Linee Guida Modello di Interoperabilità

Release Bozza in consultazione

Agenzia per l'Italia Digitale

29 mag 2018

Indice

Consultazione pubblica

La consultazione pubblica per questo documento è attiva dal 8 maggio al 7 giugno 2018.

Questo documento raccoglie il testo delle line guida del *Modello di interoperabilità 2018 per la Pubblica Amministrazione*, disponibile per la consultazione pubblica.

Leggi le istruzioni per la consultazione

Vai al testo delle linee guida:

CAPITOLO 1

Istruzioni per la consultazione pubblica

1.1 Informazioni sulla consultazione

- **Durata della consultazione:** dal 8 maggio 2018 all'7 giugno 2018
- **Settore:** ICT

1.2 Esiti della consultazione

I risultati della consultazione pubblica on line saranno presi in considerazione dall'Agenzia per l'Italia Digitale per la redazione del testo definitivo della Guida Tecnica.

1.3 Destinatari

Tutte le pubbliche amministrazioni che si trovano a sviluppare o mantenere un parco applicativo da utilizzare nell'ambito dei propri compiti istituzionali che richieda interazione con altre pubbliche amministrazioni o soggetti terzi (cittadini e imprese), gli operatori del mercato ICT (aziende, sviluppatori, integratori, etc) e tutti gli addetti ai lavori o gli interessati al tema.

1.4 Obiettivo della consultazione

Il documento rappresenta l'aggiornamento delle linee guida sul tema della cooperazione applicativa (SPCoop), pubblicate dall'allora CNIPA.

Le linee guida sono redatte per l'attuazione del Codice dell'Amministrazione Digitale ai sensi dell'art. 71 del Codice stesso.

L'esigenza di aggiornare le precedenti linee guida nasce:

- dalle novità introdotte, anche sotto l'aspetto strategico, dal «Piano Triennale per l'informatica nella Pubblica amministrazione 2017- 2019»;
- dal tempo trascorso dall'ultima revisione del documento, in relazione alla rapidità con cui notoriamente evolve il settore dell'ICT e gli standard disponibili;

1.5 Come partecipare

Le linee guida sul Modello di Interoperabilità per le PA sono pubblicate su Docs Italia ed è possibile commentarle su Forum Italia.

È possibile inviare i propri commenti fino all'7 giugno 2018.

CAPITOLO 2

Presentazione del Modello di Interoperabilità 2018

La visione generale del Modello di Interoperabilità 2018, considerando il contesto Europeo, introduce gli elementi che saranno considerati e le modalità con cui si provvederà al costante aggiornamento dello stesso.

2.1 Introduzione

Il Modello di Interoperabilità 2018[1] (nel seguito in breve ModI 2018) rappresenta il modello di supporto alla strategia di interoperabilità e cooperazione tra le Pubbliche Amministrazioni (di seguito PA), che definendo i contesti di interazione e integrazione tra le PA, i cittadini e le imprese permette di vedere la PA nella sua interezza come un unico sistema informativo (virtuale).

La definizione del ModI 2018 deve essere coerente con il nuovo *European Interoperability Framework* (EIF) oggetto della Comunicazione COM (2017)134[2] della Commissione Europea del 23 marzo 2017, al fine di assicurare anche l'interoperabilità nel contesto Europeo e per l'attuazione del *Digital Single Market* (Mercato Unico Digitale).

Gli obiettivi del nuovo Modello di Interoperabilità 2018 sono:

- definire le modalità di integrazione tra le PA;
- armonizzare le scelte architetturali delle PA;
- individuare le scelte tecnologiche che favoriscano lo sviluppo, da parte delle PA, cittadini e imprese, di soluzioni applicative innovative che abilitino l'utilizzo dei servizi individuati nelle Infrastrutture immateriali del Piano triennale per l'informatica nella PA[3];
- promuovere, quando applicabile, l'adozione dell'approccio *API first*, al fine di favorire la separazione dei livelli di back end e front end, con logiche aperte e standard pubblici che garantiscano ad altri attori, pubblici e privati, accessibilità e massima interoperabilità di dati e servizi;
- privilegiare standard tecnologici che soddisfino l'esigenza di rendere sicure le interazioni tra le PA e tra queste con i cittadini e le imprese;

[1] Il ModI 2018 è concettualmente la seconda versione (aggiornamento) del framework di interoperabilità della PA che nella prima versione fu definito nel 2005 con il nome di SPCoop - Servizio Pubblico di Cooperazione Applicativa, cf. http://www.agid.gov.it/agenda-digitale/infrastrutture-architetture/sistema-pubblico-connettivita/cooperazione-applicativa Il termine *modello* trova corrispettivo nel termine inglese framework, e pertanto nel presente documento i due termini verranno considerati sinonimi.

[2] Cf. https://ec.europa.eu/transparency/regdoc/rep/1/2017/IT/COM-2017-134-F1-IT-MAIN-PART-1.PDF

[3] Cf. https://pianotriennale-ict.italia.it/assets/pdf/Piano_Triennale_per_l_informatica_nella_Pubblica_Amministrazione.pdf

- semplificare le procedure di scambio di servizi tra le PA e, ove possibile, tra PA e privati, facilitando la realizzazione dei sistemi che le realizzano.

2.2 Il contesto europeo

Lo *European Interoperability Framework (EIF)* (in italiano Quadro Europeo di Interoperabilità - QEI[4]) fornisce orientamenti alle PA Europee su come operare le iniziative relative al tema dell'interoperabilità; tutto questo mediante una serie di raccomandazioni atte a stabilire relazioni tra le varie organizzazioni, razionalizzare i processi volti a sostenere i servizi digitali e assicurare che le norme esistenti e quelle nuove non pregiudichino gli sforzi di interoperabilità.

L'obiettivo dell'EIF è:

- orientare gli sforzi delle PA Europee nel progettare ed erogare servizi pubblici ad altre PA, cittadini e imprese che siano, per quanto possibile, *(i)* digitali per definizione, *(ii)* transfrontalieri per definizione e *(iii)* aperti per definizione;
- fornire alle PA orientamenti in merito alla progettazione e all'aggiornamento di quadri nazionali di interoperabilità o di politiche nazionali, strategie e orientamenti che promuovano l'interoperabilità;
- contribuire all'istituzione del Digital Single Market incoraggiando l'interoperabilità transfrontaliera e intersettoriale per l'erogazione di servizi pubblici europei.

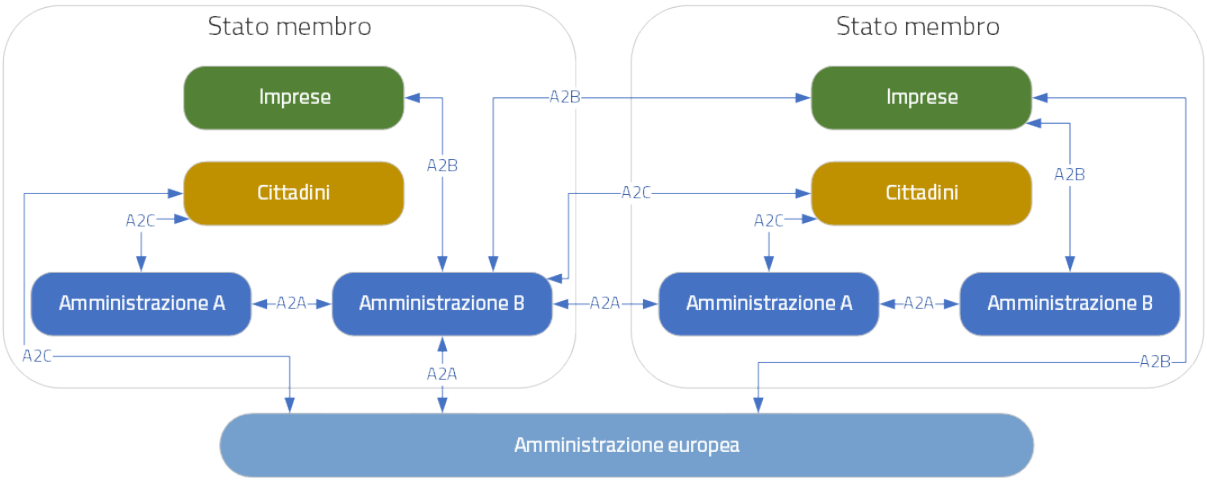


Figura 1 - Ambito di applicazione dell'EIF

L'ambito di applicazione dell'EIF comprende tre tipi di interazioni:

- A2A (*amministrazione-amministrazione*), ossia le interazioni tra PA;
- A2B (*amministrazione-impresa*), ossia le interazioni tra le PA e le imprese;
- A2C (*amministrazione-cittadino*), ossia le comunicazioni tra le PA e i cittadini.

Il modello di interoperabilità delineato nell'EIF è applicabile a tutti i servizi pubblici digitali, lo stesso include:

- quattro livelli di interoperabilità:
  - giuridico, per garantire che le organizzazioni che operano nell'ambito di diversi quadri giuridici (nazionali e settoriali), possano lavorare insieme;
  - organizzativo, per favorire l'allineamento delle procedure e processi delle organizzazioni coinvolte delineando le responsabilità e le aspettative per raggiungere obiettivi comuni concordati e reciprocamente vantaggiosi;

[4] In precedenti documenti a cura di AgID e del Team Digitale, il termine inglese framework è stato sovente tradotto in italiano come modello, ed è questo il termine utilizzato nel presente documento. La dicitura *quadro* è la traduzione letterale della Commissione Europea. Nel seguito di questo documento verrà preferito il termine modello, pur considerando i termini framework, modello e quadro come sinonimi.

  - semantico, per assicurare che il formato e il significato delle informazioni e dei dati scambiati siano mantenuti e compresi durante tutti gli scambi che avvengono tra le parti;
  - tecnico, in cui, attraverso l'adozione di specifiche di interfaccia, di servizi di interconnessione, di servizi di integrazione dei dati, la presentazione e lo scambio dei dati e i protocolli di comunicazione sicuri, si assicuri l'interoperabilità delle applicazioni e delle infrastrutture che collegano sistemi e servizi.

- una componente trasversale ai quattro livelli, denominata *governance dei servizi pubblici integrati*, per assicurare il necessario coordinamento e governance delle organizzazioni coinvolte nella erogazione di servizi pubblici in modo integrato;
- un livello di base, denominato *governance di interoperabilità*, per assicurare che le decisioni prese in merito ai quadri di interoperabilità, disposizioni istituzionali, strutture organizzative, ruoli e responsabilità, politiche, accordi e altri aspetti garantiscano e verifichino l'interoperabilità a livello nazionale e di UE.

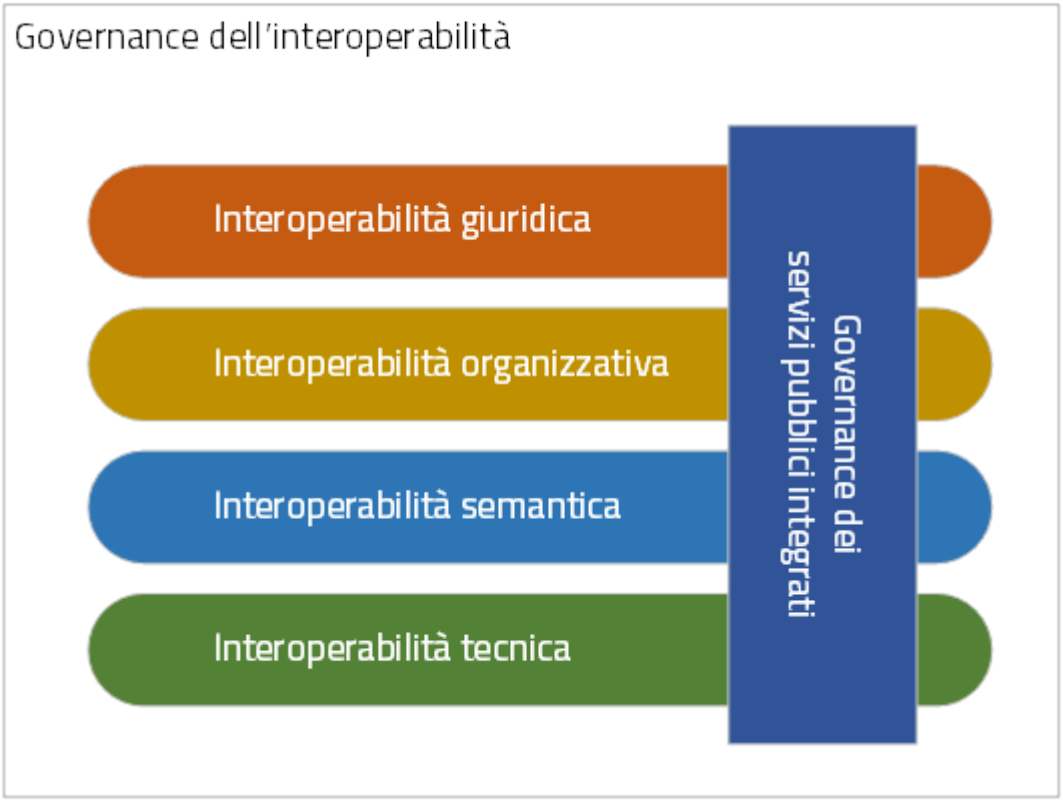


Figura 2 - Livelli di interoperabilità

Nel suo insieme il modello di interoperabilità delineato nell'EIF è stato disegnato sulla base dei 12 principi di interoperabilità, condivisi dagli Stati membri della Comunità Europea, individuati quali aspetti fondamentali per guidare le azioni tese a garantire l'interoperabilità:

1. *Sussidiarietà e proporzionalità*. Il principio di sussidiarietà prevede che le decisioni dell'UE vengano prese al livello più vicino a quello del cittadino mentre il principio di proporzionalità limita l'azione dell'UE a quanto necessario per raggiungere gli obiettivi dei trattati.
2. *Apertura*. Il principio di apertura fa riferimento principalmente ai dati, alle specifiche e al software. Nell'ottica di questo principio occorre: pubblicare i dati che si possiedono come dati aperti, fatta salva l'eventuale applicazione di determinate restrizioni; garantire condizioni di parità per il software open source e prenderne in considerazione l'utilizzo in modo attivo ed equo, tenendo conto del costo totale di proprietà della soluzione; prediligere le specifiche aperte, tenendo debitamente conto delle esigenze funzionali, del livello di maturità e del sostegno e dell'innovazione del mercato.
3. *Trasparenza*. In ottemperanza a questo principio occorre: conferire visibilità nel contesto amministrativo di una PA; assicurare la disponibilità di interfacce con i sistemi informatici interni e garantire il diritto alla tutela dei dati personali; garantire visibilità interna e fornire interfacce esterne per i servizi pubblici.

4. *Riusabilità*. Secondo tale principio si deve trarre vantaggio dal lavoro degli altri cercando le informazioni disponibili, valutandone l'utilità o la pertinenza rispetto al problema in questione e, se del caso, decidendo di usare soluzioni che si sono rivelate efficaci in altre situazioni.
5. *Neutralità tecnologica e portabilità dei dati*. Allorché istituiscono servizi pubblici, le PA devono concentrarsi sulle esigenze funzionali e posporre le decisioni in materia di tecnologia il più a lungo possibile per ridurre al minimo la dipendenza tecnologica, evitare di imporre tecnologie o prodotti specifici ai loro partner ed essere in grado di adattarsi all'ambiente tecnologico in rapida evoluzione.
6. *Centralità dell'utente*. Nel determinare quali servizi pubblici erogare e come farlo, si deve prendere in considerazione le esigenze degli utenti. Occorre perciò mettere a punto meccanismi per coinvolgere gli utenti nell'analisi, nella progettazione, nella valutazione e nell'ulteriore sviluppo dei servizi pubblici
7. *Inclusione e accessibilità*. Inclusione significa permettere a chiunque di approfittare delle opportunità offerte dalle nuove tecnologie per l'accesso e l'utilizzo dei servizi pubblici europei superando gli svantaggi e l'esclusione sociale ed economica. L'accessibilità garantisce che le persone anziane, i disabili e gli altri gruppi svantaggiati possano utilizzare i servizi pubblici alla stregua di tutti gli altri cittadini.
8. *Sicurezza e privacy*. Le interazioni con le autorità pubbliche devono svolgersi in un ambiente sicuro ed affidabile ed in totale conformità con le norme in materia di protezione dei dati, di identificazione elettronica e dei servizi fiduciari.
9. *Multilinguismo*. Occorre soddisfare le aspettative di cittadini e imprese che desiderano essere serviti nella loro lingua, o in un'altra lingua di preferenza, e la capacità delle PA di offrire servizi in tutte le lingue ufficiali.
10. *Semplificazione Amministrativa*. Le PA, laddove possibile, devono razionalizzare e semplificare le loro procedure amministrative migliorandole o eliminando quelle che non hanno utilità pubblica.
11. *Conservazione delle informazioni*. La legislazione impone che le decisioni e i dati siano conservati e che vi si possa accedere per un determinato periodo di tempo. Occorre pertanto formulare una politica di conservazione a lungo termine per le informazioni relative ai servizi pubblici.
12. *Valutazione dell'efficacia e dell'efficienza*. Esistono numerosi modi per misurare il valore offerto dall'interoperabilità dei servizi pubblici, quali le considerazioni circa il ritorno sull'investimento, il costo totale di proprietà, il livello di flessibilità e adattabilità, la riduzione degli oneri amministrativi, l'efficienza, la riduzione dei rischi, la trasparenza, la semplificazione, il miglioramento dei metodi di lavoro e il grado di soddisfazione degli utenti. Valutare l'efficacia e l'efficienza di diverse soluzioni di interoperabilità e opzioni tecnologiche, in considerazione delle esigenze dell'utente, della proporzionalità e dell'equilibrio tra costi e benefici.

L'EIF delinea uno schema concettuale per i servizi pubblici integrati al fine di orientarne la progettazione, lo sviluppo, la gestione e la manutenzione da parte degli Stati membri. Lo schema concettuale promuove l'idea di *interoperability-by-design* (*interoperabilità fin dalla fase di progettazione*). Lo schema promuove la riusabilità come motore per l'interoperabilità, riconoscendo che i servizi pubblici dovrebbero riutilizzare le informazioni e i servizi esistenti e provenienti da varie fonti, sia all'interno che all'esterno dei confini organizzativi delle PA. Le informazioni e i servizi dovrebbero essere recuperabili e resi disponibili in formati interoperabili.

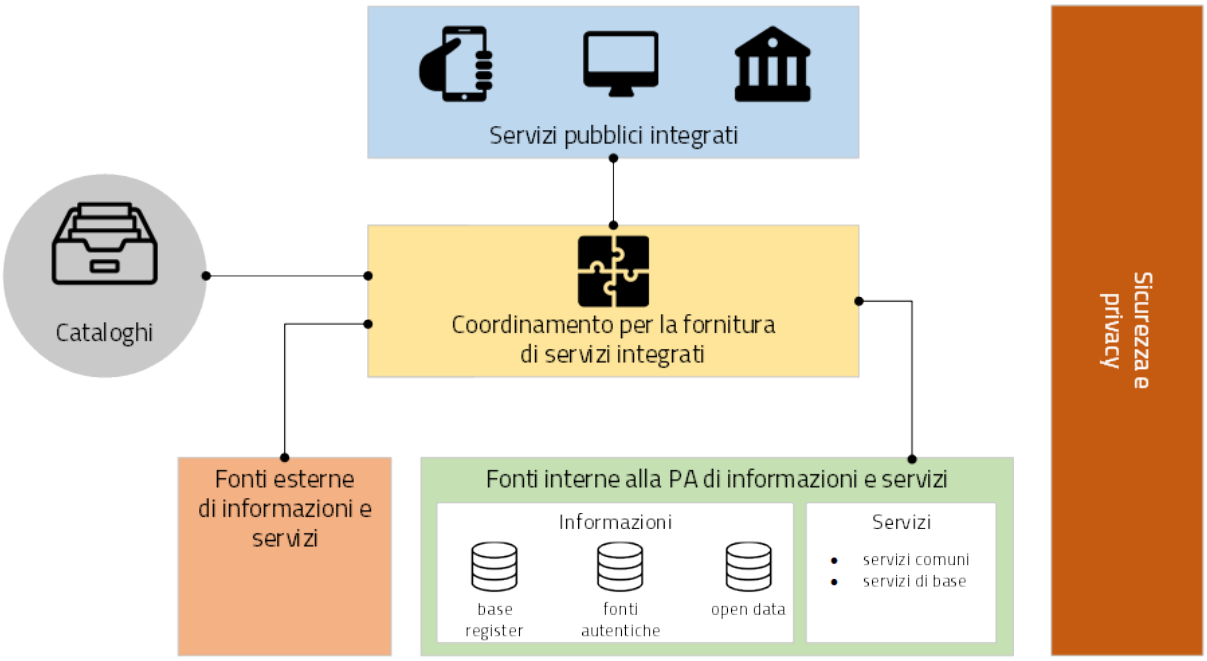


Figura 3 - Schema concettuale per i servizi pubblici integrati

La Commissione Europea ha individuato uno schema concettuale per i servizi pubblici che comprende:

- la *fornitura di servizi integrati* basata su una *funzione di coordinamento* per eliminare la complessità per l'utente finale;
- una politica di fornitura del servizio basata sul *concetto secondo cui tutte le porte sono buone* per offrire opzioni e canali alternativi per l'erogazione dei servizi, garantendo nel contempo la disponibilità di canali digitali (*digital first*);
- il riutilizzo di dati e servizi per ridurre i costi e accrescere la qualità dei servizi e l'interoperabilità;
- cataloghi che descrivono i servizi riutilizzabili e le altre risorse per aumentare la loro rintracciabilità e il loro utilizzo;
- la governance dei servizi pubblici integrati;
- la sicurezza e la tutela della privacy.

La funzione di coordinamento garantisce l'individuazione delle esigenze e il ricorso ai servizi coordinati per fornire complessivamente un servizio pubblico. Le fonti di informazioni (*base register*, portali sui dati aperti e altre fonti autorevoli di informazioni) e i servizi, disponibili non solo all'interno del sistema amministrativo ma anche in un contesto esterno, possono essere utilizzati per creare servizi pubblici integrati. Per favorire questi processi occorre sviluppare un'infrastruttura condivisa di servizi e fonti di informazioni riutilizzabili che possa essere adottata da tutte le amministrazioni pubbliche favorendo il riutilizzo, la pubblicazione e l'aggregazione dei servizi e delle fonti di informazioni.

La direttiva relativa al riutilizzo dell'informazione del settore pubblico prevede un quadro giuridico comune per il riutilizzo dei dati (*open data*); in essa l'accento è posto sulla messa a disposizione di dati *machine-readable* ad uso di terzi per promuovere la trasparenza, la concorrenza leale, l'innovazione e un'economia basata sui dati.

I *cataloghi* hanno la finalità di consentire la ricerca di servizi, dati, software e modelli di dati.

