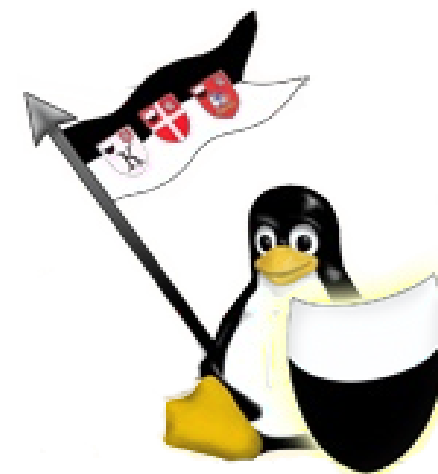


# Database spaziali e sistemi GIS

*Un assaggio di dati geografici, in salsa opensource...*



**SLUG**

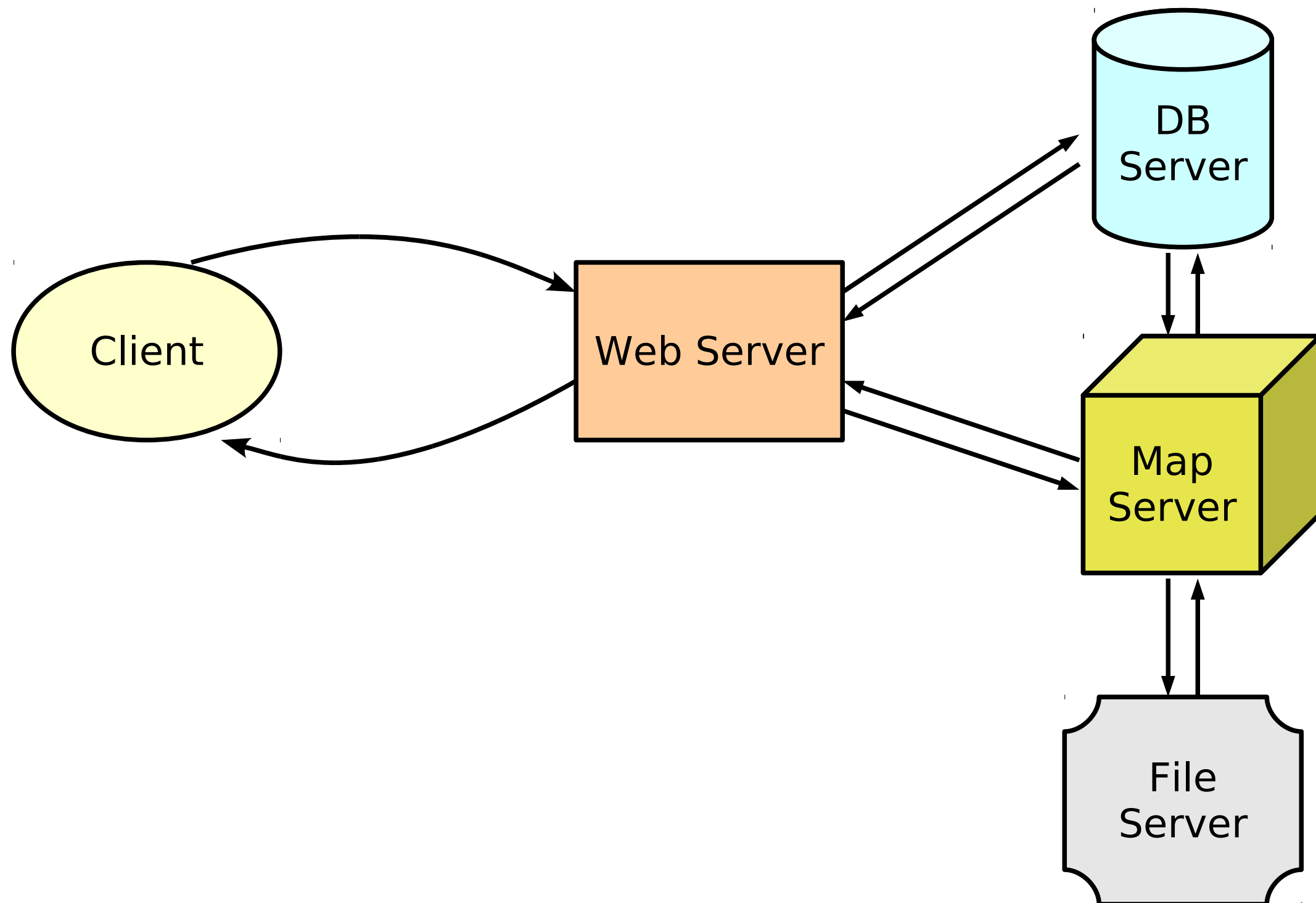


# Cosa (non) vedremo

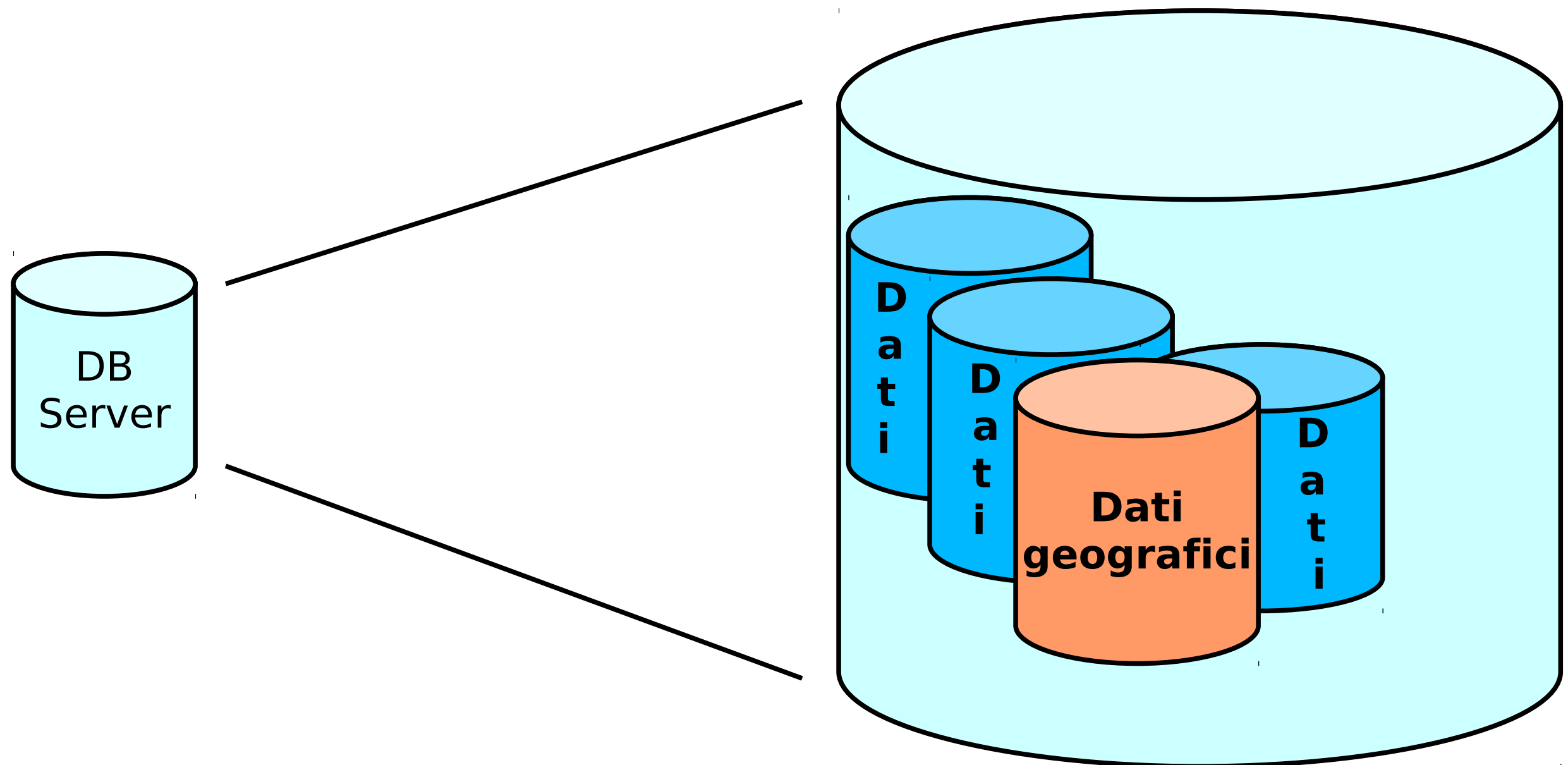
- Vista complessiva di un sistema GIS
- Cosa sono i dati geografici (cenni)
- Formati principali e il formato Shapefile
- Perché serve un database geografico?
- L'estensione PostGIS: intro, tipo di dato, funzioni, query spaziali, indice GIST
- QGIS
- Server delle mappe (cenni)
- Un assaggio di OpenLayers

