

**A.N.C.I.
FRIULI V.G.**

Manifesto per l'Integrazione e l'Interoperabilità dei Sistemi Informativi degli Enti Locali

Ver. 1.2

A cura del

Tavolo Tecnico sull'e-Government

Carlo Tasso – Università di Udine

Alessandro Garbino – Comune di Lignano

Antonio Scaramuzzi – Comune di Udine

Gianni Sumin – Comune di Pordenone

Ivano Sebastianutti – Comune di Tavagnacco

Paolo Londero – Comune di Faedis

Marco Malandrino – Comune di Maniago

Udine, 24 Settembre 2015.

Indice

Indice	2
Executive summary	3
2. Il problema	5
3. Principi generali e strategia di lavoro.....	5
4. Procedimenti amministrativi e software.....	9
5. Il ruolo fondamentale dell'opensource.....	13
6. Strumenti per l'interoperabilità in rete.....	14
7. Benefici dell'approccio proposto.....	15
8. Esempificazione dell'approccio proposto tramite un <i>caso di studio</i>	17
8.1 Definizione di un modello per la generazione di schemi dati certificati ai fini dell'interoperabilità.....	17
8.2 Funzionamento del "generatore di contratti dati"	18
9. Il Contesto FVG.....	21
10. Conclusioni	22

Executive summary

Questo documento affronta la problematica dell'interoperabilità e dell'integrazione dei Sistemi Informativi della Pubblica Amministrazione, ed in particolare degli Enti Locali.

Il problema della non integrabilità e non interoperabilità dei sistemi informativi è sentito in modo assai diffuso a tutti i livelli della Pubblica Amministrazione ed ha gravissime conseguenze:

- a. non è possibile lo scambio automatico di dati ed informazioni tra diversi sistemi informativi, sia interni ad una singola amministrazione (intra) che tra amministrazioni diverse (inter);
- b. i tempi di risposta degradano a causa della necessità di operare scambi di dati in modalità praticamente 'manuale', o quantomeno non efficacemente ed efficientemente automatizzata;
- c. non è agevole né veloce disporre in contemporanea di dati che riguardano processi organizzativi diversi e di conseguenza non si riescono a sviluppare efficienti ed efficaci sistemi informativi direzionali, rivolti a chi deve valutare andamenti, processi, fenomeni, performance gestionali/organizzative con il fine di prendere le più adeguate decisioni; e quindi, a cascata,
- d. l'informazione risulta spesso incompleta o non aggiornata, con ovvie conseguenze per lo svolgimento dei processi che ne fanno uso,
- e. gli archivi spesso vengono replicati, risultando spesso ridondanti e meno aggiornati/consistenti (minore uniformità, completezza e attendibilità del dato), con conseguente maggiore attività da parte degli operatori.

In ultima analisi, le conseguenze si ripercuotono sul cittadino e sulla qualità dei servizi.

I problemi sopra accennati sono avvertiti in modo particolare nei Comuni e le loro conseguenze saranno ancor più accentuate con l'avvio delle UTI (Unioni Territoriali Intercomunali), che di fatto porteranno alla definizione di processi inter-comunali.

Questo documento ha carattere tecnico ed illustra i risultati del Tavolo Tecnico dell'ANCI sull'e-Government. In particolare le criticità summenzionate vengono analizzate in maggior dettaglio, identificando i principi generali (linee guida/obiettivi) su cui basare la progettazione di sistemi informativi che permettano l'integrazione e l'interoperabilità. Le soluzioni proposte sono in linea con l'evoluzione tecnologica degli ultimi anni, in particolare relativa alla progettazione di complessi sistemi informativi inter-organizzazione, informatizzati e distribuiti su Web. Viene anche illustrato un Caso di Studio che aiuta a meglio comprendere l'approccio generale presentato e a coglierne i vantaggi.

Questo lavoro è rivolto per la parte generale ai decisori e per la parte squisitamente tecnica ai tecnici progettisti informatici (pur non essendo ovviamente un manuale tecnico). L'obiettivo è fornire uno strumento che indichi le direzioni e le metodologie di progettazione (ancorché non uniche ed

esclusive) che garantiscano fattibilità e flessibilità di (successivi o meno) processi di integrazione software che abbiano come possibile obiettivo il semplice scambio di dati ovvero forme di interoperabilità ed interazione più sofisticate.

Con riferimento al contesto Regionale, il Manifesto illustra in termini generali alcune delle criticità (specificamente relative al problema dell'integrazione e dell'interoperabilità dei sistemi informativi) che sono state segnalate dagli informatici degli Enti Locali, nell'ambito dei lavori del Tavolo Tecnico. Molte di tali criticità potrebbero essere superate adottando in modo condiviso e sistematico i criteri di progettazione richiamati nel presente documento.

Il Manifesto rappresenta quindi un primo fondamentale passaggio, ossia la presa d'atto dell'esistenza e criticità di una serie di problemi e la presentazione di adeguate soluzioni, basate su precisi criteri di progettazione e su moderne tecnologie ICT in grado di fornire risposte efficaci.

Il Tavolo Tecnico non ha affrontato la proposta e la redazione di un piano di lavoro per i successivi passaggi. Infatti, le problematiche da affrontare hanno una natura che va oltre l'aspetto puramente tecnico che, come già richiamato, è adeguato a fornire soddisfacenti e flessibili soluzioni. Al contrario, le criticità da affrontare sono di tipo politico, gestionale ed organizzativo (se non, da certi punti di vista, addirittura culturale). E conseguentemente, riguardano anche aspetti relativi al personale (informatico), ai costi, alle risorse finanziarie. Il Tavolo Tecnico auspica che su tale tematica la discussione si apra e si estenda ad altri importanti stakeholder in ambito regionale.

