



COMUNE DI GENOVA
DIREZIONE SISTEMI INFORMATIVI
SETTORE PROGRAMMAZIONE E REALIZZAZIONE

**ACQUISIZIONE DI SERVIZI DI MANUTENZIONE EVOLUTIVA,
MANUTENZIONE ORDINARIA E ASSISTENZA ALLA PIATTAFORMA
DEI PAGAMENTI (MIP - MODULO INCASSI E PAGAMENTI E PUNTO
UNICO DI EMISSIONE) DEL COMUNE DI GENOVA**

ALLEGATO 8 – REQUISITI DI INTEROPERABILITÀ

INDICE

1	REQUISITI DI INTEROPERABILITÀ.....	4
1.1	Introduzione.....	4
1.2	Convenzioni utilizzate	5
2	ELENCO DEI REQUISITI	6
2.1	Requisito #1: Disponibilità di API.....	6
2.2	Requisito #2: Progettazione delle API REST.....	6
2.3	Requisito #3: Documentazione delle API REST.....	6
2.4	Requisito #4: Requisiti obbligatori delle API REST	6
2.4.1	Requisito #4.1: Formato dei dati scambiati	7
2.4.2	Requisito #4.2: Definizione e naming delle interfacce	8
2.4.3	Requisito #4.3: Performance e Robustezza	9
2.5	Requisito #5: Requisiti di Privacy e di Sicurezza	10
2.5.1	Requisito #5.1: Privacy by Design e Privacy by Default	10
2.5.2	Requisito #5.2: Comunicazione dati personali solo su richieste esplicite.....	10
2.5.3	Requisito #5.3: Veicolazione dei dati tramite protocollo HTTPS.....	10
2.5.4	Requisito #5.4: Accesso circoscritto ai soli dati necessari	11
3	RISORSE.....	13
3.1	Conformance al Modello Informativo di Riferimento del Comune di Genova (R.I.M.) 13	
3.2	Elenco delle risorse	16
3.2.1	Proprietà della risorsa AvvisoPagamento.....	16
3.2.2	Proprietà della risorsa Transazione	17
3.2.3	Proprietà della risorsa RicevutaPagamento	18
3.2.4	Proprietà della risorsa Debito	19
3.2.5	Proprietà della risorsa PosizioneDebitoria	20
3.3	Elenco dei datatype complessi	21
4	PATH DELLE INTERFACCE	22
4.1	Caratteristiche generali.....	22
4.2	Interfaccia ListaTransazioni.....	23
4.3	Interfaccia DettaglioTransazione	23
4.4	Interfaccia ListaAvvisiPersona.....	24
4.5	Interfaccia ListaAvvisiTransazione	24
4.6	Interfaccia DettaglioAvviso	25

4.7	Interfaccia ListaRicevutePersona	26
4.8	Interfaccia ListaRicevuteAvviso	26
4.9	Interfaccia ListaRicevuteTransazionePagoPA	27
4.10	Interfaccia DettaglioRicevuta	27
4.11	Interfaccia ListaDebitiPersona	28
4.12	Interfaccia ListaDebitiPersonaServizio	28
4.13	Interfaccia DettaglioDebito	29
4.14	Interfaccia PosizioneDebitoriaPersona	30
4.15	Interfacce in input	30
5	ALLEGATI	31
5.1	Datatypes	31
5.1.1	SubjectIdentifier	31
5.1.2	CodeableService	33
5.2	Esempi	35
5.2.1	avvisoPagamentoExample.json	35
5.2.2	transazioneExample.json	36

1 REQUISITI DI INTEROPERABILITÀ

1.1 Introduzione

In conformità con le linee guida AGID per l'Informatica nella Pubblica Amministrazione il Comune di Genova considera la tematica dell'interoperabilità un elemento fondamentale della propria architettura di erogazione dei servizi. Il principale punto di riferimento è il Modello di Interoperabilità 2018, in breve ModI 2018, redatto da AGID e disponibile all'indirizzo

<https://docs.italia.it/italia/piano-triennale-ict/lg-modellointeroperabilita-docs/it/v2018.1/>

Ogni nuovo componente applicativo che viene introdotto nel S.I. comunale dovrà quindi produrre ed interagire con un elevato numero di microservizi realizzati da terzi.

Per la gestione dei microservizi il Comune di Genova si è inoltre dotato di una soluzione completa di API Management, con l'obiettivo di gestire l'intero ciclo di vita dei servizi erogati verso tutti i diversi attori del territorio e che:

1. Massimizza la consapevolezza generale dei servizi disponibili nel territorio favorendone la diffusione e il riuso
2. Fornisce gli strumenti per gestire l'intero ciclo di vita delle API sotto il completo controllo dell'organizzazione territoriale
3. Semplifica il compito dei progettisti nel dotare le API delle necessarie caratteristiche, funzionali e non, per rispettare i requisiti normativi vigenti (sicurezza, logging, QoS, ...)

I servizi che compongono l'ecosistema del Comune di Genova comprendono tutti i tipi di interazioni:

- Comune Genova <--> altre PA (A2A)
- Comune Genova <--> Imprese (A2B)
- Comune Genova <--> Cittadini (A2C)

In un contesto così variegato la scelta del Comune di Genova è quella di adottare l'approccio API First in quanto garantisce la massima circolazione e riuso delle interfacce dei servizi, che saranno quindi adottate dalle diverse applicazioni per un accesso regolamentato alle banche dati del territorio.

1.2 Convenzioni utilizzate

Nella definizione dei requisiti, delle specifiche e delle regole descritte sono utilizzate le parole chiave **DEVE**, **NON DEVE**, **OBBLIGATORIO**, **VIETATO**, **DOVREBBE**, **CONSIGLIATO**, **NON DOVREBBE**, **SCONSIGLIATO**, **PUÒ**, **OPZIONALE** che devono essere interpretate in particolare:

- **DEVE**, **OBBLIGATORIO**, **NECESSARIO** (**MUST**, **REQUIRED**, **SHALL**) significano che la definizione è un requisito assoluto, il requisito deve essere implementato, la consegna è inderogabile.
- **NON DEVE**, **VIETATO** (**MUST NOT**, **SHALL NOT**) significano che c'è proibizione assoluta di implementazione di un determinato elemento.
- **DOVREBBE**, **CONSIGLIATO** (**SHOULD**, **RECOMMENDED**) significa che in particolari circostanze possono esistere validi motivi per ignorare un requisito, non implementare una specifica, derogare alla consegna, ma che occorre esaminare e valutare con attenzione le implicazioni correlate alla scelta.
- **NON DOVREBBE**, **SCONSIGLIATO** (**SHOULD NOT**, **NOT RECOMMENDED**) significano che in particolari circostanze possono esistere validi motivi per cui un elemento è accettabile o persino utile, ma, prima di implementarlo, le implicazioni correlate dovrebbero essere esaminate e valutate con attenzione.
- **PUÒ**, **OPZIONALE** (**MAY**, **OPTIONAL**) significano che un elemento è a implementazione facoltativa.