*Non vedremo come ottenere i dati con una licenza d'uso ragionevole...*

# Vista complessiva di un sistema GIS



# Vista complessiva di un sistema GIS



# Dati Geografici?

- Sono dati che sono **georiferiti**, ossia definiti da coordinate rispetto ad un certo sistema di riferimento
- I sistemi che fanno uso di dati geografici sono detti **GIS** (Geographical Information Systems)
- In italiano si sente spesso parlare di **SIT** (Sistemi Informativi Territoriali)
- Non sono esattamente la stessa cosa: i sistemi GIS sono composti da strumenti software, i SIT invece comprendono solitamente anche le procedure di acquisizione, elaborazione, validazione e presentazione dei dati geografici
- *Per questo vi parlo di GIS, la questione dei dati è spinosa...*



# Dati raster

- Ortofoto (eventualmente georeferenziate)

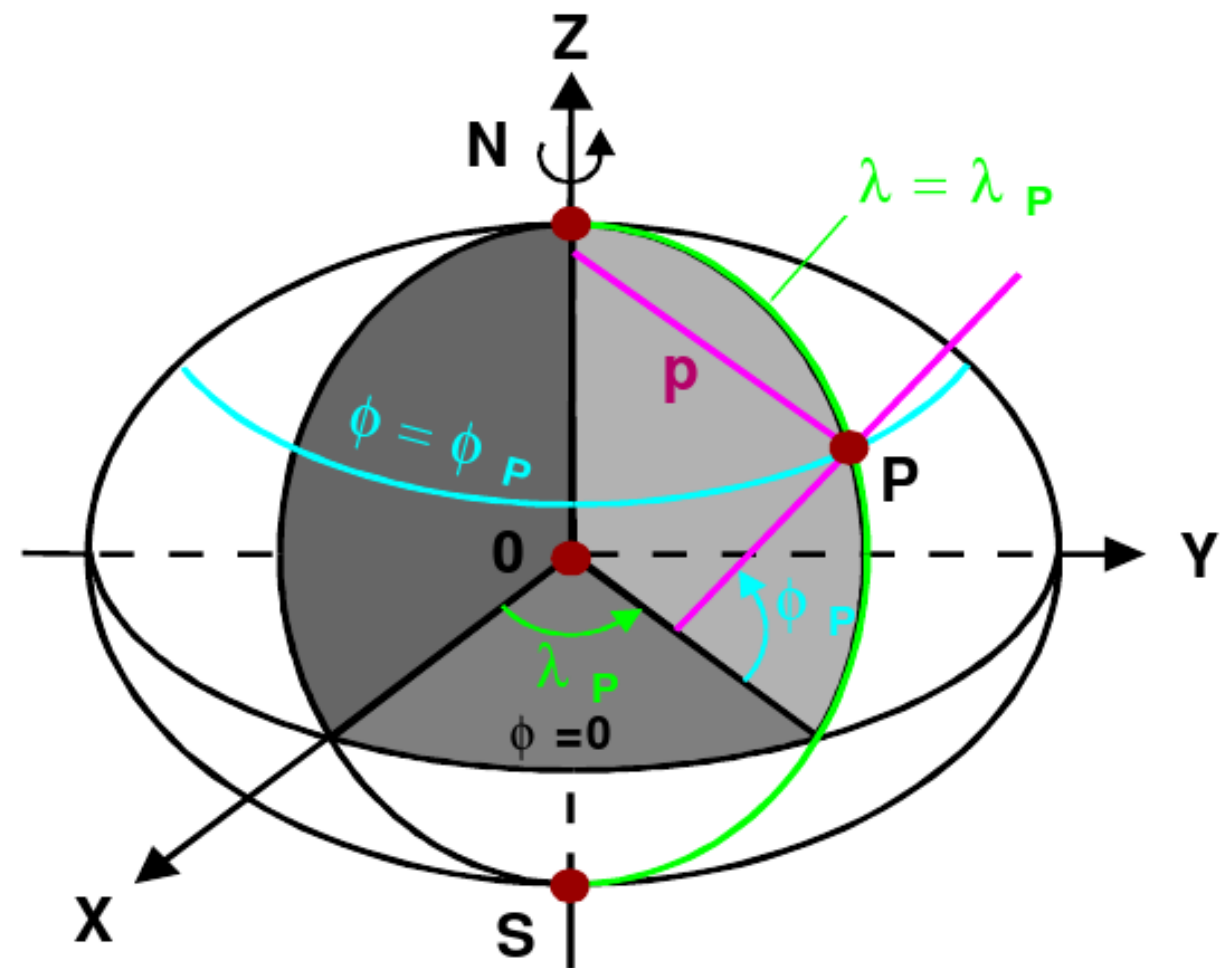


# Dati vettoriali

- Descritti per mezzo di primitive geometriche
  - Punti
  - Linee
  - Poligoni
  - Curve (Bezier, Spline ...)
- Molto compatti
- Più facili da “maneggiare” (calcolo dell'area, del perimetro ecc...) e da “trasformare”

# Sistemi di coordinate

- La superficie terrestre può essere rappresentata con vari modelli matematici, via via più sofisticati, ciascuno dei quali descrive un punto sulla superficie terrestre con un insieme di coordinate diverso
- **WGS 84**: modello di riferimento per l'ellissoide di riferimento che approssima la superficie terrestre usato ad esempio nel GPS





# Proiezioni

- Metodo matematico per trasformare le coordinate sulla superficie tridimensionale curva della terra in quelle di un supporto bidimensionale
- Ne esistono moltissime, alcune valide solo localmente, altre con l'ambizione di essere valide su tutta la superficie terrestre
- Molte sono nazionali, alcune sono storicamente legate a modelli diversi del geoide

# Proiezioni

- Dato che non è possibile rappresentare una sfera su un piano senza deformazioni (una conseguenza di un teorema di Gauss), le proiezioni possono essere classificate in base alle proprietà che riescono a preservare (aree, angoli, distanze) oppure rispetto al modo in cui sono state ottenute (sviluppo su un solido, su un piano tangente ecc...)
