2 RFObb6 RFObb6.3.1 RFObb6.3.2 RFObb6.3.3.1 RFObb6.3.3.2 RFObb7.1 RFObb7.2 RFObb10.1.1 RFObb10.1.2 RFObb10.2.1 RFDes10.2.1.2 RFObb10.2.2
AtAVi::Front-End::Controllers::LWALoginController::loginL WA()	RFObb12  RFObb12.1 RFObb12.2 RFObb12.3
AtAVi::Front-End::Controllers::ControllerPersone-Impiegati: :addCustomerRule()	RFObb9.1  RFObb9.1.1 RFObb9.1.1.1 RFObb9.1.1.1.1



Funzione	Id Requisito
	RFObb9.1.1.1.1.1 RFObb9.1.1.1.1.2 RFObb9.1.1.1.1.3
AtAVi::Front-End::Controllers::ControllerPersone-Impiegati: :updateCustomerRule()	RFObb9.1 RFObb9.1.2 RFObb9.1.2.2 RFObb9.1.2.2.1
AtAVi::Front-End::Controllers::ControllerPersone-Impiegati: :getCustomerRules()	RFObb9.1
AtAVi::Front-End::Controllers::ControllerPersone-Impiegati: :deleteCustomerRule()	RFObb9.1.3
AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::a ddEmployeeRule()	RFObb9.1 RFObb9.1.1 RFObb9.1.1.2 RFObb9.1.1.2.1 RFObb9.1.1.2.1.1 RFObb9.1.1.2.1.2 RFObb9.1.1.3
AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::u pdateEmployeeRule()	RFObb9.1.2 RFObb9.1.2.1 RFObb9.1.2.1.1 RFObb9.1.2.1.2
AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::g etEmployeeRules()	RFObb9.1
AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::d eleteEmployeeRule()	RFObb9.1.3

**Tabella 1:** Tracciamento: Funzioni-Requisiti

## 11 Tracciamento: Requisiti-Funzioni

Id Requisito	Funzione
RFObb1	AtAVi::Front-End::Controllers::ConversationController::updateConversation() AtAVi::Front-End::Controllers::AudioController::playAudio() ( )
RFObb2	AtAVi::AWS-Back-End::Skill::Util::util_addCustomer()
RFObb2.1	AtAVi::AWS-Back-End::Skill::Util::util_getCustomer()
RFObb2.2	AtAVi::AWS-Back-End::Skill::Util::util_addCustomer()
RFObb2.2.1	AtAVi::AWS-Back-End::Skill::Util::util_addCustomer()
RFObb2.2.2	AtAVi::AWS-Back-End::Skill::Util::util_addCustomer()
RFObb2.2.3	AtAVi::AWS-Back-End::Skill::Util::util_addCustomer
RFObb2.2.4	AtAVi::AWS-Back-End::Skill::Util::util_addCustomer
RFObb3	AtAVi::AWS-Back-End::Skill::atavi_core()
RFObb3.1	AtAVi::AWS-Back-End::Skill::atavi_core()
RFDdes3.2	AtAVi::AWS-Back-End::Skill::atavi_core() AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()
RFObb4	AtAVi::AWS-Back-End::Skill::atavi_core()
RFObb4.1	AtAVi::AWS-Back-End::Skill::Util::util_sendSlack()
RFObb4.2	AtAVi::AWS-Back-End::Skill::atavi_core()
RFDdes4.3	AtAVi::AWS-Back-End::Skill::atavi_core()
RFObb4.4	AtAVi::AWS-Back-End::Skill::atavi_core()
RFObb5	AtAVi::AWS-Back-End::Skill::Util::util_addCustomer() AtAVi::Front-End::Controllers::ConversationController::updateConversation() AtAVi::Front-End::Controllers::AudioController::playAudio() ( )
RFObb5.1	AtAVi::AWS-Back-End::Skill::Util::util_addCustomer() AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()

