Desarrollo en gvSIG 2.0

Introducción a las novedades de desarrollo en gvSIG 2.0

http://www.gvsig.org

Joaquín José del Cerro — IVER T.I.
César Ordiñana — DiSiD Technologies S.L.
Jorge Piera — IVER T.I.

Contenidos del taller

- Repaso a las principales novedades de cara al desarrollo en gvSIG 2.0
- Para más detalles sobre cada punto, hay disponible:
  - Documentación de desarrollo de gvSIG 2.0

    http://www.gvsig.org/web/docdev/gvsig_desktop_2_0/
  - Charlas en junio del 2009 sobre desarrollo gvSIG 2.0

    http://www.gvsig.org/web/docusr/learning/gvsig-courses/gvsig_des_2.x_d

Novedades de desarrollo en gvSIG 2.0

- Nomenclatura de paquetes, proyectos y jars normalizada
- Uso de maven
- Arquitectura común entre gvSIG desktop y mobile
- Separación entre API, SPI e implementación en parte del core. Ej:
  - geometrías
  - proyectos
  - documentos
  - etc.
- org.gvsig.tools
- División de FMap en varios proyectos
- Nueva librería de acceso a datos (DAL)
- Migración a OSOR.eu

Novedades de desarrollo en gvSIG 2.0

- Otros:
  - Primeros pasos de extracción del API de simbología
  - Migración al API de logging de SLF4J
  - Librería de compatibilidad Java SE – Java ME CDC
  - Nueva implementación del documento tabla
  - etc.

Nomenclatura de paquetes Java

- Normalización en la nomenclatura de paquetes: todos empiezan por *org.gvsig.* Ej:

  - libTools: *org.gvsig.tools.**
  - appgvSIG: *org.gvsig.app.**
  - libFMap_mapcontext: *org.gvsig.fmap.mapcontext.**

Nomenclatura de archivos .jar

- Normalización en la nomenclatura de archivos .jar
  - Estructura: *PAQUETE-VERSION[-CLASIFICADOR].jar*
    - PAQUETE: paquete principal del proyecto.
    - VERSION: versión del proyecto.
    - CLASIFICADOR: (opcional) para distinguir en proyectos que generan más de un jar.
- Ej: (libTools)
  - *org.gvsig.tools-2.0.0.jar*
- Ej: (libFMap_geometries)
  - *org.gvsig.fmap.geometry-2.0.0.jar*
  - *org.gvsig.fmap.geometry-2.0.0-impl.jar*

Nomenclatura de proyectos

- Se emplea como nombre el paquete principal del proyecto
  - Sólo siguen esta nomenclatura proyectos nuevos, desde hace un tiempo, aunque poco a poco se irán cambiando.
- Ej:
  - org.gvsig.symbology
  - org.gvsig.scripting
  - ...

Maven

- ¿Qué es maven? (http://maven.apache.org)
  - Herramienta para la gestión y construcción de proyectos software.
- Características principales de la versión 2.x:
  - Configuración por convención basado en buenas prácticas de desarrollo.
  - Estructura de proyecto por defecto.
  - Compilación y construcción de proyectos.
    Tareas típicas predefinidas como compilar, empaquetar, javadocs, ...
  - Generación de documentación técnica.
  - Configuración de proyecto basada en el archivo pom.xml
  - Uso de repositorios para la distribución de binarios, fuentes, javadocs, etc.

Maven

- Ventajas que aporta a gvSIG:
  - Configuración y construcción uniforme de todos los proyectos.
  - Eliminar archivos .jar de subversion
  - Gestión unificada de dependencias externas.
  - Versionado de los jars generados por nuestros propios proyectos.
  - Generación de informes sobre el estado del código fuente.

