Linee Guida per le attività tecniche di progettazione, sviluppo, installazione, distribuzione e manutenzione di servizi informatici

Versione: 1.3

Sommario

Indice degli allegati

1. Introduzione

Il Codice dell'Amministrazione Digitale (CAD) è un testo unico che ha riunito ed organizzato le norme riguardanti l'informatizzazione della Pubblica Amministrazione nei rapporti con i cittadini e le imprese. Adottato con il D.Lgs n.82 del 7 marzo 2005, è stato poi ripetutamente modificato e integrato, per promuovere e rendere effettivi i diritti di cittadinanza digitale.

Con gli ultimi interventi, il CAD è stato razionalizzato nei suoi contenuti ed è stata avviata un'azione di deregolamentazione, sia semplificando il linguaggio, sia sostituendo le precedenti regole tecniche con linee guida, a cura dell'Agenzia per l'Italia Digitale (AGID), la cui adozione risulta più rapida e reattiva in accordo alla continua evoluzione tecnologica.

Nella direzione indicata dal CAD e sulla spinta del processo di informatizzazione della PA si pone il Piano Triennale per l'informatica nella Pubblica Amministrazione, frutto della stretta collaborazione tra l'Agenzia per l'Italia Digitale e il Dipartimento per la Trasformazione Digitale, con lo scopo di organizzare e supportare la transizione al digitale delle Pubbliche Amministrazioni. Per l'attuazione di quanto previsto dal CAD e dal Piano Triennale per l'Informatica nella Pubblica Amministrazione, vengono adottate atti e provvedimenti con valenza esplicativa e regolatoria (es. le "Linee Guida sull'interoperabilità tecnica delle Pubbliche Amministrazioni", "Il modello di Cloud per la PA", il "Censimento del Patrimonio ICT della PA", ecc.); in questo contesto, assumono particolare rilievo le Linee guida adottate da AGID in attuazione del CAD, con efficacia vincolante non solo per i soggetti pubblici (Pubbliche Amministrazioni; gestori di servizi pubblici, ivi comprese le società quotate, in relazione ai servizi di pubblico interesse; società a controllo pubblico), ma anche, in certi casi, per quelli privati (crf. art. 2, comma 3).

Per quanto riguarda le Linee guida, che come detto sviluppano e danno attuazione alle indicazioni tecniche del CAD, guidando la PA nel processo di trasformazione digitale, è opportuno richiamare qui almeno delle principali: Linee Guida di design per i servizi web delle pubbliche amministrazioni, Linee guida per lo sviluppo del software sicuro, Linee guida su acquisizione e riuso di software per le pubbliche amministrazioni, Linee Guida OpenID Connect in SPID, Linee guida sull'accessibilità degli strumenti informatici, Linee Guida sull'interoperabilità tecnica delle Pubbliche Amministrazioni. Ne risulta un quadro regolatorio estremamente complesso, articolato ed in continua evoluzione.

In accordo con il contesto regolatorio sopra descritto, il presente documento, emanato dalla Direzione Sistemi Informativi del Comune di Firenze, ha lo scopo di raccogliere un insieme di suggerimenti, raccomandazioni e prescrizioni da osservare nell'intero ciclo di vita dei servizi e delle forniture relativi al software, durante le fasi di progettazione, sviluppo, installazione, distribuzione e manutenzione, al fine di porre in essere attività in linea con le prescrizioni e gli standard tecnici di riferimento.

Si evidenzia che l'inosservanza delle indicazioni contenute in questo documento comporta la potenziale esposizione dell'Ente a criticità come:

- il disallineamento rispetto alle norme tecniche di settore sopra richiamate;
- il disallineamento rispetto al regolamento EU 2016/679 (GDPR), relativo alla tutela della privacy;
- la creazione di situazioni di *vendor lock-in* dovute alla conoscenza tecnica (detto anche *"know-how"*) del servizio limitata al solo *vendor* erogatore;
- il possibile aumento non controllato di costi ricorrenti o una tantum di servizi;
- la difficoltà da parte della Direzione Sistemi Informativi di fornire supporto su servizi realizzati in modo indipendente da altre Direzioni dell'Ente ed inosservanti delle presenti Linee guida;

- l'esposizione dei sistemi a vulnerabilità di sicurezza, con conseguenti rischi di violazione della riservatezza, integrità e disponibilità dei servizi e dei dati dell'ente.

Pertanto, le prescrizioni contenute nel presente documento sono vincolanti per le diverse unità organizzative della Direzione Sistemi informativi, per le altre Direzioni dell'Ente (nei casi in cui, in via eccezionale, procedono autonomamente all'acquisizione di sistemi o prodotti IT, comunque in raccordo con la Direzione Sistemi informativi), e per i fornitori e devono essere inserite nei relativi contratti.

Laddove nel presente documento ricorrono i termini "DEVE", "DEVONO", "DOVRÀ", "DOVRANNO" , o le espressioni "È NECESSARIO CHE….", le relative prescrizioni sono da intendere come vincolanti.

Laddove, invece, sono utilizzate espressioni come "È CONSIGLIABILE CHE …", " È PREFERIBILE CHE…", le indicazioni sono da intendere come opzionali, nel senso che possono essere oggetto di attribuzione di uno specifico punteggio in sede di gara.

Nella stesura delle presenti Linee guida si individuano delle fasi incrementali, dalla più semplice e raggiungibile nel breve periodo a quelle più complicate e tendenti ad uno scenario TO-BE condiviso, all'interno delle quali raccogliere i singoli punti.

Le Linee guida sono suddivise per macro- fasi:

1. progettazione
2. sviluppo
3. installazione
4. distribuzione.

Il presente documento è il frutto del lavoro svolto in sinergia tra le diverse unità organizzative della Direzione Sistemi Informativi ed è stato oggetto di un primo utilizzo sperimentale in relazione alle commesse seguite direttamente da tale Direzione

Il presente documento è soggetto ad aggiornamento periodico, al fine di mantenere l'allineamento costante con il contesto regolatorio di riferimento.

2. Progettazione

2.1 - Definizione delle modalità di Interazione del fornitore con l'ente

La presente sezione si riferisce ai fornitori che operano per l'Ente al fine di erogare:

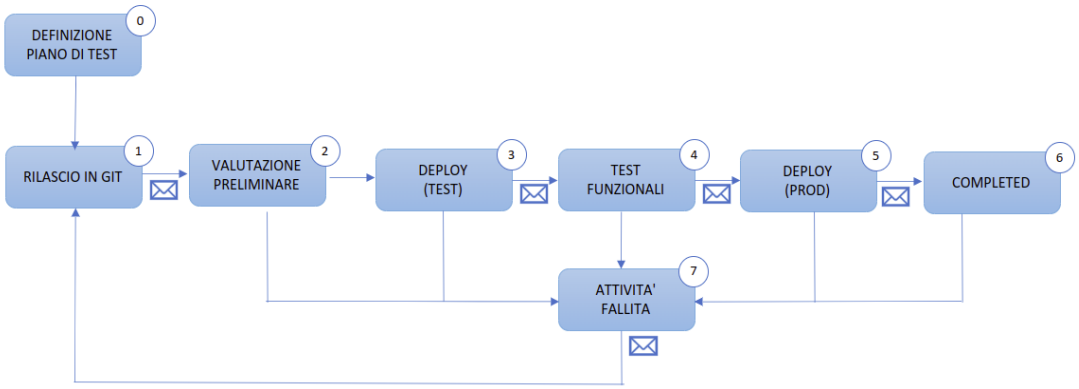
- software il cui codice sorgente è proprietà dell'Ente committente;
- servizi informatici utilizzati dall'Ente;
- servizi informatici sui quali l'Ente si appoggia per erogare servizi digitali agli utenti.