Id Requisito	Funzione
RFObb5.2	AtAVi::AWS-Back-End::Skill::Util::util_addCustomer() AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()
RFObb5.3	AtAVi::AWS-Back-End::Skill::Util::util_addCustomer() AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()
RFObb5.4	AtAVi::AWS-Back-End::Skill::Util::util_addCustomer() AtAVi::Front-End::Controllers::ConversationController::upd ateConversation() AtAVi::Front-End::Controllers::AudioController::playAudio( )
RFObb5.4.1	AtAVi::AWS-Back-End::Skill::Util::util_addCustomer() AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()
RFObb5.4.2	AtAVi::AWS-Back-End::Skill::Util::util_addCustomer() AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()
RFObb6	AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()
RFObb6.1	AtAVi::AWS-Back-End::Skill::Util::util_sendSlack()
RFObb6.2	AtAVi::AWS-Back-End::Skill::Util::util_sendSlack()
RFObb6.3	AtAVi::Front-End::Controllers::AudioController::playAudio( )
RFObb6.3.1	AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()
RFObb6.3.2	AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()
RFObb6.3.3	AtAVi::Front-End::Controllers::ConversationController::upd ateConversation() AtAVi::Front-End::Controllers::AudioController::playAudio( )
RFObb6.3.3.1	AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()

<b>Id Requisito</b>	<b>Funzione</b>
RFObb6.3.3.2	AtAVi::Front-End::Controllers::AudioController::sendAudioAlexa()
RFObb6.3.3.2.1	AtAVi::AWS-Back-End::Skill::Util::util_getSlackChannel()
RFObb6.3.3.2.1.1	AtAVi::AWS-Back-End::Skill::Util::util_getSlackChannel()
RFObb7	AtAVi::AWS-Back-End::Skill::Intent::util_searchCustomerRule() AtAVi::Front-End::Controllers::ConversationController::updateConversation() AtAVi::Front-End::Controllers::AudioController::playAudio()
RFObb7.3	AtAVi::AWS-Back-End::Skill::Util::util_searchCustomerRule()
RFObb7.1	AtAVi::Front-End::Controllers::AudioController::sendAudioAlexa()
RFObb7.2	AtAVi::Front-End::Controllers::AudioController::sendAudioAlexa()
RFObb8	AtAVi::AWS-Back-End::Admin::auth::atavi_login()
RFObb8.1	AtAVi::AWS-Back-End::Admin::auth::atavi_login() AtAVi::Front-End::Controllers::AdminLoginController::loginAdmin()
RFObb8.1.1	AtAVi::AWS-Back-End::Admin::auth::atavi_login() AtAVi::Front-End::Controllers::AdminLoginController::loginAdmin()
RFObb9	AtAVi::AWS-Back-End::Admin::auth::atavi_login() AtAVi::AWS-Back-End::Admin::auth::atavi_logout()
RFObb9.1	AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_getCustomerRule() AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_addCustomerRule() AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_updateCustomerRule() AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_getAllEmployeeRules()

Id Requisito	Funzione
	AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_getEmployeeRule() AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_addEmployeeRule() AtAVi::Front-End::Controllers::ControllerPersone-Impiegati::addCustomerRule() AtAVi::Front-End::Controllers::ControllerPersone-Impiegati::updateCustomerRule() AtAVi::Front-End::Controllers::ControllerPersone-Impiegati::getCustomerRules() AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::addEmployeeRule() AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::getEmployeeRules()
RFObb9.1.1	AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_addCustomerRule() AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_addEmployeeRule() AtAVi::Front-End::Controllers::ControllerPersone-Impiegati::addCustomerRule() AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::addEmployeeRule()
RFObb9.1.1.1	AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_addCustomerRule() AtAVi::Front-End::Controllers::ControllerPersone-Impiegati::addCustomerRule()
RFObb9.1.1.1.1	AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_addCustomerRule() AtAVi::Front-End::Controllers::ControllerPersone-Impiegati::addCustomerRule()
RFObb9.1.1.1.1.1	AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_addCustomerRule() AtAVi::Front-End::Controllers::ControllerPersone-Impiegati::addCustomerRule()