- Il discorso si fa rapidamente matematico...



# Formati per i dati geografici raster

- Per i dati raster, i formati usati sono quelli delle immagini
  - JPEG
  - TIFF
  - PNG
  - Formati basati su wavelet (JPEG 2000, ECW)
  
- La georeferenziazione avviene associando all'immagine un file di testo che descrive le coordinate di un vertice dell'immagine e le dimensioni di ogni pixel espresse nell'unità di misura del sistema di coordinate scelto

# Formati per i dati geografici

- Per i dati vettoriali esistono molti formati e diversi sono proprietari, ossia sono stati definiti dai produttori di software come formati nativi per certe applicazioni
  - Autodesk => DWG, DWF (AutoCAD)
  - ESRI => Coverage (Arc/INFO)
- Esistono anche layer di compatibilità per scrittura su database
  - Esri ArcGis
    - personal geodatabases (MS Access!)
    - ArcSDE (si interfaccia con Oracle, IBM DB2, MS SQL Server e vari altri)

# Il formato Shapefile

- Definito dalla ESRI come formato di interscambio per l'interoperabilità fra applicazioni diverse
- La sua specifica è sostanzialmente aperta
  - (<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>)
- Si è quindi affermato come formato di riferimento ed esistono molti software in grado di leggerlo e modificarlo.



# Il formato Shapefile

- Uno “Shapefile” è composto in realtà da molti files, alcuni obbligatori ed altri facoltativi, legati fra loro dal fatto di possedere lo stesso nome, a meno dell'estensione
  - Un file .shp. *Obbligatorio*, contiene le geometrie. Es. *strade.shp*
  - Un file .shx. *Obbligatorio*, contiene un indice posizionale sulle geometrie. Es. *strade.shx*
  - Un file .dbf. *Obbligatorio*, contiene un database su file (nel formato “vetusto” dBase III) con i dati scalari (attributi) legati alle geometrie. Es. *strade.dbf*
- La cosa interessante è che *non è obbligatorio* dichiarare il sistema di coordinate adottato (tale dichiarazione sta in un file opzionale). Se esso non è presente le geometrie quindi hanno coordinate arbitrarie e non possono essere riproiettate in un altro sistema di riferimento