Le parole chiave nel testo sono segnalate in maiuscolo e neretto (es. "**DEVE**").

2 ELENCO DEI REQUISITI

2.1 Requisito #1: Disponibilità di API

Tutte le nuove implementazioni software e tutti gli interventi evolutivi di software esistenti introdotti nel S.I. comunale **devono** includere interfacce applicative (API) delle risorse gestite dal sistema.

Le interfacce applicative **devono** essere realizzate secondo il paradigma REST che propone una visione del Web incentrata sul concetto di risorsa.

Le interfacce applicative **possono** essere progettate secondo il paradigma SOAP solo nel contesto di servizi preesistenti per i quali sia necessario estendere un set di API SOAP già in uso. In tal caso, comunque, le nuove API SOAP **dovranno** essere conformi alle specifiche del Modello di Interoperabilità 2018 di AGID, che prevede l'aderenza al Basic Profile 2.0.

2.2 Requisito #2: Progettazione delle API REST

Per la specifica di tali interfacce, al fine di descrivere le informazioni sulle operazioni disponibili in una API e su come devono essere strutturati i dati della richiesta e della risposta, **deve** essere adottato lo standard OpenAPI 3 (<https://www.openapis.org>).

2.3 Requisito #3: Documentazione delle API REST

Le Web API **devono** essere corredate dalla documentazione che fornisca i seguenti dettagli tecnici:

- Il modello e il formato dei dati
- Le operazioni fornite dalla API
- Eventuali caratteristiche aggiuntive quali: sicurezza, qualità del servizio erogato, ecc.

Il requisito sulla documentazione delle Web API **deve** essere soddisfatto allegando una descrizione dell'API formulata mediante il linguaggio di descrizione OpenAPI 3.

2.4 Requisito #4: Requisiti obbligatori delle API REST

Le API **devono** soddisfare i successivi requisiti di dettaglio suddivisi nelle seguenti tematiche:

- Formato dei dati scambiati
 - Definizione e naming delle interfacce
 - Performance e Robustezza
 - Schemi di sicurezza per l'accesso
-

2.4.1 Requisito #4.1: Formato dei dati scambiati

Formato JSON per i dati:

La descrizione di dati strutturati **deve** essere nel formato JSON (<http://www.rfc-editor.org/rfc/rfc7159.txt>), adottando il media type "application/json".

Naming Proprietà Oggetti JSON

Le proprietà degli oggetti JSON **devono** essere denominate utilizzando uno solo a scelta tra i metodi "snake_case" e "camelCase".

Standard per la nomenclatura

Per la denominazione delle proprietà, laddove possibile, si **dovrebbe** seguire la nomenclatura stabilita nelle "Linee Guida per la valorizzazione del Patrimonio informativo nazionale" contenente ontologie e vocabolari controllati (<https://github.com/italia/daf-ontologie-vocabolari-controllati>). Esempio di utilizzo di tali standard sono: lingue, nazioni, monete, ecc.

Boolean not null

Le proprietà booleane **non devono** ammettere il valore NULL.

Formati Data e Ora

Devono essere utilizzati i seguenti formati standard per Data ed Ora (RFC-3339 o UTC) e per gli http headers (RFC-7231).

Tipi con attributo "Format"

Si **deve** definire l'attributo "format" quando si usano i tipi Number ed Integer.

2.4.2 Requisito #4.2: Definizione e naming delle interfacce

REST Maturity Level 2

La definizione delle interfacce delle API **deve** seguire le regole del Richardson Maturity Model livello 2, che include:

- Evitare le azioni e ragionare intorno alle risorse
- Evitare i verbi nelle URL
- Usare correttamente i metodi HTTP
- Usare gli status HTTP appropriati

Uso corretto dei metodi HTTP

L'interfaccia di servizio REST **deve** utilizzare l'HTTP verb più adatto all'operazione come indicato in RFC 7231 (<https://tools.ietf.org/html/rfc7231#section-4.3>).

Uso corretto degli header HTTP

Gli header **non devono** essere utilizzati per veicolare informazioni applicative ma solo per informazioni di contesto oppure per supportare funzionalità di protocollo (autenticazione, autorizzazione, ecc). Prima di usare un header si **deve** verificare se è già disponibile nel censimento IANA (<https://www.iana.org/assignments/message-headers/message-headers.xhtml>) ed eventualmente adeguarsi.

URL Path

Si **dovrebbe** utilizzare parole separate da trattino "-" per i Path nelle URL.

Query Parameters

Si **deve** utilizzare uno tra i metodi "snake_case" e "camelCase" per i Query Parameters.

Abbreviazioni

Non si devono utilizzare abbreviazioni o acronimi non universalmente conosciuti.

Uso dei nomi al plurale

Le collezioni di risorse **devono** usare nomi al plurale.

Slash finali nei path

Si **deve** evitare slash finali nei path indicati nel descrittore OpenAPI.

Link Headers

Non si devono usare Link Headers RFC5988 se la response è in JSON.

Restituire URI assoluti

Si **deve** usare URI assoluti nei risultati al fine di indicare chiaramente al client l'indirizzo delle risorse di destinazione.

Formato Errori JSON

Si **deve** usare lo schema Problem JSON per le risposte di errore. Il payload restituito è di tipo Problem (RFC 7807) con media-type "application/problem+json".

HTTP Headers custom

Si **deve** evitare l'uso di X-Headers (deprecati in RFC-6648). Eventuali header di nuova definizione:

- **Non devono** andare in conflitto con header pubblici esistenti
- **Non devono** iniziare per "X-"
- **Devono** indicare il nome dell'organizzazione che li ha assegnati

2.4.3 Requisito #4.3: Performance e Robustezza

Rate Limiting

I servizi erogati **devono** attuare politiche di rate limiting. Gli eventuali limiti raggiunti devono essere segnalati ai fruitori tramite lo status "HTTP 429 (too many requests)". Inoltre le risposte **devono** sempre contenere gli header di controllo X-RateLimit-* e **devono** onorare l'header Retry-After, sia nella variante che espone il numero di secondi dopo cui riprovare, sia nella variante che espone la data in cui riprovare.

Sovraccarico del sistema o indisponibilità del servizio

I servizi erogati **devono** esporre un piano di continuità operativa segnalando il sovraccarico del sistema o l'indisponibilità del servizio tramite lo status "HTTP 503 (service unavailable)". **Devono** onorare l'header Retry-After analogamente al punto precedente.

Caching

Le API che supportano il caching **devono** documentare le varie limitazioni e modalità di utilizzo tramite gli header definiti in RFC-7234. Di default il caching è disabilitato tramite indicazione nell'header HTTP "Cache-Control: no-cache".

Paginazione con parametri standard

Si **deve** supportare la paginazione delle collezioni tramite:

- Paginazione classica: offset e limit.
- Paginazione con cursore: pagine con "infinite scrolling".