Quando l'attività prevede la realizzazione o la modifica di codice sorgente, allora È NECESSARIO CHE:

- Il codice sorgente abbia il repository Git ospitato all'interno del sistema Gitlab dell'Ente
- Siano attivate VPN nominali a scadenza predefinita, che consentano le attività richieste (la VPN è associata ad una persona fisica e ha una scadenza prefissata coerente con le attività pianificate)

- Sul sistema Gitlab vengano attivati gli accessi nominali tramite i quali è consentito al fornitore operare sul progetto specifico.

Le modalità di svolgimento delle attività tecniche richieste devono rispettare il seguente flusso.



Numero	Titolo	Descrizione	In carico a
0	Definizione piano di test	Fornitore e committente definiscono il piano di test, in base a caratteristiche, moduli e componenti interessati dal rilascio	Fornitore/ Committente
1	Rilascio in Git	Il Fornitore esegue il rilascio del codice prodotto sul repository Git e contestualmente avverte il committente mediante comunicazione formale. La comunicazione DEVE allegare la dettagliata documentazione richiesta al par. 3.1. del presente documento.	Fornitore
2	Valutazione preliminare	Se i test svolti dal fornitore o la relativa descrizione vengono ritenuti insufficienti o poco significativi, si passa immediatamente allo stato “Attività fallita”.	Committente
3	Deploy (Test)	Il Fornitore esegue la distribuzione di una versione specifica su ambiente di Test. Contestualmente avverte il committente mediante comunicazione formale.	Fornitore
4	Test Funzionali	Il Committente esegue le verifiche funzionali di quanto presente su ambiente di Test e informa il Fornitore dell’esito dei test mediante comunicazione formale.	Committente
5	Deploy (Prod.)	Il Fornitore esegue la distribuzione di una versione specifica su ambiente di Produzione. Contestualmente avverte il committente mediante comunicazione formale.	Fornitore
6	Completed	Il Committente controlla che quanto presente su ambiente di Produzione sia coerente con la distribuzione richiesta e informa il Fornitore dell’esito della verifica mediante comunicazione formale.	Committente
7	Attività fallita	Nel caso di esito negativo dell’attività precedente, è necessario attivare le azioni e le misure necessarie per correggere l’anomalia e permettere	Fornitore/ Committente

		di risolvere il problema. Il Fornitore invia una comunicazione formale al Committente relativa alle misure e azioni da attuare.	

2.2 - Riuso del codice sorgente

Deve essere chiarito fin dalla fase di progettazione se il codice sorgente che andremo a realizzare sarà oggetto di pubblicazione e possibile riuso.

In generale, ed in coerenza con gli artt. 68 e 69 del CAD, è buona norma realizzare il codice sorgente in modo che esso sia riusabile *by default*, ma se già in fase di progettazione è stato definito che sarà messo a "riuso", tanto più saranno rafforzate le misure necessarie a rendere il codice sorgente "riusabile".

A tal proposito È NECESSARIO:

- soddisfare le linee guida di Agid, in materia di "acquisizione e riuso di software per le pubbliche amministrazioni" (https://www.agid.gov.it/it/design-servizi/riuso-open-source/linee-guida-acquisizione-riuso-software-pa);
- assicurare che eventuali variabili di ambiente, username, passwords, stringhe di collegamento a database, ed ogni altro eventuale dato sensibile non sia incluso nei file pubblici del codice sorgente;
- assicurare che il codice sorgente sia corredato della documentazione tecnica necessaria per poterne comprendere l'utilizzo e le peculiarità, effettuarne l'installazione e la configurazione, poterne effettuare diagnosi relative a malfunzionamenti.
- Soddisfare i criteri di sicurezza nello sviluppo del codice sorgente, come indicato nella sezione 3.2 Sicurezza del Codice Sorgente.

Nel contesto dell'Ente, si definisce che comunque i progetti sono salvati sul repository interno del Comune di Firenze.

Nel caso che si abiliti il riuso, è attivo un repository pubblico che contiene i progetti, (si veda come riferimento la pagina Github del Comune di Firenze https://github.com/ComuneFI) dove tenere aggiornato il codice sorgente mediante meccanismi di *mirroring* (si veda la documentazione di Gitlab relativa al Mirroring, https://docs.gitlab.com/ee/user/project/repository/mirror/).

Per l'accesso in modifica al repository pubblico, è necessario ottenere i privilegi su un utente personale.

2.3 - Architettura del prodotto

La seguente sezione si applica nel caso di attività tecnica volta alla realizzazione di una soluzione software.

La progettazione della architettura della soluzione software deve tenere in considerazione, oltre agli aspetti indicati nelle altre sezioni di questo documento, anche la possibilità di mettere a riuso e a disposizione dell'Ente i componenti realizzati.

Per questo motivo, la progettazione dell'architettura deve prevedere, ove possibile, una separazione fra lo strato applicativo e lo strato di accesso ai dati.

In particolare, devono essere riutilizzate le API già realizzate e disponibili sui cataloghi dell'Ente (WSO2) e nazionali (PDND); nel caso si sviluppino nuove API, esse DEVONO essere pubblicate e documentate sul catalogo dell'Ente.

Lo strato applicativo crea, aggiorna e legge le informazioni disponibili tramite accesso alle API e dovrà evitare ogni accesso diretto ai database, a meno che questo non sia motivato da vincoli stringenti.

Per l'utilizzo, la realizzazione e la pubblicazione di API si prendano come riferimento le Linee Guida di Interoperabilità del Comune di Firenze, allegate al presente documento [Allegato A].

Le indicazioni precedenti sono da ritenersi valide anche nel caso di deploy su infrastruttura CST (Centro Servizi Territoriali), gestito da Firenze Smart - Silfi S.p.A. (di seguito anche Firenze Smart). In questo secondo caso, sono da considerare anche le Linee guida di Firenze Smart per il deploy su infrastruttura CST, reperibili al seguente indirizzo: http://www.lineacomune.it/requisiti-dispiegamento-servizi-cst

2.4 - Metodi di condivisione della documentazione

Per quanto riguarda la condivisione di documentazione, sia tra solo personale interno che con personale esterno, il fornitore DEVE catalogare i documenti secondo quattro diverse **categorie**:

- Documentazione progettuale
- Documentazione tecnica
- Manualistica
- Documenti interni

Si identificano inoltre alcuni **processi** che È NECESSARIO coprire per poter gestire correttamente la documentazione:

- Condivisione snella e dinamica della documentazione durante la fase di esecuzione del progetto
- Registrazione della documentazione finale di progetto o relativa al raggiungimento di milestone intermedie per fini di archiviazione e aderenza alle norme di documentazione ISO
- Archivio delle procedure tecniche di routine, utili per ritrovare soluzioni già applicate o per analisi interna.

Si evidenziano infine alcuni **punti di attenzione** che DEVONO essere tenuti in considerazione:

- Permettere la rapida ricerca di documentazione ed evitare la dispersione di documentazione all'interno di mail
- Uniformare le modalità di condivisione dei documenti
- Non creare vincoli di "lock-in" su piattaforme specifiche.