Le PA devono poter fruire dei servizi erogati da terzi al di fuori dei confini delle loro organizzazioni, quali i servizi di pagamento forniti dalle istituzioni finanziarie oppure i servizi di connettività erogati da fornitori di servizi di telecomunicazioni. Esse hanno bisogno anche di utilizzare le *fonti esterne di informazioni*, quali i dati aperti e i dati delle organizzazioni internazionali, delle camere di commercio, ecc.

Nell'EIF è raccomandato:

- rendere disponibili fonti autorevoli di informazioni a terzi, istituendo nel contempo meccanismi di accesso e controllo per garantire la sicurezza e la riservatezza in conformità con la normativa specifica in materia;

- sviluppare interfacce con i base register, pubblicare i mezzi tecnici e i documenti necessari affinché terze parti possano connettersi e riutilizzare le informazioni disponibili;
- abbinare ad ogni base register i metadati appropriati, compresi la descrizione del contenuto, la garanzia del servizio e le responsabilità, le tipologie di master data contenuti, le condizioni di accesso e le licenze, la terminologia, il glossario e le informazioni sugli eventuali master data utilizzati di altri base register;
- creare e monitorare piani di garanzia della qualità dei dati per i base register e i relativi master data;
- elaborare cataloghi di servizi pubblici, dati pubblici e soluzioni di interoperabilità e utilizzare modelli comuni per descriverli;
- adottare e riusare fonti di informazioni e servizi esterni, laddove utile e fattibile, nello sviluppo dei servizi pubblici.

La sicurezza e privacy sono aspetti che devono essere definiti in pieno accordo con l'"e-Government action plan 2016-2020 della Commissione EU[5]. Per le PA è raccomandato:

- tenendo conto dei requisiti specifici di sicurezza e riservatezza, identificare per ogni servizio le contromisure in conformità con piani di gestione del rischio;
- utilizzare i servizi fiduciari, in base al regolamento in materia di identificazione elettronica e servizi fiduciari, come meccanismi per garantire lo scambio sicuro e protetto dei dati nei servizi pubblici (Regolamento (UE) 2014/910[6]).

Per perseguire gli obiettivi dell'EIF, la Commissione Europea ha individuato i seguenti obblighi per gli stati membri.

- Le PA devono identificare, negoziare e approvare un approccio comune per i componenti di servizi integrati. Ciò è realizzato a diversi livelli amministrativi, in base all'assetto organizzativo di ogni paese, per garantire che piani nazionali e le strategie di interoperabilità siano allineati con l'EIF e, se necessario, adattati e ampliati per tenere conto del contesto e delle esigenze nazionali.
- L'accesso ai servizi e alle informazioni deve essere realizzato mediante specifiche interfacce e condizioni di accesso preventivamente definite (accordi di interoperabilità). Vanno favorite le politiche di riuso dei dati e dei servizi.
- Concordare uno schema comune per interconnettere i componenti dei servizi, nonché predisporre e mantenere l'infrastruttura necessaria per istituire e mantenere i servizi pubblici europei.
- Le PA devono documentare i propri processi lavorativi utilizzando tecniche di modellizzazione comunemente accettate per erogare un servizio pubblico.
- Percepire i dati e le informazioni come un bene pubblico che deve essere adeguatamente prodotto, raccolto, gestito, condiviso, protetto e preservato, elaborando una strategia di gestione delle informazioni al livello più alto possibile per evitare la frammentazione e la duplicazione.
- Promuovere l'istituzione di comunità di settore e intersettoriali che mirino a creare specifiche aperte sulle informazioni condividendo i propri risultati sulle piattaforme nazionali ed europee.
- Utilizzare specifiche aperte, per garantire l'interoperabilità tecnica quando si istituiscono servizi pubblici.

2.3 Il quadro di riferimento attuale

Il Piano triennale per l'informatica nella PA[7] costituisce il quadro di riferimento entro cui si colloca il ModI 2018 all'interno del *Modello strategico di evoluzione del sistema informativo della PA*.

[5] Cf. https://ec.europa.eu/digital-single-market/en/news/communication-eu-egovernment-action-plan-2016-2020-accelerating-digital-transformation
[6] Cf. http://eur-lex.europa.eu/legal-content/IT/TXT/HTML/?uri=CELEX%3A32014R0910&from=EN
[7] Cf. https://pianotriennale-ict.italia.it/

Figura 4 - Piano triennale per l'informatica nella PA

Il modello strategico, pensato per superare l'approccio a «silos», storicamente adottato dalla PA, mira a favorire la realizzazione di un sistema informativo unitario della PA ed è caratterizzato da:

1. Gli strumenti per la generazione e diffusione dei servizi digitali, indicati come **Accesso ai servizi**, che:
   (a) definiscono regole comuni per la progettazione di interfacce, servizi e contenuti, migliorando e rendendo coerente la navigazione e l'esperienza del cittadino e delle imprese;
   (b) facilitano il design, la realizzazione e la diffusione di servizi digitali;
   (c) definiscono linee guida e kit di sviluppo;
   (d) provvedono alla creazione di community di sviluppatori, di designer e di chiunque voglia scambiare informazioni, collaborare e partecipare.
2. Gli **Ecosistemi**, sono i settori o le aree omogenee in cui si svolge l'azione da parte delle PA. Ciascun ecosistema coinvolge enti e organismi pubblici, e soggetti privati che operano nella stessa area di interesse e che a vario titolo svolgono funzioni attive all'interno dell'ecosistema stesso. I soggetti interessati interagiscono per il raggiungimento di obiettivi comuni attraverso
   (a) la condivisione delle esigenze e delle modalità operative;
   (b) la condivisione delle differenti competenze;
   (c) la pianificazione e la realizzazione di progetti ICT.

3. Il **Modello di Interoperabilità,** definisce i meccanismi che facilitano e garantiscono la corretta interazione tra gli attori del sistema (cittadini, imprese e PA), favorendo la condivisione trasparente di dati, informazioni, piattaforme e servizi. Il Modello di Interoperabilità è costituito da linee guida, standard tecnologici e profili di interoperabilità che ciascuna PA dovrà seguire al fine di garantire l'interoperabilità dei propri sistemi con quelli di altri soggetti per l'implementazione complessiva del Sistema informativo della PA.
4. Le **Infrastrutture immateriali** e il **Data & Analytics Framework** (DAF) della PA, che incentivano la centralizzazione e la razionalizzazione dei sistemi per la gestione dei processi e dei dati, riducendo la frammentazione degli interventi. In particolare, le *Infrastrutture immateriali* facilitano, standardizzano e razionalizzano la creazione di servizi ICT e sono composte dalle Piattaforme abilitanti e dai Dati della PA:

   (a) nelle *piattaforme abilitanti* ricadono tutti quei servizi infrastrutturali (eg. servizio di identificazione, servizio di pagamenti, ANPR) che agevolano e riducono i costi per la realizzazione di nuovi servizi uniformando gli strumenti utilizzati dagli utenti finali durante la loro interazione con la PA;

   (b) relativamente ai *dati della PA* si distinguono: le basi di dati di interesse nazionale, gli open data, e i vocabolari controllati.

   Il *Data & Analytics Framework* è un ambiente centralizzato che acquisisce e rende più fruibili i dati pubblici di interesse e ha l'obiettivo *(i)* di rendere più semplice e meno onerosa l'interoperabilità dei dati pubblici tra PA e la distribuzione e standardizzazione dei dati aperti (open data) e *(ii)* di permettere lo studio dei fenomeni sottostanti ai dati pubblici.
5. Le **Infrastrutture fisiche,** che perseguono l'obiettivo di aumentare la sicurezza, ridurre il costo delle infrastrutture tecnologiche e migliorare la qualità dei servizi software della PA, attraverso la razionalizzazione dei data center, l'adozione sistematica del paradigma cloud e lo sviluppo della connettività, con particolare riferimento alla rete Internet nei luoghi pubblici e negli uffici della PA.
6. La **Sicurezza** che comprende:
   - le attività per la regolazione e regolamentazione della cyber-security nella PA per l'*assessment test,*
   - il CERT-PA quale strumento operativo per supportare l'adozione dei corretti livelli di sicurezza presso le PA.
7. La **Gestione del cambiamento** che è una componente definita per far fronte alle necessità di coordinamento; gestione e monitoraggio delle attività funzionali allo sviluppo del Piano.

2.4 Scenario pregresso dell'interoperabilità nella PA

Nell'ottobre 2005 il CNIPA (oggi Agenzia per l'Italia digitale - AgID) ha pubblicato un insieme di documenti che costituiscono il riferimento tecnico per l'interoperabilità fra le PA. Tali documenti delineano il quadro tecnico-implementativo del Sistema pubblico di cooperazione (SPCoop), framework di interoperabilità a livello applicativo[8].

SPCoop ha costituito il modello concettuale ed architetturale della cooperazione applicativa tra differenti Amministrazioni e/o soggetti pubblici italiani. Tale sistema era organizzato in modo da:

- supportare una modalità di erogazione dei servizi articolata per procedimenti istituzionali;
- essere paritetico fra tutti i soggetti cooperanti;
- essere indipendente dagli assetti organizzativi dei soggetti cooperanti;
- lasciare a ciascun soggetto cooperante la responsabilità dei servizi erogati e dei dati forniti;
- garantire a ciascun soggetto autonomia nella gestione dei propri sistemi e nella definizione ed attuazione delle politiche di sicurezza del proprio sistema informativo;
- lasciare a ciascun soggetto la responsabilità delle autorizzazioni per l'accesso ai propri dati e/o servizi.

In sintesi, alla base di SPCoop vi erano i seguenti principi:

[8] Cf. http://www.agid.gov.it/agenda-digitale/infrastrutture-architetture/sistema-pubblico-connettivita/cooperazione-applicativa

1. *cooperazione tra amministrazioni* attraverso la erogazione e fruizione di servizi offerti tramite un unico elemento logico denominato *Porta di Dominio;*
2. *ambito di responsabilità* delle singole Amministrazioni dei servizi erogati che costituiscono il *Dominio di servizi applicativi* della stessa Amministrazione;
3. *accordi di servizio* quale rappresentazione formale della cooperazione tra erogatore/i e fruitore/i costituiti sulla base di un fondamento normativo;
4. *tecnologie di cooperazione:* i servizi erano erogati come web service basati sugli standard che in quel momento erano consolidati ed in uso (SOAP, WSDL, UDDI).

Con l'obiettivo di assicurare agli utenti di avere una visione integrata dei servizi di ogni PA, le tematiche coperte da SPCoop sono state tutte quelle che interessano l'interoperabilità dei sistemi a diversi livelli, ovvero:

- interoperabilità applicativa;
- catalogazione dei servizi;
- semantica dei dati e dei servizi;
- identità digitale.

Lo scenario normativo di SPCoop è quello inquadrato dal DPCM 1 aprile 2008, recante regole tecniche e di sicurezza del Sistema pubblico di connettività (SPC), di cui SPCoop era un componente fondamentale, poi compiutamente delineato sul piano tecnico-implementativo da una suite di linee guida di seguito richiamate:

- Interoperabilità applicativa
  - Specifiche della busta di e-gov
  - Specifiche della porta di dominio
  - Linee guida busta di e-gov
  - Qualificazione della porta di dominio
  - Qualificazione porta di dominio con concorso delle regioni
- Catalogazione dei servizi
  - Specifiche dell'accordo di servizio
  - Specifiche del Registro SICA
  - Raccomandazioni stesura accordi di servizio
- Semantica dei dati e dei servizi
  - Nomenclatura e semantica
- Identità digitale
  - GFID - Gestione federata delle identità digitali

In particolare SPCoop prevedeva:

- Tutti i servizi applicativi di una PA erano offerti attraverso un unico elemento denominato *Porta di Dominio*, che svolgeva funzioni di proxy e dispatcher assicurando l'implementazione del protocollo applicativo denominato *Busta e-Gov*, un'estensione dello standard SOAP.
- I servizi infrastrutturali per la gestione di tutti gli aspetti legati agli *accordi di servizio*, nel loro insieme denominati *Servizi SICA*, prevedevano:
  - *Servizi di Registro*: la componente, realizzata a partire dallo standard UDDI, entro cui erano registrati gli Accordi di Servizio organizzati in modo distribuito prevedendo due livelli, ovvero Generale, che contiene la totalità degli *accordi di servizio*, e Secondario, contenente delle viste definite secondo differenti criteri;
  - *Catalogo degli Schemi/Ontologie*, che offre gli strumenti per ragionare sulla semantica dei servizi e delle informazioni da essi veicolati;

  - *Servizi di Sicurezza* assicuravano le funzionalità per la qualificazione degli elementi del sistema e garantire gli opportuni requisiti di autenticità, riservatezza, integrità, non ripudio e tracciabilità dei messaggi scambiati.

Il tempo trascorso dalla definizione del modello e il mutato quadro tecnico, organizzativo e normativo rendono necessario l'aggiornamento del modello, obiettivo appunto della presente iniziativa, come anticipato nel 2017 attraverso la Determinazione 219/2017 - Linee guida per transitare al nuovo modello di interoperabilità[9].

L'esperienza maturata con SPCoop, di seguito sintetizzata, deve essere considerata nella definizione del ModI 2018.

Cosa ha funzionato

- La definizione di un quadro comune per l'implementazione dei meccanismi di interoperabilità tra i sistemi delle Pubbliche Amministrazioni permette di orientare gli sforzi per la realizzazione di servizi pubblici sulla logica propria degli stessi.
- Il coordinamento, anche delegato ad organi intermedi quali elementi di aggregazione di un insieme omogeneo di Amministrazioni, permette di favorire l'applicazione del modello condiviso.
- Il sistema di gestione federata delle identità digitali, nonostante si ponesse come un elemento fortemente innovativo, è stato utilizzato a livello regionale e ha consentito di disegnare su tali basi tecniche il futuro SPID.

Cosa deve essere cambiato

- Le tecnologie e gli standard utilizzati dal modello SPCoop richiedono un consistente aggiornamento in considerazione delle innovazioni intervenute in tali ambiti.
- È necessario un modello di governance che permetta di gestire le specificità dei singoli domini applicativi determinati dalle caratteristiche delle amministrazioni e dei soggetti terzi coinvolti.

Cosa deve essere abbandonato

- L'adozione di un'unica modalità per attuare l'interoperabilità dei sistemi non permette di considerare la molteplicità e la specificità delle esigenze di scambio tra le Pubbliche Amministrazioni e di queste con i cittadini e le imprese.
- La necessità di componenti infrastrutturali disegnati per la sola Pubblica Amministrazione italiana (come Porta di Dominio e Registro SICA) determina che la spesa per il loro sviluppo ed evoluzione sia totalmente a carico della Pubblica Amministrazione.

2.5 Principi del nuovo modello di interoperabilità

2.5.1 Interazioni

L'ambito di applicazione del Modello di Interoperabilità 2018 comprende i tre tipi di interazioni previsti nell'EIF. Le interazioni prevedono che i soggetti coinvolti svolgano alternativamente la funzione di **erogatore** di servizio, nel caso del soggetto che mette a disposizione API o servizio utilizzati da altri, e la funzione di **fruitore**, nel caso invece del soggetto che utilizza le API o servizi messi a disposizione da altro soggetto.

[9] Cf. http://www.agid.gov.it/sites/default/files/upload_avvisi/linee_guida_passaggio_nuovo_modello_interoperabilita.pdf

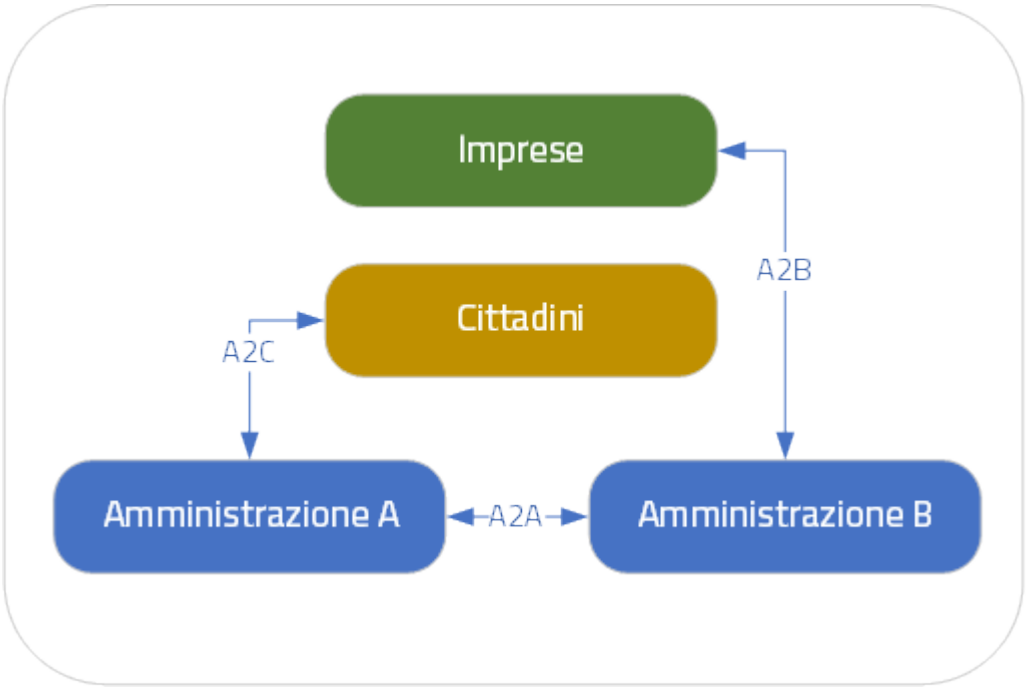


Figura 5 - Ambito di applicazione del modello di interoperabilità

I soggetti fruitori possono utilizzare le API/servizi[10] esposti dall'erogatore attraverso:

- una soluzione software attivata da un attore umano (*user agent/human*);
- un sistema applicativo *automatico*[11] (*server/machine*), anche allo scopo di definire nuovi servizi a valore aggiunto .

In considerazione di quanto sopra si individuano le seguenti possibili interazioni:

1. A2A in modalità *human-to-machine*;
2. A2A in modalità *machine-to-machine*;
3. A2B in modalità *human-to-machine*;
4. A2B in modalità *machine-to-machine*;
5. A2C in modalità *human-to-machine*.

2.5.2 Paradigmi di cooperazione

In generale, nell'integrazione dei sistemi software si individuano principalmente le seguenti tre casistiche che il modello di interoperabilità deve tener presente:

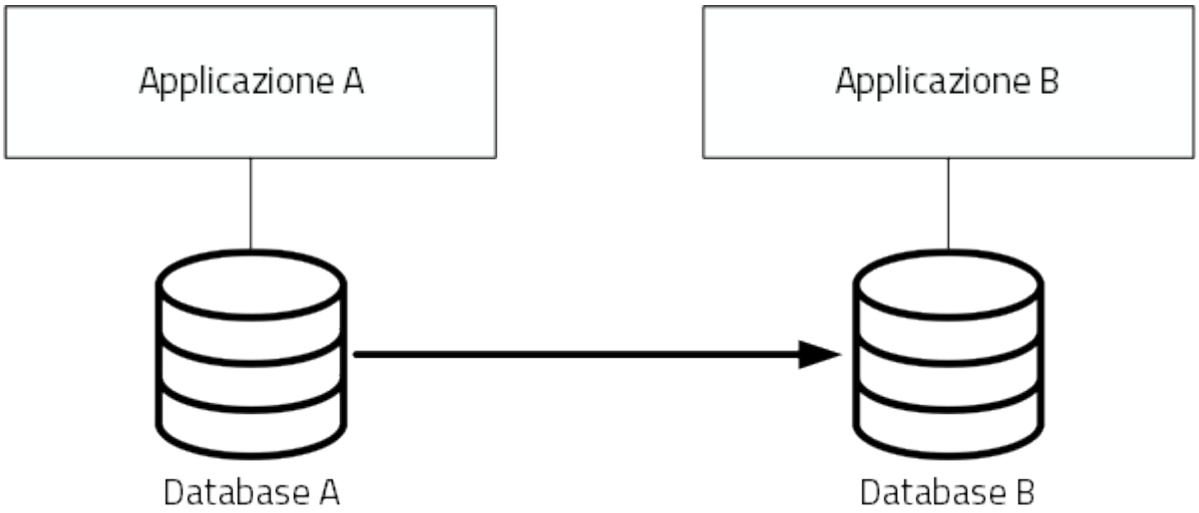
- **Condivisione di dati**: l'obiettivo è quello di tenere allineati i dati di uno o più sistemi; le applicazioni software che gestiscono (creano, aggiornano, leggono ed eventualmente cancellano[12]) tali dati, sono logicamente e fisicamente indipendenti. I processi che sovraintendono le applicazioni sono separati ed indipendenti. Il caso tipico è quello di un'Amministrazione, o soggetto privato, che per dare seguito ad una sua attività ha necessità di accesso ai dati posseduti dall'Amministrazione B, titolare degli stessi, senza che sia richiesto all'Amministrazione B nessuna elaborazione sui dati. Ad esempio, B è il Ministero delle Finanze che ha i dati del codice fiscale di ogni cittadino, ed A è un qualsiasi altro soggetto (pubblico o privato) che

[10] Con abuso di nomenclatura, ma intuitivamente chiaro, si intende nel presente documento servizio e API come sinonimo, ad indicare una componente software, esposta sul Web, che funge da servente e può essere utilizzata da client. In modo rigoroso, sia SPCoop che il ModI 2018 prevedono l'esposizione da parte di una PA di un'API accessibile sul Web come modalità base di interoperabilità e scambio di dati/informazioni, tale API permette la fruizione di un servizio offerto dalla PA stessa. La tecnologia web service è una particolare modalità con cui realizzare API che siano accessibili su Internet/intranet, da cui il termine Web. Tali concetti verranno ulteriormente approfonditi nel Modello di Interoperabilità 2018.

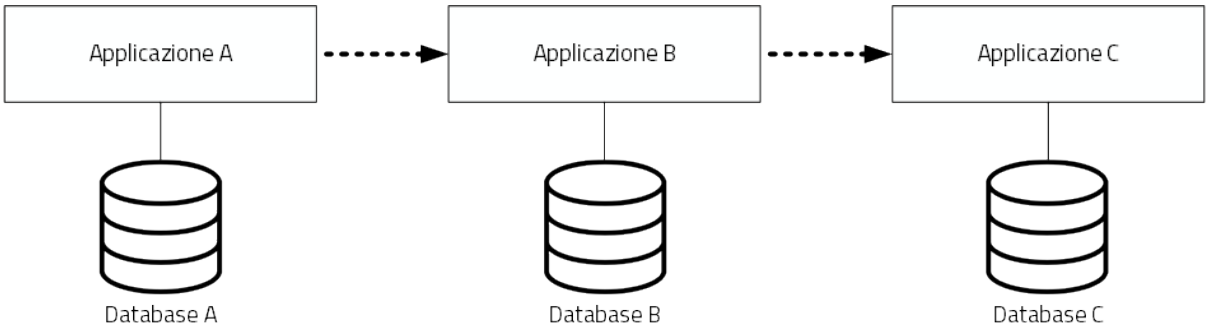
[11] Quindi non attivato da un utente umano, anche impropriamente detto *enterprise* in taluni contesti.