2.5 Requisito #5: Requisiti di Privacy e di Sicurezza

2.5.1 Requisito #5.1: Privacy by Design e Privacy by Default

Le API esposte **devono** essere progettate seguendo i due principi di base di Privacy by Design e di Privacy by Default:

- *Privacy by Design*: La progettazione delle API che gestiscono dati personali **deve** essere condotta applicando i criteri di riservatezza che troveranno di fatto nell'applicazione che fornisce l'implementazione, ma senza affidarsi ad essa. Devono quindi essere identificati chiaramente i dati personali necessari allo svolgimento delle operazioni e definite le misure di protezione in termini dei livelli di accesso necessari.
- *Privacy by Default*: Le operazioni effettuate nelle API **devono** prevedere, come comportamento di default, il livello massimo di protezione e quindi, a fronte di richieste generiche o prive di autorizzazione, condividono l'insieme minimo di informazioni personali e possibilmente nessuna.

2.5.2 Requisito #5.2: Comunicazione dati personali solo su richieste esplicite

In nessun caso **devono** essere presenti API che restituiscono dati personali fuori dal contesto di una esplicita richiesta dell'utente. I dati personali sono trattati da operazioni ben precise che possono essere attivate solo per richiesta esplicita da parte dell'utente o comunque da lui debitamente autorizzate.

2.5.3 Requisito #5.3: Veicolazione dei dati tramite protocollo HTTPS

Le API che trasmettono dati personali **devono** utilizzare per le comunicazioni esclusivamente il protocollo HTTPS, la cui cifratura dei dati previene eventuali intercettazioni delle informazioni sul percorso di trasmissione.

2.5.4 Requisito #5.4: Accesso circoscritto ai soli dati necessari

Le interfacce API **devono** consentire l'accesso mirato ai dati personali sulla base dello specifico utilizzo e quindi dell'operazione che deve essere effettuata.

Devono essere evitate le soluzioni che veicolano insieme più grandi di dati, non necessari all'esecuzione dell'operazione richiesta, o addirittura la restituzione di un intero profilo di dati personali.

Per ciascuna operazione eseguibile nel contesto della API **deve** essere definito l'insieme minimo di dati personali necessari al suo svolgimento.

Sulla base di ciascun requisito funzionale, **dovranno** essere diversificate le possibili richieste di dati personali in maniera adeguata al tipo di operazione da svolgere.

Possano essere utilizzate due differenti tecniche:

1. Associare i dati acceduti a risorse distinte per caso d'uso. Sulla base dei singoli casi d'uso, vengono individuati gli specifici insiemi di dati personali da condividere, si procede di conseguenza alla definizione di singole risorse/metodi per ciascuno dei casi. Ad esempio:
 - a. `/user-profile/{username}/indirizzo-residenza` risorsa per operare (lettura o scrittura) sull'indirizzo ove è domiciliato l'utente.
 - b. `/user-profile/{username}/dati-istruzione` risorse per operare (lettura o scrittura) sulle informazioni inerenti il titolo di studio dell'utente.
2. Parametrizzare la singola risorsa per selezionare i dati acceduti. La distinzione dei dati personali da condividere viene effettuata, a parità di risorsa, tramite specifici parametri da fornire con la richiesta. Ad esempio:
 - a. `/user-profile/{username}` risorsa che consente potenzialmente di operare sull'intero profilo utente ma che stabilisce precisamente cosa condividere in base ai seguenti parametri (tipicamente nella query string):
 - i. `indirizzo-residenza` : boolean
 - ii. `dati-istruzione` : boolean

L'impostazione a "true" di ciascun parametro corrisponde alla richiesta di uno specifico set di dati. Se entrambi i parametri sono "false", non saranno restituiti dati personali.

Se i set di dati personali sono associati a risorse/metodi distinti (caso 1), l'autorizzazione delle richieste in ingresso **può** essere effettuata indistintamente dal Resource Server che eroga il servizio o dall'API Gateway. Su quest'ultimo infatti è possibile configurare

opportune politiche di autorizzazione basate sulle risorse accedute e le relative credenziali e scope richiesti.

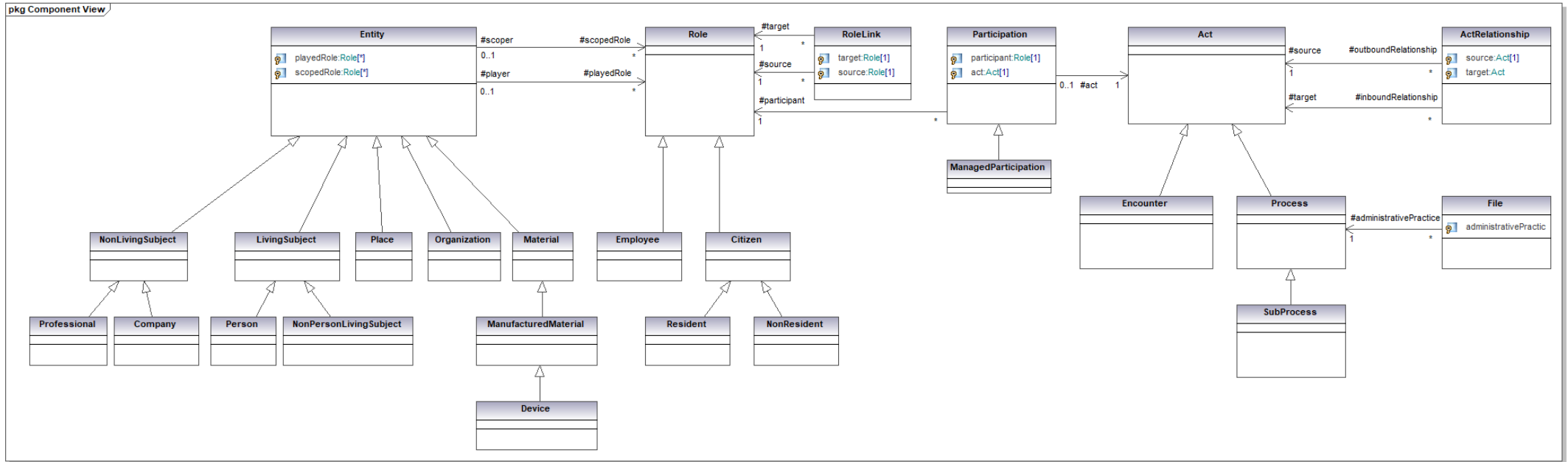
Se i set di dati personali sono distinti tramite parametri, nel contesto di una singola risorsa/metodo, l'autorizzazione **deve** essere effettuata esclusivamente dal Resource Server erogatore del servizio in quanto l'API Gateway non è in grado di effettuare valutazioni autorizzative basate sugli specifici contenuti delle richieste.

3 RISORSE

3.1 Conformance al Modello Informativo di Riferimento del Comune di Genova (R.I.M.)

Le API esposte dal sistema **devono** gestire in richiesta/risposta rappresentazioni di risorse che soddisfino il seguente diagramma delle classi:





Entity:

E' un soggetto o un oggetto in grado di partecipare (Participation) con un certo ruolo (Role) ad un atto (Act). Comprende persone, imprese, luoghi ecc.