2. Il problema

I Comuni avvertono l'esigenza di aprire gli applicativi al mercato garantendo degli standard di interoperabilità. Allo stato attuale sono stabiliti degli standard di normativa per i dati relativi alle comunicazioni e allo scambio di informazioni con l'amministrazione centrale. Non vi sono standard definiti a livello di dati comunali.

Ogni fornitore di applicativi utilizza il proprio formato di banca dati ed i propri schemi. Ciò di fatto rende difficile (spesso impossibile), anche all'interno dello stesso Comune, lo sviluppo di applicativi che siano trasversali all'ente, rendendo impossibile il riuso di componenti, l'integrazione di applicativi e l'analisi statistica utile ai fini della programmazione politica del territorio (cosiddetti sistemi informativi o cruscotti *direzionali*).

Ciò è dovuto essenzialmente ad un approccio chiuso da parte dei fornitori software tendente, anche nella competizione tecnologica, alla tutela del proprio patrimonio software (schemi delle base dati) dimenticando volutamente nel contempo che il patrimonio dei dati e le banche dati sono in realtà di proprietà della Regione e degli Enti Locali.

Per garantire una maggiore flessibilità nell'utilizzo dei dati e per svincolare gli stessi dall'applicativo è necessario identificare delle priorità nello sviluppo e nella scelta delle soluzioni da adottare. Tali priorità hanno lo scopo nel tempo di perseguire l'efficacia e l'efficienza dell'azione amministrativa, non solo in termini di riuso e tutela degli investimenti, ma anche per garantire che gli applicativi comunichino tra loro in un linguaggio comune e lo sviluppo non segua un modello a "torre di Babele", accadendo ciò persino nello sviluppo di soluzioni all'interno della stessa software house o dello stesso comune.

3. Principi generali e strategia di lavoro

Lo sviluppo di sistemi informativi sempre più complessi e l'evoluzione dei processi di informatizzazione ha fatto emergere ormai da anni espliciti requisiti di integrazione e di interoperabilità. A fronte di tali requisiti le tecniche di progettazione software si sono evolute, identificando dei principi generali che garantiscono maggiori livelli di integrabilità

ed interoperabilità del software in generale e dei sistemi informativi più specificamente. Tali principi generali sono qui proposti come linee guida/obiettivi da considerare/perseguire da parte dei tecnici nel corso dell'attività di progettazione tecnica dei sistemi informativi. I più importanti principi generali da considerare nelle scelte progettuali riguardano:

- 1) interoperabilità software e hardware aperti,
- 2) integrazione di componenti eterogenei,
- 3) astrazione e indipendenza dalle tecnologie di basso livello,
- 4) scalabilità ed estendibilità delle soluzioni,
- 5) riusabilità delle soluzioni.

Qui di seguito sono brevemente illustrati.

Per *interoperabilità* si intende, in ambito informatico, la capacità di un sistema o di un prodotto informatico di cooperare e di scambiare informazioni o servizi con altri sistemi o prodotti in maniera più o meno completa, ovviamente priva di errori, con efficacia ed affidabilità e con ottimizzazione delle risorse (efficienza, adeguati tempi di risposta, ecc.).

Obiettivo dell'interoperabilità è dunque facilitare l'interazione (intra o inter-organizzazione) fra sistemi informativi differenti, nonché lo scambio ed il riutilizzo delle informazioni anche fra sistemi informativi non omogenei (sia dal punto di vista software che hardware).

Il termine interoperabilità è utilizzato in ambito tecnologico per indicare un elevato grado di sinergia di sistemi diversi col fine di offrire servizi (ad esempio servizi che si basano su processi complessi/trasversali) o funzionalità nuove (al limite nemmeno ipotizzate nel momento in cui i sistemi e gli archivi che vengono utilizzati furono concepiti e progettati). È direttamente legato alla ormai consolidata tendenza di far convergere su alcune tecnologie evolute una vasta gamma di servizi.

L'interoperabilità va di pari passo con l'*integrazione di componenti eterogenei*. I componenti delle soluzioni software devono essere visti come fornitori e/o consumatori di servizi, e per forza di cose sono spesso caratterizzati da forti differenze tecnologiche, sia dal punto di vista hardware (che però è l'aspetto che interessa di meno), ma soprattutto dal punto di vista software: i motivi sono molteplici, diversi fornitori, epoche di sviluppo diverse, progettazione 'chiusa' e non 'aperta', ecc. E' quindi ad un buon livello di interazione fra componenti eterogenei che mira l'interoperabilità.

L'interoperabilità si ottiene ponendo opportuna attenzione alle cosiddette interfacce fra i moduli software, ossia quella parte del sistema che permette lo scambio di dati o l'invio e la ricezione delle richieste di elaborazione. Le strade possono essere diverse: utilizzare opportuni standard, definendo in modo chiaro e accessibile i contenuti (ossia i dati che vengono reciprocamente scambiati/trasferiti) e le regole da seguire per l'interazione, ed in ultima analisi, concordando modalità e soluzioni condivise fra i moduli che interoperano.

Per astrazione e indipendenza dalle tecnologie di basso livello si intende il non essere legati ad una specifica base dati, ad uno specifico linguaggio di programmazione, ad uno specifico protocollo, ad uno specifico formato o ad una specifica tecnologia e soluzione software (analogamente si potrebbe dire dell'hardware, anche se, si ribadisce, i problemi che qui affrontiamo sono oramai concentrati quasi completamente – se non del tutto – nelle componenti software). L'astrazione e l'indipendenza dal basso livello si ottengono progettando i sistemi a livelli, ove i livelli più bassi gestiscono gli aspetti più tecnici e specifici della tecnologia, nascondendoli ai livelli superiori, che invece si occupano di definire le modalità di elaborazione, i flussi/workflow, l'interazione con l'utente, ossia gli aspetti più astratti che riguardano l'applicazione specifica e il dominio in cui essa è collocata. In tal modo i livelli alti sono indipendenti dalla specifica soluzione tecnologica del problema. Una buona progettazione a livelli permette di sostituire un componente tecnico di basso livello senza che sia necessario modificare i livelli superiori. E' chiaro però che componenti rimpiazzabili in questo modo devono essere realizzati con buone caratteristiche di interoperabilità!