# Rappresentazione delle geometrie

- Point, Line, Polygon
  - Molto semplice, ma per rappresentare curve “dolci” occorrono molti punti (non c'è nulla di analogo a splines o a curve di Bézier)
- ... e loro generalizzazioni (multipoint, multiline, multipolygon)
- Ci interessano perché sono simili alle definizioni di base della specifica **Simple Features** dell'OGC (Open Geospatial Consortium) ... lo vedremo

# Vantaggi del formato Shapefile

- È un buon formato di interscambio perché le specifiche sono aperte
- Ci sono molte applicazioni, anche open source che sono in grado di leggerli e scriverli
- È un formato semplice da implementare e da utilizzare

# Svantaggi del formato Shapefile

- Alcune scelte discutibili (alcuni campi sono big-endian, altri little-endian)
- Non è possibile costruirci topologie (che possiamo immaginare essere per i dati geografici l'analogo dei *constraints* dei database)
- Il formato dBase III ha molte limitazioni (ad esempio i nomi dei campi limitati ad 8 caratteri, secondo la convenzione DOS 8+3)
- Normalmente ogni Shapefile ha un solo tipo di geometrie (punti, linee, poligoni)

# Lo svantaggio principale

- Gli elementi geometrici (file shp) e quelli scalari (file dbf) sono collegati implicitamente, ossia i due file hanno lo stesso numero di righe e il collegamento è dato dal numero!
- Ossia ... è molto facile corrompere irreparabilmente lo shapefile modificando contemporaneamente i due file (ad esempio aggiungendo una geometria e cancellando una o più righe dal database dbf)



# Perché è utile un database spaziale?

- ... per gli stessi motivi per cui è utile un database normale, ossia:
  - Velocità ed efficienza nel reperimento delle informazioni
  - Gestione degli accessi concorrenti
  - Transazioni
  - Gestione delle politiche di accesso ai dati (autenticazione, viste)

# Database con estensioni spaziali

- Alcuni hanno estensioni che permettono di manipolare dati geografici, altri sono usati come backend con tipi di dato binari
  - Oracle + Oracle Spatial
  - MS SQL Server 2005/2008
  - IBM DB2 + Spatial Extender
  - MySQL + estensione spaziale
  - **PostgreSQL + PostGIS**

# Alcuni standard di riferimento

- Open Geospatial Consortium (OGC): associazione su base volontaria che comprende più di 370 fra aziende, enti pubblici, enti di ricerca e organizzazioni no profit
- Il suo scopo è creare e favorire l'adozione di standard per la gestione di dati geografici e per l'implementazione di servizi che permettano di condividere tali dati
- Ha creato molti standard utili per la definizione di dati geografici e per l'interscambio

# Simple Features

- Quello che a noi interessa è chiamato **Simple Features**
- Esso permette di specificare dati geometrici (punti, linee, poligoni, collezioni di geometrie ecc...) secondo un formato detto WKT (**Well Known Text**)
  - POINT(6 10)
  - LINESTRING(3 4,10 50,20 25)
  - POLYGON((1 1,5 1,5 5,1 5,1 1),(2 2, 3 2, 3 3, 2 3,2 2))
  - MULTIPOINT(3.5 5.6,4.8 10.5)
  - MULTILINESTRING((3 4,10 50,20 25),(-5 -8,-10 -8,-15 -4))
  - MULTIPOLYGON(((1 1,5 1,5 5,1 5,1 1),(2 2, 3 2, 3 3, 2 3,2 2)),((3 3,6 2,6 4,3 3)))

# L'estensione PostGIS

- Nasce intorno al 2001 come prodotto dell'azienda Refractions Research
  - <http://www.refractions.net/products/postgis/>
  - <http://postgis.refractions.net/>
- Una prima versione è rilasciata nel Maggio 2001 con licenza GNU GPL
- Implementa lo standard Simple Features definito da OGC
- È efficiente (*si può fare una demo su un netbook ...*)



# Il tipo di dato Geometry

- È un tipo di dato binario molto compatto per la memorizzazione delle geometrie. Questo consente di minimizzare l'occupazione su disco e velocizzare il recupero dei dati
- Ci sono delle funzioni per convertire il dato geometrico in un formato “più umano”, quello WKT

# Funzioni geometriche

- Unioni, intersezioni
- Relazioni fra geometrie (inclusione, tangenza, intersezione ecc...)
- Funzioni di misura (aree, perimetri ecc...)
- Altre funzioni (routing, interpolazione, proiezione di un punto su una linea ecc...)

