

# Progetto IDRA REST - Manuale architetturale

Il presente documento introduce la versione web del sistema IDRA.

## *1.1 Introduzione al sistema*

Il progetto Idra è un sistema per la catalogazione dei reperti archeologici secondo la normativa ICCD. Il progetto Idra REST è un applicativo web distribuito che mette a disposizione un webservice pubblico attraverso il quale è possibile interrogare la base di dati attraverso gli standard REST ovvero lo standard de facto per i web services nel mondo web 2.0.

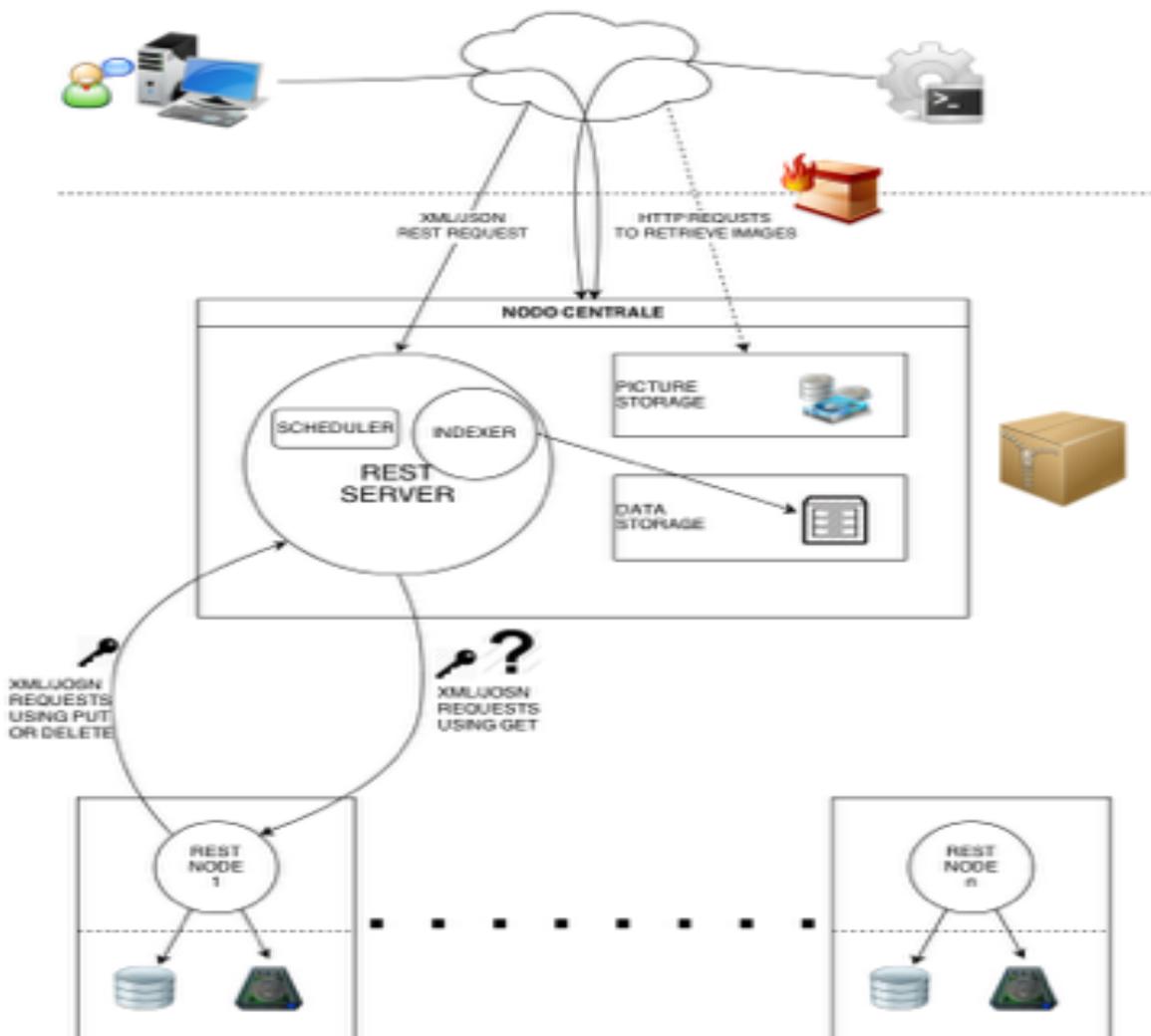
Il sistema seguirà la filosofia ROA (Rest oriented architecture) a discapito di SOA (Service oriented Architecture) per la maggiore flessibilità e facilità d'integrazione con le nuove tecnologie.

Il progetto consentirà di recuperare e visionare le varie informazioni memorizzate nei sistemi di competenza aderendo all'idea di "open data", ormai standard per tutte le pubbliche amministrazioni.

L'accesso al sistema permetterà la divulgazione dei contenuti archeologici datati e catalogati nei vari sistema sparsi su tutto il territorio creando un punto di congiunzione.