La *scalabilità* si riferisce, in termini generali, alla capacità di un sistema di gestire la crescita (o la diminuzione) di scala in funzione delle necessità (di calcolo, di prestazioni, di servizio, di numero di utenti) e delle disponibilità (hardware e reti). Anche la scalabilità si ottiene utilizzando opportuni criteri già in fase di progettazione, in modo da poter poi adottare opportuni interventi quando le necessità si presentano: aggiunta di componenti più potenti (scalabilità verticale), affiancamento di altri moduli simili ai precedenti (scalabilità orizzontale). E' chiaro che interventi per l'aumento della scalabilità sono tanto più agevoli quanto più il sistema organizzato in modo che le infrastrutture hardware sottostanti non influiscano sui livelli più alti dell'applicazione.

Per *riusabilità*, infine, ci si riferisce alla possibilità di utilizzare in tutto, ma più frequentemente anche solo in parte, componenti sviluppate precedentemente per usi e scopi anche diversi da quelli inizialmente considerati, ma che comunque ben si adattano a fornire un contributo alle elaborazioni che hanno luogo in un nuovo sistema informativo o in una parte di esso. La riusabilità garantisce quindi la possibilità di realizzare nuovi sistemi con investimenti più limitati e con componenti già collaudati (almeno in parte). Ma la riusabilità dei sistemi software non è automatica e si ottiene solo se la progettazione è aperta, non cristallizzata su sistemi che comunicano e interagiscono in modo limitato e rigido. In generale, la riusabilità del software dipende anche dalle specifiche tecniche di analisi/progettazione/programmazione utilizzate per lo sviluppo. Applicazioni sviluppate con strumenti object-oriented hanno solitamente¹ un buon livello di riusabilità.

Il concetto di riusabilità è molto generale e può andare al di là del 'semplice' riutilizzo del codice sorgente. Un'evoluzione in un certo senso legata al concetto di riusabilità e che già da diversi anni ha preso piede è la modalità di distribuzione *SaaS (Software as a Service)*, che prevede di rendere disponibili via Internet moduli software caratterizzati da precise funzionalità e residenti tipicamente su cloud, che vengono combinati (e 'riutilizzati') in modo dinamico al fine di fornire specifici servizi.

La domanda che sorge a questo punto spontanea è: perché i pacchetti software, gli applicativi e più in generale i sistemi informativi non sono stati sviluppati tenendo conto di tali principi?

Una prima risposta banale è che ogni azienda che opera nello sviluppo del software ha marcato il proprio territorio. Ma le aziende, se avessero voluto, potevano stabilire degli standard di interoperabilità, cosa che solo raramente è stato fatto, tipicamente in situazioni e mercati molto ampi, con la presenza di player molto grandi e di problematiche vaste (che coinvolgono ampissimi interessi economici, operatori su scala globale, ecc.).

Una risposta alternativa è che il trattamento dei dati dei Comuni è soggetto spesso a cambiamenti e modifiche che dipendono dalle normative (vedasi, ad esempio, le norme sulla trasparenza amministrativa), che sono complessi da trattare e che emergono in successione varia nel tempo. Va osservato però che se da un lato le informazioni di base rimangono le stesse nel tempo (una persona ha nome, cognome, codice fiscale), dall'altro lato non è stato

¹ Attenzione: 'solitamente' e non 'sicuramente'. La riusabilità dipende infatti da come lo strumento di sviluppo viene utilizzato! Usando male un linguaggio object-oriented si possono realizzare moduli software per nulla riutilizzabili!

definito e concordato un modo semplice per esportare/importare tali informazioni e poterle quindi riutilizzare.

Il problema reale è che ci si è concentrati sugli applicativi dei singoli produttori software, sull'importazione ed esportazione dei dati dagli applicativi (anche in occasione di riorganizzazione degli enti) e non ci si è concentrati sul perché questi approcci non funzionano. Questi approcci sono orientati al prodotto e non al dato, inteso come entità su cui operare.

In termini più precisi e dettagliati:

- 1) il dato deve essere descritto con delle regole contrattuali che tutti possano capire e deve essere intellegibile, interpretabile univocamente per tutti;
- 2) il dato deve essere manipolato con operazioni che tutti possano capire e che tutti possano utilizzare e generare.

Solo così si può pensare di perseguire alcuni dei principi generali delineati sopra.

Come avvicinarsi ad una soluzione? Non basta purtroppo ribadire, da tecnici, che esistono degli standard per la descrizione dei dati (ad esempio, *JSON*² o *XML*³ ben formati). Bisogna individuare una strategia per portare effettivamente i fornitori software ad usare gli standard.

Una possibile strategia adottabile per raggiungere tale obiettivo può essere basata sull'analisi dell'operato della pubblica amministrazione che funziona per procedimenti amministrativi e sulla condivisione delle modalità di definizione dei processi e dei relativi dati tra i diversi attori coinvolti: pubblica amministrazione centrale e locale, fornitori di software e fornitori di servizi. L'approccio viene analizzato nella prossima sezione.