A fronte della catalogazione dei documenti, dell'analisi dei processi e dei punti di attenzione, si conclude di utilizzare preferibilmente i seguenti strumenti per soddisfare le esigenze sopra indicate.

Processo	Strumento
Condivisione nella fase di esecuzione del progetto	Microsoft Sharepoint
Registrazione della documentazione di milestone e/o finale	Cartelle condivise dell'ente aderenti alla normativa ISO
Archivio procedure tecniche	MediaWiki

2.5 - Integrazione con servizi SIT

Quando viene progettato un nuovo servizio/applicativo, è raccomandato, nel caso tale servizio abbia degli aspetti che riguardano la posizione o quanto altro da tenere presenti, individuare se è necessaria l'interazione con i servizi del SIT. Nel caso in cui esista questa possibilità, cioè se i dati trattati dall'applicativo hanno una valenza di tipo geografico, chi sviluppa DEVE confrontarsi con la E.Q. Risorse Dati, Open Data, SIT della Direzione Sistemi informativi, per individuare la strategia migliore per tenere conto anche di tali aspetti.

Il SIT ha una infrastruttura basata su banche dati georeferenziate e messa a disposizione delle stesse anche tramite web service geografici. L'infrastruttura conta circa duemila strati informativi disponibili in queste modalità.

Sostanzialmente, a disposizione di chi sviluppa, per la consultazione o la visualizzazione dei dati geografici ci sono due strumenti principali:

a. Geoserver, l'infrastruttura che genera i servizi, che quindi consente la condivisione dei dati spaziali in diversi formati: vettoriali, raster e anche alfanumerici.

b. Geoflorentia, che è un'interfaccia Web GIS che consente di "estendere" le funzionalità di un applicativo gestionale alfanumerico in modo geografico, permettendo non solo di partire dalla parte alfanumerica e "visualizzarla" su mappa, ma anche di fare ricerche e selezioni geografiche sul territorio di Firenze.

La la E.Q. Risorse Dati, Open Data, SIT fornirà I documenti descrittivi dei due sistemi con i necessari dettagli. In base alle esigenze applicative si potrà scegliere tra le due soluzioni. Con Geoserver è possibile accedere a dati geografici attraverso i formati:

· WMS, che consente di avere immagini raster nei formati AtomPub, GIF, GeoRSS, GeoTiff, GeoTiff 8-bits, JPEG, JPEG-PNG, KML (compressed), KML (network link), KML (plain), OpenLayers, OpenLayers 2, OpenLayers 3, PDF, PNG, PNG 8 bit, SVG, Tiff, Tiff-8bits, UTF-Grid.

· WFS, che consente di avere dati alfanumerici nei formati CSV, GML 2, GML 3.1, GML 3.2, GeoJSON, KML, Shapefile.

Generalmente, negli applicativi l'utilizzo degli strati informativi attraverso i servizi necessita di chiamate verso URL intranet. Tale meccanismo è in fase di studio per l'integrazione con l'API manager di WSO2.

Le eccezioni alle chiamate intranet, ovvero le chiamate "da fuori rete aziendale" e/o in generale da Internet, vengono permesse tramite in taluni casi tramite il software Geofence, che è deputato all'implementazione delle policy di accesso ai dati sia in intranet (accesso agli strati informativi per gruppi di dominio) che in internet (eccezioni a chiamate a strati informativi).

In generale, è possibile anche utilizzare direttamente il database (vari db PostGis, ma anche Oracle); questa modalità è sicuramente necessaria in caso di editing diretto sui dati da parte del personale competente; raramente in caso di letture da parte "di chiunque nella Intranet".

Dato che le banche dati sono in continua evoluzione, per avere un elenco completo ed aggiornato dei dataset disponibili occorre rivolgersi alla E.Q. Risorse Dati, Open Data, SIT..

Con il componente Geoflorentia è possibile, attraverso una chiamata http, ricevere risultati interrogando le banche dati del SIT. Come precedentemente accennato, tale componente va vista proprio come una estensione di un applicativo alfanumerico verso la parte cartografica. Si atterra su tale interfaccia a partire dall'applicativo alfanumerico di partenza, tramite chiamate http che generalmente hanno nei propri parametri attributi, quali uk civico (identificativo univoco della toponomastica georeferenziata) o uk_particella (identificativo univoco del catasto georeferenziato). Infatti, di solito le applicazioni alfanumeriche che utilizzano catasto e civici integrano questi id, presenti nelle banche dati di riferimento, proprio per permettere un legame diretto o indiretto con le banche dati georeferenziate. Attraverso l'interfaccia messa a disposizione dall'applicativo è possibile quindi visualizzare geograficamente il dato di partenza, originariamente non georeferenziato sulla banca dati alfanumerica, ma anche effettuare interrogazioni su cartografia secondo le entità geometriche di base (punto, linea, polilinea), il sistema restituisce una risposta http che sarà elaborata dall'applicativo chiamante, restituendo indietro gli uk_civici o gli uk_particella selezionati, per permettere appunto la ricerca alfanumerica di questi oggetti.

Tramite Geoflorentia, è possibile anche implementare aspetti di editing di banche dati geografiche, permettendo quindi una completa trattazione del dato geografico. Ricordiamo, infatti, che la georeferenziazione indiretta tramite numero civico o particella catastale non è sufficiente per tutti gli oggetti del territorio (alberature, chioschi, sinistri, segnaletica, ecc. non possono essere agganciati a un numero civico). L'alternativa per gestirli è, quindi, poterli inserire in banca dati con le loro componenti geometriche. Raramente sono sufficienti le coordinate x e y, perché un oggetto può essere geometricamente più complesso. Il componente Geoflorentia permette comunque anche la gestione (o editing) tramite un meccanismo piuttosto semplice e, volendo, anche trasparente alla banca dati alfanumerica di partenza. Ovviamente per la progettazione di tale interazione, essendo questi aspetti in continua evoluzione, è opportuno contattare la E.Q. Risorse Dati, Open Data, SIT.

2.6 - Gestione dei dati su file system e policy di retention

Viste le finalità delle presenti Linee guida, in questo contesto si farà riferimento esclusivamente ai log cosiddetti *applicativi* o log *file*, ovverosia quelli prodotti da un applicativo durante il suo funzionamento per tracciare le operazioni eseguite, di tipo automatico, oppure generate da una qualsiasi interazione, sia essa di tipo uomo-macchina che machine-to-machine (ad es. Applicativi che richiamano API).

E' importante tenere in considerazione che i log file prodotti dalle applicazioni richiedono una attenzione particolare da parte del titolare del trattamento dei dati che, in linea con il GDPR, ha la responsabilità di identificare quali sono le informazioni necessarie da raccogliere, tracciare ed includere nei log applicativi. Ad esempio, se si vogliono tracciare le modifiche a determinate entità da parte di un utente, sarà necessario tracciare l'orario della modifica e il tipo di modifica, ma non invece informazioni ulteriori e non pertinenti.

Pertanto, in coerenza con quanto previsto nel presente documento, la definizione delle regole applicabili ai log applicativi (es. set di informazioni, tempi di conservazione, ecc.) DEVE essere effettuata congiuntamente tra fornitore, titolare del trattamento e personale tecnico dell'Ente.

Il fornitore DEVE rendere disponibili i log applicativi al sistema di *log management* utilizzato dall'Ente, capace di tracciare, conservare e analizzare le informazioni riguardanti le operazioni compiute sui sistemi e che impattano direttamente sui dati personali.