Act:

E' un' azione di interesse che viene svolta da una o più Entity in relazione ad un'altra o ad altre Entities. Un' istanza di un Act è un «record» di quell'azione (es. una specifica pratica associata ad un procedimento relativo ad un soggetto o oggetto).

Role:

E' una specializzazione di un Entity definita attraverso la relazione fra un «playing» Entity verso uno «scoping» Entity

Participation:

E' un' associazione fra un Role e un Act che rappresenta la funzione assunta dal Role nel contesto dell' Act. Un singolo Role può partecipare a multipli Act e un singolo Act può avere multipli participating Roles.

Ogni evento che corrisponde ad un'azione eseguita è un Act (una registrazione anagrafica, la presentazione di una istanza, il pagamento di un tributo, l'iscrizione ad un servizio ecc.)

La Participation definisce il contesto di un Act richiedente, ricorrente, titolare, firmatario, destinatario ecc. I partecipanti all'azione sono i Role (residente, contribuente, dipendente comunale, seconda casa, istituto scolastico ecc.)

I ruoli sono impersonati dalle Entity (persona, impresa, civico, edificio, strada, impianto termico ecc.)

Gli Act sono correlabili fra loro attraverso delle relazioni (ActRelationship) quali: composizione, preconditione, aggiornamento, dipendenza ecc....

I Role sono correlabili fra loro attraverso delle relazioni (RoleRelationship) quali: genitore di, coinquilino di , sposato/a con ecc...

3.2 Elenco delle risorse

Nome	Descrizione	Formato (Json/XML)
AvvisoPagamento	Atto che rappresenta la richiesta di pagamento per un determinato servizio	Json
Transazione	Atto che traccia il processo di pagamento per un determinato servizio	Json
RicevutaPagamento	Atto che certifica l'esito di pagamento	Json
Debito	Atto che rappresenta il pagamento dovuto per un determinato servizio	Json
PosizioneDebitoria	Riepilogativo che rappresenta la situazione di debito del soggetto	Json

3.2.1 Proprietà della risorsa AvvisoPagamento

nome	descrizione	obbligatoria	datatype
Debitore	soggetto a cui è intestato l'AvvisoPagamento	sì	SubjectIdentifier
CodiceAvviso	identificativo univoco a livello nazionale dell'AvvisoPagamento	sì	Integer
Importo	importo dovuto	sì	Currency (ISO 4217)
Servizio	servizio codificabile per cui è richiesto il pagamento	sì	CodeableService
Causale	dettaglio informativo del motivo del pagamento	sì	String
DataCreazioneAvviso	Data che indica giorno e ora della creazione dell'AvvisoPagamento	sì	Date (format UTC ISO 8601) <i>YYYY-MM-ddThh:mm:ss</i>
NumeroPratica	Identificativo della pratica generata dal servizio	sì	String
CodiceTransazione	identificativo univoco della Transazione	sì	Integer

3.2.2 Proprietà della risorsa Transazione

nome	descrizione	obbligatoria	datatype
Debitore	soggetto a cui è intestato l'AvvisoPagamento	sì	SubjectIdentifier
CodiceTransazione	identificativo univoco della Transazione	sì	Integer
StatoTransazione	stato di avanzamento del processo	sì	Enumeration {CREATO, INSOLUTO, INSOLUTO DOPO PAGATO, NON AUTORIZZATO, PAGATO, PAGATO DOPO INSOLUTO, PAGATO IN DIFETTO, PAGATO IN ECCESSO, QUADRATO, RICHIAMATO, RIPARTITO}
DataInizioTransazione	Data che indica giorno e ora in cui la transazione è iniziata	sì	Date (format UTC) YYYY-MM-ddThh:mm:ss
DataPagamento	Data che indica giorno e ora dell'avvenuto pagamento	Sì, valorizzato a NULL fino all'avvenuto pagamento	Date (format UTC) YYYY-MM-ddThh:mm:ss
Importo	importo dovuto	sì	Currency (ISO 4217)
Servizio	servizio codificabile per cui è richiesto il pagamento	sì	CodeableService
NumeroPratica	Identificativo della pratica generata dal servizio	sì	String
Causale	dettaglio informativo del motivo del pagamento	sì	String

MetodoPagamento	canale con cui viene effettuato il pagamento	sì	Enumeration {ALTRO, CARTA DI CREDITO, F24, MAV, NODO-PSP BANCARIO, NODO-PSP POSTA, RID BANCARIO, RID POSTALE}
CodiceAvviso	identificativo univoco a livello nazionale dell'AvvisoPagamento	sì, valorizzato a NULL ove non presente. Alternativo a CodiceTransazionePagoPA	Integer
CodiceRicevuta	identificativo univoco della RicevutaPagamento	sì, valorizzato a NULL ove non presente	String
CodiceTransazionePagoPA	identificativo della Transazione avvenuta su PagoPA	sì, valorizzato a NULL ove non presente. Alternativo a CodiceAvviso	String

3.2.3 Proprietà della risorsa RicevutaPagamento

nome	descrizione	obbligatoria	datatype
Debitore	soggetto a cui è intestato l'AvvisoPagamento	sì	SubjectIdentifier
CodiceRicevuta	identificativo univoco della RicevutaPagamento	sì	String
EsitoPagamento	descrizione testuale dell'esito del pagamento	sì	String
Importo	importo dovuto	sì	Currency (ISO 4217)
Servizio	servizio codificabile per cui è richiesto il pagamento	sì	CodeableService

NumeroPratica	Identificativo della pratica generata dal servizio	sì	String
Causale	dettaglio informativo del motivo del pagamento	sì	String
DataPagamento	Data che indica giorno e ora dell'avvenuto pagamento	sì	Date (format UTC) YYYY-MM-ddThh:mm:ss
CodiceAvviso	identificativo univoco a livello nazionale dell'AvvisoPagamento	sì, valorizzato a NULL ove non presente. Alternativo a CodiceTransazionePagoPA	Integer
CodiceTransazionePagoPA	identificativo della Transazione avvenuta su PagoPA	sì, valorizzato a NULL ove non presente. Alternativo a CodiceAvviso	String

3.2.4 Proprietà della risorsa Debito

nome	descrizione	obbligatoria	datatype
Debitore	soggetto a cui è imputato il Debito	sì	SubjectIdentifier
IdentificativoDebito	Codice univoco associato al debito	sì	Integer
Servizio	servizio codificabile per cui è richiesto il pagamento	sì	CodeableService
NumeroPratica	Identificativo della pratica generata dal servizio	sì	String