4. Procedimenti amministrativi e software

E' utile ricordare alcuni tratti caratteristici del concetto di 'Comune'. Consideriamo ad esempio quanto viene proposto in Wikipedia:

² <http://json.org/json-it.html>

³ <http://www.w3.org/XML/>

“Un Comune è un ente territoriale di base, dotato di un certo grado di autonomia amministrativa, dedicato agli interessi della popolazione locale. Si definisce, per le sue caratteristiche di centro demico nel quale si svolge la vita sociale pubblica dei suoi abitanti, l'ente locale fondamentale” [Wikipedia]

In sostanza gli enti locali ed in particolare i Comuni hanno a che fare con atti amministrativi legati al territorio e alla popolazione. Ma un atto amministrativo che cos'è? Sempre da Wikipedia:

“Un atto amministrativo è, nel diritto amministrativo, un atto giuridico posto in essere da un'autorità amministrativa nell'esercizio di una sua funzione amministrativa. Esso è espressione di un potere amministrativo, produttivo di effetti indipendentemente dalla volontà del soggetto o dei soggetti cui è rivolto.

La sequenza di atti amministrativi all'interno di un procedimento amministrativo, sotto la supervisione di un responsabile del procedimento amministrativo, porta invece all'emanazione di un provvedimento amministrativo.”

In sostanza il Comune emana atti e procedimenti attraverso procedimenti amministrativi, legati al territorio e alla popolazione.

Se cerchiamo una base di partenza per trovare un “linguaggio comune” nella babele degli applicativi (o una specie di Stele di Rosetta), conviene partire da questi concetti di base.

I procedimenti amministrativi sono seguiti da tutti gli enti, sono trasversali all'ente e sono comuni a tutti gli uffici, sia nel caso delle determinazioni (di impegno di spesa, di liquidazione degli impegni, di accertamento entrate) che nel caso delle deliberazioni, concessioni ed autorizzazioni, che prevedono comunque pareri tecnici.

I procedimenti amministrativi, in fin dei conti, sono processi aziendali che portano all'emissione di un provvedimento. Quindi una prima base condivisibile comune che è possibile individuare è costituita dal concetto di *processo*, concetto fondamentale anche nell'ambito dei sistemi informativi informatizzati. Tale è la centralità del concetto di

processo, che in ambito tecnico (informatico) sono stati sviluppate diverse specifiche tecniche standard di rappresentazione dei processi. In particolare, lo standard forse più noto esistente nel mercato del software per la rappresentazione (e quindi l'analisi e la gestione) dei processi organizzativi è *BPMN*⁴ (*Business Process Model and Notation*) che, inizialmente, era una notazione grafica in stile flow chart per la descrizione dei processi.

Inoltre, per far sì che i vari applicativi potessero scambiare le definizioni dei processi e gestire quindi alla stessa maniera i grafici rappresentanti i processi (spostare un grafico che rappresenta un processo da un programma ad un altro programma) è stato sviluppato un altro linguaggio molto diffuso, ossia *XPDL*⁵ (*XML Process Definition Language*), un linguaggio per la definizione dei processi di lavoro, che corrisponde in modo preciso alla notazione grafica. Entrambi i due linguaggi menzionati (BPMN e XPDL) offrono la possibilità di avere uno standard per la descrizione dei processi degli enti, per condividere queste definizioni tra enti, per fornire specifiche tecniche ai fornitori di software, per analizzare gli aspetti organizzativi al fine di una successiva informatizzazione, ecc.

Un'ulteriore possibilità che è emersa è quella di adottare i cosiddetti sistemi di *Workflow Management System*, ossia programmi che sono in grado di gestire l'esecuzione di un processo (quindi di supportare l'esecuzione di un procedimento amministrativo) partendo (i) da un insieme di moduli software di base che sono in grado di svolgere i vari passi dell'elaborazione e (ii) dalla descrizione di un *workflow*, ossia la definizione di uno specifico processo che sfrutta ed organizza l'elaborazione svolta dai vari moduli software. La cosa interessante è che mediante gli stessi moduli software di base si possono realizzare diverse elaborazioni, modificando (nella definizione del workflow) l'ordine di chiamata, i dati coinvolti, ecc. Di fatto, con workflow diversi, sugli stessi dati e con simili passi di elaborazione, si possono informatizzare procedimenti amministrativi anche assai differenziati. Analogamente, simili passi di elaborazione possono essere adattati, grazie a diverse definizioni del workflow, a enti che al loro interno hanno definito in modo diverso uno stesso procedimento amministrativo.

Lo standard XPDL è uno degli standard tipicamente adottati nei sistemi di Workflow Management, che quindi rendono de facto la gestione dei processi indipendente dall'applicativo utilizzato. E' anche ovvio però, che per utilizzare un qualsiasi modulo

⁴ <http://www.bpmn.org/>

⁵ <http://www.xpdl.org/>

software all'interno di un Workflow Management System, è necessario che il modulo sia conforme a precisi requisiti di interoperabilità, apertura, ecc., come precedentemente specificato.

Va osservato a questo punto che gli applicativi Regionali sviluppati sinora, non sono sistemi di workflow management, né ne utilizzano l'idea centrale. Gestiscono solo una parte dei processi e non sono facilmente personalizzabili per gestire altri processi, perlomeno per quanto è noto tramite le documentazioni disponibili. Ciò perché non si utilizzano strumenti espliciti per la rappresentazione dei processi e l'interfaccia è 'piatta' e non strutturata. In sostanza, gli applicativi rappresentano i processi deliberativi e le determinazioni nel rispetto dell'attuale normativa 'cablandoli' nell'interfaccia per mezzo di tecniche di programmazione tradizionale. Non c'è separazione fra gestione dell'interfaccia e processo/dato gestito: l'interfaccia dei programmi non viene generata a partire da uno schema concettuale del dato o da una rappresentazione del processo non essendovi alla base un sistema di workflow management. Quindi, qualora vi fossero dei cambiamenti di processo, questi dovrebbero essere riportati modificando i programmi con un rallentamento del processo di informatizzazione e inficiando la tempestività dell'azione amministrativa (ciò è successo, per esempio, con le modifiche apportate dalla "trasparenza amministrativa").