# Come si usa (query spaziali)

- In modo molto semplice, all'interno del normale linguaggio SQL (si supponga che le tabelle abbiano un campo ***the\_geom*** di tipo ***geometry***)

```
SELECT max(area(the_geom)) FROM laghi WHERE tipo='vulcanico'
```

```
SELECT *
FROM strade
WHERE Intersects(the_geom,GeomFromText('LINESTRING(3 4,10 50,20
25)'))
```

# Indice GIST

- Quale indice usare per velocizzare l'esecuzione delle query spaziali?
- L'indice GIST è stato creato appositamente per i dati geometrici.
- Può essere utilizzato per memorizzare varie strutture dati, come gli alberi B+ o gli alberi R

# Perché un indice GIST?

- Vantaggi dell'indice GIST rispetto ad un comune indice basato su R-tree
  - Gli indici basati su R-trees in PostgreSQL non possono indicizzare oggetti grandi più di 8k
  - Gli indici basati su R-trees in PostgreSQL non sono *null-safe* ossia fallirebbero in caso di geometrie nulle ma soprattutto ...
  - L'indice GIST è **lossy** (ossia l'indice non è fatto sulle geometrie ma sui *bounding box*), quindi può indicizzare geometrie arbitrariamente grandi



# Perché un indice GIST

```
SELECT * FROM particelle WHERE Intersects (the_geom,  
GeomFromText('POLYGON((1 1,5 1,5 5,1 5,1 1))'))
```

- La cosa più conveniente da fare inizialmente è cercare le particelle le cui geometrie hanno **bounding box che si interseca con quello del mio poligono** ... (poco costosa)
- ... per poi rifinire la ricerca facendo l'intersezione vera e propria (molto costosa se geometrie con molti punti)
- Infine, l'indice GIST è perfettamente integrato con il **query planner** di PostgreSQL

# Installazione di PostGIS

- Disponibile per Windows, Linux e MacOSX, sotto forma di binari precompilati o di codice sorgente.
- Ogni versione di PostGIS può essere installata come estensione per una o più versioni di PostgreSQL.

```
#aptitude install postgresql-8.3
```

```
#aptitude install postgresql-8.3-postgis
```

# Abilitazione di PostGIS

- L'estensione PostGIS deve essere abilitata a *livello di singolo database*, non per l'intero DBMS
- È quindi possibile che nello stesso server PostgreSQL convivano database abilitati e non abilitati

*Se da un lato questo costringe ad un lavoro extra per ogni database, dall'altro ci consente di avere i tipi di dato e le funzioni geografiche solo dove servono*