<b>Id Requisito</b>	<b>Funzione</b>
RFObb9.1.1.1.1.2	AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_addCustomerRule() AtAVi::Front-End::Controllers::ControllerPersone-Impiegati::addCustomerRule()
RFObb9.1.1.1.1.3	AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_addCustomerRule() AtAVi::Front-End::Controllers::ControllerPersone-Impiegati::addCustomerRule()
RFObb9.1.1.2	AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_addEmployeeRule() AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::addEmployeeRule()
RFObb9.1.1.2.1	AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_addEmployeeRule() AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::addEmployeeRule()
RFObb9.1.1.2.1.1	AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_addEmployeeRule() AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::addEmployeeRule()
RFObb9.1.1.2.1.2	AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_addEmployeeRule() AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::addEmployeeRule()
RFObb9.1.1.3	AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_addEmployeeRule() AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::addEmployeeRule()
RFObb9.1.2	AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_updateCustomerRule() AtAVi::AWS-Back-End::Admin::Rules::Customer::updateEmployeeRule() AtAVi::Front-End::Controllers::ControllerPersone-Impiegati::updateCustomerRule()

Id Requisito	Funzione
	AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::updateEmployeeRule()
RFObb9.1.2.1	AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_updateEmployeeRule() AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::updateEmployeeRule()
RFObb9.1.2.1.1	AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_updateEmployeeRule() AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::updateEmployeeRule()
RFObb9.1.2.1.2	AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_updateEmployeeRule() AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::updateEmployeeRule()
RFObb9.1.2.2	AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_updateCustomerRule() AtAVi::Front-End::Controllers::ControllerPersone-Impiegati::updateCustomerRule()
RFObb9.1.2.2.1	AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_updateCustomerRule() AtAVi::Front-End::Controllers::ControllerPersone-Impiegati::updateCustomerRule()
RFObb9.1.3	AtAVi::AWS-Back-End::Admin::Rules::Customer::rules_deleteCustomerRule() AtAVi::AWS-Back-End::Admin::Rules::Employee::rules_deleteEmployeeRule() AtAVi::Front-End::Controllers::ControllerPersone-Impiegati::deleteCustomerRule() AtAVi::Front-End::Controllers::ControllerImpiegati-Slack::deleteEmployeeRule()
RFObb9.2	AtAVi::AWS-Back-End::Admin::auth::atavi_logout()
RFObb9.2.1	AtAVi::AWS-Back-End::Admin::auth::atavi_logout()
RFObb9.3	AtAVi::AWS-Back-End::Admin::auth::atavi_changePsw()

<b>Id Requisito</b>	<b>Funzione</b>
RFObb9.3.1	AtAVi::AWS-Back-End::Admin::auth::atavi_changePsw()
RFObb9.3.2	AtAVi::AWS-Back-End::Admin::auth::atavi_changePsw()
RFObb9.3.3	AtAVi::AWS-Back-End::Admin::auth::atavi_changePsw()
RFObb9.3.4	AtAVi::AWS-Back-End::Admin::auth::atavi_changePsw()
RFObb10	AtAVi::AWS-Back-End::Skill::Util::util_addCustomer() AtAVi::Front-End::Controllers::ConversationController::updateConversation() AtAVi::Front-End::Controllers::AudioController::playAudio() ( )
RFObb10.1	AtAVi::Front-End::Controllers::ConversationController::updateConversation() AtAVi::Front-End::Controllers::AudioController::playAudio() ( )
RFObb10.1.1	AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()
RFObb10.1.2	AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()
RFObb10.2	AtAVi::Front-End::Controllers::ConversationController::updateConversation() AtAVi::Front-End::Controllers::AudioController::playAudio() ( )
RFObb10.2.1	AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()
RFDdes10.2.1.1	AtAVi::Front-End::Controllers::ConversationController::updateConversation() AtAVi::Front-End::Controllers::AudioController::playAudio() ( )
RFDdes10.2.1.2	AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()
RFObb10.2.2	AtAVi::Front-End::Controllers::AudioController::sendAudio Alexa()
RFDdes11	AtAVi::Front-End::Controllers::ConversationController::updateConversation()