Maven

- Repositorios de maven:
  - Repositorio local:
    - $USER_HOME/.m2/repository
  - Repositorio oficial de maven:
    - http://repository.sonatype.org/index.html
  - Repositorio de gvSIG:
    - http://gvsig-desktop.forge.osor.eu/downloads/pub/projects/gvSIG-desktop/maven-repository/

Maven

- Objetivos habituales
  - mvn compile: compilar
  - mvn test: lanzar los tests unitarios
  - mvn package: generar los archivos .jar
  - mvn install: copiar los archivos .jar al repositorio local
  - mvn deploy: copiar los archivos .jar al repositorio remoto
  - mvn clean: borrar todos los artefactos generados por maven en el proyecto

Maven

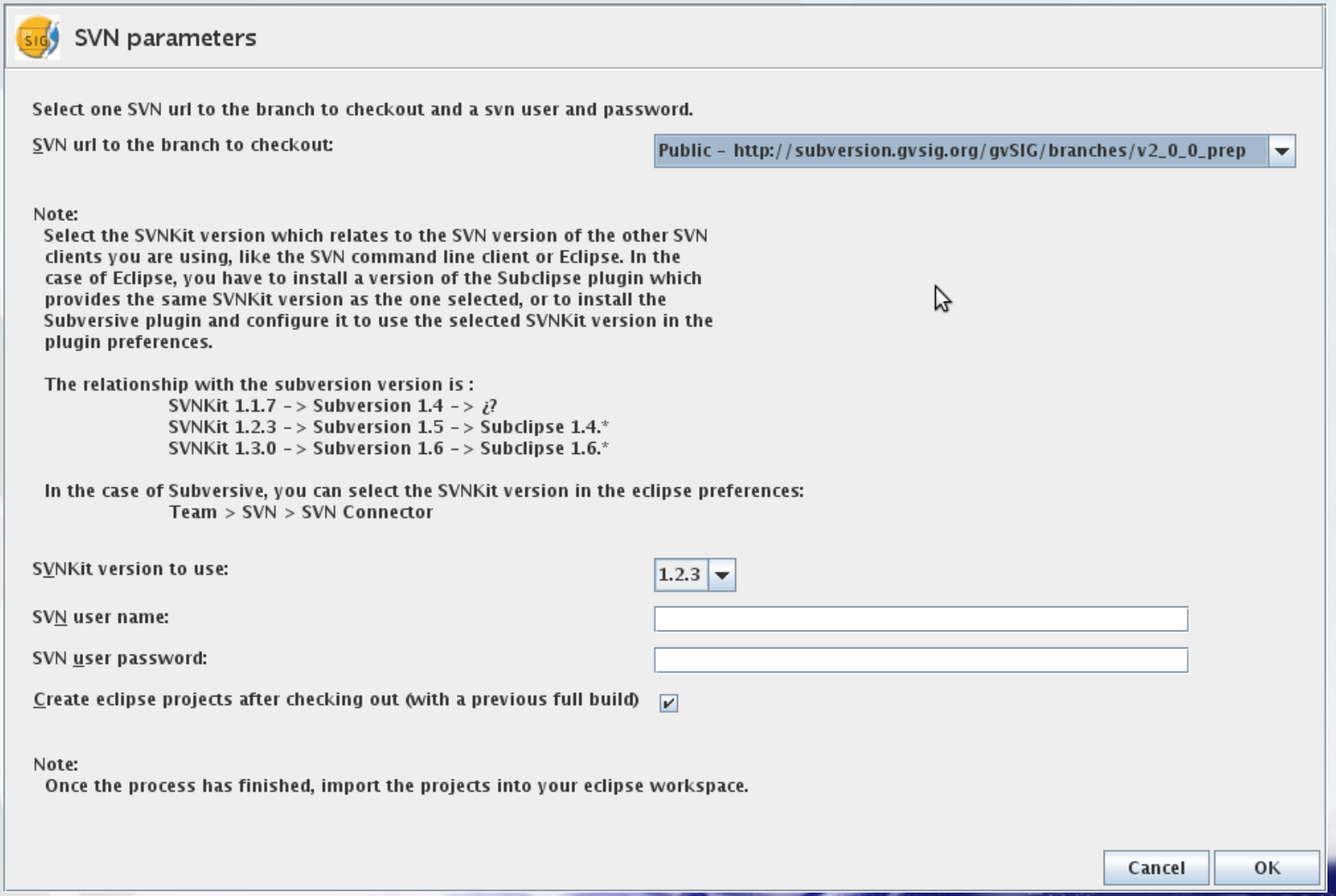
- Configuración de maven en gvSIG 2.0
  - Configuración común en el proyecto org.gvsig.maven.base
  - Configuración base para proyectos de tipo:
    - librería
    - librería nativa
    - extensión
  - Podemos heredar fácilmente de esta configuración en nuestros proyectos.
  - Librerías nativas: compilación y descarga de dependencias integrada en maven