### 1.2 Architettura del sistema

Il diagramma sotto servirà ad illustrare la soluzione architettonica nell'ambito del progetto Idra per la divulgazione delle informazioni legate ai reperti archeologici



La struttura è divisa in due macro blocchi architettureali che corrispondono alle due componenti da sviluppare nel suddetto progetto.

Le componenti sono:

- Rest Server
- Rest Node

Tali strutture sono state disegnate per mantenere un alto grado di disaccoppiamento seguendo la filosofia ROA.

Non c'è una dipendenza diretta delle componenti ma vengono esposti degli endpoint in grado di far comunicare i vari sistemi.

La componente Rest Server o anche nodo centrale renderà omogeneo uno strato applicativo che altrimenti risulterebbe di difficile accesso. Tale componente ha una serie di strutture interne che permettono la memorizzazione e la divulgazione delle informazioni contenute all'interno di esso.

Le componenti facenti parte del sistema sono le seguenti:

- Rest Server
  - Indexer
  - Scheduler
- Picture Storage
- Data Storage

### *1.2.1 Rest Server*

La componente Rest Server è il cuore del progetto Idra, è il collante per tutte le altre componenti all'interno del sistema.

Di base si tratta di una componente REST in grado di esporre tutta una serie di endpoint sia verso l'esterno (utente finale) che verso le varie componenti interne all'architettura. La comunicazione avviene tra i vari nodi interni (es. Idra) e il Rest Server in maniera bidirezionale.

I compiti principali sono quelli di leggere le varie informazioni all'interno dei vari nodi e di permettere, grazie a un sistema di autenticazione, di inserire o cancellare varie informazioni direttamente dalla sorgente al nodo centrale.

Le comunicazioni seguono gli standard più recenti quali json e xml. All'interno del nodo centrale viene effettuato il marshalling e l'unmarshalling dei vari oggetti.

Rest Server esporrà un API sia verso client/applicazioni esterne che nodi interni

### *1.2.1.1 Scheduler*

Lo scheduler all'interno del Rest Server permetterà di sincronizzare le informazioni contenute nelle varie sorgenti con quelle in possesso del Nodo Centrale. Il tutto avverrà attraverso la schedulazione di chiamate asincrone verso le varie sorgenti che esporranno un API comune a tutte le sorgenti di dati ovvero i nodi interni.

### *1.2.1.2 Indexer*

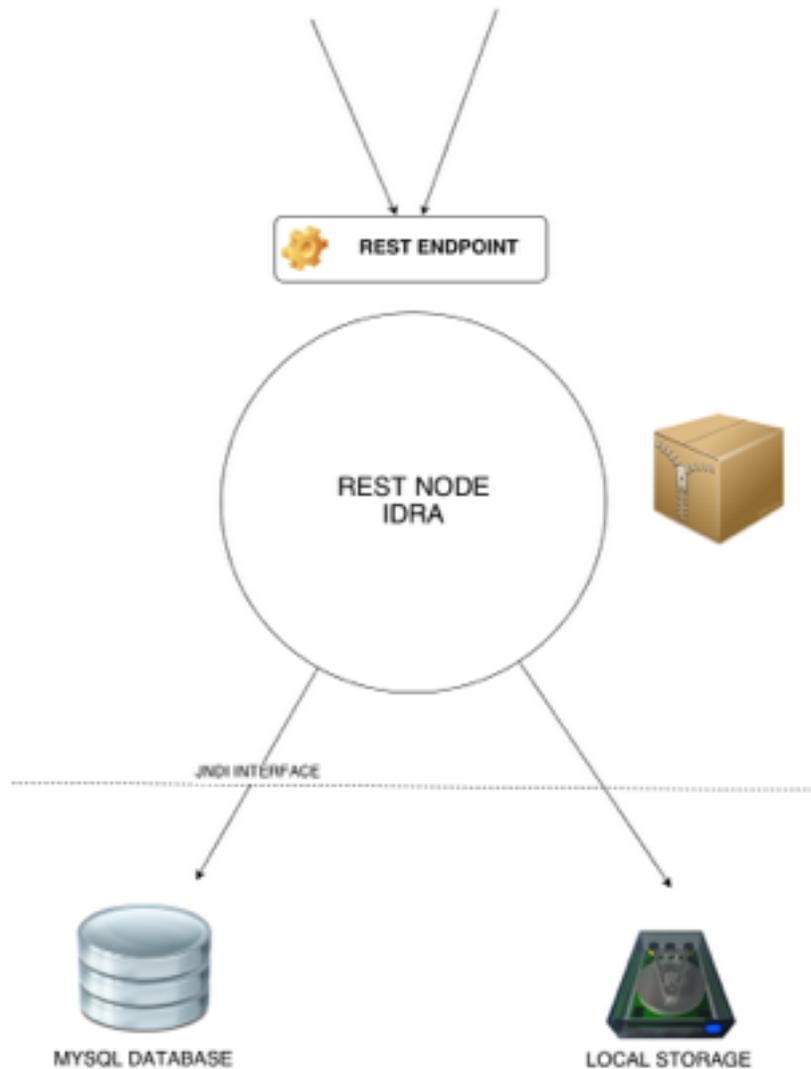
Le informazioni recuperate dalle varie sorgenti saranno memorizzate e indicizzate per ottenere alte prestazioni nel recupero delle stesse informazioni e per evitare di interrogare i nodi periferici del sistema che detengono l'informazione nel momento in cui si fa una ricerca o un interrogazione al nodo centrale tramite web service pubblico.