Id Requisito	Funzione
	AtAVi::Front-End::Controllers::AudioController::playAudio() ( )
RFDdes11.1	AtAVi::Front-End::Controllers::ConversationController::updateConversation() AtAVi::Front-End::Controllers::AudioController::playAudio() ( )
RFObb12	AtAVi::Auth-Back-End::lwa::setTokens() AtAVi::Auth-Back-End::lwa::deleteTokens() AtAVi::Auth-Back-End::lwa::getToken() AtAVi::Auth-Back-End::lwa::getRefreshToken() AtAVi::Auth-Back-End::lwa::authRefresh() AtAVi::Auth-Back-End::interaction::sendAudio() AtAVi::Auth-Back-End::interaction::synchronizeState() AtAVi::Auth-Back-End::interaction::establishDownChannel() ( ) AtAVi::Front-End::Controllers::LWALoginController::loginLWA() WA()
RFObb12.1	AtAVi::Front-End::Controllers::LWALoginController::loginLWA() WA()
RFObb12.2	AtAVi::Front-End::Controllers::LWALoginController::loginLWA() WA()
RFObb12.3	AtAVi::Front-End::Controllers::LWALoginController::loginLWA() WA()
RFObb12.4	AtAVi::Auth-Back-End::lwa::setTokens() AtAVi::Auth-Back-End::lwa::deleteTokens() AtAVi::Auth-Back-End::lwa::getToken() AtAVi::Auth-Back-End::lwa::getRefreshToken() AtAVi::Auth-Back-End::lwa::authRefresh() AtAVi::Auth-Back-End::interaction::sendAudio() AtAVi::Auth-Back-End::interaction::synchronizeState() AtAVi::Auth-Back-End::interaction::establishDownChannel() ( )

Id Requisito	Funzione
	AtAVi::Front-End::Controllers::LWALoginController::loginLWA()

**Tabella 2:** Tracciamento: Requisiti-Funzioni

## 12 Tracciamento: Componenti-Requisiti

Componente	Id Requisito
AtAVi::AWS-Back-End::Admin	RFObb8
	RFObb8.1
	RFObb8.1.1
	RFObb9
	RFObb9.1
	RFObb9.1.1
	RFObb9.1.1.1
	RFObb9.1.1.1.1
	RFObb9.1.1.1.1.1
	RFObb9.1.1.1.1.2
	RFObb9.1.1.1.1.3
	RFObb9.1.1.2
	RFObb9.1.1.2.1
	RFObb9.1.1.2.1.1
	RFObb9.1.1.2.1.2
	RFObb9.1.1.3
	RFObb9.1.2
	RFObb9.1.2.1
	RFObb9.1.2.1.1
	RFObb9.1.2.1.2
	RFObb9.1.2.2
	RFObb9.1.2.2.1
	RFObb9.1.3
	RFObb9.2
	RFObb9.2.1
	RFObb9.3
	RFObb9.3.1
	RFObb9.3.2

Componente	Id Requisito
	RFObb9.3.3 RFObb9.3.4
AtAVi::AWS-Back-End::Skill	RFObb2 RFObb2.1 RFObb2.2 RFObb2.2.1 RFObb2.2.2 RFObb2.2.3 RFObb2.2.4 RFObb2.2.5 RFDes2.2.6 RFObb3 RFObb3.1 RFDes3.2 RFObb4 RFObb4.1 RFObb4.2 RFDes4.3 RFObb4.4 RFObb5 RFObb5.1 RFObb5.2 RFObb5.3 RFObb5.4 RFObb5.4.1 RFObb5.4.2 RFObb6 RFObb6.1 RFObb6.2 RFObb6.3

Componente	Id Requisito
	RFObb6.3.1 RFObb6.3.2 RFObb6.3.3 RFObb6.3.3.1 RFObb6.3.3.2 RFObb6.3.3.2.1 RFObb6.3.3.2.1.1 RFDdes6.3.3.2.2 RFObb7 RFObb7.1 RFObb7.2 RFObb7.3 RFObb10 RFObb10.1 RFObb10.1.1 RFDdes10.1.1.1 RFObb10.1.2 RFObb10.2 RFObb10.2.1 RFDdes10.2.1.1 RFDdes10.2.1.2 RFDdes10.2.1.3 RFObb10.2.2
AtAVi::Auth-Back-End	RFObb12 RFObb12.1 RFObb12.2 RFObb12.3 RFObb12.4
AtAVi::Front-End::Controllers	RFObb1 RFDdes3.2