# Abilitazione di PostGIS

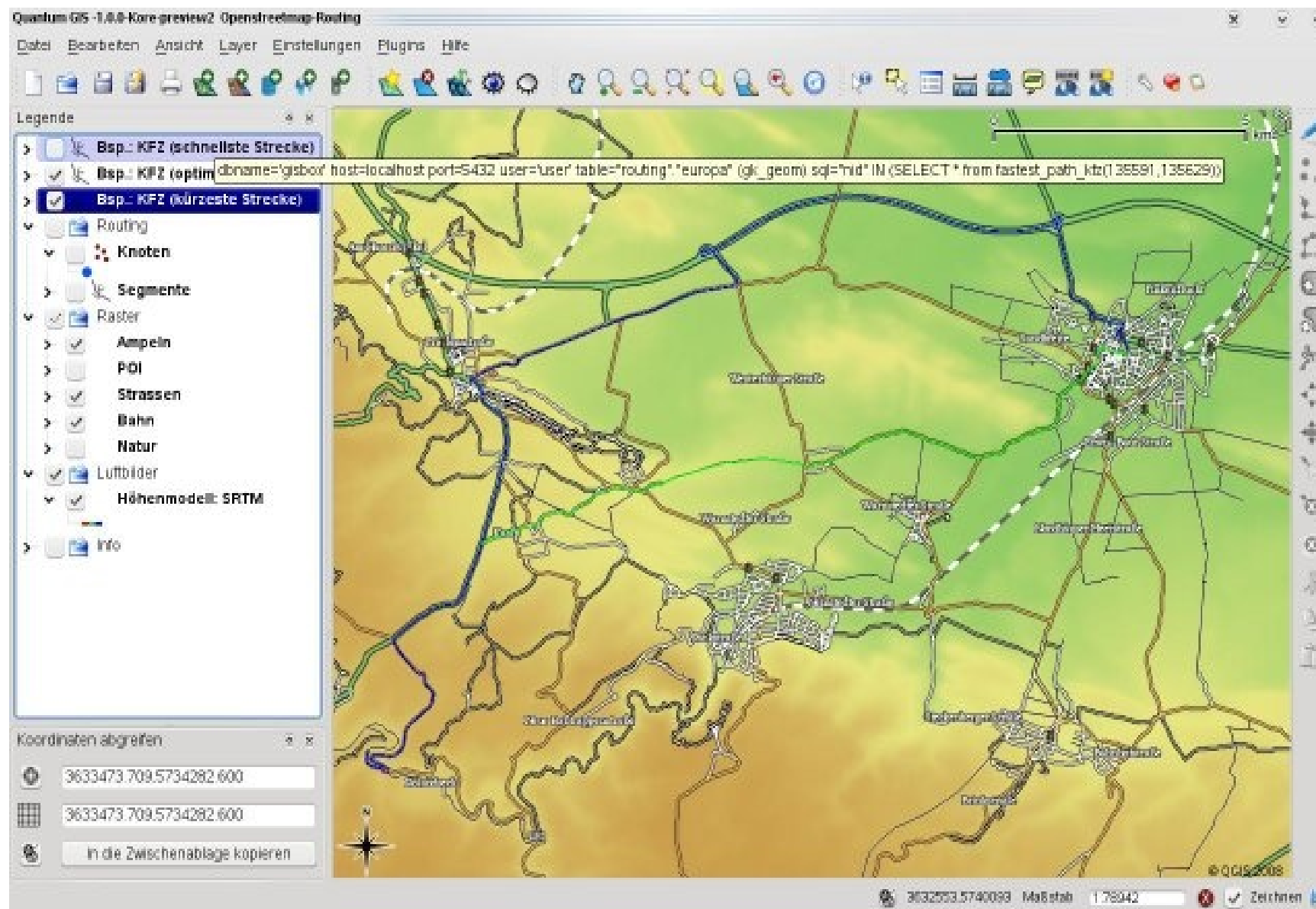
- `createlang plpgsql yourtestdatabase`
  - Aggiunge il linguaggio procedurale plpgsql al database
- `psql -d yourtestdatabase -f lwpostgis.sql`
  - Aggiunge le funzioni proprie dell'estensione
- `psql -d yourtestdatabase -f spatial_ref_sys.sql`
  - Aggiunge la tabella dei sistemi di riferimento (oltre 4000 righe...)
- ... et voilà!!

# Importazione ed esportazione

- Come si fa a inserire ed esportare dati geografici nel nostro database?
- Ci sono due comode utility per importare uno shapefile in una tabella ed esportare una tabella con un campo geometria in uno shapefile.
- Però abbiamo anche il vantaggio di poter fare tutto anche tramite SQL

# Visualizzazione dei dati

- QGIS (<http://www.qgis.org/>) - Win & Linux & MacOSX (GPL)



# Piccola demo

- Shapefile e sua visualizzazione in QGIS
- Abilitazione PostGIS di un database
- Importazione shapefile
- Visualizzazione campo the\_geom e AsText(the\_geom)
- Visualizzazione dati su db in QGIS
- Creazione Indice GIST e Vacuum Analyze
- Nuova visualizzazione da QGIS