[12] Cf. le cosiddette operazioni CRUD - Create, Read, Update, Delete

all'interno della propria applicazione ha necessità di verificare la correttezza dei codici fiscali del proprio database, per poi utilizzarli in proprie elaborazioni.



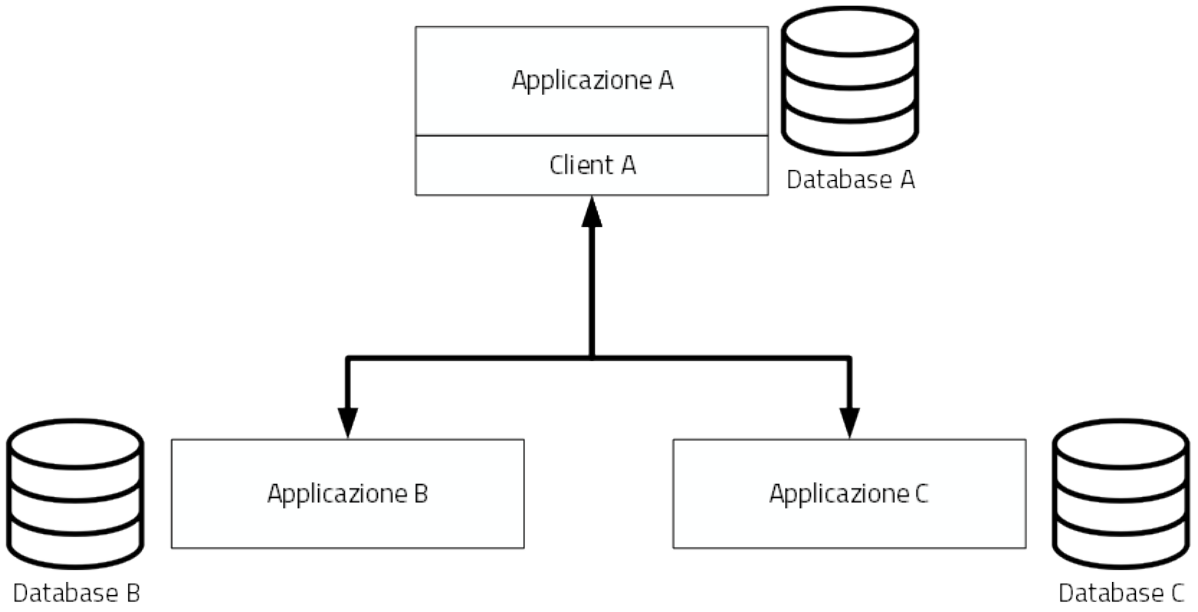
- **Notifica inter-PA**: in questo caso un'applicazione in un soggetto scatena un evento compie un'operazione che deve essere propagata sincronizzata con altre applicazioni di altri soggetti. Le applicazioni sono fisicamente indipendenti ma non logicamente, in quanto esiste un processo inter-organizzativo che sovraintende a tutte le organizzazioni che devono cooperare[13]. Il caso tipico è quello in cui il presentarsi di un evento all'interno di un'Amministrazione A debba essere comunicato ad altri soggetti B e C, pubblici e privati, che devono dare seguito a proprie procedure interne in relazione all'evento stesso, per vincoli normativi, ecc. Ad esempio, la registrazione di una nascita in un Comune è un evento che deve essere propagato all'Agenzia delle Entrate, per il rilascio di un nuovo codice fiscale, all'AUSL di riferimento per l'iscrizione al Servizio Sanitario Nazionale, ecc.



- **Composizione inter-PA**: in questo caso un insieme di applicazioni comunicano, anche in maniera bidirezionale, al fine di comporre una nuova logica applicativa ottenuta dalla loro interazione, ed erogare questa a sua volta come servizio a valore aggiunto. Talvolta questa nuova logica viene indicata come servizio/applicazione composito/a (o composto/a). Come nel caso precedente, esiste un processo inter-organizzativo che sovraintende a tutte le organizzazioni che vengono composte. Il caso tipico, nel mondo commerciale, è quello di un servizio che definite delle date ed una destinazione, propone all'utente voli aerei, hotel e noleggio auto, ecc, andando appunto a comporre servizi per la bigliettazione aerea, prenotazione alberghiera, noleggio auto, ecc. Nel caso della PA, un caso è una conferenza di servizi telematica[14] in cui diverse Amministrazioni compongono un un nuovo servizio per dare seguito ad una istanza di un cittadino o di un'impresa.

[13] Nel caso della PA, questo processo inter-organizzativo corrisponde al concetto di macro-processo o di processo inter-amministrazione: M Mecella, C Batini (2001), Enabling italian e-government through a cooperative architecture. IEEE Computer 34 (2), pp. 40-45.

[14] La conferenza di servizi, cf. http://www.italiasemplice.gov.it/conferenza/guida-alle-novita-della-conferenza-di-servizi/ , è l'istituto che facilita l'acquisizione da parte della PA di autorizzazioni, atti, licenze, permessi e nulla-osta o di altri elementi comunque denominati, finalizzati all'emissione di un provvedimento amministrativo, coordinando differenti soggetti coinvolti. La conferenza semplificata in modalità sincrona è l'esempio di composizione di servizi, mentre la conferenza semplificata in modalità asincrona costituisce un altro caso della modalità precedente (notifica inter-PA).



È importante analizzare le analogie e differenze con il caso precedente: nel caso della notifica inter-PA, c'è una relazione peer-to-peer tra i vari soggetti coinvolti, e si parla di *coreografia* tra le applicazioni coinvolte[15]. Nel caso invece della composizione, una delle applicazioni ha un ruolo di *orchestrazione* nei confronti delle altre, e quindi c'è una relazione uno (l'orchestratore, che fa da *master*) a molti (le applicazioni orchestrate, che sono *slave*).

In entrambe le situazioni, esiste a livello concettuale (dovuto a norme, accordi, ecc.) un processo inter-organizzativo che sovraintende alle varie applicazioni, e l'espletamento del quale è l'obiettivo del servizio composto offerto.

La differenza tra i due casi risiede quindi nel grado di autonomia che i soggetti che concorrono al processo inter-organizzativo mantengono: se si sceglie un approccio completamente decentralizzato, si è nel caso notifica inter-PA, se si opta per un approccio per cui uno dei soggetti prende in carico la fornitura del servizio finale composto a valore aggiunto, allora si è nel caso composizione inter-PA.

2.5.3 Incrementalità del modello

In base alle considerazioni precedenti, il Modello di Interoperabilità si concretizza nella definizione, lo sviluppo, il miglioramento, la resa operativa, il mantenimento e la promozione di servizi, strumenti, norme tecniche e specifiche per l'interoperabilità delle soluzioni ICT basata su un'architettura modulare che include componenti interconnessi con l'ausilio di infrastrutture comuni. Questo modello, al fine di evitare le problematiche di possibile obsolescenza, e fronteggiare la necessità di continui aggiornamenti, si estrinsecherà concretamente in rilasci successivi e cadenzati nel tempo, di una serie di 5 documenti, in particolare:

1. **Presentazione del Modello di Interoperabilità 2018**, che è il documento attuale, rilasciato nella prima versione a maggio 2018.
2. **Tecnologie ed approcci all'Integrazione ed Interoperabilità**, che nella prima versione (maggio 2018) viene rilasciato contestualmente al presente documento. Ha come oggetto l'individuazione delle possibili tecnologie ed approcci che possono essere utilizzati dalle PA.
3. **Pattern e Profili di Interoperabilità**, che fornirà indicazioni concrete, a livello tecnico, su differenti modalità operative per realizzare l'interoperabilità, tenendo conto delle possibili tecnologie ed approcci disponibili. La prima release di questo documento è prevista per l'estate 2018. Il Modello introduce il concetto di profilo di interoperabilità e come esso possa essere evoluto nel tempo; si introduce anche il concetto di pattern di interoperabilità. Infine questo documento si occuperà di discutere anche l'aspetto della QoS - Quality of Service e degli SLA - Service Level Agreement.

[15] Approfondimenti sui concetti di orchestrazione e coreografia possono essere trovati in: https://stackoverflow.com/questions/4127241/orchestration-vs-choreography (C Peltz (2003), Web Services Orchestration and Choreography. IEEE, Computer 36(10), pp. 46-52 e R M Dijkman, M Dumas (2004), Service-Oriented Design: A Multi-Viewpoint Approach. Int. J. Cooperative Inf. Syst. 13(4), pp. 337-368)

4. **Governance del Modello di Interoperabilità**, che presenterà compiutamente la governance dell'intero modello e le sue modalità di evoluzione, ed è previsto in una prima versione per l'estate 2018.
5. **Registri e Cataloghi**, che si occuperà di definire le linee guida per i registri e cataloghi necessari a supportare il modello stesso. Anche per questo documento è prevista una prima versione per l'estate 2018.

Gli interventi mirano, in coordinamento con le altre azioni presenti nel Piano Triennale per l'Informatica nella PA, a:

- definire e attuare specifiche comuni sui termini e le condizioni per gestire e accedere ai *base register*;
- attuare e promuovere modelli comuni per descrivere e classificare i servizi pubblici;
- individuare misure volte a creare sicurezza, tracciabilità e SLA - Service Level Agreement nell'erogazione dei servizi;
- analizzare i dati contenuti e i sistemi esistenti per l'informatizzazione delle PA;
- individuare gli ostacoli al reciproco riconoscimento, sviluppare mappature e sostenere gli sforzi di armonizzazione.

Gli **standard tecnologici** adottati, in particolare per i web service REST e SOAP, rispecchiano l'attuale stato di evoluzione delle tecnologie ed il loro utilizzo è consolidato nelle pratiche adottate nell'ambito dell'interoperabilità dei sistemi informativi.

2.5.4 Profili e pattern di interoperabilità

Il nuovo modello introduce i concetti di **caso d'uso**, **pattern** e **profilo di interoperabilità**.

Un caso d'uso di interoperabilità è la formalizzazione di una specifica esigenza di interoperabilità, che si manifesta frequentemente tra PA, o che può manifestarsi in particolari contesti applicativi. Tale necessità viene descritta mostrandone il contesto di applicazione, i problemi progettuali che ne derivano, i possibili schemi di soluzione e le implicazioni di ognuno di essi.

Ogni caso d'uso può essere risolto in vari modi, ognuno di questi schemi verrà indicato come pattern di interoperabilità. Esso fornisce una serie di linee guida per l'implementazione e l'interoperabilità che raccomandano come utilizzare una specifica tecnologia od approccio, e permette eventualmente di risolvere eventuali ambiguità/punti non adeguamente definiti in alcune tecnologie possibili con cui le PA possono interoperare.

Un profilo infine, in maniera trasversale rispetto ai casi d'uso ed ai pattern, risolve le diverse opzionalità o aspetti non adeguatamente specificati dagli standard tecnologici.

L'applicazione dei casi d'uso, pattern e profili agevola l'azione nello sviluppo e nella distribuzione di API/servizi. Il nuovo Modello proporrà un catalogo *di casi d'uso*, *profili* e *pattern di interoperabilità* messi a disposizione delle PA, popolato maniera incrementale sulla base di esigenze individuate dall'Agenzia per l'Italia Digitale anche a fronte dell'evidenza di nuovi bisogni per le PA.

Ogni PA che offre un'API/servizio deve, nel nuovo modello, offrire un insieme di artefatti che lo accompagnano, in particolare:

- meccanismi di controllo delle versioni;
- documentazione coordinata alla versione;
- Software Development Kit - SDK - per l'interfacciamento e un ambiente di test (in analogia a quanto avviene per alcuni servizi commerciali di largo utilizzo in applicazioni Web[16]);
- dichiarazione sulla qualità del servizio che si impegna a rispettare. In questo secondo caso, deve anche definire le modalità di misurazione e deve offrire un'opportuna modalità di monitoraggio, che i fruitori possono sfruttare per la verifica.

Nello scambio informativo tra PA mediante API/servizi, le soluzioni che verranno adottate devono assicurare: *(i)* autenticità, *(ii)* integrità e *(iii)* non ripudio. In questo contesto il Regolamento (UE) 2014/910 fornisce una base

[16] Ad es., Paypal, cf. https://developer.paypal.com/ , offre SDK ed un servizio di prova, cosiddetta sandbox, che permette agli sviluppatori che si vogliono integrare con Paypal di provare le interazioni prima di rilasciare i propri sistemi.

normativa comune per le interazioni elettroniche sicure fra cittadini, imprese e PA; le soluzioni software conformi al Modello di Interoperabilità devono applicare i principi indicati in esso.

2.5.5 Catalogo delle API/servizi

Il Modello di Interoperabilità prevede la presenza del *Catalogo* quale componente che assicura alle parti coinvolte nel rapporto erogazione/fruizione la consapevolezza sulle interfacce e i livelli di servizio dichiarati.

La presenza del Catalogo è funzionale a:

- facilitare l'interoperabilità tra le PA e tra queste e i soggetti privati interessati;
- contenere la spesa della PA riducendo la replicazione di API/servizi;
- manifestare gli impegni dei fornitori o erogatori di API/servizi.

La realizzazione del Catalogo deve, fatti salvi i principi comuni che saranno emanati dall'Agenzia per l'Italia Digitale al fine di permettere una normalizzazione a livello nazionale, tener conto della:

- specificità dei territori e dei diversi ambiti entro cui la PA opera che potrà determinare la specializzazione del catalogo, prevedendo contenuti con un livello di aggregazione territoriale (eg. su base regionale) e/o relativamente agli ambiti tematici entro cui opera la PA (eg. giustizia). Tale scelta è ulteriormente giustificata dalla opportunità di favorire momenti di aggregazione di soggetti omogenei.
- esigenza di assicurare la governance del Catalogo, quale presupposto per garantire una semantica univoca e condivisa, per evitare ridondanze e/o sovrapposizioni in termini di competenze e contenuti.
- esigenza di assicurare una descrizione formale delle API/servizi che, attraverso l'utilizzo di *interfacce description language*, permetta di descrivere le interfacce degli stessi in maniera indipendente dal linguaggio di programmazione adottato dall'erogatore e dai fruitori degli stessi. L'attuale stato di evoluzione degli standard tecnologici indicati in precedenza determina la scelta di *WSDL* per i *web service SOAP* e *OpenAPI v3* per i *web service REST.*

2.5.6 Governance del modello

L'Agenzia per l'Italia Digitale è responsabile delle attività di *governance* del ModI 2018 con l'obiettivo di definire, condividere ed assicurare l'aggiornamento continuo dei seguenti aspetti:

- l'*insieme delle tecnologie* che abilitano l'interoperabilità tra le PA, e tra queste e cittadini ed imprese;
- i *casi d'uso di interoperabilità*;
- i *pattern di interoperabilità*;
- i *profili di interoperabilità*;
- il *catalogo* dei servizi resi disponibili dalle PA.

I progetti che realizzano gli Ecosistemi, previsti nel Piano Triennale per l'Informatica nella PA, si basano sul Modello di Interoperabilità, e possono determinare l'esigenza di nuovi *casi d'uso*, *pattern* e *profili di interoperabilità* che verranno definiti con un approccio collaborativo.

Nel precedente SPCoop, l'uso di servizi/API richiedeva un accordo tra amministrazioni anche tramite la firma di convenzioni bilaterali. Questo non sarà più necessario nel nuovo modello, in cui l'adesione si estrinsicherà nell'atto di registrazione da parte della PA di un'API/servizio nel catalogo. In ottemperanza al principio «once-only» definito nell'EU eGovernment Action Plan 2016-2020[17], l'erogatore si impegna a fornire l'accesso alle proprie API/servizi a qualunque soggetto registrato ne faccia richiesta[18]. Gli erogatori devono descrivere le loro API/servizi classificando le informazioni scambiate ove possibile collegandole ai vocabolari controllati e a concetti semantici predefiniti, utili anche a determinare l'impatto rispetto ai regolamenti in tema privacy e GDPR, e applicando tag di categoria. Il Catalogo può facilitare questo processo attraverso opportune euristiche.

[17] Cf. EU eGovernment Action Plan 2016-2020, https://ec.europa.eu/digital-single-market/en/news/communication-eu-egovernment-action-plan-2016-2020-accelerating-digital-transformation]

[18] Cf. Codice dell'Amministrazione Digitale Capo 1 Sez. 2 Art. 3 http://cad.readthedocs.io/it/v2017-12-13/_rst/capo1_sezione2_art3.html

In virtù degli articoli 12 e 14 del Codice dell'Amministrazione Digitale, AgID è formalmente incaricata della gestione di tutto il catalogo e di garantire il rispetto delle regole suddette e per farlo si avvale della collaborazione di alcuni enti, che vengono indicati come Capofila.

Gli enti Capofila si proporranno per eseguire questo compito su porzioni del catalogo; ci saranno enti che si occupano della gestione di aree geografiche e, allo stesso tempo, enti che si occupano della gestione di particolari aree tematiche.

In prima istanza si prevede che gli enti Capofila possano essere:

- a livello territoriale, le Regioni (e.g., la Regione per conto delle ASL regionali)
- a livello di ecosistema, gli enti individuati dai GdL descritti nel Piano Triennale al capitolo 6 Ecosistemi.

A tal fine, sul fronte delle aree tematiche il Piano Triennale 2017-2019 introduce:

- gli Ecosistemi[19], settori o aree di intervento in cui si svolge l'azione delle PA, che raggruppano i vari enti per aree tematiche;
- i Gruppi di Lavoro[20] che, all'interno degli Ecosistemi, indirizzano il vero e proprio lavoro di standardizzazione coinvolgendo sia tecnici che esperti dei rispettivi domini applicativi.

I Gruppi di Lavoro devono formalizzare le specifiche di dettaglio, attraverso il meccanismo dei profili e dei pattern di interoperabilità, e revisionare periodicamente le specifiche rilasciate.

Il nuovo Modello opera in assenza di elementi centralizzati che mediano l'interazione tra le entità comunicanti (erogatore e fruitore del servizio), pur prevedendo la presenza di un catalogo dei servizi disponibili allo scopo di permettere a tutti i soggetti interessati, pubblici e privati, di acquisire conoscenza dei servizi disponibili e delle loro modalità di erogazione/fruizione.

L'Agenzia per l'Italia Digitale ha il ruolo di:

- recepire le esigenze, anche applicative, delle PA, astrarre tali esigenze ed eventualmente formalizzare i casi d'uso ed i pattern di interoperabilità;
- coordinare il processo di definizione dei profili di interoperabilità;
- rendere disponibile il catalogo, attraverso un'interfaccia di accesso unica per permettere a tutti i soggetti interessati, pubblici e privati, di assumere consapevolezza dei servizi disponibili;
- verificare il rispetto delle regole del Modello di Interoperabilità, quale condizione di accesso al catalogo, e controllare con continuità il rispetto dei requisiti per l'iscrizione al catalogo.

[19] Cf. http://pianotriennale-ict.readthedocs.io/it/latest/doc/06_ecosistemi.html
[20] Cf. http://pianotriennale-ict.readthedocs.io/it/latest/doc/06_ecosistemi.html#linee-di-azione

CAPITOLO 3

Tecnologie ed Approcci all'Integrazione ed Interoperabilità

Il secondo documento del Modello di Interoperabilità 2018, così come introdotto nella Visione generale, funge da guida alle possibili tecnologie che possono essere considerate dalle PA per l'integrazione e l'interoperabilità. Data la veloce evoluzione tecnologica, questo documento verrà continuamente aggiornato ed approfondito, e costituisce il riferimento per «Pattern e Profili di Interoperabilità». Questa versione si focalizza prevalentemente sugli approcci e tecnologie basati su SOAP, REST e Message Broker, ed accenna ad altre possibili scelte che in futuro potrebbero essere valutate.

3.1 Introduzione alle interfacce di servizio

3.1.1 Il concetto di servizio

I servizi sono sempre più rilevanti nella nostra vita e nei paesi di tutto il mondo. Il concetto di servizio copre un ampio spettro di aspetti nelle relazioni moderne tra amministrazioni pubbliche, fornitori privati e utenti finali.

Introduciamo il concetto di **servizio**[1] così come intuitivamente percepito nella vita quotidiana. Interagiamo ogni giorno con le persone e le imprese per soddisfare i nostri bisogni, facendo uso di transazioni, ad esempio di tipo economico in cui, dato un pagamento, possiamo acquisire un bene, o utilizziamo un bene che non è nostro, per raggiungere un obiettivo. Nel secondo caso stiamo parlando dell'uso di un servizio. Ad esempio, quando compriamo un biglietto ferroviario Milano-Roma, stiamo utilizzando un bene non nostro (il treno) per soddisfare la nostra necessità di mobilità. Arrivati a Roma, il nostro bisogno è soddisfatto e nulla ci rimane per l'uso, in termini di possesso del bene (il treno) utilizzato.

Un servizio consiste quindi in un'attività, o in una serie di attività, di natura più o meno intangibile, che si svolgono in uno scambio tra un fornitore e un cliente, in cui l'oggetto della transazione è un bene immateriale.

I servizi sono espletati in un sistema di servizi. Un ecosistema di servizi è l'insieme delle regole, delle componenti sociali, delle organizzazioni, dei processi, delle risorse umane, dei materiali e delle tecnologie che nella società coincidono con la produzione e l'uso dei servizi. Un sistema di servizi è caratterizzato da tre tipi di utenti finali: cittadini, imprese e l'ambiente circostante. Vari tipi di produttori forniscono servizi; e per questo verranno indicati più in generale come *erogatori* di servizi.

Per fornire servizi, gli erogatori devono eseguire una serie di attività, indicate come *processi di servizio*. Un processo di servizio è un insieme di attività la cui esecuzione, in base a un determinato flusso di controllo, produce in uscita un servizio fornito a un utente che ne ha bisogno. Se, ad esempio, vogliamo prenotare e beneficiare di un

[1] La trattazione si basa in parte su C. Batini, M. Castelli, M. Comerio, M. Cremaschi, L. Iaquinta, A. Torsello, G. Viscusi (2015): The Smart methodology for the life cycle of services. Cf. https://boa.unimib.it/retrieve/handle/10281/98632/144883/SmartBook-0315.pdf

viaggio in treno, la prenotazione e l'acquisizione di un biglietto sono la prima fase del processo corrispondente, che avviene presso un'agenzia di viaggi o, sempre più spesso, attraverso Internet. La seconda fase è il momento del viaggio, in cui sono coinvolte risorse quali il treno che ci trasporta, il materiale rotabile, il personale a bordo, il personale nelle stazioni. Nell'esecuzione del processo di servizio, ci sono delle interazioni tra il cliente e l'erogatore; tali interazioni, dal punto di vista del cliente, sono percepite come *operazioni* a sua disposizione, con cui è possibile richiedere e modificare aspetti specifici del servizio. Nell'esempio del servizio di trasporto ferroviario, le operazioni sono quelle che permettono al cliente di acquisire un biglietto, modificare la prenotazione (modifica del posto, del giorno, ecc.), accedere al treno, ecc. La *granularità* delle operazioni offerte al cliente dipende dall'erogatore dal processo di servizio che viene messo in atto per espletare il servizio.