Componente	Id Requisito
	RFObb5
	RFObb5.1
	RFObb5.2
	RFObb5.3
	RFObb5.4
	RFObb5.4.1
	RFObb5.4.2
	RFObb6
	RFObb6.3
	RFObb6.3.1
	RFObb6.3.2
	RFObb6.3.3
	RFObb6.3.3.1
	RFObb6.3.3.2
	RFObb7
	RFObb7.1
	RFObb7.2
	RFObb8.1
	RFObb8.1.1
	RFObb9.1
	RFObb9.1.1
	RFObb9.1.1.1
	RFObb9.1.1.1.1
	RFObb9.1.1.1.1.1
	RFObb9.1.1.1.1.2
	RFObb9.1.1.1.1.3
	RFObb9.1.1.2
	RFObb9.1.1.2.1
	RFObb9.1.1.2.1.1
	RFObb9.1.1.2.1.2

Componente	Id Requisito
	RFObb9.1.1.3
	RFObb9.1.2
	RFObb9.1.2.1
	RFObb9.1.2.1.1
	RFObb9.1.2.1.2
	RFObb9.1.2.2
	RFObb9.1.2.2.1
	RFObb9.1.3
	RFObb10
	RFObb10.1
	RFObb10.1.1
	RFObb10.1.2
	RFObb10.2
	RFObb10.2.1
	RFDes10.2.1.1
	RFDes10.2.1.2
	RFObb10.2.2
	RFDes11
	RFDes11.1
	RFObb12
	RFObb12.1
	RFObb12.2
	RFObb12.3

**Tabella 3:** Tracciamento: Componenti-Requisiti

## 13 Tracciamento: Requisiti-Componenti

Id Requisito	Componente
RFObb1	AtAVi::Front-End::Controllers
RFObb2	AtAVi::AWS-Back-End::Skill
RFObb2.1	AtAVi::AWS-Back-End::Skill
RFObb2.2	AtAVi::AWS-Back-End::Skill
RFObb2.2.1	AtAVi::AWS-Back-End::Skill
RFObb2.2.2	AtAVi::AWS-Back-End::Skill
RFObb2.2.3	AtAVi::AWS-Back-End::Skill
RFObb2.2.4	AtAVi::AWS-Back-End::Skill
RFObb2.2.5	AtAVi::AWS-Back-End::Skill
RFDes2.2.6	AtAVi::AWS-Back-End::Skill
RFObb3	AtAVi::AWS-Back-End::Skill
RFObb3.1	AtAVi::AWS-Back-End::Skill
RFDes3.2	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb4	AtAVi::AWS-Back-End::Skill
RFObb4.1	AtAVi::AWS-Back-End::Skill AtAVi::AWS-Back-End::Skill
RFObb4.2	AtAVi::AWS-Back-End::Skill
RFDes4.3	AtAVi::AWS-Back-End::Skill
RFObb4.4	AtAVi::AWS-Back-End::Skill
RFObb5	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb5.1	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb5.2	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb5.3	AtAVi::AWS-Back-End::Skill



<b>Id Requisito</b>	<b>Componente</b>
	AtAVi::Front-End::Controllers
RFObb5.4	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb5.4.1	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb5.4.2	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb6	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb6.1	AtAVi::AWS-Back-End::Skill
RFObb6.2	AtAVi::AWS-Back-End::Skill
RFObb6.3	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb6.3.1	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb6.3.2	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb6.3.3	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb6.3.3.1	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb6.3.3.2	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb6.3.3.2.1	AtAVi::AWS-Back-End::Skill
RFObb6.3.3.2.1.1	AtAVi::AWS-Back-End::Skill
RFObb6.3.3.2.2	AtAVi::AWS-Back-End::Skill
RFObb7	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb7.3	AtAVi::AWS-Back-End::Skill

<b>Id Requisito</b>	<b>Componente</b>
RFobb7.1	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFobb7.2	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFobb8	AtAVi::AWS-Back-End::Admin
RFobb8.1	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFobb8.1.1	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFobb9	AtAVi::AWS-Back-End::Admin
RFobb9.1	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFobb9.1.1	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFobb9.1.1.1	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFobb9.1.1.1.1	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFobb9.1.1.1.1.1	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFobb9.1.1.1.1.2	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFobb9.1.1.1.1.3	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFobb9.1.1.2	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFobb9.1.1.2.1	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFobb9.1.1.2.1.1	AtAVi::AWS-Back-End::Admin