Secondo quanto stabilito dall'Autorità Garante per la Protezione dei dati personali, i log file devono rispondere alle seguenti caratteristiche:

- completezza;
- inalterabilità;
- verificabilità.

Ciò premesso, è evidente come i log file rappresentino uno strumento fondamentale al fine di garantire la sicurezza dell'infrastruttura informatica e la lecita gestione dei dati personali, in quanto capace di "fotografare" le operazioni compiute sui sistemi informatici e quindi sui dati personali. Pertanto, una corretta organizzazione e gestione di tali log è fondamentale per il perseguimento dei fini suddetti.

Partendo dalla struttura delle singole righe dei file di log, si riporta di seguito un set minimo di informazioni che il fornitore DEVE riportare nei log:

- IP chiamante
- Userid
- Data
- Operazione/funzionalità richiamata
- Esito operazione
- Identificativo unico (per esempio il session ID)

È CONSIGLIABILE la suddivisione per categorie:

- Info
- Debug
- Error

È CONSIGLIABILE la rotazione e successiva storicizzazione dei log, con base temporale da definire in sede di commessa, valutata sulla base delle effettive necessità.

La nomenclatura da rispettare per ogni file di log DEVE permettere una rapida e corretta individuazione di quanto ricercato; pertanto, DOVRA' ESSERE rispettato il pattern:

YYYY-MM-DD-log-<nome-applicativo>-<[se pertinente:]modulo-applicativo><tipologia-log>

I file di log seguiranno a tutti gli effetti le regole applicate ai documenti informatici, pertanto, DEVE essere prevista la suddivisione in "live documents" da mantenere sui sistemi e "cold documents", nei casi in cui debbano essere soddisfatte le regole di conservazione secondo norma vigente.

Per quanto concerne i periodi di conservazione, a parte quelli dettati da normativa, DEVE essere effettuata, in fase di elaborazione del Piano di Progetto Definitivo, una valutazione congiunta tra fornitore, titolare del trattamento e personale tecnico dell'Ente.

E' attualmente in fase di valutazione l'integrazione dei sistemi di logging con uno stack ELK (Elastic-Logstash-Kibana) che faciliti la gestione e l'interrogazione di questa base dati così eterogenea e variegata.

I log derivanti da sistemi e sensori seguiranno, per quanto possibile e compatibilmente con le possibilità tecniche degli stessi, le stesse regole stabilite nel presente documento.

L'accesso ai log avverrà sempre ed esclusivamente in modalità di lettura, con opportuna configurazione delle utenze, per evitare eventuali alterazioni e/o manomissioni, pena la decadenza dell'effetto probatorio.

2.7 - Modalità di integrazione con Single Sign-On (SSO)

Il Comune di Firenze utilizza un Active Directory centralizzato per autenticare gli utenti aziendali e abilitarli all'utilizzo della postazione locale. Si è pertanto pensato di utilizzare lo stesso meccanismo anche per gli applicativi web, centralizzando in tal modo l'identità dell'utente. Il fornitore DEVE attenersi alle indicazioni riportate di seguito.

Il meccanismo di autenticazione centralizzato si avvale di un url di autenticazione, denominato "login-url", settato in modo da poter essere interrogato dall'applicativo chiamante. L'operazione di controllo dell'avvenuta autenticazione può avvenire in 2 modalità:

1. Modalità http semplice (CGI)
2. Modalità web service (SOAP)

Nel caso che l'applicativo abbia necessità di gestire documenti presenti nel sistema di gestione documentale dell'Ente (Alfresco) è possibile chiamare un servizio di SSO per richiedere l'autenticazione al documentale. Anche in questo caso sono previste 2 modalità di servizio:

1. Modalità http semplice (CGI)
2. Modalità web service (SOAP)

Per consentire il logout dell'utente bisognerà richiamare un'altra url.

Per tutti i dettagli implementativi si rimanda alla documentazione completa presente in \\condivisioni\CONDIVISE\SISTEMI_Informativi\PROGETTI\SSO\MTO PSQ 8.5-4 Specifiche integrazione SSO.pdf

2.8 - Modalità di integrazione con Servizi di Autenticazione Firenze Smart

Gli eventuali servizi di autenticazione SPID, CIE e CNS da attivare sul servizio software oggetto delle attività tecniche devono avvalersi dei servizi forniti da Firenze Smart.

Per usufruire dei servizi di autenticazione SPID, CIE e CNS forniti da Firenze Smart, è necessario seguire le specifiche fornite sulla pagina http://www.lineacomune.it/requisiti-dispiegamento-servizi-cst nella sezione "Integrazione Identity Server WSO2".

I servizi di autenticazione si basano sulla piattaforma WSO2 e implementano il protocollo Oauth 2.0.

Per accendere i servizi di autenticazione su un nuovo servizio software, è necessario mettersi preventivamente in contatto con Firenze Smart e procedere alla richiesta di attivazione.

2.9 - Progettazione dei servizi sicuri dal punto di vista architetturale e di framework

Le architetture e i framework utilizzati DEVONO essere scelti prendendo la versione stabile degli stessi più recente possibile; DEVE essere inoltre garantita l'aggiornabilità futura degli stessi, con particolare attenzione alle patch critiche di sicurezza.

Sono, quindi, da evitare architetture/framework fuori dal periodo di supporto (EOS - End Of Support o EOL - End Of Life del prodotto) o all'interno di un periodo di supporto che però determini un aggravio di costo per l'Ente.

A tal fine sono fortemente consigliate le versioni Long Term Support (LTS).

Inoltre, la scelta della versione e l'installazione di queste componenti devono essere fatte privilegiando la possibilità di attivare sistemi di aggiornamento automatico (sistemi di aggiornamento automatico già attivi per i Sistemi Operativi, client e server, e per le loro componenti principali).

A tal fine, il Fornitore dovrà porre particolare attenzione a quelle componenti che sono considerate deprecate (ancorché funzionanti) e che potrebbero risultare non più funzionanti in caso di aggiornamento automatico del Sistema.

DEVONO essere, inoltre, privilegiate quelle architetture/framework che possono essere installate e manutenute in autonomia dai tecnici dell'Ente, in modo da poter mantenere pieno controllo delle stesse. Qualora non si tratti di architetture standard (es. diverse da Apache, Tomcat, PHP, etc...), il fornitore DOVRÀ presentare una guida di installazione delle architetture/framework richiesti.

Nel caso in cui l'installazione debba necessariamente essere effettuata dal fornitore, quest'ultimo fornirà dettagliata documentazione dell'attività effettuata, in modo da rendere autonomi i tecnici dell'Ente nella manutenzione delle architetture/framework installate.

Per quanto riguarda i protocolli di comunicazione, l'applicativo DEVE funzionare con protocolli sicuri (HTTPS, LDAPS, etc...). L'Amministrazione si riserva tuttavia di poter attivare protocolli non sicuri in caso di integrazione dell'applicativo all'interno della Rete dei Servizi.

Gli applicativi installati nell'infrastruttura dell'Ente, in caso di accesso a risorse esterne alla rete comunale, DEVONO permettere la configurazione di un Proxy Server.

2.10 - Progettazione dei servizi in aderenza al GDPR

La progettazione dei servizi DEVE essere svolta in accordo alle norme in materia di privacy, che recepiscono il GDPR Regolamento UE 2016/679.