La PA, in tutto il mondo, è fornitrice di una vasta gamma di servizi. Le differenze tra PA ed erogatori privati sono molteplici. Soprattutto, la fornitura di servizi è un obbligo legale per la PA.

Ad esempio, in Italia, sulla base di una legge emanata nel 1950, i Comuni sono responsabili dei registri della popolazione residente. Pertanto, se un cittadino ha bisogno di un certificato di residenza, deve andare al comune, che è responsabile per la validità e la correttezza delle informazioni che contiene il certificato. I fornitori privati forniscono servizi in base alla convenienza economica. Secondo il contesto, i prezzi dei servizi sono generalmente regolati nella PA da leggi, decreti o direttive che mantengono nella società una forma di equità sociale. A volte la PA fornisce servizi gratuitamente, mentre di fatto li finanzia attraverso le tasse. I fornitori privati forniscono servizi a pagamento e le entrate sono la ragione della loro attività come azienda. Infine, la PA nella pianificazione della produzione e della fornitura di servizi si ispira a criteri che tengono conto delle esigenze delle comunità, ed è quindi ispirata da una visione sociale, mentre le società private sono indirizzate dal mercato.

Il concetto di servizio include una grande quantità di aspetti. Di conseguenza, è necessario determinare il perimetro di osservazione del concetto di servizio nella PA, il dominio considerato nel Modello di Interoperabilità. Delineiamo nel seguito diverse classificazioni di servizi.

1. Classificazione in base alla natura del fornitore di servizi. In questo caso, abbiamo:

   (a) servizi amministrativi (o abilitanti), la cui fornitura non ha carattere discrezionale da parte della PA, in quanto derivano da procedimenti amministrativi definiti dalla legge;

   (b) servizi che chiamiamo orientati al mercato o facilitanti, che la PA può decidere o meno di fornire, in base alla presenza di un obbligo procedurale, e che sono più spesso erogati da fornitori privati del mercato dei servizi.

   I servizi amministrativi forniti dalla PA sono di primario interesse, ma è anche importante attirare l'attenzione su servizi orientati al mercato, che sono parte delle aspettative e dei bisogni degli utenti e potrebbero essere forniti da soggetti pubblici o privati.

2. Classificazione in base alla natura finale del servizio prodotto. In questo caso, abbiamo:

   (a) servizi che rispondono alle esigenze degli utenti che modificano il loro stato. Verranno indicati come servizi che modificano lo stato (dell'utente e/o del mondo) o semplicemente servizi;

   (b) servizi il cui scopo è quello di fornire informazioni e/o conoscenze che l'utente non possiede e che sono utili per un'attività operativa o un processo decisionale. Verranno indicati come servizi informativi o semplicemente informazioni.

   Un esempio della prima categoria è la fornitura di una licenza commerciale che consente a un'azienda di vendere la propria merce; questo servizio modifica lo stato dell'azienda perché consente una nuova attività commerciale. Un esempio di servizio informativo è l'informazione resa disponibile sugli orari di apertura di un laboratorio, che non cambia lo stato del soggetto che ha richiesto l'informazione, ma gli dà la possibilità di intraprendere un'azione o di prendere una decisione per andarci.

3. Classificazione in base al consumatore. In questo caso, possiamo distinguere tra:

   (a) servizi esterni, quando il servizio è focalizzato al di fuori della PA, verso la comunità di cittadini e imprese;

   (b) servizi interni, quando il servizio è dedicato agli utenti interni all'organizzazione erogatrice, sia essa PA che erogatore di servizi privato.

Oltre ai servizi, sappiamo che altri tipi di oggetti coinvolti nelle transazioni sono beni e informazioni; abbiamo visto che l'informazione può essere vista come un tipo specifico di servizio, quindi non c'è una chiara distinzione

tra servizi e informazioni, nel senso che entrambi i tipi di concetti appartengono a un concetto di servizio più generale. Allo stesso modo, tra prodotti e servizi non è possibile distinguere una linea precisa.

Consideriamo il caso in cui dobbiamo viaggiare in India, e il nostro obiettivo immediato è ottenere un visto per l'India; contattiamo due agenzie che, quando richiesto per le condizioni che applicano per fornire il visto, rispondono come mostrato nella tabella seguente:

Obiettivo del servizio	**Agenzia 1**	**Agenzia 2**
Necessità di un visto per andare in India	*nella nostra agenzia rilasciamo il visto in 7 giorni, al costo di € 30, e la penalità per un giorno di ritardo è di € 2*	*nella nostra agenzia facciamo il possibile per rilasciare il visto in 2 settimane, il costo è di € 20*

Guardando le due specifiche, il nostro obiettivo ora è fornire loro una struttura, distinguendo le diverse parti che hanno ruoli diversi.

Possiamo identificare i tipi di proprietà:

- proprietà funzionale, che esprime «cosa» otteniamo dal servizio;
- qualità del servizio, riferito a caratteristiche (ad es., tempo di consegna) che specificano vantaggi o utilità percepita, associati al servizio;
- proprietà non funzionali, esprimendo «come» il servizio ci viene consegnato.

La tabella seguente mostra la classificazione delle proprietà applicate all'esempio di cui sopra:

Tipo di proprietà	**Agenzia 1**	**Agenzia 2**
funzionale	rilascio del visto	rilascio del visto
qualità del servizio	in 7 giorni	il possibile in 2 settimane (*best effort*)
altra non funzionale	prezzo : € 30 penale : € 2 / giorno ritardo	prezzo : € 20

Le proprietà funzionali di un servizio descrivono cosa fa il servizio per il cliente. Una proprietà funzionale consente un cambiamento di stato del mondo reale, coerentemente con gli obiettivi espressi dal cliente. Le proprietà non funzionali di un servizio definiscono il modo in cui il servizio esegue le proprietà funzionali. Lo schema dei dati del servizio (talvolta chiamato *information model*) descrive i tipi di dati che rappresentano lo stato del mondo reale quando il servizio viene eseguito. I servizi possono essere visti come cambiamenti di stato del mondo reale ad un alto livello di astrazione, quindi un modo di descrivere i tipi di dati coinvolti in tali cambiamenti sono gli schemi concettuali, ad esempio diagrammi Entity Relationship o UML Class Diagram.

Quindi l'esempio mostra che i servizi possono essere descritti in termini delle seguenti caratteristiche:

1. un nome;
2. un insieme di proprietà funzionali, le operazioni appunto discusse in precedenza;
3. un insieme di proprietà non funzionali, tra cui quelle relative alla qualità del servizio;
4. uno schema di dati di servizio.

Finora abbiamo introdotto un modello che ci consente di descrivere un singolo servizio. Nei nostri eventi della vita quotidiana, per raggiungere i nostri obiettivi, abbiamo bisogno di invocare un numero elevato di servizi, facendo riferimento a un numero elevato di proprietà funzionali (operazioni). Consideriamo cosa accade in corrispondenza a un cambio di indirizzo di abitazione. Quando cambiamo il nostro indirizzo di casa, dobbiamo scegliere un nuovo medico, un nuovo fornitore di elettricità e acqua, dobbiamo cambiare il nostro indirizzo nella patente di guida, ecc. Inoltre, la procedura amministrativa è diversa nel caso in cui ci si trasferisce da un comune ad un altro comune, o se cambiamo il nostro indirizzo a causa della partenza dal nostro paese per andare a vivere all'estero.

I servizi interessati sono ovviamente concettualmente correlati. Ci concentriamo su due relazioni concettuali fondamentali, *part-of* e *is-a*. Una relazione *part-of* vale tra due servizi quando la specifica di uno ha come componente la specifica dell'altro. Nell'esempio, i servizi che (offrono le operazioni che) aggiornano l'indirizzo di casa nella patente di guida, scelgono il nuovo medico e scelgono il nuovo fornitore di energia elettrica, sono

tutti legati al servizio «cambio di indirizzo di casa». Diciamo che «cambio di indirizzo di casa» è un servizio composito, e i quattro servizi *part-of* con esso sono servizi elementari. Un servizio è elementare quando non siamo interessati a rappresentarlo ulteriormente in termini di componenti più atomici.

Fondamentalmente, un *servizio* è *elementare* se e solo se non esiste un altro servizio con una relazione *part-of* con esso, altrimento è un *servizio composito*.

Il costrutto *part-of*, pur essendo efficace nel relazionare servizi elementari e compositi, non ci aiuta ad esprimere la relazione esistente tra i diversi tipi di servizi relativi al «cambio di indirizzo di casa» nei diversi contesti in cui si applicano. Abbiamo bisogno per questo scopo di un nuovo costrutto. Una relazione *is-a* vale tra un servizio s_i (servizio figlio/specifico) e un servizio s_j (servizio padre/generale) quando s_i è una specializzazione (caso specifico) di s_j. Secondo la proprietà di ereditarietà dell'*is-a*, s_i eredita tutte le proprietà (funzionali e non funzionali) di s_j. Inoltre, s_i eredita tutte le relazioni tra s_j e le sue componenti. s_i può avere proprietà aggiuntive, non in s_j. Ad esempio, tre servizi che cambiano indirizzo tra due comuni, cambiano indirizzo tra Italia e estero, e cambiano indirizzo tra due paesi stranieri, possono essere considerati casi specifici del servizio generico di «cambio di residenza». Le caratteristiche comuni a tutti e quattro i servizi sono la necessità di aggiornare due basi di dati, mentre i database specifici cambieranno in base ai luoghi coinvolti nel cambio di indirizzo. Inoltre, quando ci si sposta dall'Italia all'estero, possiamo immaginare che verranno attivate ulteriori procedure amministrative specifiche, ad es., per questioni relative alla cittadinanza.

Concludiamo questa breve introduzione sui servizi, rimarcando che i servizi sono erogati attuando dei processi. Un processo pubblico è un processo che definisce le interazioni tra i partecipanti (nel processo) e le attività che sono visibili al pubblico per ogni partecipante. Un processo privato è un processo che, oltre alle interazioni e alle attività definite nei processi pubblici, definisce le interazioni e le attività interne ai singoli partecipanti.

3.1.2 Servizio digitale, API e Interfaccia di servizio

Un **servizio digitale** (talvolta anche indicato come *electronic service* o *e-service*) è un servizio che *viene erogato via Internet o in una rete, la fornitura è essenzialmente automatizzata o comporta solo un intervento umano minimo, ed è impossibile da garantire in assenza di tecnologia informatica*[2]. Quanto detto per i servizi, vale anche per quelli digitali, essendo questi una specializzazione.

La trasposizione di un *servizio* in un *servizio digitale* non si riduce al solo utilizzo di tecnologie informatiche ma, per ottenere la totalità dei vantaggi conseguenti da tale possibilità, richiede la necessità di ridefinire i processi attraverso una riprogettazione degli stessi (*Business Process Reengineering*, in breve BPR). Il BPR deve, tra le altre, assicurare:

- la formazione degli atti amministrativi direttamente in digitale, per ridurre gli oneri legati alla gestione degli originali analogici;
- superare una visione document-oriented favorendo una visione record-oriented, al fine di agevolare la circolarità delle informazioni in possesso della PA;
- efficientare le azioni realizzate da parte della PA, per razionalizzare le proprie funzioni e compiti;
- mettere al centro dell'azione amministrativa i cittadini ed imprese, per l'attuazione della semplificazione amministrativa.

Nella progettazione di sistemi software, tipicamente si distinguono tre strati logici di funzionalità in comunicazione tra loro:

- logica di presentazione (presentation layer) o front-end (ad es., un'applicazione web, una APP mobile, ecc.), ha il compito di presentare i risultati dell'elaborazione all'utente umano ed inviare le richieste di questi verso la parte centrale/elaborativa del sistema, facendo dunque da interfaccia uomo-macchina;

[2] Cf. Wikipedia, https://en.wikipedia.org/wiki/E-services] Rowley (Rowley J. (2006): An analysis of the e-service literature: towards a research agenda. Internet Research, 16 (3), 339-359) defines e-services as » [...] deeds, efforts or performances whose delivery is mediated by information technology. Such e-service includes the service element of e-tailing, customer support, and service delivery». This definition reflect three main components - service provider, service receiver and the channels of service delivery (i.e., technology). For example, as concerned to public e-service, public agencies are the service provider and citizens as well as businesses are the service receiver. The channel of service delivery is the third requirement of e-service. Internet is the main channel of e-service delivery while other classic channels (e.g. telephone, call center, public kiosk, mobile phone, television) are also considered. [...] The provision of services via the Internet (the prefix "e" standing for "electronic", as it does in many other usages), thus e-service may also include e-commerce, although it may also include non-commercial services (online), which is usually provided by the government».

- logica applicativa (application layer o business layer);
- logica di accesso ai dati (access data layer) o back-end, interroga il database o il sistema legacy[3].

Tale architettura viene poi spesso mappata a livello fisico-infrastrutturale in altrettanti strati fisici (*tier*) corrispondenti all'unità di computazione su cui risiede lo strato logico. Tali strati sono intesi interagire fra loro secondo le linee generali del paradigma client/server (il presentation layer è cliente della logica applicativa, e questa è cliente del modulo di gestione dei dati) e utilizzando interfacce ben definite. In questo modo, ciascuno dei tre strati può essere modificato o sostituito indipendentemente dagli altri, conferendo scalabilità e manutenibilità al sistema. Nella maggior parte dei casi, si intende anche che i diversi strati fisici (*tier*) siano distribuiti su diversi nodi di una rete anche eterogenea. Questa architettura di base può anche essere estesa ipotizzando che gli strati siano a loro volta «stratificati»; in questo caso si giungerebbe a una architettura multi-layer/tier.

Nello specifico dei servizi digitali, che appunto vengono erogati su Internet, il presentation layer verso l'utente può essere rappresentato da un Web server e da eventuali contenuti dinamici e statici (es. pagine di scripting che producono HTML visualizzato nel browser dell'utente), oppure da applicazioni mobili (*App*) che risiedono sul device mobile dell'utente (cellulare, tablet); la logica applicativa corrisponde a una serie di moduli integrati in un server applicativo, ed i dati sono depositati in maniera persistente su un DBMS o su un sistema legacy.

Con **application programming interface** (in acronimo **API**) si indica ogni *insieme di procedure/funzionalità/operazioni disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito*. Spesso con tale termine si intendono le librerie software disponibili in un certo linguaggio di programmazione. Una buona API fornisce una «scatola nera», cioè un livello di astrazione che evita al programmatore di sapere come funziona l'implementazione dell'API ad un livello più basso. Questo permette di ri-progettare o migliorare le funzioni all'interno dell'API senza cambiare il codice che si affida ad essa. Una API che non richiede il pagamento di diritti per il suo accesso ed utilizzo è detta «aperta» (open). La finalità di un'API è di ottenere un'astrazione a più alto livello, di solito tra lo strato sottostante l'API e quello che la utilizza (client).

Per realizzare un servizio digitale, come detto, è necessario progettare e realizzare i tre strati; lo strato di logica applicativa offre la sua API affinchè chi sviluppa lo strato di presentazione all'utente possa utilizzarla come se la logica applicativa fosse una libreria; estendendo, se vari sistemi esportano le proprie logiche applicative come API, la logica di presentazione può utilizzarle insieme, mischiandole (*mash-up*), esattamente come nello sviluppo di software moderno si programma riutilizzando le librerie offerte nel linguaggio di programmazione, sistema operativo, ecc. Quando il servizio digitale è erogato su Internet, e prevalentemente sul Web che si basa sul protocollo HTTP,) si parla di Web API. Per le Web API l'erogatore potrebbe decidere di rendere disponibile l'API non soltanto a chi sviluppa la logica di presentazione, ma «aperta» anche ad altre organizzazioni che volessero collaborare con l'erogatore, in questo caso si parla di Open API . In molti contesti, con abuso di nomenclatura, ma intuitivamente chiaro, i due termini vengono confusi e considerati sinonimi (dato che l'apertura è spesso associata al Web/Internet).

Per il W3C un **web service** è qualsiasi software che si rende disponibile su Internet e standardizza la sua interfaccia tramite la codifica XML[4]. Un client richiama un'operazione offerta da un web service inviando una richiesta (solitamente sotto forma di un messaggio XML) e il web service invia una risposta XML. I web service invocano la comunicazione su una rete, con HTTP come protocollo più comune. I web service si basano principalmente su standard come XML-RPC e SOAP (Simple Object Access Protocol). Quindi un web service è un possibile modo di realizzare una Web API. Il termine web service (originatosi intorno ai primi anni 2000) è nato proprio per indicare la logica applicativa, esposta sul web, sottostante ad un servizio digitale. A partire dalla seconda metà degli anni 2000, creando possibili confusioni, il termine Web API è stato utilizzato come alternativa a web service per indicare altri approcci/protocolli/tecnologie (come REST) per realizzare API senza utilizzare XML-RPC e SOAP. Ma anche una Web API indica la logica applicativa, esposta sul web, sottostante ad un servizio digitale.

Al fine di evitare ogni possibile ambiguità, spesso dovuta semplicemente all'utilizzo di termini differenti per indicare gli stessi concetti, nel seguito del documento si utilizza il termine **interfaccia di servizio** per indicare **l'esposizione delle funzionalità applicative che sono necessarie per realizzare un servizio digitale**. Tutte le classificazioni e considerazioni presentate per i servizi, valgono per i servizi digitali e quindi per le interfacce di servizio. In particolare come queste classificazioni e considerazioni si calano in specifiche tecnologie/protocolli/standard è

[3] Un sistema legacy (letteralmente «ereditato», che è un lascito del passato) è un sistema informatico, un'applicazione o un componente obsoleto, che continua ad essere usato poiché l'utente (di solito un'organizzazione) non intende o non può rimpiazzarlo. Legacy equivale a versione «retrodatata» (rispetto ai sistemi/tecnologie correnti).

[4] Cf. https://www.w3.org/TR/ws-arch/#whatis

uno degli obiettivi del presente documento. Un'interfaccia di servizio si compone in generale di varie operazioni, e può essere realizzata come un web service, un'API, una Web API, ecc.

Ogni qualvolta c'è un servizio, si può immaginare che nella moderna spinta all'innovazione, si giunga prima o poi ad una controparte digitale.
Un servizio digitale, se sviluppato seguendo i più moderni approcci di ingegneria del software, deve essere organizzato separando la logica di presentazione da quella applicativa, dove quest'ultima deve esporre le proprie operazioni tramite una interfaccia di servizio. Una interfaccia di servizio è l'esposizione delle funzionalità applicative che sono necessarie per realizzare un servizio digitale; tale esposizione deve essere operata con un approccio/tecnologia/standard che ne permetta l'invocazione da un modulo software client.
Emerge in ultima analisi che ogni qualvolta c'è un servizio digitale, ci può essere una interfaccia di servizio equivalente, e viceversa ogni qualvolta c'è una interfaccia di servizio, è immediato ipotizzare il servizio digitale equivalente.
Una interfaccia di servizio può offrire più operazioni (almeno una). Una interfaccia di servizio può essere realizzata utilizzando approcci/tecnologie/standard web service, API, Web API, REST API, ecc.

Nel prosieguo di questo documento, ci si focalizza solamente sulle interfacce di servizio, che sono il fondamento del Modello di Interoperabilità 2018.

3.1.3 Caratteristiche delle interfacce di servizio

Interfacce semplici e complesse In prima istanza, le interfacce di servizio possono essere distinte in due categorie: semplici e complesse.

Una interfaccia di servizio semplice implementa operazioni atomiche come ad esempio:

- Fornire contenuti puri, ad esempio informazioni dettagliate riguardo una risorsa (come le informazioni fiscali riguardanti una azienda) oppure le notizie del giorno;
- Effettuare una aggregazione semplice di informazioni provenienti da diversi sistemi back-end;
- Effettuare operazioni con effetti circoscritti ad un unico sistema di back-end in maniera atomica (che non richieda supporto alle transazioni).

Le interfacce di servizio semplici eseguono unità di lavoro atomiche che lasciano i sistemi sottostanti in uno stato consistente. Le operazioni non necessitano del mantenimento di uno stato tra una chiamata e l'altra e perciò sono anche note come interfacce di servizio stateless (senza stato). Si noti come il concetto di stato sia espresso in relazione all'interazione tra i due sistemi (client ed erogatore) e non alla persistenza di informazioni circa le risorse di interesse.

Le interfacce di servizio complesse coinvolgono l'utilizzo e la composizione di altre interfacce di servizio (in alcuni casi esposte da organizzazioni diverse) richiedendo il supporto all'esecuzione di processi e funzionalità di tipo transazionale. Questo significa che, rispetto alle interfacce di servizio semplici, in quelle complesse le operazioni hanno una granularità alta (meno fine) e richiedono il mantenimento di uno stato condiviso; per questo motivo vengono anche definite interfacce di servizio stateful (con stato). Concetti potenzialmente connessi a quello di stato sono il mantenimento di una sessione o conversazione.

Interfacce sincrone ed asincrone Un altro modo di classificare le interfacce di servizio è lo stile di interazione richiesto dalle diverse operazioni disponibili: sincrono (eg. di tipo Remote Procedure Call - RPC, chiamata remota a procedura) o asincrono (eg. basato sullo scambio di messaggi o documenti). Nelle operazioni sincrone, un client esprime la sua richiesta nella forma di una chiamata ed attende una risposta prima di continuare l'esecuzione. Nelle operazioni asincrone, invece, il client invia un documento/messaggio ma non si aspetta nessuna risposta (se non in alcuni casi il fatto che la richiesta è stata presa in carico). La risposta da parte dell'interfaccia di servizio, nei casi in cui ci sia, può apparire ore o anche giorni più tardi.

Interfacce semplici e mission-critical Un modo ulteriore di classificare le interfacce di servizio è quello di distinguere quelle sostituibili da quelle mission-critical. Una interfaccia di servizio sostituibile può essere fornita da diverse organizzazioni e la produttività è impattata in maniera limitata nel caso di disservizi. Una interfaccia di servizio mission-critical è invece di solito fornita da un'unica organizzazione e la indisponibilità della stesso può provocare dei forti disservizi.

Caratteristiche funzionali e non funzionali delle interfacce Le classificazioni introdotte non sono strette poiché a seconda delle operazioni fornite, una interfaccia di servizio può essere catalogata in una posizione qualsiasi tra i due estremi delle stesse.