Maven

- Integración de maven en eclipse para gvSIG 2.0
  - Ver guía de desarrollo:
    - http://www.gvsig.org/web/docdev/gvsig_desktop_2_0/guia-de-desarrollo/
  - Abrir apartado "*Cómo montar un workspace de gvSIG para Eclipse*"

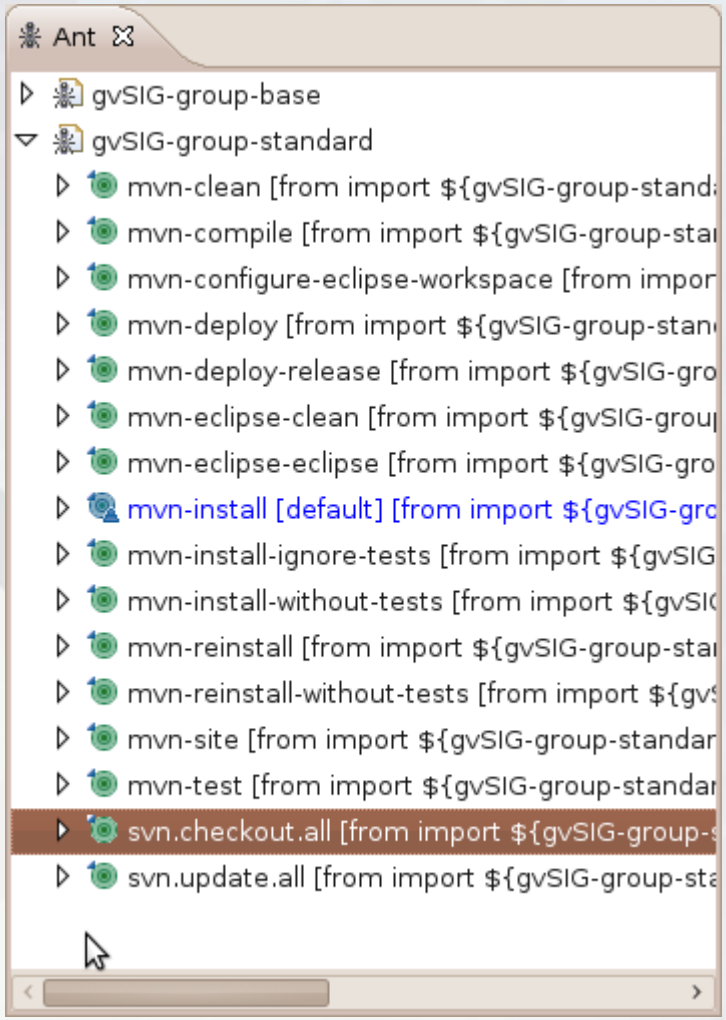
Maven

Checkout de los proyectos de gvSIG