3. Sviluppo

3.1 - Validazione e test del prodotto

L'insieme dei test eseguibili comprende quanto segue:

- Test automatici da eseguire a fronte di ogni rilascio. Nel caso sia stata attivata una procedura automatica di CI/CD presso la piattaforma del committente, che prevede anche l'esecuzione dei test automatici, non è necessario fornire documentazione apposita, in quanto sarà la stessa procedura automatica che procederà o meno con i passaggi successivi nel caso di superamento / non superamento dei test automatici. In ogni caso, ne va fornita evidenza all'Ente.
- Test di validazione del software dal punto di vista della sicurezza (vedi anche sezione 3.2 Sicurezza del codice sorgente)
- Test funzionali volti a verificare la correttezza del prodotto rispetto ai requisiti
- Test di accessibilità in conformità ai livelli richiesti nelle linee guida sull'accessibilità di AGID (https://www.agid.gov.it/it/design-servizi/accessibilita/linee-guida-accessibilita-strumenti-informatici)
- Test di utilizzo del prodotto sotto stress (concorrenza dell'utilizzo, operazioni massive sul servizio, …).

Il committente e il fornitore condividono, prima di ogni rilascio, un piano dei test che DEVE essere applicato dal fornitore stesso al prodotto realizzato. Il piano di test potrà essere, di volta in volta, più o meno esaustivo a seconda delle componenti e delle funzionalità interessate.

Una volta che il personale incaricato dal fornitore ha completato le attività di test pre-rilascio, rilascia altresì la documentazione che attesta il superamento dei test concordati, eventualmente integrati da ulteriori test che il fornitore abbia ritenuto necessari. In quest'ultimo caso, non è obbligatorio attestarne il superamento.

3.2 - Sicurezza del codice sorgente

Le attività di realizzazione di codice sorgente DEVONO tenere in considerazione le misure necessarie per assicurarne, oltre alla correttezza, anche la sicurezza.

AGID pubblica ed aggiorna le Linee guida per lo sviluppo di software sicuro, corredandole di allegati per quanto attiene ai seguenti ambiti:

- Linee guida per l'adozione di un ciclo di sviluppo di software sicuro (Allegato A)
- Linee Guida per lo sviluppo sicuro di codice (Allegato B)
- Linee Guida per la configurazione per adeguare la sicurezza del software di base (Allegato C)
- Linee Guida per la modellazione delle minacce ed individuazione delle azioni di mitigazione conformi ai principi del Secure/Privacy by Design (Allegato D).

Il personale incaricato della realizzazione del software DEVE aver visionato le linee guida e rispettare ed applicare quanto di interesse per le attività coinvolte; DEVE inoltre assicurarsi di essere aggiornato con l'ultima versione disponibile pubblicata da AGID.

Le Linee guida AGID sono reperibili al seguente indirizzo:
https://www.agid.gov.it/it/sicurezza/cert-pa/linee-guida-sviluppo-del-software-sicuro

3.3 - Script automatici

Per "script automatici" si intendono tutti gli script dispiegati in modo indipendente all'interno delle infrastrutture dell'Ente o comunque all'interno dei componenti di pertinenza dell'Ente (quindi anche su cloud) capaci di avviare le loro attività in modo automatico e in accordo ad una pianificazione pre-impostata.

Al fine della corretta catalogazione e gestione, la realizzazione di script automatici DEVE soddisfare i seguenti requisiti:

- **Requisito di documentazione**
  - la documentazione dello script DEVE contenere:
    - Gli *orari pianificati* per l'esecuzione dello script;
    - Le *motivazioni* per le quali è richiesta l'esecuzione dello script in accordo agli orari pianificati;
    - Le *conseguenze* nel caso di modifica degli orari pianificati;
    - Quale è il *piano di notifiche* che lo script produce (quante e quali notifiche vengono inviate, a quali destinatari, in quale spazio orario).
- **Requisito di monitoraggio**
  - Il monitoraggio prevede che lo script DEVE loggare la propria attività e inviarla come notifica sull'esito della propria esecuzione (al momento tramite e-mail indicando in modo esplicito nell' oggetto l'esito dell'esecuzione).

4. Installazione

4.1 - Installazione di un nuovo database

Nel caso il servizio da costruire richieda l'utilizzo di un database relazionale, il fornitore DEVE tenere in considerazione l'elenco di preferenza seguente (in ordine discendente dalla più preferibile alla meno preferibile):

Lista di preferenza delle tipologie di database
Postgres
MariaDB (MySQL)
Microsoft SqlServer
Oracle

Generalmente le informazioni seguenti vengono scelte dal personale tecnico dell'Ente in relazione al nome col quale l'applicativo è conosciuto tra il personale interno, in particolare gli utenti finali:

- i nomi degli utenti, login e ruoli,
- i nomi dei database, dei tablespace e dei datafile,
- gli alias DNS per i db server e application server,
- altri nomi di oggetti attinenti.

Il personale tecnico dell'Ente individua con il fornitore anche le password in accordo agli standard di sicurezza adottati.
Il personale tecnico e i fornitori DEVONO escludere nelle stringhe di connessione l'uso degli indirizzi ip e dei nomi veri delle macchine, così da favorire gli alias (del tipo db-nome, dove "nome" risponde alle scelte di cui sopra).
Per le connessioni a database di tipo Oracle il personale tecnico dell'Ente e il fornitore adottano la risoluzione dei nomi basata su LDAP, e NON DEVONO utilizzare stringhe del tipo NomeMacchina:Porta:Istanza o IndirizzoIP:Porta:Istanza o l'uso del file tnsnames.ora.
E' NECESSARIO che tutte le definizioni delle stringhe di connessione siano note ai tecnici del Comune e siano modificabili in caso di necessità dai medesimi.
Eventuali aggiornamenti degli standard di sicurezza adottati comportano interventi di manutenzione ed aggiornamento degli oggetti precedentemente definiti (come ad esempio le password).
Generalmente l'Ente non fornisce db server dedicati (in particolare per motivi di licenza). Seguendo il principio del minor privilegio, gli utenti applicativi NON DEVONO avere privilegi amministrativi.

Per quanto riguarda la richiesta di RDBMS Oracle, il richiedente deve verificare con la Direzione Sistemi Informativi la possibilità di inserire il nuovo Database in un'istanza Oracle già esistente (solo per progetti rendicontabili e afferenti al progetto "PON Metro"). In caso contrario il richiedente deve prendere in carico il costo delle licenze Oracle e sostenerne le spese per tutta la durata della fornitura. Il richiedente deve inoltre considerare e prendersi in carico il costo della licenza anche per gli anni successivi al termine della durata della fornitura.

Di seguito si riportano i dettagli per le singole tecnologie scelte per soddisfare l'erogazione di un database.

Database Postgres

Cosa viene configurato dall'Ente?	• un ROLE non di login che sarà il proprietario del tablespace e del database, • il tablespace ed il database che si appoggia su di esso: come detto questi due oggetti sono di proprietà (clausola authorization) del ROLE di cui sopra, • un LOGIN ROLE cui si assegna il ROLE di cui sopra, • nel database si lascia al ruolo public il solo "usage" dello schema public *L'Ente preferisce infatti che gli oggetti dell'applicativo non risiedano nello schema public ma in uno schema dedicato cui si accede con un utente dedicato che ha privilegi limitati al suo schema: si usa la variabile search_path con la variabile $user, per cui tale utente e tale schema sono omonimi. L'uso del search_path permette di non dover prefissare gli oggetti col nome dello schema.*
Cosa consiglia l'Ente al fornitore?	• È utile per le finalità diagnostiche che gli applicativi, specialmente quelli su tecnologia container, popolino i campi della vista pg_stat_activity, ad esempio mediante comando "set application_name to", niente esclude di usare la variabile Application_Name in caso di connessioni JDBC.