Le interfacce di servizio devono essere accompagnate da una descrizione delle operazioni offerte il cui linguaggio dipende dalla tecnologia con cui l'interfaccia è implementata (si veda a partire dalla Sezione 3 per maggiori dettagli). La descrizione di una interfaccia di servizio di solito include caratteristiche funzionali e non funzionali. La descrizione funzionale si concentra sulle caratteristiche operative dell'interfaccia di servizio che descrivono il funzionamento in termini di operazioni offerte, i parametri richiesti da ognuna, gli 'endpoint[5] da utilizzare, il formato dei messaggi ed i protocolli di rete da utilizzare. La descrizione non funzionale si concentra invece sulla *qualità del servizio* (o qualità dell'interfaccia di servizio) in termini di limiti di utilizzo, costi e metriche di performance quali scalabilità, disponibilità, tempo di risposta, accuratezza, transazionalità, sicurezza e affidabilità.

3.1.4 Qualità del servizio

Il concetto di *quality of service - QoS*, fa riferimento alla descrizione non funzionale di una interfaccia servizio, cioè la capacità di una interfaccia di servizio di soddisfare le aspettative dei fruitori. Assicurare la QoS nell'ambito Internet e quindi ai fini dell'interoperabilità è una sfida critica a causa della natura dinamica ed impredicibile del contesto applicativo. Cambiamenti negli schemi di traffico, la presenza di transazioni business-critical, gli effetti dei problemi di rete, le performance dei protocolli e degli standard di rete richiedono una definizione precisa della QoS offerta da una interfaccia di servizio.

Gli elementi chiave a supporto della QoS possono essere riassunti come segue:

- *Disponibilità*. La probabilità che una interfaccia di servizio sia disponibile e funzionante in un istante casuale. Associato al concetto di disponibilità è quello di Time-To-Repair (TTR), cioè il tempo necessario a ripristinare una interfaccia di servizio una volta che questa diventa indisponibile. La disponibilità di una interfaccia di servizio dovrebbe potere essere verificata tramite l'esposizione di un'altra interfaccia di servizio di monitoraggio, dedicata ed a basso impatto (e quindi ad elevata disponibilità).
- *Accessibilità*. Misura la capacità di una interfaccia di servizio di essere contattabile da un elevato numero di richieste.
- *Prestazioni*. Le prestazioni vengono misurate solitamente rispetto a due valori: il *throughput* e la *latenza*. Il throughput rappresenta il numero di richieste soddisfatte in un dato intervallo. La latenza rappresenta la quantità di tempo che passa tra l'invio di una richiesta e la ricezione di una risposta. Una interfaccia di servizio con buone prestazioni ha un elevato throughput ed una bassa latenza.
- *Affidabilità*. Rappresenta la capacità di una interfaccia di servizio di funzionare correttamente e consistentemente fornendo la stessa QoS a dispetto di malfunzionamenti di diversa natura. Di solito viene espressa in termini di fallimenti in un dato lasso di tempo.
- *Scalabilità*. L'abilità di servire in maniera consistente le richieste a dispetto di variazioni nel numero delle richieste[6]. È strettamente connesso al concetto di accessibilità, ma qui il concetto fondamentale è il mantenimento delle prestazioni.
- *Sicurezza*. La sicurezza implica aspetti quali confidenzialità, integrità, autorizzazione ed autenticazione che saranno oggetto della Sezione 2.
- *Transazionalità*. Ci sono alcuni casi (ad es., interfacce di servizio stateful) in cui è necessario assicurare l'esecuzione transazionale di una operazione. La capacità di una operazione di rispettare questa proprietà è parte della QoS.

Gli erogatori devono prendere tutte le iniziative necessarie a mantenere i requisiti di QoS richiesti dal caso d'uso. Questo include anche l'utilizzo di buone pratiche. Ad esempio, per assicurare prestazioni e scalabilità il risparmio

[5] Con il termine endpoint si indica l'identificativo unico da utilizzare per richiamare un'interfaccia di servizio. Ad esempio, nel caso della tecnologia SOAP è l'URL del web service, nel caso di REST le URL (che hanno tutte un suffisso comune) delle risorse offerte, nel caso dei Message Broker il nome univoco della coda di messaggi o un topic nella stessa.

[6] In ambito cloud, si utilizzano i termini di scale-up/scale-down per indicare la scalabilità ottenuta incrementando o riducendo le risorse di singoli sistemi (ad es., memoria RAM), di scale-out/scale-in per indicare la scalabilità ottenuta mediante distribuzione, aggiungendo o diminuendo il numero dei sistemi utilizzati.

della banda è una condizione fondamentale. Le interfacce di servizio dovrebbero quindi implementare meccanismi di compressione del payload[7] e supportare la paginazione[8].

Quando si utilizzano meccanismi di caching, essi devono essere documentati nelle specifiche delle interfacce di servizio, ed essere conformi alle specifiche RFC-7234[9].

Questa sezione si è concentrata sul concetto di QoS nel campo delle interfacce di servizio. Misure di QoS possono essere introdotte anche per quanto riguarda i servizi digitali utilizzando metriche introdotte nei campi della Interazione Uomo-Macchina. Queste ultime sono fuori dagli obiettivi di questo documento.

Service Level Agreement - SLA

L'integrazione può coinvolgere numerose organizzazioni e erogatori esterni di interfacce di servizio. Al fine di accordarsi sulla QoS, erogatori di interfacce di servizio e fruitori utilizzano quelli che vengono definiti *Service Level Agreement - SLA*, ovvero *accordi sul livello di servizio*. Uno SLA può contenere le parti seguenti:

- *Scopo*. Le ragioni che hanno portato alla definizione dello SLA.
- *Parti*. I soggetti interessati nello SLA con i loro rispettivi ruoli (ad es., l'erogatore dell'interfaccia di servizio e il fruitore).
- *Periodo di validità*. L'intervallo di tempo, espresso mediante data e ora di inizio e data e ora di fine, per il quale si ritiene valido un particolare termine di accordo all'interno dello SLA.
- *Perimetro*. Quali sono operazioni interessate dallo specifico SLA.
- *Service Level Objectives - SLO*, ovvero *obiettivi sul livello di servizio*. I singoli termini di accordo all'interno di uno SLA. Di solito vengono definiti utilizzando dei *Service Level Indicators - SLI*, ovvero *indicatori sul livello di servizio*, che quantificano i singoli aspetti di QoS come indicato in questa sezione (ad es., disponibilità).
- *Penalità*. Le sanzioni che si applicano nel caso che l'erogatore dell'interfaccia di servizio non riesca ad assicurare gli obiettivi specificati nello SLA.
- *Esclusioni*. Gli aspetti della QoS non coperti dallo SLA.
- *Amministrazione*. I processi mediante i quali le parti possono monitorare la QoS.

Gli SLA possono essere statici o dinamici. Negli SLA dinamici, gli SLO (con associati SLI) variano nel tempo ed i periodi di validità definiscono gli intervalli di validità di questi ultimi (ad es., in orario lavorativo gli SLO possono essere differenti di quelli imposti durante la notte). La misurazione dei livelli di QoS all'interno di uno SLA richiedono il tracciamento delle operazioni effettuate in un contesto infrastrutturale multi-dominio (geografico, tecnologico e applicativo). In uno scenario tipico, ogni interfaccia di servizio può interagire con molteplici altre interfacce di servizio, cambiando il suo ruolo da erogatore a fruitore in alcune interazioni, ognuna governata da un differente SLA.

Recentemente, gli SLA hanno iniziato ad includere non soltanto vincoli relativi all'erogatore, ma anche vincoli che impongono ai singoli fruitori delle interfacce di servizio dei limiti relativi al ritmo ed alla quantità delle richieste. A tal fine gli erogatori devono definire ed esporre ai fruitori politiche di throttling[10] (anche noto come rate limiting) segnalando eventuali limiti raggiunti. Gli erogatori dovrebbero far rispettare le quote anche se se il sistema non è in sovraccarico, incentivando i fruitori a rispettarle.

Esempi di SLI sono i seguenti:

- dimensione massima di ogni richiesta accettata. Le richieste più grandi possono essere rifiutate;
- latenza al 90° percentile. Utilizzata per calcolare la responsività;
- percentuale di minuti negli ultimi 30 gg in cui l'interfaccia di servizio è stata disponibile;

[7] Il payload è il contenuto informativo di un messaggio di rete (eliminando la parte relativa al protocollo). Per compressione del payload si intende applicare un algoritmo di compressione (molto spesso gzip) al payload in modo da ridurre il traffico di rete.

[8] Per paginazione si intende la capacità di una operazione nell'interfaccia di servizio di fornire un risultato composto da molte voci per singole pagine sfruttando un qualche criterio di ordinamento.

[9] Cf. https://tools.ietf.org/html/rfc7234

[10] Con il termine throttling (o rate limiting) si intendono le politiche intraprese dalle interfacce di servizio al fine di limitare la frequenza con cui i fruitori possono chiamare l'interfaccia o specifiche operazioni all'interno della stessa.

- valori a 1 giorno e 30 giorni del success rate (ad es., il numero di chiamate terminate con successo rispetto al numero totale di chiamate);
- percentuale di minuti negli ultimi 30 gg in cui l'interfaccia di servizio è stata responsiva (ad es., il numero di chiamate con latenza inferiore ad un certo limite);
- tempo di risposta medio delle richieste totali (includendo le richieste rifiutate causa throttling) nell'ultimo giorno e negli ultimi 30 giorni;
- throughput misurato in bytes/s.

Gli SLI calcolati devono includere la latenza aggiuntiva dovuta ad eventuali componenti infrastrutturali e di rete (ad es., proxy-gateway).

Essi inoltre devono:

- utilizzare unità di misura del sistema internazionale (ad es., secondi, bytes);
- indicare nel nome identificativo l'eventuale periodo di aggregazione coi soli suffissi s (secondi), m (minuti), d (giorni) e y (anni) utilizzando al posto dei mesi il numero di giorni.

Ove possibile, gli SLO e gli SLA dovrebbero essere in relazione diretta con i valori associati (ad es., indicare success rate anzichè l'error rate), in modo che a valori più alti corrispondano risultati positivi.

3.1.5 Middleware

Con il termine middleware si intende lo strato software che separa le risorse informative dai fruitori delle interfacce di servizio, di fatto permettendo la realizzazione delle interfacce stesse. In tal senso un middleware gestisce la complessità e l'eterogeneità tipica dei sistemi distribuiti. Le risorse informative di cui si parla in questo caso possono essere nel caso più semplice della basi di dati, ma più comunemente includono altre interfacce di servizio (che a loro volta possono essere implementati utilizzando dei middleware) e sistemi legacy a cui il middleware contribuisce a fornire interfacce moderne. A tale fine i middleware forniscono una serie di funzionalità:

- Il supporto a framework per l'esposizione di interfacce di servizio implementati in differenti tecnologie e secondo differenti schemi di interazione. In questo senso essi nascondono agli sviluppatori le complessità legate all'esposizione di interfacce di servizio secondo specifici protocolli di rete.
- Facilitano il riuso di componenti software.
- Forniscono una serie di funzionalità di supporto alla sicurezza dei sistemi informatici che includono autenticazione ed autorizzazione.
- Forniscono funzionalità di scalabilità che sfruttano la distribuzione su risorse hardware.
- Aiutano in generale a soddisfare i requisiti di QoS dichiarati negli SLA.
- Integrano funzionalità utili quali il throttling, logging e caching.

Oltre a mascherare l'eterogeneità dell'hardware, i middleware mirano anche a mascherare l'eterogeneità delle piattaforme software permettendo di sviluppare i diversi componenti del sistema distribuito secondo i linguaggi e framework più adatti.

API Management

Gli API Management System sono dei middleware che concentrano tutte le funzionalità necessarie ad una organizzazione per gestire le loro interfacce di servizio su infrastrutture on-premises e cloud pubblici e privati. Essi si concentrano sullo sviluppo delle interfacce di servizio, la gestione del ciclo di vita delle stesse, il controllo degli accessi (tramite meccanismi di autorizzazione ed autenticazione), il throttling, il caching e le analitiche (utili al controllo degli SLA).

Un API management system può essere utilizzato ad esempio come strato di accesso alle API interne ad una amministrazione, rilasciando solo una parte delle stesse e con politiche personalizzate verso l'esterno e verso l'intranet.

Oltre alle funzionalità richieste nelle sezioni precedenti, alcuni API management system permettono di definire processi di automazione ed orchestrazione di breve durata (dette soft-orchestration). Si tratta di orchestrazioni molto semplici in cui non ci si aspetta intervento umano nel processo, la durata è brevissima e le regole definite sono molto semplici.

Logging

Il logging riveste un ruolo fondamentale nella progettazione e sviluppo di interfacce di servizio. Le moderne piattaforme middleware, oltre ad integrare meccanismi di logging interni, possono connettersi ad interfacce di servizio esterne che permettono la raccolta (log collection), la ricerca e la produzione di analitiche utili tra l'altro all'identificazione di problemi e al monitoraggio del sistema e della QoS. L'utilizzo di log collector permette di centrallizzare non solo i log relativi all'utilizzo dell'interfaccia di servizio, ma anche quelli di eventuali digital service e componenti di rete (ad es., proxy e application-gateway). I messaggi applicativi possono, ai fini di non ripudio (vedi Sezione 2.1.4) essere memorizzati assieme alla firma digitale e quindi archiviati periodicamente nel rispetto delle direttive sulla privacy.

L'erogatore deve documentare il dettaglio del formato della tracciatura e le modalità di consultazione e reperimento delle informazioni.

L'erogatore deve inoltre tracciare un evento per ogni richiesta, contenente almeno i seguenti parametri minimi:

- data e ora della richiesta in formato RFC3339[11] in UTC e con i separatori Z e T maiuscolo. Questa specifica è fondamentale per l'interoperabilità dei sistemi di logging ed auditing, evitando i problemi di transizione all'ora legale e la complessità nella gestione delle timezone nell'ottica dell'interoperabilità con altre PA europee;
- URI che identifica erogatore ed operazione richiesta;
- tipologia di chiamata (ad es., HTTP method per i protocolli basati su HTTP, basic.publish per AMQP);
- esito della chiamata (ad es., HTTP status per i protocolli basati su HTTP, SOAP fault nel caso di web services SOAP, OK/KO in assenza di specifici requisiti, eventuali messaggi di errore);
- identificativo del fruitore;
- ove applicabile, identificativo del consumatore o altro soggetto operante la richiesta comunicato dal fruitore - è cura del fruitore procedere a codifica e anonimizzazione ove necessario;
- ove applicabile, l'Indirizzo IP del client;
- ove applicabile, un identificativo univoco della richiesta, utile ad eventuali correlazioni tra chiamate diverse.

3.1.6 Attori e Interazioni

Come anticipato in «Presentazione del Modello di Interoperabilità 2018», l'obiettivo a tendere è quello di una PA in cui le singole amministrazioni offrono interfacce di servizio, in corrispondenza ai servizi digitali che erogano, e possono a loro volta cooperare attraverso l'invocazione di interfacce di servizio offerte da altre PA.

L'EIF riprende la classificazione delle interazioni possibili in generale in Administration-to-Citizen (A2C), Administration-to-Business (A2B) e Administration-to-Administration (A2A), ulteriormente distinguendo se il fruitore del servizio è un soggetto umano od un modulo software, arrivando quindi a definire le seguenti possibili interazioni:

1. A2A in modalità *human-to-machine*;
2. A2A in modalità *machine-to-machine*;
3. A2B in modalità *human-to-machine*;
4. A2B in modalità *machine-to-machine*;
5. A2C in modalità *human-to-machine*.

[11] Cf. https://tools.ietf.org/html/rfc3339#section-5.6

In base al precedente confronto tra servizio digitale e interfaccia di servizio, la classificazione suddetta deve essere meglio specificata, al fine di individuare i giusti contesti di intervento.

A2A in modalità human-to-machine. In questo caso c'è una interazione tra due amministrazioni, di cui una offre un servizio digitale e l'altra, per il tramite di un suo operatore umano, ne fruisce al fine di espletare le proprie procedure. Ad es., un operatore di un Comune accede ad un servizio digitale dell'Agenzia delle Entrate per verificare la correttezza del codice fiscale. In questo caso, l'interfaccia di servizio viene sollecitata dalla logica di presentazione che l'erogatore offre agli operatori delle altre amministrazioni, ma non c'è un'invocazione diretta (si ricordi che un'interfaccia di servizio viene invocata solamente da altri moduli applicativi client, non è fruibile direttamente da utenti umani)

A2A in modalità machine-to-machine. In questo caso c'è una interazione tra due amministrazioni, in cui una offre un servizio digitale, ed espone una interfaccia di servizio, e l'altra realizza una propria applicazione/sistema/procedura digitale il cui software ha bisogno di invocare l'interfaccia offerta. Ad es., in un Comune viene realizzato un software (che utilizzano gli operatori allo sportello anagrafico) che durante la sua esecuzione invoca l'interfaccia di servizio dell'Agenzia delle Entrate per la verifica del codice fiscale. In questo caso l'interfaccia di servizio dell'erogatore è invocata direttamente dal module software del fruitore.

Va notata una differenza tra le due modalità. Nel primo caso, una esigenza operativa che richieda l'utilizzo di più servizi digitali per essere espletata, prevede l'utilizzo da parte degli operatori di più servizi digitali, e gli utenti hanno il compito di coordinare i vari servizi digitali, eventualmente muovere i dati/risultati da uno all'altro, ecc. Ovvero la composizione dei servizi digitali non può essere automatizzata, ma rimane in carico all'utente che utilizza i servizi digitali. Nel secondo caso, la composizione di servizi digitali può essere invece facilmente realizzata andando a sviluppare un nuovo servizio digitale, che compone le interfacce applicative degli erogatori e realizza la logica di coordinamento, a sua volta possibilmente offerta come interfaccia di servizio composta, al di sopra della quale offrire la logica di presentazione.

A2B in modalità human-to-machine. In questo caso c'è una interazione tra un'impresa ed un'Amministrazione che offre un servizio digitale. L'impresa sfrutta il servizio digitale per il tramite di un suo addetto umano che interagisce con il servizio. Ad es., un addetto di un'azienda accede ad un servizio digitale dell'Agenzia delle Entrate per verificare la correttezza dei codici fiscale.

A2B in modalità machine-to-machine. In questo caso c'è una interazione tra un'impresa ed un'Amministrazione a livello applicativo, ovvero una procedura software di un'impresa richiama le funzionalità offerte da un'interfaccia di servizio erogata da un'Amministrazione.

Tutte le considerazioni fatte sulle interazioni A2A human-to-machine e machine-to-machine si applicano anche a questi casi, fatta salva la trasposizione operatore di un'Amministrazione con addetto di un'azienda.

L'ultimo caso **A2C in modalità human-to-machine** è quello in cui un cittadino utilizza un servizio digitale erogato da un'Amministrazione.

Un cittadino non interagirà mai con l'interfaccia di servizio erogata, ma sempre con una logica di presentazione che a sua volta invoca, nel caso auspicabile di software progettato in modo stratificato, l'interfaccia di servizio.

Dal punto di vista funzionale (cf. Sezione 1.1) tutte le modalità machine-to-machine sono analoghe: per l'interfaccia di servizio, l'essere invocata da un modulo software è funzionalmente indipendente dalla natura dell'utente che siede di fronte alla logica di presentazione che si attesta su quel modulo (sia esso un operatore di un'altra Amministrazione o di un'azienda). La differenza è negli aspetti non funzionali, in particolare QoS e sicurezza, in quanto a seconda di chi è l'organizzazione fruitrice, l'erogatore potrebbe offrire differenti livelli di servizio, autorizzazioni, garanzie di sicurezza, ecc. L'utilizzo che il fruitore farà dell'interfaccia di servizio ha un impatto, soprattutto in termini di responsabilità, framework legale, ecc.; ad esempio, nel caso A2B, il caso in cui l'azienda fruitrice utilizza l'interfaccia all'interno di un proprio modulo applicativo, ovvero il caso in cui offre un servizio a valore aggiunto, devono essere differenziati; ma questo non ha impatti sugli aspetti tecnologici dell'interfaccia di servizio, bensì su quelli di governance, e verranno ripresi in «Governance del Modello di Interoperabilità». Tutti i casi human-to-machine sono analoghi: in questo caso non c'è interazione diretta con l'interfaccia di servizio, ma sempre per il tramite di una qualche logica di presentazione e la differenza è nella natura dell'utente umano che siede di fronte al modulo software che realizza tale logica di presentazione.

Emerge come la modalità di progettazione dei servizi digitali che stratifica chiaramente le interfacce di servizio separandole dalle logiche di presentazione, è la modalità corretta per supportare le possibili interazioni offerte da un'Amministrazione: a seconda della modalità diventa agevole stratificare la corretta logica di presentazione, ovvero moduli client, al di sopra della stessa interfaccia di servizio.

La tabella seguente riassume le considerazioni presentate.

Interazione	servizio digitale	interfaccia di servizio	richiede logica di presentazione	composizione di più servizi[12]
A2A human-to-machine	✓		✓	-
A2A machine-to-machine		✓		+
A2B human-to-machine	✓		✓	-
A2B machine-to-machine		✓		+
A2C	✓		✓	-

3.1.7 Uniformità dei dati

Uno degli aspetti maggiormente critici quando si espongono interfacce di servizio è la modellazione dei dati. Come anticipato nella Sezione 1.1, l'information model sottostante ad un servizio (e quindi anche ad un servizio digitale e interfaccia di servizio) serve a rappresentare sia il modello dei dati relativo ai cambiamenti di stato che il servizio opera, sia i dati che «transitano» (input/output) attraverso il servizio. Nel seguito ci soffermiamo sul caso delle interfacce di servizio. Facendo un parallelo con la programmazione orientata agli oggetti, oltre a definire i metodi offerti dalle classi del programma (nel parallelo corrispondenti alle operazioni dell'interfaccia di servizio), bisogna definire correttamente il numero e soprattutto il tipo dei parametri di input ed output. Non a caso, l'aspetto metodologico cruciale su cui si soffermano tutte le metodologie di progettazione e programmazione basate sul design-by-contract[13] è la definizione della segnatura dei metodi, al giusto livello di granularità, che comprende sia il nome del metodo che i parametri.

Il livello di granularità dipende da vari aspetti dell'interfaccia di servizio, in particolare se questa è atomica o composta, se il servizio a cui corrisponde è informativo o transazionale (cf. Sezione 1.1). Nella tabella seguente si forniscono delle indicazioni qualitative, da utilizzare come linee guida nella definizione delle interfacce di servizio. In «Profili e pattern di interoperabilità», esse saranno utilizzate nella definizione di vari possibili pattern che rispondono ad esigenze specifiche.