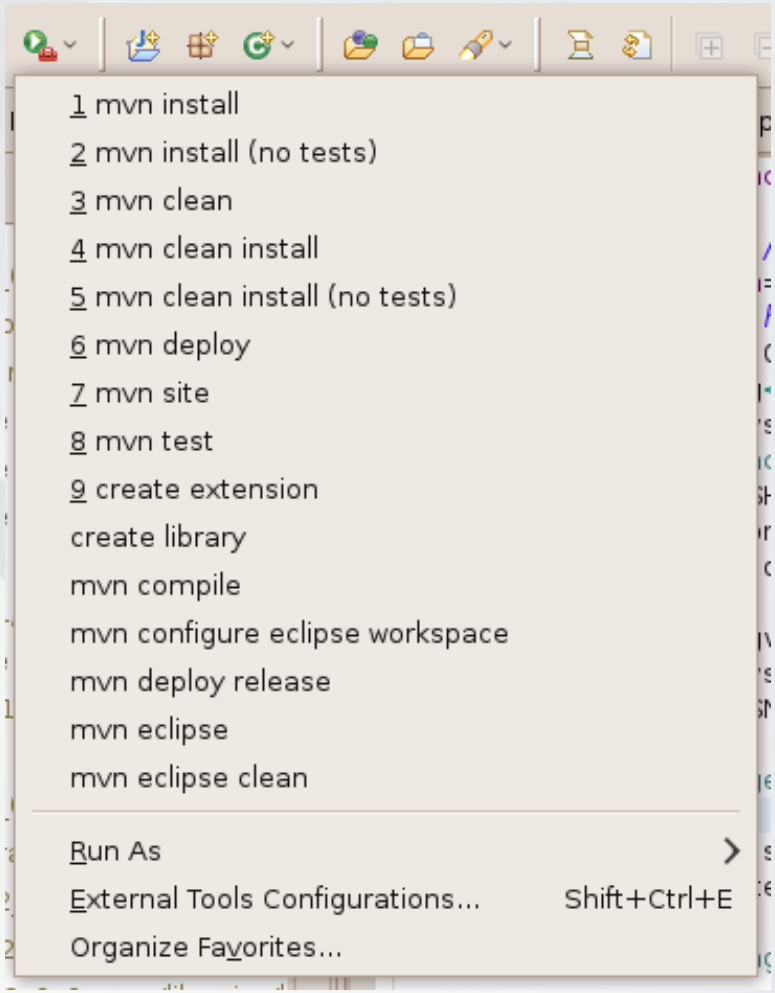
Maven

- Objetivos de construcción sobre todos los proyectos



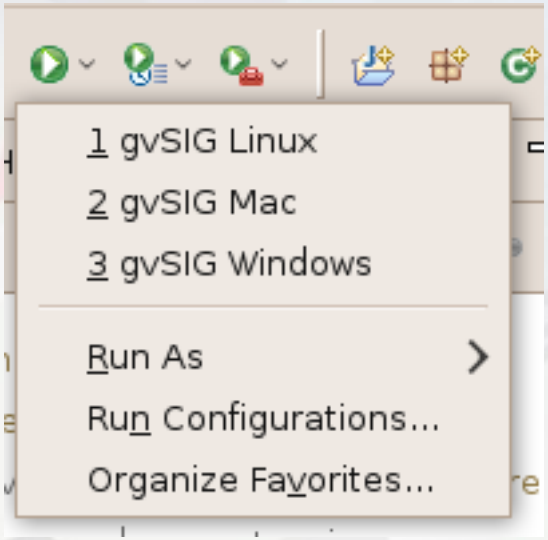
Maven

- Objetivos de construcción sobre un proyecto concreto (eclipse external tools)



Maven

- Arrancar gvSIG (eclipse *launchers*)