Database MariaDB (MySql)

Cosa viene configurato dall'Ente?	• un utente (in realtà almeno due: uno utente@localhost, l'altro utente@[Rete del Comune]), • un database e gli assegnati privilegi pieni all'utente su quel database, • Viene utilizzata la configurazione con attivo il parametro innodb_file_per_table (innodb_file_per_table = ON) *Esistono casi in cui un applicativo abbia più utenti e/o databases ma nessun utente ha privilegi amministrativi. Generalmente l'Ente non fornisce db server dedicati. I privilegi degli utenti non escono dall'ambito dei databases creati per l'applicativo.*
Cosa consiglia l'Ente al fornitore?	

Microsoft SQLServer

Cosa viene configurato dall'Ente?	• un login dedicato all'applicativo, o più d'uno se serve ma sempre nella logica dello "One Big Application User", • un database con al suo interno uno user, che sia dbOwner di quel database, mappato al login creato in precedenza, • i datafile vengono creati in autoestensione e con dimensione massima fissata, • un piano di manutenzione schedulato che, contenendo il rebuild di indici, ci avvisa anche quando lo spazio nei datafiles è in esaurimento (il rebuild fallisce perché manca lo spazio per tenere due copie dell'indice, non completa la rebuild e cancella la copia temporanea). *Esistono casi in cui un applicativo abbia più login, utenti, databases ma nessun login ha privilegi amministrativi.*
Cosa consiglia l'Ente al fornitore?	• si sconsiglia l'uso del privilegio di BULK INSERT in quanto non è limitabile al contesto dei singoli databases (i privilegi dei login e degli utenti NON DEVONO uscire dall'ambito dei databases creati per l'applicativo)

Oracle

Cosa viene configurato dall'Ente?	• un utente/schema che sarà dedicato all'applicativo nella logica dello "One Big Application User", per applicativi più complessi gli schemi possono anche essere più d'uno; abbiamo avuto alcuni casi, motivati dal fornitore, in cui uno stesso utente ha più tablespace, o l'applicativo ha vari utenti e ruoli. • un tablespace (o più ove richiesto), l'utente non ha limiti di quota sui propri tablespace, • I datafile sono configurati in autoestensione e con dimensione massima fissata (si rimanda al sistema di monitoraggio dell'Ente per gli avvisi sull'esaurimento dello spazio),
Cosa consiglia l'Ente al fornitore?	• È utile per le finalità diagnostiche che gli applicativi, specialmente quelli su tecnologia container, popolino i campi della vista v$session, ad esempio mediante le procedure del package DBMS_APPLICATION_INFO, primo fra tutti il campo client_info • La separazione tra tablespaces di dati, tablespaces di indici e tablespaces per campi lob, per una manutenzione più efficace.

4.2 - Standard di lavoro su macchine

L'installazione di architetture, framework, middleware sulle macchine dell'Ente DEVE necessariamente avvenire secondo regole prestabilite.

Alcune di queste regole sono generali per tutte le architetture, mentre altre sono specifiche.

Tra le regole generali, ricordiamo:

1. L'accesso ai server DEVE avvenire attraverso credenziali nominali fornite a soggetti determinati e opportunamente autorizzati; non è consentito l'accesso ai server con credenziali altrui.
2. Principio del minor privilegio: le installazioni DEVONO essere fatte in modo tale che i privilegi assegnati alle varie componenti siano i minimi necessari per il funzionamento degli applicativi richiesti. A titolo esemplificativo, è da evitare l'assegnazione di un privilegio completo di lettura, scrittura ed esecuzione di un file per tutti gli utenti senza un giustificato motivo.
3. Le varie componenti DEVONO essere il più possibile installate ciascuna all'interno di directory dedicate, seguendo percorsi standard e separando i file di configurazione (che contengono ad esempio password, e che quindi devono avere minimi privilegi di accesso) dai file di log (a cui possono aver accesso anche utenti diversi) e dai file applicativi. Per determinate architetture/framework esistono delle directory standard, che dovranno essere concordate con i tecnici della Direzione Sistemi Informativi.
4. L'installazione delle componenti del Sistema DEVE avvenire quanto più possibile utilizzando l'installatore standard del sistema operativo (es. Yum, apt, etc...), lasciando le impostazioni di default ove non espressamente indicato di cambiarle.
5. I servizi (quindi applicativi che restano in continua esecuzione) DEVONO essere installati privilegiando sistemi automatici di gestione degli stessi (es. Systemd su Linux o Servizi di Windows), in modo da rendere standard all'interno del sistema operativo l'avvio, l'arresto e il monitoraggio degli stessi. Va quindi ridotta al minimo la presenza di applicativi che vengono eseguiti lanciando script manuali, script di cron con watchdog a frequenza elevata, etc...
6. I comandi che vanno eseguiti periodicamente DEVONO ESSERE censiti più possibile sotto il crontab dell'utente che li esegue.
7. Nel caso in cui un utente di S.O. debba accedere ad una directory proprietà di un altro utente, si consiglia di privilegiare le mount di tipo bind, effettuando il mount della directory interessata possibilmente sotto la home dell'utente che deve avere accesso. Nella mount si specificheranno i permessi di lettura o scrittura che dovranno essere concessi.

Per quanto riguarda alcune specifiche piattaforme (es. Apache, Tomcat, PHP, etc...) e anche per il posizionamento degli applicativi sul server, l'installazione, qualora non venga effettuata da parte dei tecnici della Direzione Sistemi Informativi, DEVE comunque essere concordata con i tecnici stessi, che forniranno le direttive per installare correttamente i software necessari.

5. Distribuzione

5.1 - Procedure automatiche di build e distribuzione

La parte di distribuzione di un prodotto è un aspetto cruciale per la corretta erogazione di servizi funzionanti e aderenti a criteri imposti di continuità di servizio.

Le operazioni di distribuzione eseguite da un operatore umano, sebbene svolte con la necessaria attenzione, sono soggette a possibili errori e richiedono spesso delle attività ripetitive a basso valore aggiunto.

Per questi motivi È PREFERIBILE implementare un processo di build e distribuzione completamente automatico e basato sulle moderne pratiche del CI/CD.

La procedura automatica di build e distribuzione sarà realizzata considerando le caratteristiche seguenti:

- La procedura automatica DOVRA' avviarsi in base ad eventi che si verificano sul codice sorgente (push di codice sorgente, creazione di un tag, chiusura di una merge request, …) e in accordo con quanto stabilito con il committente;
- La procedura automatica POTRA' contenere l'esecuzione di batterie di test automatici sul prodotto sviluppato (si veda anche sezione 3.1 relativa alla "Validazione e test del prodotto");
- La procedura automatica DOVRA' creare una build del software e orchestrare la sua distribuzione sui vari ambienti di sviluppo e test per i relativi controlli, prima di procedere alla distribuzione in produzione;
- La procedura automatica DOVRA' interrompere la propria elaborazione in caso di fallimento di ogni step preliminare alla distribuzione del servizio, così da permettere una diagnosi ed un intervento da parte del personale incaricato;
- La procedura automatica POTRA' richiedere una conferma da parte di un operatore umano del committente per la distribuzione sull'ambiente di produzione;
- La procedura automatica DOVRA' tenere in considerazione le linee guida OWASP DevSecOps (https://owasp.org/www-project-devsecops-guideline/), assicurandosi di soddisfare almeno i punti: secrets management, linting code, static application security testing (SAST), dynamic application security testing (DAST), Container vulnerability testing, privacy.