Tipo di interfaccia	Granularità[14]
Elementare	*fine-grained*
Composta	*coarse-grained*
Informativa	*fine-grained*
Transazionale	*coarse-grained*

Per quanto riguarda gli aspetti di formato dei dati delle interfacce di servizio, è importante

- omologare ove possibile i nomi delle variabili alle consuetudini europee abilitando l'interoperabilità con i servizi erogati dagli altri paesi;
- associare ai nomi dei campi dei metadati utili alla classificazione dei servizi;

[12] L'uso del +/- nell'ultima colonna da un'indicazione qualitativ a di quanto sia agevole comporre elementi nella specifica interazione. Come discusso, nel caso di servizi digitali la composizione è a cura dell'utente finale, che agisce da *human-ware* (ovvero deve farsi carico di realizzare, attraverso l'interazione stessa, la logica di composizione ed il passaggio di dati), mentre la composizione di interfacce di servizio è più semplice da automatizzare, e soprattutto può poi essere riusata più volte esponendo a sua volta come interfaccia di servizio composta. In quest'ultimo caso va però realizzata una logica di presentazione per il servizio digitale composta, se si vuole offrirlo agli utenti umani.

[13] Cf.

Meyer, Bertrand: *Design by Contract*, Technical Report TR-EI-12/CO, Interactive Software Engineering Inc., 1986

Meyer, Bertrand: *Design by Contract*, in *Advances in Object-Oriented Software Engineering*, eds. D. Mandrioli and B. Meyer, Prentice Hall, 1991, pp. 1–50

Meyer, Bertrand: *Applying «Design by Contract»*, in Computer (IEEE), 25, 10, October 1992

Meyer, Bertrand (1997). Object-Oriented Software Construction, second edition. Prentice Hall. ISBN 0-13-629155-4.

[14] La granularità è il livello di dettaglio con cui i dati sono esposti e scambiati. *Coarse-grained* significa un livello di dettaglio «basso», in quanto molti dettagli possono o devono rimanere interni all'implementazione dell'interfaccia di servizio. *Fine-grained* significa invece che il dato deve essere specificato ad un dettaglio massimo, poichè che il fruitore ha bisogno di una visione puntuale del dato stesso.

- facilitare la validazione automatica delle specifiche dei vari servizi[15].

Inoltre è auspicabile che la specifica del formato sia coerente, od addirittura la stessa, tra varie tecnologie di esposizione delle interfacce di servizio[16].

Le indicazioni generali sono:

- per gli schemi dei dati, utilizzo di nomi basati su riferimenti europei (ad es., Core Vocabularies/Dizionari Controllati, Direttiva Europea INSPIRE 2007/2/CE[17]) e standard de facto e de iure eventualmente disponibili sulla specifica tematica;
- UTF-8 come codifica di default[18];
- URI come identificatore del servizio e dell'erogatore[19];
- per i formati di serializzazione, semplicità di integrazione con strumenti di validazione (ad es. parsing);
- paesi, lingue e monete[20]: ISO 3166-1-alpha2 country[21], ISO 4217 currency codes[22];
- data e ora in RFC3339[23], un sottoinsieme dell'ISO8601 ottimizzato per il web;
- aree amministrative NUTS 1 e successive: nomenclature NUTS[24] (per il livello NUTS 0 - entità nazionali si fa riferimento ai codici ISO).

3.2 Concetti di Sicurezza

La sicurezza dei sistemi informatici è l'insieme di pratiche messe in atto al fine di impedire l'accesso non autorizzato, l'uso, la divulgazione, l'interruzione dell'accesso, la modifica, l'ispezione e la distruzione delle informazioni.

Questa sezione si concentra sui meccanismi di sicurezza che vadano oltre il semplice filtraggio di pacchetti basato su indirizzi IP, tipo di protocollo (anche detto circuit-level filtering) o contenuto del dato applicativo (application-level gateway o antivirus)[25]. In particolare la sezione si concentra sull'utilizzo di protocolli e tecniche di sicurezza basate sulla manipolazione dei messaggi di rete. La sezione farà inoltre riferimento a come i requisiti di sicurezza possano essere variabili a seconda dello scenario applicativo e del caso d'uso.

3.2.1 Meccanismi di base

Diversi sono i concetti chiave dietro al mondo della sicurezza. In origine il termine faceva riferimento al concetto di triade CIA (Confidenzialità, Integrità e Availability - Disponibilità). Nel tempo altri concetti si sono aggiunti quali l'autenticazione e il non ripudio.

Questa sezione descrive questi concetti introducendo le principali tecniche impiegate per assicurarli.

[15] Come anticipato in "Presentazione del Modello di Interoperabilità 2018" ed approfondito in "Governance del Modello di Interoperabilità", la modellazione e specifica dei dati avviene nei Gruppi di Lavoro interni agli Ecosistemi, che indirizzano il lavoro di standardizzazione.

[16] Ad esempio, la serializzazione in JSON di un dato dovrebbe essere la medesima sia se viene esposto esternamente tramite REST API sia se transita da un messaging system interno all'amministrazione. Una rappresentazione opportuna permette quindi la fruizione del dato da sistemi diversi limitando il ricorso alle conversioni.

[17] Cf. https://joinup.ec.europa.eu/page/core-vocabularies e http://eur-lex.europa.eu/legal-content/IT/ALL/?uri=CELEX:32007L0002

[18] Vedi Linee Guida Patrimonio Pubblico. Architettura dell'Informazione del Settore Pubblico, http://lg-patrimonio-pubblico.readthedocs.io/it/latest/arch.html#formati-aperti-per-i-dati-e-documenti

[19] Gli URI vengono utilizzati anche dal gruppo DAF-Semantic per la nomenclatura delle ontologie e dei dataset

[20] Si noti che questi standard sono già usati nelle specifiche AgID sulle firme elettroniche e sul formato della fattura PA.

[21] Cf. https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2

[22] Cf. https://en.wikipedia.org/wiki/ISO_4217

[23] Cf. https://tools.ietf.org/html/rfc3339#section-5.6

[24] Cf. https://it.wikipedia.org/wiki/Nomenclatura_delle_unit%C3%A0_territoriali_statistiche

[25] Per questi si faccia riferimento alla letteratura, ad es., William Stallings (2017): Cryptography And Network Security, 7th edition.

Disponibilità

Il concetto di disponibilità è stato introdotto nella Sezione 1.3 parlando della QoS. Il concetto di disponibilità è legato strettamente anche a quello di sicurezza, poiché la disponibilità di una interfaccia di servizio può essere legata non solo a cause di natura tecnica ma anche a specifici tipi di attacco (ad es., denial of service).

Confidenzialità

Con il termine confidenzialità si intende la protezione dei dati e delle informazioni scambiate tra un mittente e un destinatario. La confidenzialità, declinata per il canale di comunicazione, è la proprietà di assicurare che l'informazione scambiata tra due entità colloquianti in rete non possa essere acceduta da soggetti terzi.

La confidenzialità è ottenuta tramite la cifratura dei dati e delle informazioni (sicurezza di messaggio) o del canale di comunicazione (sicurezza del canale).

In un metodo di cifratura, un messaggio in chiaro (anche chiamato plain text) viene trasformato in un messaggio codificato e viceversa. Gli algoritmi di cifratura si distinguono in meccanismi a chiave simmetrica (o privata o condivisa) e chiave asimmetrica (o pubblica). In entrambi i casi la lunghezza delle chiavi influenza la sicurezza della comunicazioni (chiavi più lunghe sono più sicure) perché proteggono maggiormente da attacchi a forza bruta. Si suppone infatti che ogni meccanismo di cifratura possa essere rotto tramite enumerazione a patto che il tempo necessario (esponenziale nella lunghezza della chiave) non sia troppo lungo rispetto agli scopi dell'attaccante. Un'altra tipologia di attacco ai metodi di cifratura (che si applica in particolar modo ai metodi a chiave simmetrica in cui le password sono generate da umani) sono quelli di tipo dizionario, basati sull'uso di parole di uso comune.

Nei meccanismi di cifratura a chiave privata, entrambe le parti (il mittente ed il destinatario) nel canale di comunicazione condividono la stessa chiave di cifratura che viene impiegata sia per cifrare che per decifrare il messaggio. La cifratura a chiave simmetrica è molto efficiente e viene utilizzata per la riservatezza di grandi quantità di dati (ad es., interi file). È necessario che le due parti abbiano condiviso la chiave privata con un metodo sicuro (ad es., scambiandola fisicamente di persona oppure tramite un meccanismo di cifratura a chiave pubblica, come si vedrà nella Sezione 2.4). Algoritmi noti di cifratura a chiave simmetrica sono RC4, DES, Triple DES, AES, IDEA e Camellia.

Nei meccanismi di cifratura a chiave pubblica, vengono utilizzate due chiavi diverse per la cifratura e la decifratura dei messaggi. In particolare si supponga che il destinatario abbia una coppia di chiavi di cui una è privata (conosciuta solo al destinatario) ed una è pubblica (conosciuta a tutti e liberamente inviata sulla rete anche in chiaro). Al fine di inviare un messaggio su di un canale sicuro, il mittente cifra il messaggio utilizzando la chiave pubblica del destinatario, ma questo potrà essere decifrato solo dal destinatario utilizzando la chiave privata. Per il destinatario infatti chiave pubblica e chiave privata sono state generate in modo da essere complementari. Il meccanismo a chiave pubblica risolve il problema della condivisione delle chiavi poiché la chiave pubblica può essere inviata su Internet senza pericolo (non può essere utilizzata per decifrare il messaggio). Come difetto, la crittografia a chiave pubblica soffre di basse prestazioni e per questo motivo viene utilizzata o nelle fasi preliminari necessarie a concordare una chiave privata di sessione condivisa (come nel caso di TLS) oppure per i meccanismi di firma digitale (quindi non a scopo di cifratura). L'algoritmo più diffuso per la cifratura a chiave pubblica è RSA (dai nomi degli inventori Rivest Shamir e Adleman).

Integrità e Firma Digitale

Un messaggio in transito su una rete informatica può subire delle modifiche (ad esempio tramite attacchi di tipo man-in-the-middle). I meccanismi a chiave pubblica possono essere utilizzati ai fini di produrre delle prove, dette firme digitali, utili a di verificare che il messaggio ricevuto sia uguale a quello inviato.

Il meccanismo di firma digitale prevede di inviare assieme al messaggio, un secondo messaggio (detto firma digitale) ottenuto dal primo:

- calcolando un riassunto (digest) del messaggio tramite tecniche cosiddette di hashing;
- cifrando il riassunto utilizzando la chiave privata del mittente.

Le tecniche di hashing utilizzate per la firma digitale sono progettate secondo diversi criteri. Tra cui:

- devono essere funzioni cosiddette one-way. Deve cioè essere facile calcolare il riassunto ma difficile risalire dal riassunto al testo originale. Questo viene anche facilitato dal fatto che i riassunti hanno solitamente lunghezza fissa;
- devono fare si che piccolissime modifiche al messaggio in input generino significative differenze nel riassunto.

La tecnica di hashing più utilizzata per la firma digitale è Secure Hash Algorithm - SHA (disponibile in diverse versioni). Nel momento in cui un messaggio viene ricevuto, il destinatario utilizza la chiave pubblica del mittente per decifrare la firma digitale e verificare che essa corrisponda al riassunto del messaggio. La combinazione di tecniche di hashing e di cifratura a chiave pubblica assicura che un attaccante non possa modificare il messaggio e generare una firma valida per lo stesso, assicurando quindi l'integrità del messaggio stesso.

Non Ripudio e Public Key Infrastructure - PKI

Il meccanismo di firma digitale descritto in Sezione 2.1.3 assicura l'integrità del messaggio ma non ne assicura l'autenticità della fonte. In pratica, chi riceve un messaggio è sicuro che esso non ha subito modifiche durante il transito ma non è sicuro dell'identità del mittente. Il messaggio ricevuto non potrà quindi essere utilizzato ai fini del non ripudio, cioè come prova che uno specifico soggetto è il vero mittente del messaggio. Il problema principale risiede nella maniera in cui la chiave pubblica di un soggetto viene distribuita. Essa, come detto, viene posta pubblicamente su Internet ma niente vieta ad un attaccante di creare una coppia chiave pubblica / chiave privata e distribuire quest'ultima fingendosi un altro soggetto ed inviare per conto di questo, in maniera fraudolenta, dei messaggi. In altre parole chi riceve il messaggio non ha modo di verificare l'autenticità della chiave pubblica che sta utilizzando. A tal fine il meccanismo introdotto è quello della Public Key Infrastructure - PKI.

Nella PKI oltre al mittente ed al destinatario del messaggio, viene aggiunto una terza parte detta Certification Authority (Autorità di Certificazione) la quale emette dei certificati. Un certificato è un documento in chiaro contenenti informazioni riguardanti l'identità dell'intestatario del certificato e la sua chiave pubblica e viene firmato dalla certification authority utilizzando la propria chiave privata.

La chiave pubblica della certification authority è installata nei sistemi operativi (e distribuita solitamente tramite gli aggiornamenti degli stessi), viene utilizzata per verificare che la chiave pubblica del mittente sia autentica. Il mittente invia assieme al messaggio firmato il suo certificato che viene validato utilizzando la chiave pubblica della certification authority che ha emesso il certificato stesso.

Il meccanismo PKI è sicuro fino a quando un attaccante non è in grado di installare sulle macchine del destinatario una public key fasulla per le certification authority. Per ovviare a questi problemi sono necessari dei meccanismi di sicurezza a livello di macchina che sono fuori dal perimetro di questo documento. Lo standard comunemente usato per i certificati è X.509.

Nel Modello di Interoperabilità 2018, le amministrazioni dovranno acquistare certificati commerciali. Negli ultimi anni alternative all'approccio PKI sono stati proposti (ad es., Web of Trust) ma il Modello attualmente ne vieta l'utilizzo.

Autenticazione

In un ambiente di calcolo distribuito, l'autenticazione è il meccanismo tramite il quale client e erogatore accertano le identità degli specifici utenti e sistemi per conto dei quali stanno operando. Quando la prova di autenticazione è bidirezionale si parla di mutua autenticazione.

L'autenticazione è spesso ottenuta in due fasi:

1. Si definisce un contesto di autenticazione effettuando una chiamata ad una entità di autenticazione diversa dall'erogatore;
2. Il contesto di autenticazione è impiegato per autenticarsi con l'altra parte della comunicazione.

Si noti come il meccanismo di non ripudio basato su PKI e firma digitale presentato in Sezione 2.1.4 sia un metodo di autenticazione ed in tal modo è usato in protocolli di strato di trasporto quali TLS (vedi Sezione 2.4) al fine di garantire non ripudio. Esistono poi dei protocolli di autenticazione a livello applicativo che forniscono dei vantaggi rispetto all'autenticazione basata su PKI:

- L'autenticazione basata su PKI solitamente non autentica solo i soggetti ma anche le macchine coinvolte (ad es., il certificato di un sito Internet contiene anche i nomi DNS su cui il sito risponderà);
- Possibilità di Single-Sign On - SSO. Il contesto di autenticazione definito con protocolli di strato applicativo può essere riutilizzato nell'interazione con diverse interfacce di servizio. Questo è dovuto al fatto che il client assume l'identità della persona o del soggetto per cui è stato creato il contesto di autenticazione;
- L'utilizzo di certificati è scomodo per l'utente finale e questo rende la mutua autenticazione basata su firma digitale meno adatta ai casi in cui siano utenti umani ad autenticarsi;
- Non sempre la funzionalità di non ripudio è richiesta e l'uso di certificati lato client risulta costoso.

A seconda dell'interfaccia di servizio utilizzata, l'autenticazione può essere debole o forte. Per autenticazione forte si intende una autenticazione che richiede almeno due fattori (ad es., nome utente/password e one-time password - OTP). I protocolli per autenticazione ed autorizzazione a livello applicativo più diffusi sono oggetto della Sezione 2.3.

Autorizzazione

I meccanismi di autorizzazione in ambienti distribuiti definiscono quali risorse possono essere accedute da uno specifico utente. Tipiche politiche di autorizzazione permettono l'accesso a specifiche collezioni a specifici gruppi di utenti autenticati sulla base di ruoli, gruppi e privilegi. L'autenticazione degli utenti è quindi una componente fondamentale nell'autorizzazione anche se i requisiti di autenticazione (forte o debole) possono cambiare a seconda del protocollo. Le politiche di autorizzazione sono le più svariate e possono interessare ad esempio l'ora del giorno in cui specifici utenti possono accedere a specifiche risorse oppure il rate massimo di chiamate concesse ad un utente.

3.2.2 Minacce alla sicurezza dei sistemi informatici

Nelle sezioni precedenti alcune minacce alla sicurezza sono state accennate. In questa sezione approfondiamo le diverse tipologie di attacchi. Non ci soffermeremo sugli attacchi basati su malware, ma ci limiteremo agli attacchi basati sull'uso dei protocolli di rete. I tipi di attacchi più comuni sono i seguenti:

- *Eavesdropping*. E" un tipo di attacco passivo (senza modifica dei dati) in cui un attaccante riesce a rubare informazioni leggendo dati da una connessione non cifrata. I protocolli che assicurano confidenzialità difendono da questo tipo di attacco.
- *Modifica dei dati*. Un attaccante potrebbe riuscire a modificare i pacchetti in transito nella rete. I meccanismi di firma digitale difendono da questo tipo di attacco.
- *Identity spoofing*. In questo tipo di attacco, l'attaccante finge di essere un altro utente. Questo tipo di attacco è risolto mediante meccanismi di autenticazione.
- *Attacchi su base password*. In questo caso l'attaccante cerca di ottenere delle password, utilizzate ad esempio ai fini di autenticazione ed autorizzazione. Come già anticipato, gli attacchi basati su password si basano o su forza bruta oppure su metodi di tipo dizionario. Questo tipo di attacchi si evitano impostato politiche forti riguardo alle password utilizzate e metodi di autenticazione forte (a più fattori).
- *Denial of service - DoS*. In questo tipo di attacco l'attaccante mira a rendere non operativa una interfaccia di servizio inondandola di richieste e minandone quindi l'accessibilità. Difendersi da questi tipi di attacchi è in genere molto difficile (specialmente nella variante distribuita DDoS).
- *Attacchi man-in-the-middle*. In questo caso un attaccante si intromette come terza parte in una conversazione tra mittente e destinatario modificando i messaggi scambiati. Gli attacchi man-in-the-middle si combattono tramite tecniche di cifratura ed integrità degli scambi.

In alcuni casi, gli attaccanti possono sfruttare delle falle scoperte nei protocolli o nelle implementazioni. E" quindi di fondamentale importanza tenere aggiornati i sistemi ed utilizzare quando possibile versioni aggiornate dei protocolli.

3.2.3 Protocolli per autenticazione e autorizzazione

Nel caso di autenticazione ed autorizzazione, occorre distinguere gli approcci utilizzati nello scenario human-to-machine e quelli utilizzati nello scenario machine-to-machine. I protocolli più comuni in ambito Web per autenticazione ed autorizzazione nel caso human-to-machine sono:

- OAuth2[26] è uno standard per l'autorizzazione;
- OpenID[27] è uno standard pensato per la sola autenticazione. L'ultima versione, denominata OpenID Connect[28], è costruita su OAuth2 in termini di scambio di messaggi;
- Security Assertion Markup Language - SAML[29] (la versione corrente è la 2) è il protocollo più vecchio in circolazione e copre l'autenticazione e in parte l'autorizzazione;
- eXtensible Access Control Markup Language - XACML[30] complementare a SAML per la gestione esaustiva degli aspetti di autorizzazione.

Nei protocolli human-to-machine, un client riceve autorizzazioni ad usare un certo tipo di risorsa per conto di un utente umano tramite le credenziali di quest'ultimo. La richiesta del token/assertion è effettuate per mezzo di uno user-agent (cioè un browser o una app mobile) che funge da intermediario.

Il ModI 2018 obbliga all'utilizzo di SPID per l'autenticazione human-to-machine o degli altri metodi indicati nell"art. 64 del Codice per l'Amministrazione Digitale[31] che includono anche la Carta d'Identità Elettronica - CIE e la Carta Nazionale dei Servizi - CNS.

SPID[32] è attualmente basato su SAML ma il supporto per OpenID Connect è in fase di definizione al fine di supportare in maniera più semplice l'autenticazione da piattaforme mobili.

In questo senso vale la pena esplorare le differenze principali tra SAML ed OpenID Connect (in breve Connect). Dal punto di vista della terminologia i due protocolli utilizzano termini differenti per gli stessi componenti:

- Identity Provider (SAML) o OpenID Provider (Connect) sono le entità che certificano l'identità dell'utente;
- Service Provider (SAML) o Relying Party (Connect) sono le interfacce di servizio, le app mobili o i siti presso cui l'utente vuole autenticarsi;
- Asserzioni (SAML) o Token (Connect) sono dei documenti firmati dall'Identity Provider (SAML) o dall'OpenID Provider (Connect) che contengono le informazioni circa l'utente identificato e le autorizzazioni che possiede.

La tabella seguente riassume le caratteristiche dei protocolli per l'interazione human-to-machine:

	OpenId Connect	**SAML + XACML**
Formato token/assertion	JSON	XML
Autorizzazione		✓
Autenticazione	✓	✓
Rischi per la sicurezza	Phishing[33]	XML Signature Wrapping[34]

Uno scenario interessante nell'ambito dell'integrazione A2A e A2B è quello legato alla federazione di domini (ad es., due diverse amministrazioni) in cui alcuni utenti di un dominio devono essere autenticati ed autorizzati per accedere a risorse dell'altro dominio (una federazione può includere anche più di due domini). In ambito SOAP,

[26] Cf. https://tools.ietf.org/html/rfc6749
[27] Cf. http://openid.net/developers/specs/
[28] Cf. http://openid.net/connect/
[29] Cf. http://saml.xml.org/saml-specifications
[30] Cf. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html
[31] Cf. http://www.agid.gov.it/cad/art-64-sistema-pubblico-gestione-identita-digitali-modalita-accesso-ai-servizi-erogati-rete
[32] Cf. http://spid-regole-tecniche.readthedocs.io/en/latest/
[33] Per phishing si intende il tentativo di un attaccante di fingersi qualcun altro. Nel caso di OpenId Connect, in particolare, sia per quanto riguarda OpenId che OAuth2, diversi attacchi sono stati rivelati che permettono ad una relying party di redirezionare l'utente verso un identity provider falso.
[34] L'XML Signature Wrapping è una vulnerabilità non legata direttamente al protocollo ma presente in alcune implementazioni ed in diverse forme (cf., https://blog.netspi.com/attacking-sso-common-saml-vulnerabilities-ways-find/). Il tool SAML Raider può essere utilizzato per verificare la presenza della vulnerabilità.

gli standard più utilizzati sono WS-Federation[35] & WS-Trust[36] (vedi Sezione 3 per l'inquadramento nello stack WS-*). Soluzioni su altre tecnologie vengono sviluppate ad-hoc.

Per quanto riguarda lo scenario machine-to-machine invece, come si vedrà nella sezione 2.4, l'autenticazione può avvenire a livello di trasporto utilizzando TLS.

Per quanto riguarda l'autorizzazione machine-to-machine invece è possibile utilizzare il protocollo OAuth2 nello specifico del flusso Client Credential Grant[37]. Tale flusso a differenza di quello standard non richiede la presenza di uno user-agent. Il client possiede invece delle proprie credenziali che vengono utilizzate per richiedere il token all'authorization server.