I procedimenti amministrativi lavorano su dati connessi alle persone e al territorio. Questi dati vanno considerati nei processi amministrativi. Come definirli e come standardizzarli? Si può ragionare come fatto in sede Regionale dal SUAP, ovvero definire dei modelli standard di dati e di modulistica, come i modelli dei documenti approvati a livello regionale, mediante dei comitati tecnici che definiscano tale base di partenza.

L'ideale per garantire la neutralità della definizione dei dati (ossia la sua indipendenza da fornitori specifici o da specifiche soluzioni tecniche non note) è che questa definizione parta dai Comuni, dove il dato viene originato e manipolato. Ciò potrebbe essere svolto anche con l'ausilio dell'Università, ente neutro e percepito neutro rispetto agli applicativi di terzi. La Società Insiel, in-house regionale, potrebbe porsi come prestatore di servizio in un'ottica di integrazione, di progettazione e sviluppo di specifiche con il compito di individuare soluzioni esistenti nel mercato ed impostando, anche grazie alla diffusione sul territorio, strategie di integrazione dei prodotti esistenti.

Comunque si voglia/si riesca a procedere nella definizione dei dati, è importante che le definizioni prodotte (dei dati, piuttosto che dei procedimenti amministrativi) vengano poi rappresentate mediante dei linguaggi standard (JSON, XML, XPDL, BPMN, ...) e soprattutto rese pubblicamente disponibili (ovviamente a livello dei tecnici informatici). In relazione a ciò, giova ricordare il ruolo dell'opensource e le normative che lo trattano per l'utilizzo in ambito pubblico (in Italia).

5. Il ruolo fondamentale dell'opensource

Un altro elemento essenziale di questo processo per l'apertura del dato e l'interoperabilità dei moduli software è che quanto viene sviluppato per conto della Regione e dei Comuni sia messo a disposizione come sorgente opensource, come peraltro previsto dalla normativa (D.lgs 82/2005 Codice dell'amministrazione digitale):

art. 69 Riutilizzo dei programmi informatici

1. Le pubbliche amministrazioni che siano titolari di programmi **informatici** realizzati su specifiche indicazioni del committente pubblico, hanno obbligo di darli in formato sorgente, completi della documentazione disponibile, in uso gratuito ad altre pubbliche amministrazioni che li richiedono e che intendano adattarli alle proprie esigenze, salvo motivate ragioni.

2. Al fine di favorire il riutilizzo dei programmi informatici di proprietà delle pubbliche amministrazioni, ai sensi del comma 1, nei capitolati o nelle specifiche di progetto e' previsto ove possibile, che i programmi appositamente sviluppati per conto e a spese dell'amministrazione siano facilmente portabili su altre piattaforme **e conformi alla definizione e regolamentazione effettuata da DigitPA, ai sensi dell'articolo 68, comma 2.**

3. Le pubbliche amministrazioni inseriscono, nei contratti per l'acquisizione di programmi informatici **o di singoli moduli**, di cui al comma 1, clausole che garantiscano il diritto di disporre dei programmi ai fini del riutilizzo da parte della medesima o di altre amministrazioni.

4. Nei contratti di acquisizione di programmi informatici sviluppati per conto e a spese delle amministrazioni, le stesse possono includere clausole, concordate con il fornitore, che tengano

conto delle caratteristiche economiche ed organizzative di quest'ultimo, volte a vincolarlo, per un determinato lasso di tempo, a fornire, su richiesta di altre amministrazioni, servizi che consentono il **riuso dei programmi o dei singoli moduli**. Le clausole suddette definiscono le condizioni da osservare per la prestazione dei servizi indicati.

È essenziale il controllo sul software sviluppato per conto della Regione. Mettere effettivamente a disposizione il software Regionale darà l'opportunità a chi vuole offrire servizi ai cittadini di innestare su quanto esistente lo sviluppo di ulteriori soluzioni innovative (con possibili interessanti ricadute anche per quanto riguarda lo sviluppo economico e la nascita e la crescita di una APP economy).

6. Strumenti per l'interoperabilità in rete

In maniera analoga a quanto gli strumenti JSON, XML, XPD L e BPMN permettono di fare sui dati e sui processi, fornendo la possibilità di accedere e interscambiare dati e definizioni di processi in modo aperto e trasparente, è anche necessario disporre di strumenti che permettano di accedere ai servizi di elaborazione (sia che si tratti di semplici elaborazioni ed archiviazioni, sia che si tratti di transazioni complesse). In altri termini, si deve garantire la possibilità di eseguire sul dato le necessarie elaborazioni (anche complesse, basate su una successione di passi elementari, possibilmente svolti da moduli software distinti) ed è quindi necessario disporre di strumenti che permettano di lanciare il modulo 'giusto' (ovunque esso sia residente), al momento 'giusto' (ovvero nel passo giusto di un determinato workflow) ed ovviamente sul dato 'giusto' (ossia reperito opportunamente attribuendo ai suoi vari elementi costituenti la giusta semantica/il giusto significato).

Questo scenario è affrontato da anni nei cosiddetti WIS (Web Information System), ossia nei sistemi informativi che vengono realizzati mediante diversi moduli residenti in rete e che operano sui diversi archivi anch'essi accessibili in rete. Si tratta di sistemi *distribuiti*, per la cui programmazione è necessario disporre di strumenti che da un server permettano di invocare l'esecuzione di un modulo software residente in un altro server accessibile in rete (magari a mille chilometri di distanza o all'interno dello stesso sistema cloud(!)) su dei dati

estratti da banche date accessibili su altri (database) server, scaricando i risultati su ulteriori archivi, anch'essi in rete o remoti o in cloud.