6. Informazioni sul documento

Storia delle modifiche

Versione	Contenuto delle modifiche	Autori	Data
DRAFT	Prima stesura di draft	Gruppo di lavoro del Comune	2022/01/16
0.1	Revisione dei contenuti ed inclusione punti raccolti da conf. Call 2/2/2022	Gruppo di lavoro del Comune	2022/02/02
0.2	Aggiornamento sezione 2.1	R.Vannetti, in seguito a conf. Call 2022/02/28	2022/03/02
0.3	Aggiornamento sezione 2.5	F. Foresta	
0.4	Condivisione e finalizzazione sezioni 2.1 e 2.5	Gruppo di lavoro del Comune	2022/03/14
0.5	Aggiornamento sezione 2.2, “Riuso del Codice sorgente”	R.Vannetti	2022/03/22
0.6	Aggiornamento sezione 2.4, in base a conf. Call 23/3	R. Vannetti	2022/03/28
0.7	Revisione dei contenuti della sezione 2.2 e 2.4	Gruppo di lavoro del Comune	2022/03/28
0.8	Stesura draft 3.2 Sicurezza del codice sorgente	R. Vannetti	2022/04/06
0.9	Revisione dei contenuti delle sezioni 2.6 e 3.2	Gruppo di lavoro del Comune	2022/04/06
0.10	Aggiornamento sezione 3.1	R. Vannetti	2022/04/28
0.11	Revisione dei contenuti sezione 3.1	Gruppo di lavoro del Comune	2022/04/29
0.12	Typo fix	R.Vannetti	2022/04/29
0.13	Aggiornamento sezione Architettura del Prodotto	R.Vannetti	2022/05/10
0.14	Revisione dei contenuti della sezione 2.3	Gruppo di lavoro del Comune	2022/05/11
0.15	Revisione sezioni 2.3, 2.8	R.Vannetti	2022/05/18
0.16	Revisione sezioni 2.3, 2.7, 28	Gruppo di lavoro del Comune	2022/05/18
0.17	Revisione sezione 5.1	Gruppo di lavoro del Comune	2022/06/06
0.18	Revisione in base a DevSecOps OWASP guidelines	R. Vannetti	2022/06/14
0.19	Revisione sezione 5.1, Aggiunta della sezione 2.9	Gruppo di lavoro del Comune	2022/06/15

0.20	Modifiche alla sezione 2.1 e correzioni ortografiche sparse	B. Femia	2022/09/05
0.21	Aggiornamento del flusso di interazione fornitori / committente della sezione 2.1	R.Vannetti	2022/09/09
1.0	Merge del documento finalizzato da B. Femia sul documento del gruppo di lavoro ed aggiunta delle informazioni del capitolo 4, sezione 4.1	R.Vannetti, F.Carnasciali	2022/09/15
1.1	Revisione sezione 3.3 Script Automatici, 2.10 Privacy GDPR, rimozione sezione 5.2 containers	R.Vannetti	2022/10/07
1.2	Revisione condivisa del gruppo di lavoro per quanto riguarda le sezoni 2.9 e 4.2	Armellini A., F. Carnasciali e Gruppo di lavoro	2022/10/14
1.3	Revisione per contestualizzare il documento da un punto di vista normativo e di direzione nazionale, in accordo con il gruppo di lavoro	R. Vannetti	2023/01/09
1.3	Condivisione con gruppo di lavoro e aggiunta di alcuni punti relativi alle criticità di sicurezza, inosservanza GDPR,	Gruppo di Lavoro LGAT	2023/01/12
1.3	Modifica sezione 2.5 in accordo ad indicazioni del gruppo Geri, aggiunta di alcune piccole precisazioni, piccole correzioni	Gruppo di Lavoro LGAT, R. Crocchini	2023/01/18
1.3	Revisione finale della versione 1.3 del documento	Direzione Sistemi Informativi	2023/05/31

Autori

Gli autori del documento rappresentano in modo trasversale i team di lavoro della Direzione Sistemi Informativi.

Allegato A

Linee Guida di Interoperabilità

Versione: 1.3

Introduzione

Con questo documento la Direzione Sistemi Informativi del Comune di Firenze si pone l'obiettivo di stabilire delle linee guida da seguire al fine di avere uno standard ed una uniformità di gestione dei servizi API.

In tal modo si rendono espliciti gli scenari in cui poter utilizzare WSO2 in modo esteso oppure utilizzarlo solamente come gateway unico. Si evidenziano inoltre gli standard da applicare nello sviluppo e nell'esposizione delle API.

Il presente documento è il frutto del lavoro svolto in sinergia tra le diverse unità organizzative della Direzione Sistemi Informativi ed è stato oggetto di un primo utilizzo sperimentale in relazione alle commesse seguite direttamente da tale Direzione

Il presente documento è soggetto ad aggiornamento periodico, al fine di mantenere l'allineamento costante con il contesto regolatorio di riferimento.

Le **Linee guida sull'interoperabilità tecnica delle Pubbliche Amministrazioni** e le **Linee guida Tecnologie e standard per la sicurezza dell'interoperabilità tramite API dei sistemi informatici** adottate da AGID sono disponibili al link https://www.agid.gov.it/it/infrastrutture/sistema-pubblico-connettivita/il-nuovo-modello-interoperabilita .

Criteri per la creazione di una nuova API: dentro WSO2 o fuori da WSO2?

Quando ci si appresta a realizzare una nuova API, è necessario individuare in fase di progetto se realizzarla esternamente a WSO2 e poi pubblicarla mediante API Manager, oppure realizzarla integralmente su WSO2 tramite gli strumenti forniti da tale piattaforma.

Per poter supportare la scelta, si riportano di seguito dei criteri da soppesare, che possono indirizzare su una metodologia o sull'altra.

Complessità della API

L'API necessita di operazioni semplici (operazioni CRUD) di complessità ridotta? WSO2 fornisce degli strumenti automatici per la realizzazione di semplici CRUD. È possibile anche inserire facilmente delle componenti per la gestione di semplici deviazioni rispetto al comportamento di base. Se si necessita di inserire operazioni più complesse la situazione si complica e potrebbe essere preferibile lavorare esternamente per avere un controllo maggiore su quanto deve essere realizzato.

Tipologia della base dati

L'API coinvolge una base dati distribuita su molte tabelle? Nel caso in cui la base dati sulla quale si appoggiano le API da costruire sia composta da molteplici tabelle con foreign key di collegamento fra le tabelle stesse, è preferibile costruire le API fuori da WSO2 e poi procedere alla loro pubblicazione.

Interventi futuri di manutenzione evolutiva e correttiva

Si prevede che gli interventi futuri di manutenzione correttiva ed evolutiva verranno realizzati da personale esterno? WSO2 è una piattaforma conosciuta e diffusa. Essa si pone in un punto di mezzo per quanto riguarda "Completness of vision" e "Ability of execute" (si veda Magic Quadrant for Full Life Cycle API Management del 2020, Gartner).