3.2.4 Protocolli per integrità e confidenzialità

Per ragioni storiche lo stack TCP/IP non ha di base funzionalità di sicurezza. I messaggi viaggiano in chiaro sulla rete. Poiché le tecnologie per l'integrazione che verranno introdotte utilizzano HTTP come principale protocollo di trasporto o applicativo[38], è importante che il canale di comunicazione sia protetto. La IETF definisce come standard per la securizzazione di TCP il protocollo Transport Layer Security - TLS. Con il termine HTTPS si definisce l'utilizzo di HTTP su canale TLS. Tutti le interfacce di servizio esposte nel ModI 2018 devono essere basate su HTTPS. Il protocollo TLS (ed il suo predecessore deprecato Secure Sockets Layer - SSL) assicurano su TCP confidenzialità (tramite cifratura) ed integrità (tramite firma digitale e PKI). Come introdotto in Sezione 2.1.5, il meccanismo di firma digitale assicura anche autenticazione ma questa è fatta machine-to-machine.

Il protocollo TLS (versione stabile corrente 1.2, draft 1.3 presentato a Marzo 2018) si basa come detto sull'utilizzo della firma digitale per lo scambio di una chiave di sessione da utilizzare come chiave simmetrica.

Per quanto riguarda i singoli algoritmi utilizzati:

- Per lo scambio della chiave di sessione, TLS supporta numerose tecniche. Tra quelle proposte, si impone l'uso di tecniche che evitano attacchi man-in-the-middle e forniscono la cosiddetta forward secrecy (cioè che la scoperta di una chiave privata usata nello scambio non permette di scoprire la chiave di sessione). Gli algoritmi di scambio delle chiavi permessi sono quindi ephemeral Diffie–Hellman - DHE ed ephemeral Elliptic Curve Diffie–Hellman - ECDHE.
- Per la cifratura TLS supporta numerosi algoritmi. Si suggeriscono i protocolli attualmente supportati nello standard TLS 1.3 e che sono considerati sicuri: Advanced Encryption Standard - AES (nella versioni GCM e CCM).
- Per l'integrità si suggerisce l'uso SHA almeno a 256 bit (quindi a partire dal cosiddetto SHA-2).

Nel Modello di Interoperabilità 2018, a prescindere dal profilo di autenticazione ed autorizzazione scelta (che dipende dal caso d'uso), il protocollo di trasmissione:

- DEVE essere basato su HTTP >= 1.1;
- DEVE essere cifrato tramite TLS >= 1.2;
- DEVE essere firmato con SHA-256 o superiore
- DEVE essere conforme alle misure minime AgID Basic Security Controls[41];
- Gli erogatori di interfacce di servizio DEVONO utilizzare l'header HSTS (HTTP Strict Transport Security) per evitare attacchi di tipo SSL Strip (tipo di attacco Man-in-the-middle).

Inoltre, ogni certificato TLS utilizzato per erogare interfacce di servizio:

- NON DEVE essere self-signed (ad es., CA:true);
- DEVE contenere i seguenti elementi Subject, Key Identifier, Serial Number ed Issuer;
- DEVE avere il parametro keyUsage con i seguenti bit: *digitalSignature, keyEncipherment*[42];
- DOVREBBE contenere i riferimenti al DNS dei domini serviti;
- Un certificato usato ai fini di non ripudio DEVE avere inoltre il parametro keyUsage con il bit nonRepudiation settato.

[35] Cf. http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.html
[36] Cf. http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.html
[37] Cf. https://tools.ietf.org/html/rfc6749#section-4.4
[38] Ai fini dell'interoperabilità su Internet, la scelta di HTTP permette integrazione senza necessitare di regole particolari di inoltro o di definire Virtual Private Network - VPN.

Numerose sono le minacce alla sicurezza a cui è esposto TLS (in special modo con vecchie versioni del protocollo accoppiate ad algoritmi per cifratura ed integrità vulnerabili). L'IETF nel 2015 ha rilasciato a riguardo una RFC informativa[43]. Per questo motivo, in determinati scenari che richiedono elevati standard di sicurezza, si aggiunge talvolta un ulteriore strato di sicurezza a livello applicativo.

Nel modello SPCoop si richiedeva che in ogni caso HTTPS fosse utilizzato con autenticazione mutual-TLS (vedi Sezione 2.3). Nel tempo sono emersi scenari di interazione con requisiti di sicurezza inferiori (ad es., solo HTTPS non-mutual-TLS), che non giustificano la complessità di un sistema a mutua autenticazione (ad es., accessi in sola consultazione, applicazioni Web o sistemi IoT[44]) a livello di trasporto. Fermo l'obbligo di usare HTTPS, nasce l'esigenza di venire incontro a diversi scenari e definire per essi modelli di autenticazione e di trust differenziati. Questi aspetti verranno definiti in «Pattern e Profili di Interoperabilità».

3.3 SOAP

Il protocollo SOAP (Simple Object Access Protocol) è stato sviluppato per superare le limitazioni imposte dai protocolli precedenti per l'interazione distribuita basata su oggetti (CORBA, Java/RMI, DCOM) relative alla distribuzione a livello Internet delle macchine interessate ed ai vincoli imposti dal punto di vista delle tecnologie di implementazione.

La versione corrente della specifica SOAP è la 1.2 del 27 Aprile 2007[45]. La specifica definisce due stili di comunicazione (communication modes):

- quello basato su chiamata a procedura (RPC-like),
- e quello basato su scambio di documenti (document style).

In combinazione ad essi, il protocollo definisce delle modalità di scambio dell'informazione:

- interazioni one-way (dal client al server),
- interazioni request/response,
- invio di notifiche (interazione one-way dal server al client)
- e solicit/response (interazione request/response in cui la request è inviata dal server).

Le ultime due modalità sono poco utilizzate in pratica e fuori dai profili di interoperabilità standard, quindi il loro utilizzo è vietato.

Il protocollo SOAP definisce tre componenti fondamentali:

- una envelope (letteralmente «busta da lettere») che definisce la struttura del messaggio e come processarlo;
- un insieme di regole di codifica per esprimere istanze di tipi di dato definiti a livello applicativo;
- una convenzione per rappresentare lo stile di interazione RPC.

La definizione del protocollo è pensata per essere indipendente dal protocollo sottostante. In particolare, SOAP può operare (tramite i cosiddetti binding) su diversi protocolli di trasporto inclusi HTTP, SMTP, TCP, UDP o JMS. Sebbene implementazioni sono state proposte per ognuno di questi casi (in special modo JMS per interazioni asincrone), il mercato ha premiato principalmente soluzioni sincrone basate su HTTP.

Una delle caratteristiche che contraddistinguono il protocollo SOAP è la sua estensibilità. In particolare si indica con WS-* lo stack di estensioni costruite su SOAP, molte delle quali hanno avuto grande successo in termini di implementazioni disponibili. Queste estensioni permettono di avere su SOAP una serie di funzionalità che su

[41] Circolare AgiD 18 aprile 2017, n.2/2017 http://www.gazzettaufficiale.it/eli/id/2017/05/05/17A03060/sg

[42] Cf. https://tools.ietf.org/html/rfc5280#section-4.2.1.3

[43] Cf. https://tools.ietf.org/html/rfc7457

[44] Un esempio potrebbe essere una interfaccia di servizio di un comune che permette di avere in tempo reale la situazione dei posti liberi nei parcheggi comunali. Un sistema di trasporto integrato regionale accede al dato su tutti i parcheggi dei comuni della regione e mostra in tempo reale la situazione aggregata dei parcheggi disponibili. In questo scenario, l'informazione scambiata (numero posti liberi) è poco sensibile e eventuali apparati installati presso i parcheggi non giustificano il costo necessario di una configurazione a prova di non ripudio ed una mutua autenticazione TLS. Esempi di tali scenari (con standard diversi da SPCoop) sono emersi in E015, sviluppato in occasione di Expo nella Regione Lombardia.

[45] Cf. https://www.w3.org/TR/soap12-part1/

altri protocolli devono essere costruite ad-hoc. Lo svantaggio di questa soluzione è che il protocollo introduce un overhead di processamento che fa preferire altre soluzioni in determinati contesti.

Tra le estensioni supportate dai framework più diffusi abbiamo:

- WS-Addressing è un modo standard per includere informazioni circa l'instradamento dei messaggi (ad es., l'interfaccia di servizio a cui inviare la risposta o da contattare in caso di errore).
- WS-Security è la specifica che descrive le politiche di sicurezza implementate a livello applicativo dalle interfacce di servizio. In particolare, WS-Security include meccanismi per autenticazione e autorizzazione, confidenzialità, integrità e firma digitale.
- WS-Trust è una estensione a WS-Security che permette di richiedere, rinnovare e validare token di sicurezza. Permette inoltre di verificare la relazione di mutua fiducia su un canale sicuro.
- WS-Federation è una estensione che permette a differenti domini di sicurezza di scambiare informazioni circa identità, attributi di autorizzazione ed autenticazione.
- WS-ReliableMessaging permette di consegnare in maniera affidabile (ad es., nell'ordine corretto) messaggi SOAP in presenza di problemi di rete e di inattività di componenti software e di sistema.
- WS-AtomicTransaction è una estensione che permette di ottenere la proprietà tutto o niente per un gruppo di operazioni. Essa definisce tre protocolli (completamento, two-phase commit volatile e two-phase commit durevole) che sono implementati dal framework
- WS-Coordination.
- WS-Choreography è la specifica per la definizione di coreografie. Una coreografia specifica i passi relativi allo scambio di messaggi tra diversi soggetti che si integrano.
- WS-BPEL è la specifica per la definizione di orchestrazioni.
- WS-Coordination è un framework estensibile per il coordinamento di web service (corrispondenti alle interfacce di servizio). In particolare esso spiega come implementare (e quindi è preso a riferimento dalle varie implementazioni dello stack WS-*) i protocolli di coordinamento inclusi quelli descritti da WS-AtomicTransaction.

La specifica delle interfacce di servizio SOAP è effettuata tramite Web Services Description Language - WSDL[46]. Oltre ad indicare le funzionalità offerte dall'interfaccia di servizio dal punto di vista funzionale, esso permette anche di definire le caratteristiche non funzionali tramite le estensioni WS-Policy[47] che permettono di specificare le varie componenti della QoS.

3.3.1 Indicazioni di utilizzo

La specifica SOAP permette la definizione di specifici profili di interoperabilità, imponendo alcune restrizioni circa i tipi ed i formati scambiati. Il profilo di interoperabilità secondo il quale interfacce di servizio di tipo SOAP andranno implementati è la versione 2.0 del Basic Profile[48] (nel seguito BP2) definito dal WS-I (Web Services Interoperability Organization) ed ora confluito in OASIS. BP2 è basato su SOAP 1.2 e WS-Addressing (per il dispatching dei messaggi a livello applicativo, in particolare nel caso di interazioni asincrone). Tra le molte indicazioni, BP2 definisce anche la modalità di gestione degli errori. In particolare, oltre all'utilizzo dei codici di errore HTTP si richiede che il ricevente sia in grado di gestire le SOAP fault[49] che quindi devono, obbligatoriamente, essere emesse dall'erogatore a fronte di errori.

[46] Cf. https://www.w3.org/TR/wsdl20-primer/

[47] Cf. https://www.w3.org/TR/ws-policy/

[48] Cf. http://docs.oasis-open.org/ws-brsp/BasicProfile/v2.0/cs01/BasicProfile-v2.0-cs01.html

[49] Le SOAP fault devono essere accompagnate anch'esse da un appropriato codice di errore HTTP. Per SOAP fault comuni si può fare riferimento a "https://www.w3.org/TR/2007/REC-soap12-part2-20070427/#tabresstatereccodes" <https://www.w3.org/TR/2007/REC-soap12-part2-20070427/#tabresstatereccodes>_.

3.3.2 Sicurezza

Per quanto riguarda la sicurezza, l'ultimo profilo standard definito da OASIS è il Basic Security Profile 1.1[50]. Il profilo è datato ma le considerazioni sono ancora valide. Per quanto riguarda le versioni dei protocolli, si devono rispettare i vincoli imposti dal Modello di Interoperabilità 2018 in questo documento.

E" importante, nel caso si richiedessero funzionalità di autorizzazione, autenticazione e non ripudio, oltre che di riservatezza (coperta dall'utilizzo obbligatorio di HTTPS[51]) fare affidamento alle tecnologie di autenticazione ed autorizzazione a livello applicativo. Il Basic Security Profile 1.1, basato sull'estensione WS-Security, suggerisce l'uso di SAML 2.0. Come detto, rispetto alle tecnologie di autenticazione ed autorizzazione, ci sono alcuni domini applicativi per i quali OAuth2 o OpenId sono più appropriati. In questi ultimi casi, fermo restando l'utilizzo della XML Signature definita in WS-Security per quanto riguarda il non ripudio, l'utilizzo di token di autorizzazione ed autenticazione non SAML richiede la definizione di request header custom[52].

3.3.3 Uniformità e naming

Non esistono standard riguardanti il naming in ambito SOAP. Le best-practice prevedono l'utilizzo di CamelCase[53] (con prima lettera maiuscola, anche noto come PascalCase) per endpoint, porte, operazioni e parametri.

Quando le risorse contengono link e riferimenti a risorse esterne, si dovrebbero usare le specifiche indicate in IANA registered link relations[54] trasformando il Kebab Case[55] utilizzato con il CamelCase.

3.4 REST

REpresentational State Transfer (REST) è uno stile architetturale, proposto da Roy Fielding[55], che consente di accedere e manipolare rappresentazioni testuali di risorse web usando un insieme predefinito di operazioni stateless. Le interfacce di servizio che seguono lo stile architetturale REST sono dette RESTful o semplicemente REST. Con il termine «risorsa web» si intendevano inizialmente documenti e file identificati da una URL sul World Wide Web. Oggi il termine ha un'accezione molto più generica ed astratta, andando ad indicare ogni cosa o entità che possa essere identificata tramite una URI (si noti il passaggio da URL ad URI che indica l'indipendenza dal protocollo di recupero dei dati). Nel caso dell'applicazione di questo stile architetturale ad HTTP, le operazioni stateless a cui si fa riferimento sono GET, POST, PUT, DELETE a cui corrispondono operazioni di tipo Create-Read-Update-Delete - CRUD sulla risorsa. Questo approccio favorisce l'uniformità delle interfacce di servizio.

Il termine «state transfer» indica che è il client a dovere riportare tutte le informazioni necessarie al soddisfacimento di una richiesta, e il server non memorizza alcun tipo di informazione circa la sessione; quindi le interfacce di servizio sono, per definizione, stateless. Questo tipo di approccio favorisce l'introduzione di meccanismi di caching. In particolare, le risposte del server devono contenere una indicazione sul fatto che le risposte possano essere messe in cache o meno. Opzionalmente, inoltre, è possibile per il server richiedere l'esecuzione di alcune funzionalità al client tramite il passaggio di codice da eseguire (ad es., codice JavaScript da eseguire nel browser).

Talvolta, il termine Resource Oriented Architecture - ROA è usato per denotare l'architettura REST in opposizione alle Service Oriented Architecture - SOA, indicando la predilezione della prima per l'accesso basato su risorsa più che sulla chiamate ad operazioni di tipo RPC. Il dibattito sulla correttezza o meno di implementare operazioni RPC utilizzando REST è molto acceso, ma come dato di fatto numerose iniziative di API commerciali e non, utilizzano interfacce di servizio REST anche per effettuare RPC. Il concetto di REST è inoltre molto spesso legato, anche se non per definizione, alle architetture dette a microservizi[56], caratterizzate da elevata modularità, per via della leggerezza del protocollo.

[50] Cf. http://www.ws-i.org/Profiles/BasicSecurityProfile-1.1.html
[51] HTTPS è richiesto dal modello di interoperabilità ma non da BP2.
[52] Cf. https://developers.google.com/adwords/api/docs/guides/call-structure
[53] Cf. https://it.wikipedia.org/wiki/Notazione_a_cammello
[54] Cf. http://www.iana.org/assignments/link-relations/link-relations.xml
[55] Cf. https://it.wikipedia.org/wiki/Kebab_case
[55] Cf. http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
[56] Cf. Sam Newman (2015): Building Microservices.

A differenza delle interfacce di servizio SOAP, per cui una serie di standard è definita e mantenuta da OASIS (cf. stack WS-*), per le interfacce REST sono disponibili singoli standard e best-practice.

Per la specifica delle interfacce REST esistono due grandi iniziative: OpenAPI e RAML. Sebbene simili dal punto di vista dello sviluppatore di interfacce di servizio, la specifica RAML è più indirizzata alla creazione automatica di server e di client per API, mentre OpenAPI (attualmente nella versione OpenAPI v3) contiene elementi più descrittivi per la documentazione e la catalogazione (che invece sono disponibili in RAML come estensioni ad-hoc) e si sta imponendo come standard de-facto. Altri standard proposti in passato, quali Web Application Description Language - WADL, hanno avuto scarso successo e nei framework in cui sono stati utilizzati si sta optando per il passaggio ad OpenAPI v3[57]. Per queste ragioni il ModI 2018 impone l'uso di OpenAPI v3.

E' possibile assicurare la conversione tra le differenti rappresentazioni delle interfacce REST tramite tool automatici.

Legato al concetto di specifica nel mondo REST è quello di *Hypermedia As The Engine Of Application State - HATEOAS*. Secondo questo approccio, accedendo ad una risorsa, la risposta del server contiene hyperlink ad altre azioni che possono essere eseguite sulla risorsa[58]. HATEOAS permette di scoprire dinamicamente le operazioni presenti in una interfaccia di servizio e può essere utilizzato come approccio complementare (non sostitutivo) alla specifica.

3.4.1 Indicazioni di utilizzo

L'interfaccia di servizio REST deve utilizzare l'HTTP verb più adatto all'operazione come indicato in RFC 7231[59]. In particolare i metodi:

- GET, HEAD, DELETE: non devono avere un payload.
- GET, HEAD: devono essere «safe», cioè devono essere essenzialmente read-only. Il client in questo caso non si aspetta e non richiede un cambiamento dello stato della risorsa.
- GET, HEAD, PUT, DELETE: devono essere idempotenti, cioè chiamate multiple con richieste identiche si comportano come singole richieste.
- POST: dovrebbe implementare un meccanismo di idempotenza per evitare di duplicare eventuali entry.

Ove necessario, specialmente ai fini del caching e l'accesso concorrente alle risorse[60], occorre fare leva sugli ETag[61] (degli identificatori univoci di versione delle risorse). Infine l'utilizzo di eventuali header HTTP non deve sostituire i parametri da passare in una GET.

3.4.2 Sicurezza

Lo standard di riferimento per la firma e la crittografia in ambito JSON/REST è Javascript Object Signing and Encryption[62] (di seguito JOSE), menzionato nelle Linee Guida AgID[63] ed in «European Telecommunications Standards Institute - Security of the mission critical service»[64]. JOSE è un framework per la sicurezza comprendente diverse componenti tra cui centrale è il JSON Web Token[65] (di seguito JWT). JWT è uno standard per la definizione di token di accesso basato su JSON Web Signature[66] (di seguito JWS)) e JSON Web Encryption[67] (si seguito JWE) di cui eredita ed estende gli header. Il token JWT è passato in REST tramite l'header HTTP

[57] Cf. https://www.openapis.org/

[58] Si supponga ad esempio una operazione HTTP GET http://api.domain.com/management/departments che restituisce informazioni circa i reparti. Il singolo reparto può contenere link relativi ad altre operazioni come quella per ottenere gli impiegati del reparto:{«departmentId»: 10,»departmentName»: «Administration»,»links»: [{«href»: «[[http://api.domain.com/management/departments/10/employees]{.underline}](http://api.domain.com/management/departments/10/employees)»,»rel»: «employees», «type» : «GET» }]}

[59] Cf. https://tools.ietf.org/html/rfc7231#section-4.3

[60] C.f. https://en.wikipedia.org/wiki/Optimistic_concurrency_contro

[61] Cf. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/ETag

[62] Cf. http://www.etsi.org/deliver/etsi_ts/118100_118199/118103/02.04.01_60/ts_118103v020401p.pdf

[63] Cf. http://www.agid.gov.it/agenda-digitale/infrastrutture-architetture/cert-pa/linee-guida-sviluppo-sicuro

[64] Cf. http://www.etsi.org/deliver/etsi_ts/133100_133199/133180/14.02.00_60/ts_133180v140200p.pdf

[65] Cf. https://tools.ietf.org/html/rfc7519

[66] Cf. https://tools.ietf.org/html/rfc7515

[67] Cf. https://tools.ietf.org/html/rfc7516

Authorization utilizzando lo schema Bearer[68]. Il token in OpenID Connect è espresso per esempio direttamente come JWT.

Per ulteriori dettagli sulla sicurezza, si vedano anche:

- OWASP REST Security Cheat-Sheet[69];
- OWASP API Security Project[70];
- JWS - Security Considerations[71].

3.4.3 Uniformità e Naming

In questa sezione introduciamo le best practice da utilizzare per interfacce di servizio REST. In prima istanza, ogni endpoint deve essere univocamente associato alle componenti Scheme, Authority e Path di un URL[72].

La componente Authority dell'URL:

- dovrebbe essere associata al dominio del sito Istituzionale dell'erogatore presente su IndicePA, anche tramite il prefisso «api», ad esempio un erogatore con sito istituzionale «erogatore.gov.it», potrebbe usare come authority «api.erogatore.gov.it»;
- può essere associata al dominio di un ente che l'erogatore ha delegato (ad es., una società in-house, un consorzio di comuni).

Per quanto riguarda la componente Path, i nomi utilizzati non devono usare abbreviazioni e acronimi non universalmente riconosciuti[73]. Inoltre, il Path dovrebbe essere semplice, intuitivo e coerente[74].

Il campo Query dovrebbe:

- essere in snake_case minuscolo;
- non essere in camelCase;
- utilizzare ove possibile dei nomi comuni per le funzionalità di paginazione, ricerca ed embedding/resource-expansion (ad es., limit, offset, q, sort).

Le risposte in formato JSON[75], devono restituire sempre oggetti strutturati con attributi chiave-valore, non semplici liste. Questo permette di estendere facilmente le risposte introducendo in versioni successive ulteriori attributi (ad es., di paginazione).

[68] Lo schema Bearer, inizialmente introdotto nella specifica OAuth2 ma poi utilizzato in altri contesti, ha la forma «Authorization: Bearer <token>» dove il token JWT è codificato in base64.