Gli strumenti per realizzare siffatte tipologie di elaborazione distribuita (attraverso la rete) si sono evolute negli ultimi decenni, e attualmente esistono alcuni standard de-facto, molto diffusi e che risolvono in modo 'semplice', sicuro ed efficace le problematiche considerate. Ci si riferisce ad esempio a strumenti molto noti come SOAP⁷, Web Service, XML-RPC⁸ e alle tecnologie REST⁹ (Representational State Transfer).

7. Benefici dell'approccio proposto

L'adozione dei principi generali di progettazione proposti nelle sezioni precedenti permette di ottenere molteplici benefici, richiamati qui di seguito:

- A. INTEROPERABILITÀ DEI SISTEMI

- B. INTEGRABILITÀ (CON ADEGUATI LIVELLI DI SICUREZZA) DI COMPONENTI E MODULI ETEROGENEI E SVILUPPATI IN MOMENTI DIVERSI, DA SVILUPPATORI DIVERSI, CON STRUMENTI DIVERSI

- C. MINOR DIPENDENZA DA SPECIFICI FORNITORI, MERCATO PIÙ APERTO, MAGGIORI OPPORTUNITÀ DI BUSINESS PER SOFTWARE HOUSE E SVILUPPATORI DI APP, MAGGIORE CONCORRENZA

- D. MAGGIOR CONSISTENZA E LIVELLO DI AGGIORNAMENTO DEI DATI

- E. FLESSIBILITÀ NELLA PROGETTAZIONE

- F. DIMINUZIONE DEI TEMPI DI SVILUPPO E RIDUZIONE DEI COSTI

⁷ <http://www.w3.org/TR/soap/>

⁸ <http://xml-rpc.net/>

⁹ https://en.wikipedia.org/wiki/Representational_state_transfer

- G. MODIFICABILITÀ ED ESTENDIBILITÀ DEI SISTEMI E RIDUZIONE DEI COSTI DI INTERVENTO DI MANUTENZIONE (ORDINARIA E STRAORDINARIA)
- H. MIGLIORAMENTI ORGANIZZATIVI ED OPERATIVI, CON POSSIBILITÀ DI CONFRONTO ED INTERAZIONE TRA DIVERSI ENTI SUL MIGLIORAMENTO NELLA GESTIONE DEI PROCEDIMENTI E DELLA STRUTTURA UTI, CON UN'OMOGENEIZZAZIONE DELLE MODALITÀ OPERATIVE E DOCUMENTALI E CONSENTENDO UNA PIÙ AGEVOLE INTERSCAMBIABILITÀ DEL PERSONALE
- I. POSSIBILITÀ DI PROGETTAZIONE E SVILUPPO DI APPLICATIVI CHE RIFLETTONO MODALITÀ INTER-ORGANIZZATIVE DI OPERARE TRA PIÙ ENTI, NON DEDICATI QUINDI AD UN SINGOLO ENTE E CHE AL CONTRARIO PERMETTONO DI USARE I SERVIZI E LE FUNZIONI DI ALTRI ENTI, ANCHE IN AVVALIMENTO.

Il punto I. evidenzia un fondamentale vantaggio utile nell'impostare le strutture organizzative delle UTI. Infatti nelle UTI, attraverso gli applicativi gestionali, sarà importante sostenere i servizi in avvalimento, ovvero consentire un modello di servizio delegato che ben si sposa con la distribuzione di tipo SaaS e che è pienamente realizzabile grazie ai modelli di interoperabilità qui proposti. E' ben vero che il servizio in avvalimento e il dover gestire pratiche per altri enti si potrebbe risolvere mediante scambio di PEC tra enti, ma essendo coinvolti nel processo enti diversi dotati di propri protocolli, verrebbe meno la tracciabilità del processo, con tutte le complicazioni derivanti dalla complessità di alcuni processi inter-ente, per esempio quelli relativi al governo del territorio, che richiedono più passaggi tra enti e anche pareri tecnico contabili per la predisposizione degli atti nei singoli comuni, oltretutto in sede UTI.

8. Esempificazione dell'approccio proposto tramite un *caso di studio*

Ci siamo soffermati nelle precedenti sezioni ad esaminare i problemi e ad indicare possibili soluzioni, ossia modalità che, seppur molto generali, permettono di progettare e di realizzare sistemi informativi caratterizzati da un buon livello di integrabilità ed interoperabilità.

Nel corso dei lavori del Tavolo Tecnico sono state analizzate delle esperienze specifiche che sono state già condotte con successo nell'ambito di alcuni Comuni della Regione. Si tratta di casi di studio che hanno concretizzato e reso palese i benefici dell'approccio proposto. Nel seguito verrà descritto un modello proposto e sperimentato dai Servizi Informatici del Comune di Lignano.

Si noti che l'aspetto da considerare non è relativo agli specifici strumenti software utilizzati, bensì all'approccio in generale, che si basa sui principi e le tecnologie illustrate nelle sezioni precedenti. Non si tratta quindi di una indicazione rigida, che prevede soluzioni e fornitori specifici, ma di un'esemplificazione che adottando le linee guida proposte risolve elegantemente le problematiche di integrazione e di interoperabilità dei sistemi informativi.