Figure 1. Magic Quadrant for Full Life Cycle API Management

Source: Gartner (September 2020)

L'intervento da parte di personale esterno e la facilità di poter incaricare aziende diverse per l'espletazione della manutenzione, dovrebbe far prediligere la creazione delle API interamente su WSO2. Infatti, la creazione di API all'interno di WSO2 rende possibile l'intervento da parte di personale esterno formato sulla piattaforma WSO2. Tuttavia, va tenuto in considerazione che le aziende "certificate" WSO2 in Italia ad oggi (2021) sono solamente 2; mentre esistono tanti fornitori che sono formati su WSO2 e forniscono assistenza e supporto, pur non essendo certificati.

Realizzazione della API internamente all'Ente

Si prevede che l'API venga realizzata e manutenuta da personale interno all'ente? Se le API vengono realizzate internamente all'ente, allora esistono team specifici che a seconda del processo, realizzano le API secondo propri stack tecnologici di team. La formazione sulla piattaforma WSO2 non è completa per tutto il personale, e dunque, per consentire una intercambiabilità di risorse da allocare sui progetti, è preferibile proseguire con la realizzazione di API esternamente a WSO2 e poi procedere con la loro pubblicazione sulla piattaforma WSO2, almeno fintantoché la formazione del personale non sia stata completata in modo sufficiente da operare sulla piattaforma. Sul punto della formazione,

potranno avere un ruolo importante le persone che ad oggi sono state formate su WSO2 e che, dunque, potranno diffondere le conoscenze acquisite.

Facilità di abbandono di WSO2

In caso di cessazione dell'utilizzo di WSO2, quanto è oneroso attuare una migrazione della API ad altra piattaforma/modalità di fruizione? La costruzione della API internamente a WSO2 crea una dipendenza con la piattaforma (espone ad un vendor lock-in). In fase di progettazione è necessario chiedersi quanto può essere facile e snello attuare delle procedure di migrazione dalla piattaforma WSO2 ad altra piattaforma (exit-strategy), combinato con l'importanza nevralgica della API per l'Ente. Nel caso di una API strategica per l'ente, deve essere facile e sicuro poterla migrare su altra locazione per poter decidere di svilupparla interamente su piattaforma WSO2.

Sviluppo API tramite Enterprise Integrator

Come per la gestione standard dei progetti di sviluppo software, sarà necessario utilizzare un repository per il codice prodotto, che rende possibile archiviare il codice sorgente, condividerlo internamente, gestirne le versioni e ripristinarlo in caso di necessità.

Per i progetti di questo tipo verrà utilizzata una repository interna basata su **Git**. Ogni progetto di integrazione realizzato sul tool di sviluppo WSO2 Integration Studio corrisponderà ad un repository Git.

Durante la fase di sviluppo su WSO2, si suggerisce l'applicazione di un approccio "Git Flow", per rendere lineare e controllato il flusso di lavoro.

Per mantenere la separazione degli ambienti (produzione, staging, etc) si consiglia di separare in sotto progetti specifici le entità i cui valori dipendono dall'ambiente di deployment (ad es. URL, riferimenti ai database, etc.).

Disposizioni specifiche per il deploy automatico tramite procedure di CI/CD potrebbero essere sviluppate in versioni successive di questo documento.

Sviluppo API esterne a WSO2 ma compatibili per la pubblicazione tramite API Manager

Le API devono essere rappresentate mediante un Interface Description Language standard (IDL). Nello specifico:

- per REST, swagger 2.0 e successive, raccomandato OpenAPI 3.0;
- per SOAP, WSDL 1.1 e successive.

Si può applicare un meccanismo di autenticazione per le API che rientrano nei seguenti casi:

- espongono dati personali

- espongono servizi di aggiornamento dei dati che possono alterare in modo non controllato le relative risorse

Tale meccanismo impedisce che siano liberamente interrogabili dalla rete locale. Ad esempio, si possono implementare Basic Authentication per le API di tipo REST e WS-Security con utilizzo di UsernameToken per le API di tipo SOAP.

Esposizione API tramite API Manager

Tutte le API vengono esposte tramite WSO2 API Manager che opera come "reverse proxy evoluto" con le funzioni di autenticazione e autorizzazione, throttling delle richieste e gestione del lifecycle. Come raccomandato dalle linee guida AGID per l'interoperabilità[1], il numero di versione non deve essere presente all'interno del nome della API esposta perché viene gestito tramite API Manager.

Nell'esposizione tramite API Manager si indicano il numero di versione e la tecnologia nell'endpoint delle API secondo la seguente struttura:

https://api.comune.fi.it/[rest|soap]/<nome-api>/v<major>[.<minor>[.<patch>]]/<risorsa>

dove:

- [rest|soap] indica la tecnologia della API;
- <nome-api> indica il servizio che contiene le API relative. Potrebbe essere il nome di una applicazione specifica ("sigedo") oppure il nome di un servizio generico ("organigramma");
- v<major>[.<minor>[.<patch>]] indica il numero di versione in coerenza con Semantic Versioning 2.0.0;
- <risorsa> è il nome della risorsa specifica.

Ad esempio, una API Rest relativa al servizio "anagrafe" potrebbe esporre i seguenti end point:

Risorsa	Endpoint
Lettura anagrafica di un cittadino	GET https://api.comune.fi.it/rest/anagrafe/v1.0.0/anagrafica/{codice_fiscale}
Lettura dei nuovi nati in uno specifico anno	GET https://api.comune.fi.it/rest/anagrafe/v1.0.0/nuovi-nati/{anno}

Sicurezza delle API: servizi di Autenticazione ed Autorizzazione per le API

Le API esposte tramite API Gateway saranno protette tramite il protocollo Oauth2; per ogni applicazione verranno rilasciate una coppia di chiavi (consumer key e consumer secret) utili per l'accesso a tutte e sole le API necessarie per quell'applicazione.

[1] Linee guida: https://www.agid.gov.it/it/infrastrutture/sistema-pubblico-connettivita/il-nuovo-modello-interoperabilita
Raccomandazioni:
https://www.agid.gov.it/sites/default/files/repository_files/04_raccomandazioni_di_implementazione.pdf

Non è consentito l'utilizzo di una stessa coppia di chiavi da parte di più di una applicazione. Per ogni applicazione che vuole utilizzare i servizi esposti da WSO2, è necessario effettuare una sottoscrizione specifica ed ottenere una coppia di chiavi personalizzata e non cedibile.

Robustezza e sicurezza del sistema

L'infrastruttura è stata realizzata con una struttura di tipo cluster sfruttando le configurazioni messe a disposizione da WSO2, sono state esposte su internet solo le URL strettamente necessarie e comunque attraverso un bilanciatore centralizzato (in configurazione cluster active-passive) con funzionalità tipiche di Intrusion Detection, Intrusion Prevention e Web Application Firewall.

Revisione	Autore/i	Data
1.0 - Prima stesura del documento	V. Carboncini, R. Vannetti	2021/05/06
1.1 - Revisione dei contenuti e della struttura del documento, ancora in Draft	V. Carboncini, R. Vannetti	2021/05/26
1.2 - Revisione ed inserimento sezione “Realizzazione della API internamente all’ente”	R. Vannetti	2021/06/07
1.3 - Introduzione sezione “Robustezza e sicurezza del sistema”	V. Carboncini	2021/06/30
1.3: Revisione finale del documento	Direzione Sistemi Informativi	2023/05/31

Firmato digitalmente da:
MUNTONI DANIELE
Firmato il 27/05/2024 11:48
Seriale Certificato: 1845356
Valido dal 18/10/2022 al 18/10/2025
InfoCamere Qualified Electronic Signature CA