<b>Id Requisito</b>	<b>Componente</b>
	AtAVi::Front-End::Controllers
RFObb9.1.1.2.1.2	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFObb9.1.1.3	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFObb9.1.2	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFObb9.1.2.1	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFObb9.1.2.1.1	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFObb9.1.2.1.2	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFObb9.1.2.2	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFObb9.1.2.2.1	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFObb9.1.3	AtAVi::AWS-Back-End::Admin AtAVi::Front-End::Controllers
RFObb9.2	AtAVi::AWS-Back-End::Admin
RFObb9.2.1	AtAVi::AWS-Back-End::Admin
RFObb9.3	AtAVi::AWS-Back-End::Admin
RFObb9.3.1	AtAVi::AWS-Back-End::Admin
RFObb9.3.2	AtAVi::AWS-Back-End::Admin
RFObb9.3.3	AtAVi::AWS-Back-End::Admin
RFObb9.3.4	AtAVi::AWS-Back-End::Admin
RFObb10	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb10.1	AtAVi::AWS-Back-End::Skill

Id Requisito	Componente
	AtAVi::Front-End::Controllers
RFObb10.1.1	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFDes10.1.1.1	AtAVi::AWS-Back-End::Skill
RFObb10.1.2	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb10.2	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb10.2.1	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFDes10.2.1.3	AtAVi::AWS-Back-End::Skill
RFDes10.2.1.1	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFDes10.2.1.2	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFObb10.2.2	AtAVi::AWS-Back-End::Skill AtAVi::Front-End::Controllers
RFDes11	AtAVi::Front-End::Controllers
RFDes11.1	AtAVi::Front-End::Controllers
RFObb12	AtAVi::Auth-Back-End AtAVi::Front-End::Controllers
RFObb12.1	AtAVi::Auth-Back-End AtAVi::Front-End::Controllers
RFObb12.2	AtAVi::Auth-Back-End AtAVi::Front-End::Controllers
RFObb12.3	AtAVi::Auth-Back-End AtAVi::Front-End::Controllers
RFObb12.4	AtAVi::Auth-Back-End

**Tabella 4:** Tracciamento: Requisiti-Componenti

## A Design Pattern

Per la progettazione di *AtAVi* è stata richiesta un'architettura perlopiù serverless, ciò porta non pochi problemi nell'applicazione di design pattern standard in quanto il prodotto è basato principalmente sull'utilizzo di funzioni lambda e la mancanza di classi ed oggetti. Nonostante ciò, in alcune componenti del sistema, il gruppo è riuscito ad individuare i seguenti pattern: *MVW* e *Façade*.

### A.1 Model View Whatever

Il vantaggio principale che si riscontra nell'adottare questo pattern è la possibilità di effettuare cambiamenti alle singole componenti senza dover intervenire sulle altre, se si cambiano i dati non è necessario cambiare anche la loro rappresentazione.

All'interno di *AtAVi* questo pattern architetturale è adottato per l'utilizzo di *AngularJS* nella parte di Front-end dell'applicazione. Model-View-Whatever è un pattern architetturale che propone la suddivisione di un'interfaccia in tre componenti principali:

- **Model:** si occupa della gestione dei dati, nel nostro caso i dati sono salvati nel database Dynamo presente nei servizi Amazon. I dati sono acceduti mediante una chiamata alle Api rest effettuata dal controller di Angular;
- **View:** si occupa della presentazione dei dati. Per la nostra applicazione la view è costituita da delle pagine HTML unite alle direttive di Angular che consentono di mantenerla aggiornata;
- **Controller-modelview:** gestisce l'interazione tra il Model e la View. Il controller della nostra applicazione si occupa di effettuare le chiamate alle API rest di Amazon che prelevano i dati dal model (Database di Dynamo) e di aggiornare l'oggetto \$scope. Viene così aggiornata anche la View riflettendo le modifiche avvenute all'interno del controller.

### A.2 Façade

Façade è un design pattern che mira alla semplificazione dell'interfaccia offerta a un utente, nascondendo la complessità che può esserci all'interno di un sistema. Permette inoltre di separare l'implementazione degli elementi che fanno delle richieste ad un'interfaccia da quelli che da essa sono richiamati.

Nel nostro capitolato abbiamo adottato questo design pattern nell'utilizzo di API-Gateway il quale si pone fra il Front-End e le varie lambda functions.