8.1 Definizione di un modello per la generazione di schemi dati certificati ai fini dell'interoperabilità

Basandosi sulle tecnologie richiamate precedentemente, il Tavolo di Lavoro ha ipotizzato uno schema di sviluppo neutro e open source, che automatizzi la definizione di contratti sui dati, facilitando il compito della stesura di questi contratti. La tecnica dei *contratti* è utilizzata nella progettazione software e si basa sul concetto di contratto, che stabilisce specifiche obbligazioni sui dati forniti da un generico modulo A che richiede l'esecuzione di un servizio ad un altro modulo B: le obbligazioni caratterizzano sia i dati forniti da A che le modalità di fornitura del servizio da parte di B. In altri termini il contratto governa le relazioni tra moduli che interoperano nello svolgimento di un processo e garantisce la corretta integrazione e funzionalità dello specifico passo di interoperabilità. Ovviamente, l'utilizzo di tale tecnica di progettazione è un metodo molto utile atto a garantire l'interoperabilità di moduli software,

il corretto utilizzo dei dati e l'integrazione di più moduli per cooperare nell'ambito di workflow più complessi.

Lo schema sperimentato prevede l'utilizzo di moduli software di medio livello (middle tier), indipendenti dai database. Tali moduli, a partire da una descrizione del dato, generano i contratti che gli applicativi di terze parti possono/devono successivamente utilizzare per poter creare/modificare/cancellare gli stessi dati.

Questa struttura informatica genera anche una interfaccia base via Web che consente la manipolazione diretta del dato.

In sostanza, si può utilizzare uno schema di base dati definito dai Comuni/Regione/Università e un motore applicativo che genera in maniera automatica:

- 1) una interfaccia Web utilizzabile direttamente dagli utenti in modalità di "amministrazione";
- 2) una interfaccia Web utilizzabile dai programmi. L'interfaccia Web generata automaticamente, utilizzabile dai programmi, è di tipo REST e funge da "contratto" per chi deve operare sui dati esponendo ai programmi sia il formato dei dati, sia le operazioni disponibili.

Da notare che la parte importante di quanto sopra è il formato del dato e le operazioni sul dato che vengono generate secondo standard internazionali e sono neutre, ossia non hanno alcuna dipendenza né dal formato e dalle modalità di archiviazione del dato, né dagli algoritmi e dalle modalità di elaborazione, né dallo specifico fornitore che ha fornito i moduli software di archiviazione e di elaborazione. In futuro è possibile modificare il livello sottostante, ma il formato dei dati e le definizioni delle operazioni rimangono gli stessi.

8.2 Funzionamento del "generatore di contratti dati"

Il funzionamento del generatore di contratti dei dati, la cui architettura stratificata a livelli indipendenti è illustrata nella figura seguente, si può sintetizzare come segue:

- 1) i dati possono essere definiti mediante modelli object oriented, utilizzando per esempio il framework opensource django¹⁰ (3a);

¹⁰ <https://www.djangoproject.com/>

- 2) Il layer applicativo django di interoperabilità è il cuore del sistema (2.a) ed espone gli oggetti JSON o XML ad applicativi sviluppati da terze parti;
- 3) Il layer applicativo, oltre ad esporre gli oggetti propri della base dati si interfaccia anche con un applicativo di workflow management (2.b), utilizzando Web service, garantendo quindi un disaccoppiamento tra il livello superiore di gestione degli specifici workflow applicativi ed i livelli implementativi sottostanti;
- 4) I plugin per l'aggancio alle base dati sono realizzati dallo sviluppatore 'finale', che potrà essere un servizio interno al Comune ovvero una software house terza parte (indicata con fornitore A e fornitore B in figura).

Il vantaggio dell'approccio è che i Comuni si concentrano sulla definizione dei modelli dei dati, lasciando buona parte del lavoro di costruzione dei formati al framework django. Infatti il sistema mette a disposizione automaticamente le specifiche dei dati, le relazioni e le modalità di operare sui dati, ovvero dei "contratti" a cui tutti i fornitori devono sottostare, specificati in una lingua 'franca' standard.

Anche il primo plugin di interoperabilità è generato in automatico dal modello object oriented dei dati permettendo di operare direttamente sui dati, grazie all'interfaccia generata automaticamente dal framework.