Causale	dettaglio informativo del motivo del pagamento	sì	String
EsitoPagamento	descrizione testuale dell'esito del pagamento	sì, valorizzato a NULL ove non presente	String
ImportoPagato	importo dovuto e pagato	sì	Currency (ISO 4217)
ImportoDaPagare	importo dovuto e non ancora pagato	sì	Currency (ISO 4217)
CodiciTransazione	Collezione di identificativi univoci delle Transazioni legate al debito	sì, valorizzato a NULL ove non presente	Lista(codiceTransazione)
DataCreazioneDebito	Data che indica giorno e ora della creazione del debito	sì	Date (format UTC) <i>YYYY-MM-ddThh:mm:ss</i>
DataPagamento	Data che indica giorno e ora dell'avvenuto pagamento	sì, valorizzato a NULL ove non presente	Date (format UTC) <i>YYYY-MM-ddThh:mm:ss</i>

3.2.5 Proprietà della risorsa PosizioneDebitoria

nome	descrizione	obbligatoria	Datatype
Debitore	soggetto a cui è imputato il Debito	sì	SubjectIdentifier
Debiti	collezione di debiti	sì	Lista(Debito)
Saldo	importo complessivo dovuto	Sì	Currency (ISO 4217)

3.3 Elenco dei datatype complessi

Nome	Descrizione	Basic/schema
SubjectIdentifier	Identificativo di persona fisica o giuridica	schema Identifier
CodeableService	Servizio espresso in uno o più sistemi di codifica	schema CodeableConcept

4 PATH DELLE INTERFACCE

4.1 Caratteristiche generali

Le URL e gli URI devono sempre essere espressi nel formato:

https://baseUrl/nomeInterfaccia/[versione]/nome_risorsa/{pathparameters}

dove:

- baseUrl comprende:
 - host (o dominio)
 - porta

nel formato: host:porta

- nomeInterfaccia: è il nome assegnato al contesto della risorsa richiesta, ovvero dell'interfaccia API esposta
- versione (opzionale): contiene l'indicazione del numero di versione dell'interfaccia espresso nella forma: m.n.p (es. 1.0.0; 1.1.0; 2.0.1 ecc..)
- nome_risorsa: è il nome della risorsa REST che viene coinvolta.

Tutte le stringhe presenti nella URL **devono** essere case-sensitive.

4.2 Interfaccia ListaTransazioni

L'interfaccia "ListaTransazioni" è un'interfaccia che serve per ottenere la lista delle transazioni in corso per un determinato soggetto.

La risorsa coinvolta è la risorsa "Transazione", che rappresenta l'atto che traccia il processo di pagamento per un determinato servizio.

Questa interfaccia implementa solo il metodo GET del protocollo http e restituisce come response una collezione di transazioni a fronte del parametro in request codice fiscale del soggetto debitore.

Nome Interfaccia	ListaTransazioni
Descrizione	Restituisce la lista delle transazioni per un determinato soggetto
Metodi supportati	GET
Media Types	application/json
Parametri in ingresso	Codice fiscale del soggetto
Messaggi di risposta	HTTP 200 OK. Collezione di transazioni HTTP 400 BAD REQUEST. HTTP 429 TOO MANY REQUESTS.
Path dell'interfaccia	https://baseUrl/mip-api/[versione]/transazioni/{codiceFiscale}
Criteri di sicurezza	

4.3 Interfaccia DettaglioTransazione

L'interfaccia "DettaglioTransazione" è un'interfaccia che serve per ottenere il dettaglio di una transazione specifica.

La risorsa coinvolta è la risorsa "Transazione", che rappresenta l'atto che traccia il processo di pagamento per un determinato servizio.

Questa interfaccia implementa solo il metodo GET del protocollo http e restituisce come response una transazione specifica a fronte del suo codice identificativo.

Nome Interfaccia	DettaglioTransazione
Descrizione	Restituisce il dettaglio di una transazione per uno specifico codice di transazione
Metodi supportati	GET
Media Types	application/json
Parametri in ingresso	Codice della transazione
Messaggi di risposta	HTTP 200 OK. Dettaglio completo della transazione

	HTTP 400 BAD REQUEST. HTTP 429 TOO MANY REQUESTS.
Path dell'interfaccia	https://baseUrl/mip-api/[versione]/transazione/{codiceTransazione}
Criteri di sicurezza	

4.4 Interfaccia ListaAvvisiPersona

L'interfaccia "ListaAvvisiPersona" è un'interfaccia che serve per ottenere la lista degli avvisi di pagamento in corso per un determinato soggetto.

La risorsa coinvolta è la risorsa "AvvisoPagamento", che rappresenta l'atto che identifica la richiesta di pagamento per un determinato servizio.

Questa interfaccia implementa solo il metodo GET del protocollo http e restituisce come response una collezione di avvisi di pagamento a fronte del parametro in request codice fiscale del soggetto debitore.

Nome Interfaccia	ListaAvvisiPersona
Descrizione	Restituisce la lista degli avvisi di pagamento per un determinato soggetto
Metodi supportati	GET
Media Types	application/json
Parametri in ingresso	Codice fiscale del soggetto
Messaggi di risposta	HTTP 200 OK. Collezione di avvisi di pagamento HTTP 400 BAD REQUEST. HTTP 429 TOO MANY REQUESTS.
Path dell'interfaccia	https://baseUrl/mip-api/[versione]/avvisiPagamento/{codiceFiscale}
Criteri di sicurezza	

4.5 Interfaccia ListaAvvisiTransazione

L'interfaccia "ListaAvvisiTransazione" è un'interfaccia che serve per ottenere la lista degli avvisi di pagamento per una determinata transazione.

La risorsa coinvolta è la risorsa "AvvisoPagamento", che rappresenta l'atto che identifica la richiesta di pagamento per un determinato servizio.

Questa interfaccia implementa solo il metodo GET del protocollo http e restituisce come response una collezione di avvisi di pagamento a fronte del parametro in request codice transazione della transazione associata.

Nome Interfaccia	ListaAvvisiTransazione
Descrizione	Restituisce la lista degli avvisi di pagamento per una determinata transazione
Metodi supportati	GET
Media Types	application/json
Parametri in ingresso	Codice transazione
Messaggi di risposta	HTTP 200 OK. Collezione di avvisi di pagamento HTTP 400 BAD REQUEST. HTTP 429 TOO MANY REQUESTS.
Path dell'interfaccia	https://baseUrl/mip-api/[versione]/avvisiPagamento/{codiceTransazione}
Criteri di sicurezza	

4.6 Interfaccia DettaglioAvviso

L'interfaccia "DettaglioAvviso" è un'interfaccia che serve per ottenere il dettaglio di un avviso di pagamento.

La risorsa coinvolta è la risorsa "AvvisoPagamento", che rappresenta l'atto che identifica la richiesta di pagamento per un determinato servizio.

Questa interfaccia implementa solo il metodo GET del protocollo http e restituisce come response il dettaglio di un avviso di pagamento a fronte del suo codice identificativo.