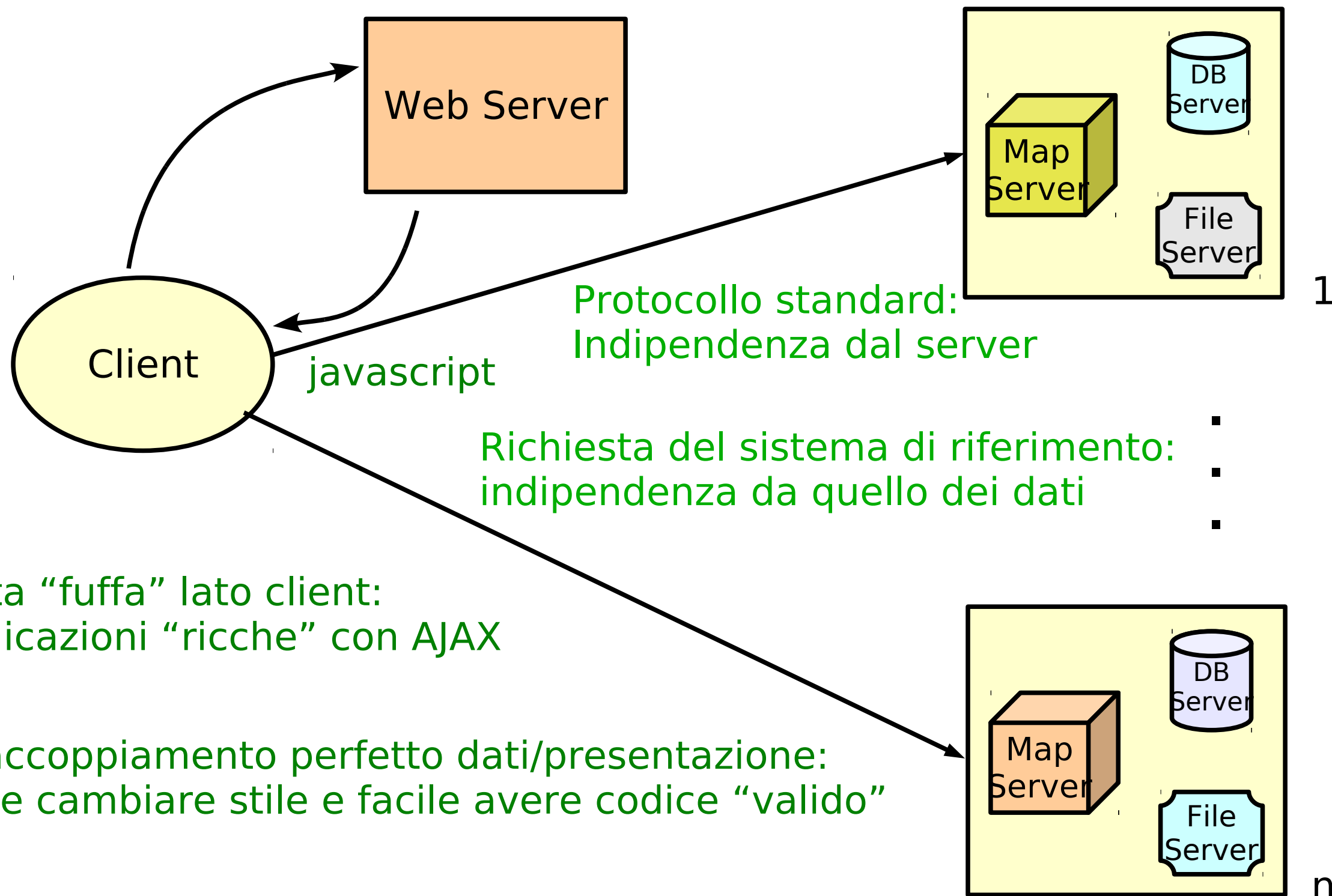
# Cosa ci manca ancora?

- ... il server delle mappe!
- Mapserver
- Geoserver
- Autodesk MapGuide Open Source

*Ognuno ha i suoi pregi ed i suoi difetti, e ne esistono altri con licenza GPL o LGPL.*

*Questa lista vuole essere solo un punto di partenza*

# Un approccio furbo: OpenLayers



# Addendum

- Nulla sarebbe divertente se non ci si potesse sporcare le mani. Meglio ancora se è possibile farlo con dati veri.
- Esistono molti dati d'esempio liberamente scaricabili da Internet.
- Tra essi, mi piace segnalare i seguenti (licenza CC, some right reserved):
  - [http://grass.osgeo.org/sampled/data/north\\_carolina/nc\\_shape.tar.gz](http://grass.osgeo.org/sampled/data/north_carolina/nc_shape.tar.gz)

**GRAZIE PER L'ATTENZIONE**