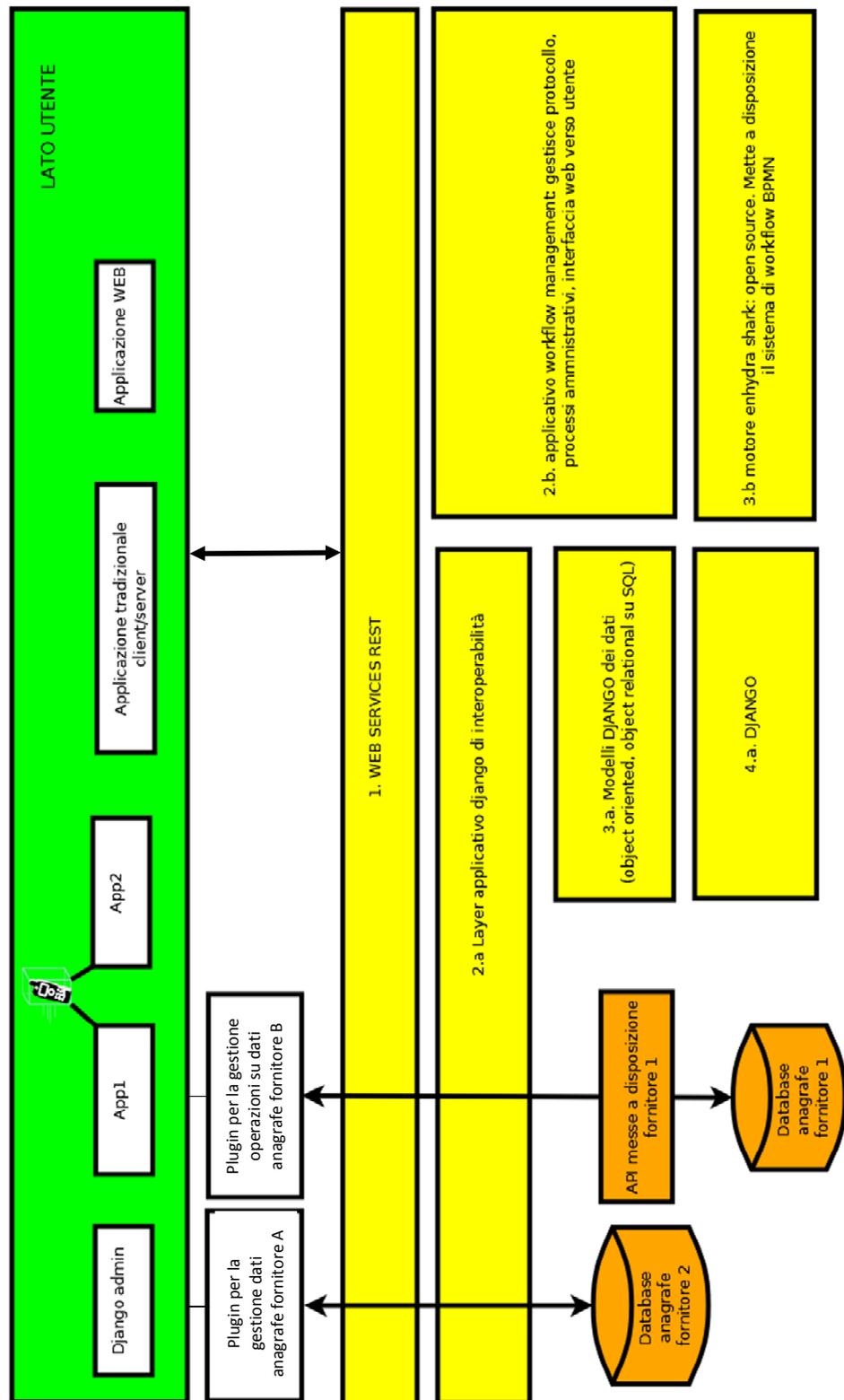


Figura: Architettura Esempiativa del Generatore di Contratti.

9. Il Contesto FVG

Il contesto del Friuli Venezia Giulia vive da anni i problemi descritti in questo Manifesto. Durante gli incontri del Tavolo Tecnico sono emerse più volte da parte dei Comuni molteplici criticità di varia natura:

- mancanza di apertura dei sistemi,
- scarsa flessibilità della progettazione software e quindi estrema rigidità (se non impossibilità) al cambiamento (ad esempio per modifiche successive) e tempi di risposta inaccettabili,
- scarsità di documentazione disponibile,
- assenza di condivisione e di diffusione di informazioni tecniche specifiche sui formati dei dati e sui meccanismi (raramente disponibili) di interoperabilità,
- e così via.

Esempi delle criticità illustrate nel Manifesto sono emersi con riferimento a tutti i fornitori software operanti in Regione: Società in-house Regionale INSIEL, società software più diffuse nel mercato nazionale, società private operanti in ambito regionale.

L'istituzione delle UTI porterà certamente ad affrontare fra breve ulteriori problematiche di integrazione ed interoperabilità dei sistemi informativi. L'approccio proposto in questo manifesto indica delle strade possibili¹¹per costruire, anche dal basso¹², all'interno delle singole UTI, utili ed efficaci sistemi informativi che si integrano ed interoperano.

¹¹ Strade possibili, ma non esclusive. Vogliamo ribadire come qui è stata sia fornita un'indicazione puntuale di alcuni degli strumenti più noti e diffusi sul mercato, ma questi non sono gli unici e i criteri generali di progettazione indicati possono essere realizzati anche con altri strumenti disponibili nel mercato ICT (opensource o meno).

¹² Partendo per esempio da piccole aggregazioni all'interno dell'UTI, su progetti anche limitati, che permettono di entrare proficuamente e progressivamente nella nuova modalità di progettazione e sviluppo.

...

10. Conclusioni

In questo Manifesto per l'Integrazione e l'Interoperabilità dei Sistemi Informativi sono state illustrate le negative conseguenze dei Sistemi Informativi (e più in generale dei sistemi software) che non sono in grado di interoperare. Sono stati successivamente illustrati dei principi generali di progettazione da utilizzare come linee guida/obiettivi dell'attività di progettazione software, al fine di ottenere sistemi aperti, facilmente integrabili ed interoperabili. Mediante tali modalità di progettazione sarà quindi possibile realizzare effettivamente dei Sistemi Informativi inter-organizzazione. Per illustrare più in dettaglio come utilizzare i principi introdotti e per dimostrarne la fattibilità ed i vantaggi, è stato sinteticamente proposto un caso di studio realizzato dal Comune di Lignano.

E' convinzione dei partecipanti al Tavolo Tecnico che la discussione sulle tematiche illustrate possa essere affrontata con due orizzonti temporali diversi:

- Nel breve termine, specifici progetti di integrazione di sistemi informativi possono trarre beneficio dall'adozione dei principi e delle soluzioni qui presentati, come già successo in alcuni casi (Monfalconese, Comunità della Carnia, Pordenone, Tavagnacco, ed altri).
- Nel medio termine, nell'ambito di un processo di aggiornamento e ri-organizzazione del digitale in Regione, possono essere intrapresi percorsi condivisi di revisione e innovazione delle metodologie di sviluppo delle applicazioni rivolte agli Enti Locali, superando gli ostacoli e le criticità oggi presenti.

Il lavoro è stato svolto con la convinzione che l'attivazione delle UTI, ed in particolare le problematiche relative ai Sistemi Informativi delle UTI, possano trarre soluzioni concrete, di agevole e veloce applicazione e con vantaggi significativi.