Nome Interfaccia	DettaglioAvviso
Descrizione	Restituisce il dettaglio di un avviso di pagamento per uno specifico codice di avviso pagamento
Metodi supportati	GET
Media Types	application/json
Parametri in ingresso	Codice avviso
Messaggi di risposta	HTTP 200 OK. Dettaglio completo di un avviso di pagamento HTTP 400 BAD REQUEST. HTTP 429 TOO MANY REQUESTS.
Path dell'interfaccia	https://baseUrl/mip-api/[versione]/avvisoPagamento/{codiceAvviso}
Criteri di sicurezza	

4.7 Interfaccia ListaRicevutePersona

L'interfaccia "ListaRicevutePersona" è un'interfaccia che serve per ottenere la lista delle ricevute di pagamento in corso per un determinato soggetto.

La risorsa coinvolta è la risorsa "RicevutaPagamento", che rappresenta l'atto che certifica l'esito di pagamento.

Questa interfaccia implementa solo il metodo GET del protocollo http e restituisce come response una collezione di ricevute di pagamento a fronte del parametro in request codice fiscale del soggetto debitore.

Nome Interfaccia	ListaRicevutePersona
Descrizione	Restituisce la lista delle ricevute di pagamento per un determinato soggetto
Metodi supportati	GET
Media Types	application/json
Parametri in ingresso	Codice fiscale del soggetto
Messaggi di risposta	HTTP 200 OK. Collezione di ricevute di pagamento HTTP 400 BAD REQUEST. HTTP 429 TOO MANY REQUESTS.
Path dell'interfaccia	https://baseUrl/mip-api/[versione]/ricevutePagamento/{codiceFiscale}
Criteri di sicurezza	

4.8 Interfaccia ListaRicevuteAvviso

L'interfaccia "ListaRicevuteAvviso" è un'interfaccia che serve per ottenere la lista delle ricevute di pagamento in corso per un determinato avviso di pagamento.

La risorsa coinvolta è la risorsa "RicevutaPagamento", che rappresenta l'atto che certifica l'esito di pagamento.

Questa interfaccia implementa solo il metodo GET del protocollo http e restituisce come response una collezione di ricevute di pagamento a fronte del parametro in request "codice avviso di pagamento".

Nome Interfaccia	ListaRicevuteAvviso
Descrizione	Restituisce la lista delle ricevute di pagamento per un determinato avviso di pagamento
Metodi supportati	GET
Media Types	application/json
Parametri in ingresso	Codice avviso di pagamento

Messaggi di risposta	HTTP 200 OK. Collezione di ricevute di pagamento HTTP 400 BAD REQUEST. HTTP 429 TOO MANY REQUESTS.
Path dell'interfaccia	https://baseUrl/mip-api/[versione]/ricevutePagamento/{codiceAvviso}
Criteri di sicurezza	

4.9 Interfaccia ListaRicevuteTransazionePagoPA

L'interfaccia "ListaRicevuteTransazionePagoPA" è un'interfaccia che serve per ottenere la lista delle ricevute di pagamento in corso per una determinata transazione di PagoPA.

La risorsa coinvolta è la risorsa "RicevutaPagamento", che rappresenta l'atto che certifica l'esito di pagamento.

Questa interfaccia implementa solo il metodo GET del protocollo http e restituisce come response una collezione di ricevute di pagamento a fronte del parametro in request codice della transazione di PagoPA.

Nome Interfaccia	ListaRicevuteTransazionePagoPA
Descrizione	Restituisce la lista delle ricevute di pagamento per una determinata transazione PagoPA
Metodi supportati	GET
Media Types	application/json
Parametri in ingresso	Codice transazione PagoPA
Messaggi di risposta	HTTP 200 OK. Collezione di ricevute di pagamento HTTP 400 BAD REQUEST. HTTP 429 TOO MANY REQUESTS.
Path dell'interfaccia	https://baseUrl/mip-api/[versione]/ricevutePagamento/{codiceTransazionePagoPA }
Criteri di sicurezza	

4.10 Interfaccia DettaglioRicevuta

L'interfaccia "DettaglioRicevuta" è un'interfaccia che serve per ottenere il dettaglio di una ricevuta di pagamento.

La risorsa coinvolta è la risorsa "RicevutaPagamento", che rappresenta l'atto che certifica l'esito di pagamento.

Questa interfaccia implementa solo il metodo GET del protocollo http e restituisce come response il dettaglio di una ricevuta di pagamento a fronte del suo codice identificativo.

Nome Interfaccia	DettaglioRicevuta
Descrizione	Restituisce il dettaglio di una ricevuta di pagamento per uno specifico codice ricevuta di pagamento
Metodi supportati	GET
Media Types	application/json
Parametri in ingresso	Codice ricevuta
Messaggi di risposta	HTTP 200 OK. Dettaglio completo di una ricevuta di pagamento HTTP 400 BAD REQUEST. HTTP 429 TOO MANY REQUESTS.
Path dell'interfaccia	https://baseUrl/mip-api/[versione]/ricevutaPagamento/{codiceRicevuta}
Criteri di sicurezza	

4.11 Interfaccia ListaDebitiPersona

L'interfaccia "ListaDebitiPersona" è un'interfaccia che serve per ottenere la lista dei debiti in corso per un determinato soggetto.

La risorsa coinvolta è la risorsa "Debito", che rappresenta l'atto che rappresenta il pagamento dovuto per un determinato servizio.

Questa interfaccia implementa solo il metodo GET del protocollo http e restituisce come response una collezione di debiti a fronte del parametro in request codice fiscale del soggetto debitore.

Nome Interfaccia	ListaDebitiPersona
Descrizione	Restituisce la lista dei debiti per un determinato soggetto
Metodi supportati	GET
Media Types	application/json
Parametri in ingresso	Codice fiscale del soggetto
Messaggi di risposta	HTTP 200 OK. Collezione di debiti HTTP 400 BAD REQUEST. HTTP 429 TOO MANY REQUESTS.
Path dell'interfaccia	https://baseUrl/mip-api/[versione]/debiti/{codiceFiscale}
Criteri di sicurezza	

4.12 Interfaccia ListaDebitiPersonaServizio

L'interfaccia "ListaDebitiPersona" è un'interfaccia che serve per ottenere la lista dei debiti in corso per un determinato soggetto e per un determinato servizio.

La risorsa coinvolta è la risorsa “Debito”, che rappresenta l’atto che rappresenta il pagamento dovuto per un determinato servizio.

Questa interfaccia implementa solo il metodo GET del protocollo http e restituisce come response una collezione di debiti a fronte dei parametri in request codice fiscale del soggetto debitore e il servizio.

Nome Interfaccia	ListaDebitiPersonaServizio
Descrizione	Restituisce la lista dei debiti per un determinato soggetto e per un determinato servizio
Metodi supportati	GET
Media Types	application/json
Parametri in ingresso	Codice fiscale del soggetto, servizio
Messaggi di risposta	HTTP 200 OK. Collezione di debiti HTTP 400 BAD REQUEST. HTTP 429 TOO MANY REQUESTS.
Path dell’interfaccia	https://baseUrl/mip-api/[versione]/debiti/{codiceFiscale}/{servizio}
Criteri di sicurezza	

4.13 Interfaccia DettaglioDebito

L’interfaccia “DettaglioDebito” è un’interfaccia che serve per ottenere il dettaglio di un debito.