### 1.2.2 Rest Node

La componente Rest Node rappresenta l'informazione che deve essere distribuita sui vari sistemi (interni ed esterni).

Anche il nodo sorgente come il nodo centrale sarà una componente REST che si interfacerà direttamente con Rest Server per alimentare l'informazione centralizzata.

In basso il diagramma schematico del nodo periferico:



Nello specifico il Nodo Idra si interfacerà direttamente con la base di dati MySQL di Idra per il recupero delle informazioni. La comunicazione verso la base di dati avverrà tramite interfaccia jndi verso il data source.

### *1.3 Comunicazione tra nodi*

La comunicazione tra i vari nodi e quindi la sincronizzazione delle informazioni è gestita dal nodo centrale del sistema che si occupa di recuperare l'informazione originariamente mantenuta dai nodi periferici, di indicizzarla in un formato comune ed uniformato e di renderla disponibile all'esterno tramite un'insieme di api REST che forniscono la ricerca e la consultazione dei dati in oggetto.

Il nodo centrale, detto anche REST server, deve ottenere l'informazione dai nodi periferici utilizzando a sua volta delle chiamate REST verso i nodi in maniera attiva oppure esponendo in maniera passiva degli endpoint per l'aggiornamento dei dati. Questi due meccanismi vengono denominati pull e push.

#### **1.3.1 Meccanismo pull**

Il meccanismo pull è usato per i client che espongono servizi rest in sola lettura. In tale meccanismo il nodo centrale si occupa di recuperare l'informazione memorizzata nel nodo periferico facendo delle chiamate ai webservice che il nodo espone. Questa modalità verrà utilizzata al momento della messa in esercizio del nodo centrale per popolare i dati con i dati iniziali. Questa modalità inoltre verrà utilizzata per aggiornare periodicamente, mediante l'uso di uno scheduler, per aggiornare i dati provenienti da quei nodi che non supporteranno la modalità push definita in seguito.

#### **1.3.2 Meccanismo push**

Dopo l'import iniziale del dataset di un determinato nodo il modo più efficiente, sia in termini di banda che di risorse computazionali, per aggiornare la base di dati condivisa si ha quando un nodo interno si preoccupa di notificare la base di dati dei cambiamenti in esso avvenuti, possibilmente anche in realtime. In questa modalità, detta appunto push, il nodo interno può richiedere l'aggiornamento dei propri dati in modalità realtime, ovvero fare push dei dati aggiornati verso il nodo centrale al cambiamenti di questi (per esempio in caso di aggiunta, modifica o cancellazione di schede)

### **1.3.3 Gestione delle immagini**

Per la gestione delle immagini collegate alle schede ci sono due possibili soluzioni nell'architettura proposta. Di seguito la descrizione delle due modalità previste

1. Il nodo interno espone tramite protocollo http le immagini delle schede in suo possesso ed inoltra al nodo centrale l'url al quale l'immagine risulta disponibile
2. Il nodo interno invia l'immagine stessa al nodo centrale (oppure l'url dove essa è raggiungibile) ed il nodo centrale salva (o recupera) l'immagine e la salva nel suo storage interno e fornisce ai client esterni un indirizzo http esternamente accessibile per il recupero dell'immagine da parte dell'utilizzatore finale.

La soluzione 1 è sicuramente di più facile implementazione ed anche computazionalmente meno onerosa per il nodo centrale in quanto non deve occuparsi di servire i files grafici in questione, ma pone un vincolo architetturale per i nodi che vogliono partecipare alla rete, infatti in alcuni caso l'approccio proposto potrebbe non essere fattibile. Si pensi ad esempio ad un nodo interno che si trova dietro un firewall e che non è pubblicamente accessibile.

Viceversa l'approccio descritto nella soluzione 2 è più generale e non pone alcun vincolo per i nodi interni.

La soluzione da utilizzare per risolvere questo problema va concordata con la Lait anche in funzione delle risorse messe a disposizione per l'esecuzione del nodo centrale.

### *1.4 Deploy componenti*

Lo sviluppo produrrà varie componenti software. Il tutto sarà composto da due WAR (Web Archive) e vari jar (il numero esatto è ancora in fase di definizione).

I due War saranno rispettivamente la componente centrale (Rest Server) e il nodo sorgente (Idra). I vari jar saranno le librerie custom condivise tra le varie componenti REST.

Tutte le componenti Software sviluppate verranno rilasciate come Enterprise Archive (EAR) per agevolare il rilascio in produzione su application server.