[69] Cf. https://www.owasp.org/index.php/REST_Security_Cheat_Sheet

[70] Cf. https://www.owasp.org/index.php/OWASP_API_Security_Project

[71] Cf. https://tools.ietf.org/html/rfc7515#section-10

[72] Cf. https://tools.ietf.org/html/rfc3986

[73] Cf. https://linee-guida-cataloghi-dati-profilo-dcat-ap-it.readthedocs.io/it/latest/catalogo_elementi_obbligatori.html#titolo-dct-title Ad esempio,

- sono ammesse stringhe come «id», «args» o «stdin» ed abbreviazioni come «tcp» ed «udp»;
- stringhe come «codice fiscale» andrebbero espresse per esteso con «codice_fiscale» o «tax_code», e non con «cod_fiscale», «cod_fisc» o «cf».

[74] Alcune indicazioni in questo senso:

- usare parole minuscole separate da trattino «-«;
- usare nomi al plurale per le risorse e al singolare per l'accesso alla singola risorsa;
- ispirarsi alle convenzioni utilizzate a livello europeo (ad es., Core Vocabularies/Dizionari Controllati, Direttiva Europea INSPIRE 2007/2/CE);
- non contenere verbi (ad es., api.example.com/ospedale/prenota/);
- uniformarsi a quello di altre interfacce di servizio a livello Europeo quando ciò vada nella direzione dell'interoperabilità e della semplicità.
- In generale tutte le stringhe in inglese dovrebbero utilizzare la dizione US per evitare ambiguità come ad es., «color» vs «colour», «flavor» vs «flavour»).

[75] Cf. https://tools.ietf.org/html/rfc7159

In caso di errore, le risposte devono usare schemi standard come quello definito nella RFC 7807 - Problem Details for HTTP APIs - IETF Tools[76] in particolare utilizzando il content type application/problem+json nella risposta.

Quando le risorse contengono link e riferimenti a risorse esterne, si dovrebbero usare le specifiche indicate in IANA registered link relations[77].

Tutti i riferimenti dovrebbero contenere URL comprensivi di schema.

3.4.4 Throttling ed indisponibilità del servizio

Nelle API basate su REST, meccanismi di throttling vengono implementati al fine di garantire l'accessibilità delle interfacce di servizio ed evitare in alcuni casi la raccolta non autorizzata (web-harvesting) dei dati.

Poiché l'RFC 6585 prevede per la gestione del throttling il solo status code 429, nel Modl2018 si richiede di notificare al fruitore lo stato del throttling ed eventuali limiti come segue:

- restituire in ogni risposta valida i valori globali di throttling tramite i seguenti header HTTP:
  - X-RateLimit-Limit: limite massimo di richieste per un endpoint;
  - X-RateLimit-Remaining: numero di richieste rimanenti fino al prossimo reset;
  - X-RateLimit-Reset: il numero di secondi mancanti al momento in cui il limite verrà reimpostato.
- utilizzare gli HTTP status code nelle risposte:
  - HTTP 429 (too many requests), insieme ai rate limit di cui al punto precedente, se il rate limit viene superato;
  - HTTP 503 (service unavailable) se l'infrastruttura non può erogare le operazioni offerte nei tempi attesi (definiti dalla SLA associata all'interfaccia di servizio).
- nei casi 429 e 503 gli erogatori dovrebbero notificare al client dopo quanti secondi ripresentarsi tramite l'header Retry-After[78] (pratica anche detta "circuit breaker"), anche implementando meccanismi di exponential back-off. L'RFC prevede che questo header possa essere utilizzato sia in forma di data che di secondi, ma il Modl2018 vieta l'utilizzo del formato data poiché se non implementato correttamente potrebbe aggravare lo stato dei sistemi.

I fruitori dell'interfaccia di servizio devono impegnarsi a rispettare le indicazioni provenienti dagli header ed dagli status code di cui sopra.

3.5 Message Broker

Un message broker è un modulo software che permette l'integrazione asincrona tramite scambio di messaggi. Questo tipo di interazione è fortemente disaccoppiata perché l'invio del messaggio avviene su un canale in cui è responsabilità del message broker consegnare il messaggio ai soggetti interessati. Il compito del message broker non è però solo quello di passare dati, in quanto esso si occupa anche di aspetti legati alla sicurezza, priorità dei messaggi, inoltro ordinato.

I middleware focalizzati sul fornire integrazione basata su messaggi vengono detti Message Oriented Middleware - MOM.

Un message broker supporta solitamente diverse modalità di interazione:

- Publish/Subscribe. In questo scenario un publisher invia dei messaggi sul canale ed il message broker li invia a diversi ricevitori sulla base di sottoscrizioni. Questo tipo di interazione supporta diversi scenari tra cui uno a molti o molti a molti;
- Queuing. In questo caso un richiedente invia una richiesta su una coda specifica (corrispondente all'erogatore) e l'erogatore invia la risposta sulla medesima coda; di fatto è una realizzazione asincrona della modalità request/reply;

[76] Cf. https://tools.ietf.org/html/rfc7807
[77] Cf. http://www.iana.org/assignments/link-relations/link-relations.xml
[78] Cf. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Retry-After

- Store/Forward. In questo caso il broker memorizza i messaggi e quindi inoltra agli interessati.

Un caso particolare di message broker è costituito dagli integration broker. Rispetto ad un message broker, questi si occupano anche della trasformazione di messaggi dai formati sorgente a quelli manipolabili dai riceventi/sottoscrittori.

L'utilizzo di message broker è consigliato in alcuni casi d'uso in cui l'interazione è asincrona o di tipo publish/subscribe (ad es., Internet-of-Things - IoT, aggregatori di dati pubblici).

Varie tecnologie e realizzazioni di message broker hanno storicamente supportato svariati protocolli quali STOMP[80], XMPP[81], MQTT[82], OpenWire[83] e AMPQ[84]. Oggigiorno, sebbene in determinati contesti essi vengano attualmente ancora utilizzati (ad es., in contesti intra-dominio o in casi particolari quali l'IoT in cui si preferiscono protocolli binari efficienti come MQTT), si preferiscono, in ambito di integrazione di sistemi, approcci in cui l'interfacciamento con i message broker avviene tramite interfacce di servizio REST. In particolare sono disponibili sia soluzioni native che wrapper per implementazioni di altri protocolli.

I vantaggi di questo approccio includono la possibilità di utilizzare le modalità di autenticazione, autorizzazione, throttling ed accounting già discussi riguardo alla tecnologia REST, e la risoluzione di possibili problematiche legate all'attraversamento di firewall e proxy.

Sebbene, a seconda delle implementazioni, le diverse interfacce di servizio REST per l'accesso a message broker differiscano per funzionalità offerte e modi di modellare code, topic/sottoscrizioni, si possono astrarre e seguenti comportamenti dei metodi HTTP:

- il metodo POST viene utilizzato per l'invio di messaggi e la creazione di topic/sottoscrizioni e code;
- il metodo GET viene utilizzato per consumare messaggi da code e topic/sottoscrizioni;
- il metodo DELETE viene utilizzato per l'eliminazione di topic/sottoscrizioni e code ed in alcuni casi per segnalare il fatto che un messaggio è stato consumato.

Il metodo PUT viene di solito utilizzato per modificare le proprietà di topic/sottoscrizioni e code.

3.6 Considerazioni comparative

Un primo criterio per orientarsi tra le tecnologie di integrazione presentate nelle Sezioni 3, 4 e 5 è quella di distinguere le tecnologie adatte per una interazione sincrona da quelle adatte ad una interazione asincrona. Riguardo a questa distinzione si può fare riferimento alla seguente tabella:

	SOAP	**REST**	**Message Broker**
Interazione Sincrona	✓	✓	
Interazione Asincrona	✓*	✓*	✓

SOAP (inteso come stack WS-*), come si evince dalla tabella, può essere utilizzato sia per interazioni sincrone che per interazioni asincrone.

In particolare, in SOAP, l'interazione asincrona può essere ottenuta sia su protocolli di trasporto sincroni che su protocolli di trasporto asincroni. Nonostante la specifica supporti questo genere di interazioni, il supporto di middleware e framework di sviluppo a queste funzionalità è limitato. Per quanto riguarda REST invece, nonostante non originariamente previsto dalla specifica, si è visto in Sezione 5 come esso venga utilizzato come interfaccia di servizio per message broker.

Per quanto riguarda l'interazione sincrona (stile chiamata a procedura o accesso CRUD a risorsa), diverse considerazioni tecnologiche devono essere effettuate. SOAP e REST utilizzano HTTP in due modi differenti.

[80] Cf. https://stomp.github.io/
[81] Cf. https://xmpp.org/
[82] Cf. http://mqtt.org/
[83] Cf. http://activemq.apache.org/openwire.html
[84] Cf. https://www.amqp.org/

Mentre SOAP lo utilizza come un protocollo di trasporto, REST lo utilizza come un protocollo applicativo. La diffusione dell'accesso alla rete ha aumentato il carico sulle infrastrutture IT, inoltre reti migliori hanno aumentato le aspettative in termini di latenza. L'IETF nel tempo ha risposto a queste esigenze:

- migliorando la semantica di HTTP, facendo leva sui campi Header, Status e Method RFC7230[85], RFC7231[86];
- codificando le semantiche di caching RFC7234[87] e controllo della concorrenza RFC7232[88] per facilitare l'implementazione di interfacce di servizio stateless, che possano scalare senza che i bilanciatori conoscano la logica applicativa;
- orientandosi verso formati più leggeri (ad es., JSON).

Queste innovazioni fanno preferire l'approccio REST:

- Quando è possibile esprimere la logica applicativa tramite la semantica HTTP, poiché si guadagna:
  - espressività (ad es., il contesto d'utilizzo è chiarito da Method e Status);
  - mobile-ready (esporre un'API in un'app con un http-wrapper);
  - performance e scalabilità (ad es., possibilità di ruotare le chiamate in base al Method, senza inspection applicativa).
- Quando le API devono essere fruibili anche da mobile/web;
- L'accesso avviene in maniera prevalente con operazioni di tipo CRUD sui dati.

Quindi rispetto a quanto discusso in "Presentazione del Modello di Interoperabilità 2018" sui paradigmi di cooperazione, questo suggerisce l'uso di REST nei casi di condivisione di dati e di composizione applicativa, quando le operazioni componenti sono prevalentemente orientate a fornire dati. Il servizio digitale corrispondente all'interfaccia di servizio è prevalentemente informativo (cf. Sezione 1).

L'utilizzo di SOAP è suggerito:

- Quando la semantica HTTP non è sufficiente ad esprimere la logica applicativa ed è necessario usare un protocollo di messaging ulteriore con dei propri header;
- Se la specifica applicazione richiede la creazione di interfacce di servizio principalmente *stateful*, cioè l'accesso ad informazioni di contesto o la gestione dello stato della conversazione[89]. SOAP prevede estensioni (eg. relative al concetto di transazione) che con altri approcci (ad es., REST) devono essere costruite ad-hoc per la specifica applicazione.
- Nel caso si necessiti di processamento asincrono che non sia possibile implementare con semantiche HTTP;
- Quando servono specifiche assicurazioni circa la QoS (quali quelle fornite dall'estensione WS-ReliableMessaging).

Quindi rispetto a quanto discusso in "Presentazione del Modello di Interoperabilità 2018" sui paradigmi di cooperazione, questo suggerisce l'uso di SOAP nei casi di processo inter-PA e di composizione applicativa quando le operazioni componenti offrono delle logiche complesse.

La tabella seguente riporta alcuni aspetti tecnologici che devono essere tenuti in considerazione (le celle in cui è presente «-» indicano che l'aspetto in questione non è considerato e standardizzato, e quindi è a cura dello specifico progetto/applicazione indirizzarlo attraverso sviluppi ad-hoc)

[85] Cf. https://tools.ietf.org/html/rfc7230
[86] Cf. https://tools.ietf.org/html/rfc7231
[87] Cf. https://tools.ietf.org/html/rfc7234
[88] Cf. https://tools.ietf.org/html/rfc7232
[89] Come nel caso di processi amministrativi sia completamente automatizzati (short-running) sia con intervento umano o comunque long-running.

	SOAP (WS-*)	REST
Formato Payload	XML	Tutti (JSON nella maggior parte dei casi)
Identificazione delle operazioni	URI, WS-Addressing	URI
Descrizione delle interfacce di servizio	WSDL	RAML, OpenAPI
Affidabilità	WS-ReliableMessaging	-
Sicurezza	HTTPS, WS-Security	HTTPS, JWT
Transazioni	WS-AtomicTransaction, WS-BusinessActivity	-
Composizione di interfacce di servizio	WS-Choreography WS-BPEL	-

In letteratura, talvolta si indica con contract-first una metodologia di progetto che parte dalla specifica dell'interfaccia senza considerare possibili vincoli di implementazione, e successivamente si occupa di come realizzare tale interfaccia al di sopra di eventuali realizzazioni esistenti. In alternativa, si parta di contract-last (che potremmo anche indicare come implementation-first) quando invece eventuali vincoli di realizzazione guidano la specifica dell'interfaccia. SOAP supporta naturalmente entrambi gli approcci, in quanto lo sviluppo di un'interfaccia di servizio origina dalla definizione dell'interfaccia o dalle segnature dei metodi utilizzati nello sviluppo, mentre in REST l'interfaccia è definita dagli http verb associati alle operazioni CRUD, riportando il contratto alla definizione delle risorse. La differenza appare ininfluente nel caso di progettazione e realizzazione di sistemi nuovi, ma non in presenza di sistemi legacy. Quando l'interfaccia di servizio è vincolata dalla presenza di un sistema esistente o legacy, essa è definita a posteriori rispetto all'implementazione. In questo caso non essere limitati dai verb http (eg. usando SOAP) appare semplificare il lavoro di modellazione e realizzazione dell'interfaccia di servizio evitando di mappare risorse su procedure legacy.

Talvolta si parla di REST indicandolo come contract-less (REST)[90], proprio ad indicare il fatto che l'interfaccia è definita dagli http verb; a rigore però vanno comunque progettate le giuste risorse da esporre su cui effettuare operazioni CRUD, e quindi più che essere senza contratto, è il contratto che ha una forma differente.

Nel modo REST, il principio secondo cui l'interfaccia di servizio (in questo caso l'API) deve essere il primo artefatto di progettazione, viene recentemente indicato come API-first[91] ed è largamente adottato da molte organizzazioni private, ed anche framework di interoperabilità nazionali come quello inglese[92].

Nel caso invece di nuovi sistemi, la progettazione dell'interfaccia può essere effettuata sia in un'ottica contract-first che contract-less. In un'ottica contract-first, la specifica dell'interfaccia viene effettuata a tavolino a partire dalle macro-operazioni che si vogliono offerte dal sistema finale. Nel caso di accesso basato su risorsa (in ottica ROA), essendo in realtà le operazioni da effettuare già predefinite (operazioni CRUD), il tipo di progettazione è contract-less. Vanno però definite le risorse che il sistema deve esporre, quindi una qualche forma di progettazione preventiva all'implementazione è comunque prevista (cioè, la specifica delle risorse).

Nel progetto di interfacce di servizio SOAP occorre definire, oltre alle macro-operazioni, anche le strutture XML per la rappresentazione dei dati. Le operazioni possono essere raggruppate in base a caratteristiche quali: area funzionale (o area di business), requisiti di sicurezza (ad es. meccanismi di autenticazione ed autorizzazione), oppure in base a fattori organizzativi quali la frequenza dei cambiamenti o pattern di gestione delle versioni. Il principio alla base di questo raggruppamento è quello di impattare il minor numero di fruitori quando avviene un cambiamento.

Nel progetto di interfacce di servizio REST invece occorre:

[90] Cf. Cesare Pautasso, Olaf Zimmermann, Frank Leymann: Restful web services vs. «big»" web services: making the right architectural decision. WWW 2008: 805-814.

[91] Cf. https://www.programmableweb.com/api-university/understanding-api-first-design In termini colloquiali, il principio può essere parafrasato in questi termini:

- L'API è la prima interfaccia dell'applicazione
- L'API viene prima dell'implementazione
- L'API deve essere descritta (ed addirittura essere auto-descrittiva, se possibile e fattibile)

[92] Cf. https://www.programmableweb.com/news/why-uks-government-data-service-takes-api-first-approach-to-datagovuk/elsewhere-web/2016/09/02

- Identificare le risorse che l'interfaccia di servizio manipolerà. Queste risorse sono solitamente i concetti base che stanno dietro ad un processo (ad es., un ordine di acquisto).
- Progettare gli URI seguendo i principi introdotti nella sezione relativa alla tecnologia REST.
- Scegliere il tipo di operazione disponibile per ognuna delle URI.
- Scegliere i collegamenti tra risorse da fornire nelle risposte. In quest'ottica l'approccio HATEOAS può risultare utile.
- Progettare le strutture JSON per la rappresentazione dei dati.

Il ModI 2018, come discusso nella Sezione 1, prevede che la progettazione parta della definizione delle interfacce di servizio, indipendentemente dalla tecnologia di realizzazione sia SOAP che REST, anche se con accorgimenti tecnici differenti nella sua realizzazione.

3.7 Altri approcci e tecnologie di integrazione

Nelle precedenti sezioni, sono state introdotte le principali tecnologie di integrazione. Accanto a queste, stanno emergendo altre modalità di integrazione che potrebbero essere proposte in futuro in affiancamento in casi d'uso molto specifici.

3.7.1 Datastore distribuiti

L'applicazione di tecnologie per datastore distribuiti è strettamente connessa, in ambito integrazione di sistemi, al mantenimento di database multi-tenant in cui, ad esempio, si richiede data locality per basi di dati di grandi dimensioni. In questo contesto, vanno considerati principalmente i file system ed i database distribuiti.

I file system distribuiti offrono interfacce basate su API per la memorizzazione di file e di oggetti e sono oggigiorno disponibili sia in soluzioni cloud pubbliche sia private. La sicurezza di queste soluzioni è soggetta agli stessi vincoli visti per l'utilizzo di interfacce di servizio nelle sezioni precedenti.

Tra i database distribuiti, grande interesse è stato suscitato da quelli basati su blockchain[93]. L'obiettivo di una blockchain è il mantenimento di un *libro mastro distribuito* (distributed ledger) mediante una rete peer-to-peer di nodi[94]. L'obiettivo è quello di avere un datastore capace di certificare transazioni e vincoli contrattuali, in cui il meccanismo di distribuzione certifica la validità degli stessi. In particolare, è possibile appurare la validità di *smart contract* (contratti intelligenti), certificando le precondizioni degli stessi. Il termine contratto spazia dal semplice scambio di denaro, ad es., la piattaforma BitCoin in cui la precondizione all'invio di denaro è il possesso del denaro stesso, a contratti complessi dove le precondizioni possono assumere una qualunque forma. L'integrità dei dati memorizzati è certificata da meccanismi basati su chiave pubblica. La maggior parte dei protocolli disponibili per la realizzazione di blockchain sono basati su scambio di messaggi su TCP/TLS o HTTPS.

In Estonia, il modello X Road[95] (equivalente al ModI 2018) ha promosso l'utilizzo di un ledger distribuito nell'ambito della Pubblica Amministrazione, anche se più a scopo di marketing[96] che per l'utilizzo degli aspetti precipui di una blockchain. Quello che è interessante è l'idea di un tracciamento distribuito delle decisioni prese da una Pubblica Amministrazione.

La tecnologia blockchain non è esente da rischi in quanto diversi tipi di attacco sono stati formulati che permettono la modifica dei contenuti e la creazione di ramificazioni della catena di transazioni alla base del libro mastro[97].

In conclusione, sebbene si tratti di una tecnologia che sta suscitando interesse, attualmente blockchain non sono considerate abbastanza mature per l'utilizzo nella Pubblica Amministrazione in settori strategici e il ModI 2018 ne sconsiglia al momento l'utilizzo. Inoltre deve ancora essere definito il modo di integrare ed interoperare tra PA utilizzando smart contract come interfacce di servizio, e le tipologie di transazioni che effettivamente hanno bisogno di requisiti tali per cui la blockchain sia la giusta soluzione.

[93] Cf. https://it.wikipedia.org/wiki/Blockchain

[94] Una rete di calcolatori si definisce peer-to-peer, quando le macchine componenti (i nodi) non sono organizzati gerarchicamente ma svolgono delle funzionalità paritarie.

[95] Cf. https://e-estonia.com/solutions/security-and-safety/ksi-blockchain

[96] Cf. https://techcrunch.com/2018/04/19/do-you-need-a-blockchain/

[97] Cf. https://www.multichain.com/blog/2017/05/blockchain-immutability-myth/

L'utilizzo di datastore distribuiti potrebbe in futuro affiancare l'integrazione basata su altre tecnologie più consolidate. Le future linee guida dovranno tenere in considerazione per queste tecnologie:

- requisiti di latenza e consistenza (ad es., eventual consistency, autoritatività);
- le modalità di logging e auditing associate alla trasmissione dei dati;
- le modalità operative di manutenzione;
- la standardizzazione delle interfacce di accesso.

3.7.2 Esposizione di open data

Una modalità di integrazione, importante specialmente negli scenari A2B e A2C, è quella basata sull'esposizione da parte delle PA di *open data*. Gli open data devono essere fruibili, ed essere inseriti ove possibile nel contesto dei Base Register definiti nell"EIF[98], standardizzando gli schemi e le modalità di fruizione.

Vista la progressiva crescita dei dataset, gli open data dovrebbero essere erogati in modo da ridurre gli impatti infrastrutturali sull'erogatore.

Come indicato nelle linee guida nazionali per la valorizzazione del patrimonio informativo pubblico[99] pubblicate da AgID nel 2014, l'obiettivo è quello di mettere a disposizione i dati aperti in formato Linked Open Data - LOD ai fini dell'integrazione, il che prevede l'esposizione di dati in formato W3C RDF e SPARQL (secondo il cosiddetto modello del *Semantic Web*). A tal fine gli SPARQL endpoint costituiscono le interfacce di servizio. Le query in formato SPARQL vengono inviate su endpoint HTTP. Un altro approccio possibile, sempre nel rispetto dei dizionari comuni, è quello di utilizzare un approccio ROA basato su interfacce REST[100].

Un'interessante evoluzione dell'approccio REST (di cui eredita molti dei vantaggi, quali ad esempio la leggerezza e l'utilizzo dei verbi HTTP) che può risultare utile nell'esposizione di open data è quello basato su GraphQL[101]. In particolare, mentre per l'estrazione di dati complessi l'approccio basato su interfacce di servizio REST richiede diverse chiamate, GraphQL introduce un linguaggio che permette l'esecuzione di interrogazioni complesse sulle risorse.

In tutti i casi presentati, restano valide le indicazioni contenute nelle sezioni precedenti circa la sicurezza nell'esposizione delle interfacce di servizio.

[98] Cf. https://joinup.ec.europa.eu/asset/eia/description
[99] Cf. http://lg-patrimonio-pubblico.readthedocs.io/it/latest/index.html
[100] Cf. Massimo Mecella, Francesco Leotta (2017): Migliorare l'accesso agli open data pubblici: tecnologie e approcci, https://www.agendadigitale.eu/cittadinanza-digitale/pa-tecnologie-e-approcci-per-migliorare-laccesso-ai-dati-aperti/
[101] Cf. https://graphql.org{]}(https://graphql.org/