La risorsa coinvolta è la risorsa “Debito”, che rappresenta l’atto che certifica pagamento dovuto per un determinato servizio.

Questa interfaccia implementa solo il metodo GET del protocollo http e restituisce come response il dettaglio di debito a fronte del suo codice identificativo.

Nome Interfaccia	DettaglioDebito
Descrizione	Restituisce il dettaglio di debito per uno specifico codice debito
Metodi supportati	GET
Media Types	application/json
Parametri in ingresso	Codice debito
Messaggi di risposta	HTTP 200 OK. Dettaglio completo di un debito HTTP 400 BAD REQUEST. HTTP 429 TOO MANY REQUESTS.
Path dell’interfaccia	https://baseUrl/mip-api/[versione]/debito/{identificativoDebito}
Criteri di sicurezza	

4.14 Interfaccia PosizioneDebitoriaPersona

L'interfaccia "PosizioneDebitoriaPersona" è un'interfaccia che serve per ottenere la posizione debitoria di un determinato soggetto.

La risorsa coinvolta è la risorsa "PosizioneDebitoria", che rappresenta il riepilogativo che rappresenta la situazione di debito del soggetto.

Questa interfaccia implementa solo il metodo GET del protocollo http e restituisce come response la posizione debitoria a fronte del codice fiscale.

Nome Interfaccia	PosizioneDebitoriaPersona
Descrizione	Restituisce la posizione debitoria per un determinato soggetto
Metodi supportati	GET
Media Types	application/json
Parametri in ingresso	Codice fiscale
Messaggi di risposta	HTTP 200 OK. Posizione debitoria HTTP 400 BAD REQUEST. HTTP 429 TOO MANY REQUESTS.
Path dell'interfaccia	https://baseUrl/mip-api/[versione]/posizioneDebitoria/{codiceFiscale}
Criteri di sicurezza	

4.15 Interfacce in input

Il sistema dovrà prevedere, per sviluppi futuri, la possibilità di ricevere delle chiamate per l'inserimento di informazioni e/o dati nei propri database.

Ad esempio, si prevede la necessità di ricevere e gestire

- un insieme di dati relativi alla creazione di un debito per un determinato cittadino o una determinata impresa
- un insieme di dati per l'aggiornamento del debito di un determinato cittadino o una determinata impresa

e restituire, se necessario, l'avviso di pagamento relativo al debito.

Le interfacce dovranno necessariamente essere implementate basandosi sulle specifiche indicate nelle sezioni 1-4 del presente documento e integrandosi con il sistema di API Management (basato sulla piattaforma WSO2) attualmente in uso all'Ente.

5 ALLEGATI

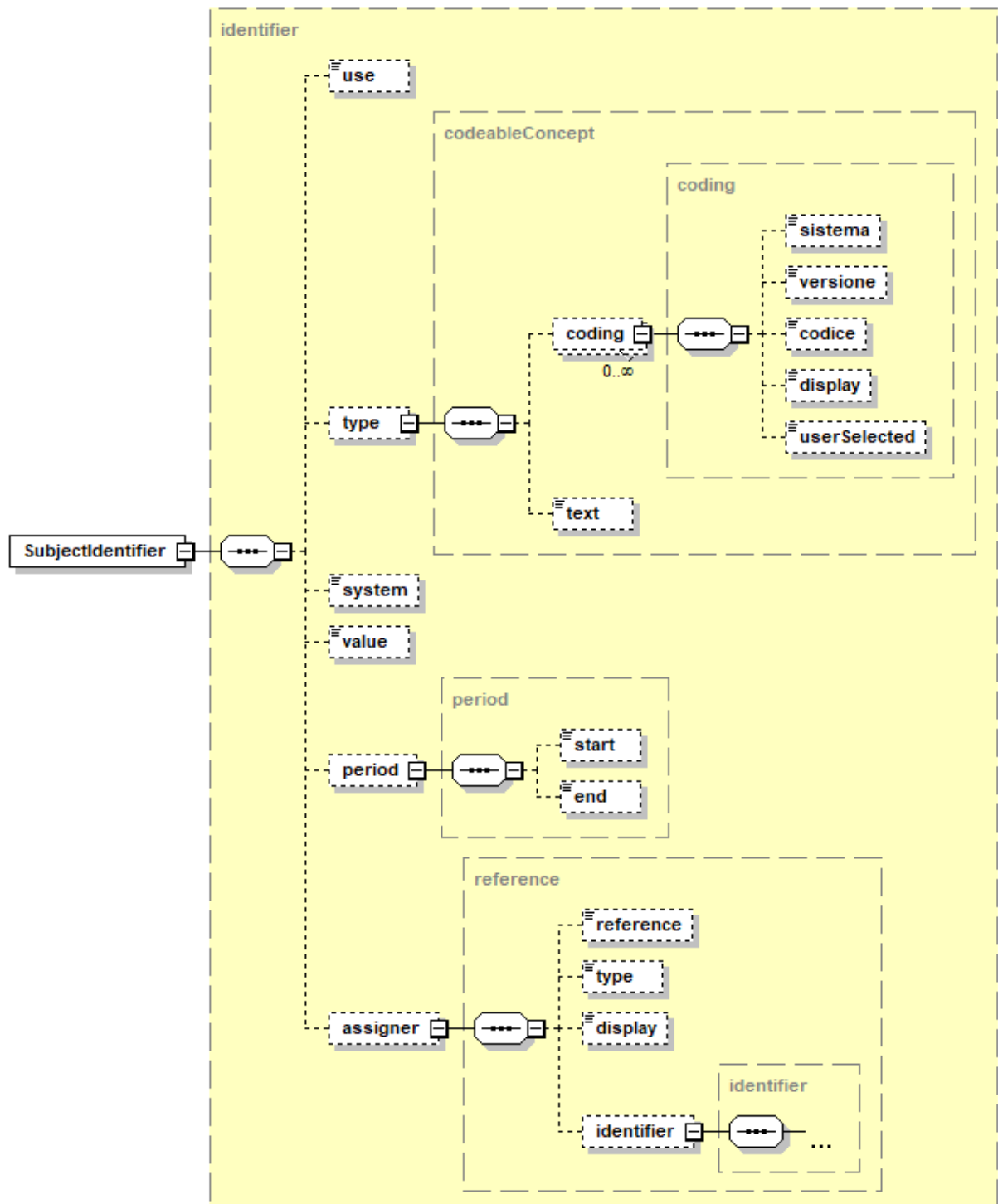
5.1 Datatypes

5.1.1 SubjectIdentifier

Il datatype SubjectIdentifier è un tipo di dati complesso (di tipo Identifier) che serve per rappresentare uno o più identificativi del soggetto.

La struttura del tipo è quella mostrata nel diagramma seguente:





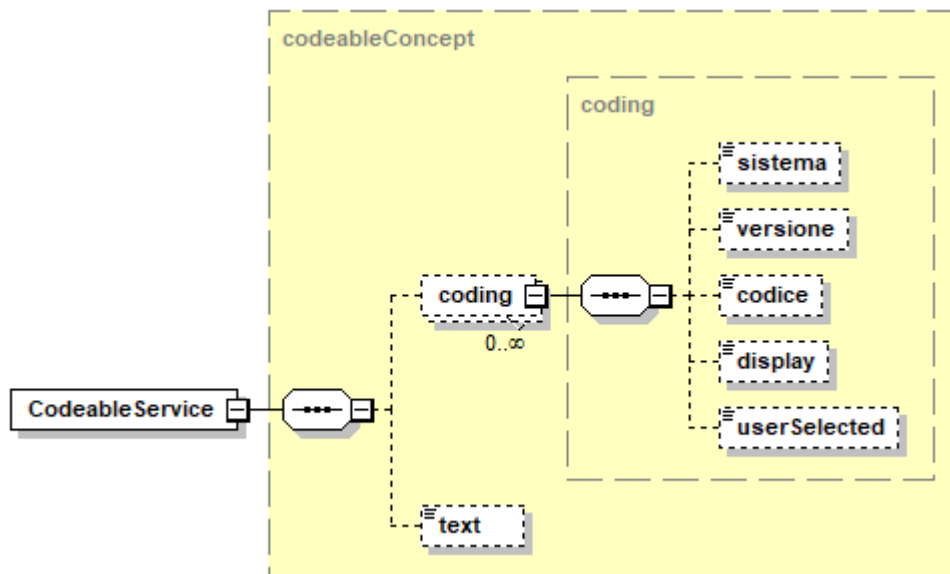
- *use (opz.)* Da compilare con uno dei seguenti valori: usual | official | temp | secondary | old

- *type*: vedere la descrizione del tipo complesso CodeableConcept nel datatype CodeableService
- *system (opz)*: namespace (uri) per il valore dell'identificativo. Valorizzare con 2.16.840.1.113883.2.9.4.3.2 se l'identificativo è il codice fiscale della persona
- *value*: il valore dell'identificativo (esempio il valore del codice fiscale)
- *Period (opz.)*: è il periodo di validità temporale (compreso fra start e end) per cui è valido l'identificativo. Se non valorizzati l'identificativo è inteso al momento attuale
- *Assigner (opz)*: serve per rappresentare opzionalmente l'organizzazione che certifica l'identificativo. Comprende:
 - Reference: nome caratteristico dell'organizzazione (ad esempio MEF)
 - Type: namespace (uri) per l'organizzazione (ad esempio se MEF usare 2.16.840.1.113883.2.9.4.3)
 - Display: descrizione testuale dell'organizzazione (ad esempio Ministero dell'Economia e Finanza)
 - Identifier: è una relazione circolare per rappresentare un eventuale identificativo dell'organizzazione

5.1.2 CodeableService

Il datatype CodeableService è un tipo di dati complesso (di tipo CodeableConcept) che serve per rappresentare attraverso un concetto semantico codificato il servizio per cui è richiesto il pagamento.

La struttura del tipo CodeableConcept è quella mostrata nel diagramma seguente:



- *coding (opz.)* esprime uno o più codifiche per il servizio. E' composto da:
 - *sistema*: un identificativo del sistema di codifica utilizzato
 - *versione*: la versione (se esiste) del sistema di codifica utilizzato
 - *codice*: il codice espresso nel sistema di codifica
 - *display*: nome (testuale) che corrisponde al codice
 - *userSelected*: (boolean) true se il sistema di codifica è stato scelto direttamente dall'utente (default = false)
- *text*: descrizione (estesa) testuale

5.2 Esempi

5.2.1 avvisoPagamentoExample.json

```
{
  "avvisoPagamento": {
    "debitore": {
      "use": "usual",
      "type": {
        "coding": {
          "sistema": "http://terminologie.comune.genova.it/entity/livingSubject/person/personRole",
          "versione": "1.0",
          "codice": "DEB",
          "display": "Cittadino titolare di un generico debito nei confronti dell'Amministrazione Comunale"
        },
        "text": "Soggetto intestatario dell'avviso"
      },
      "system": "http://terminologie.comune.genova.it/entity/livingSubject/person/codiceFiscale",
      "value": "PLNPRN80A01D969L",
      "assigner": {
        "reference": "http://terminologie.comune.genova.it/entity/organization/ministeri/mef",
        "display": "Ministero Economia e Finanza"
      }
    },
    "codiceAvviso": 123456,
    "importo": {
      "value": 12.34,
      "currency": "EUR"
    },
    "servizio": {
      "coding": {
        "sistema": "http://terminologie.comune.genova.it/entity/organization/comune/serviziCivici",
        "versione": "1.0",
        "codice": "SC",
        "display": "Stato Civile"
      },
      "text": "Matrimoni/unioni civili"
    },
    "causale": "Matrimonio/Unione Civile palazzo Ducale/Acquario",
    "dataCreazioneAvviso": "2019-03-12T10:15:32",
    "numeroPratica": "PR9876543",
    "codiceTransazione": 987654321
  }
}
```

5.2.2 transazioneExample.json

```
{
  "transazione": {
    "debitore": {
      "use": "usual",
      "type": {
        "coding": {
          "sistema": "http://terminologie.comune.genova.it/entity/livingSubject/person/personRole",
          "versione": 1.0,
          "codice": "DEB",
          "display": "Cittadino titolare di un generico debito nei confronti dell'Amministrazione Comunale"
        },
        "text": "Soggetto intestatario della transazione"
      },
      "system": "http://terminologie.comune.genova.it/entity/livingSubject/person/codiceFiscale",
      "value": "PLNPRN80A01D969L",
      "assigner": {
        "reference": "http://terminologie.comune.genova.it/entity/organization/ministeri/mef",
        "display": "Ministero Economia e Finanza"
      }
    },
    "codiceTransazione": 13579,
    "statoTransazione": "PAGATO",
    "dataRichiestaPagamento": "2019-02-09T12:05:37",
    "dataPagamento": "2019-02-11T07:32:51",
    "importo": {
      "value": 13.20,
      "currency": "EUR"
    },
    "servizio": {
      "coding": {
        "sistema": "http://terminologie.comune.genova.it/entity/organization/comune/serviziCivici",
        "versione": 1.0,
        "codice": "SC",
        "display": "Stato Civile"
      },
      "text": "Matrimoni/unioni civili"
    },
    "causale": "Matrimonio/Unione Civile palazzo Ducale/Acquario",
    "metodoPagamento": "BANCARIO",
    "codiceAvviso": 1234,
    "codiceRicevuta": "566986543",
    "codiceTransazionePagoPA": "97654987682",
    "numeroPratica": 123
  }
}
```