Maven

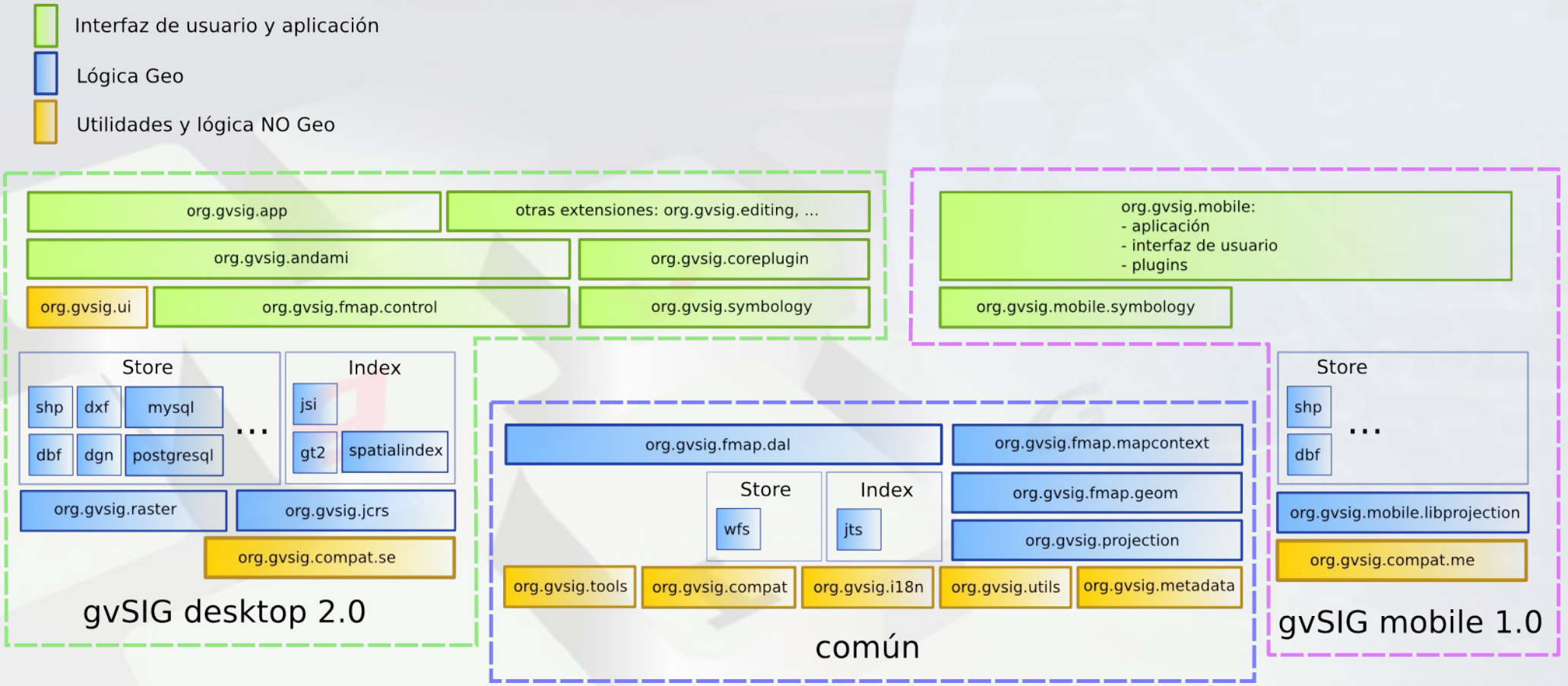
- Charla sobre maven realizada en julio
  - Presentación:
    - http://gvsig-desktop.forge.osor.eu/downloads/pub/documents/learning/gvsig-courses/gvsig_des_2.x_d/Maven-workspace.pdf
  - Video:
    - http://gvsig-desktop.forge.osor.eu/downloads/pub/documents/learning/gvsig-courses/gvsig_des_2.x_d/videos/maven.mp4

Evolución de la arquitectura de gvSIG

gvSIG Mobile

- Versión de gvSIG para Java ME perfil CDC
- El perfil CDC es compatible con Java 1.4, con la salvedad de algunos paquetes, clases y métodos que no existen. Ej:
  - java.lang.String :: String[] split(String input, String regex)
  - java.lang.Math:: double log10(double value)
  - java.nio
  - javax.swing
- ¿Qué tiene que ver gvSIG Mobile con gvSIG 2.0 (desktop)?

Arquitectura común

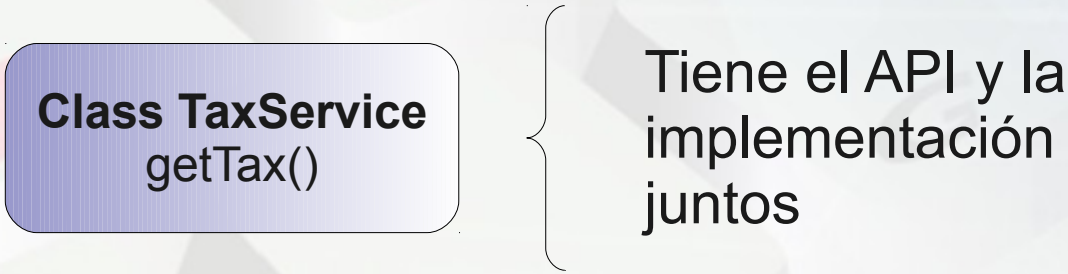


Ventajas al compartir entre desktop y mobile

- Facilidad para un desarrollador que pasa de gvSIG desktop a mobile.
- Se comparte parte de la implementación.
- Desarrollos, al menos en la parte de lógica geo, pueden servir tanto para gvSIG desktop como mobile, con relativamente poco esfuerzo de cara a la compilación.
- Los desarrollos e implementaciones hechas para mobile, que no estén a nivel de interfaz de usuario, generalmente pueden emplearse en desktop.

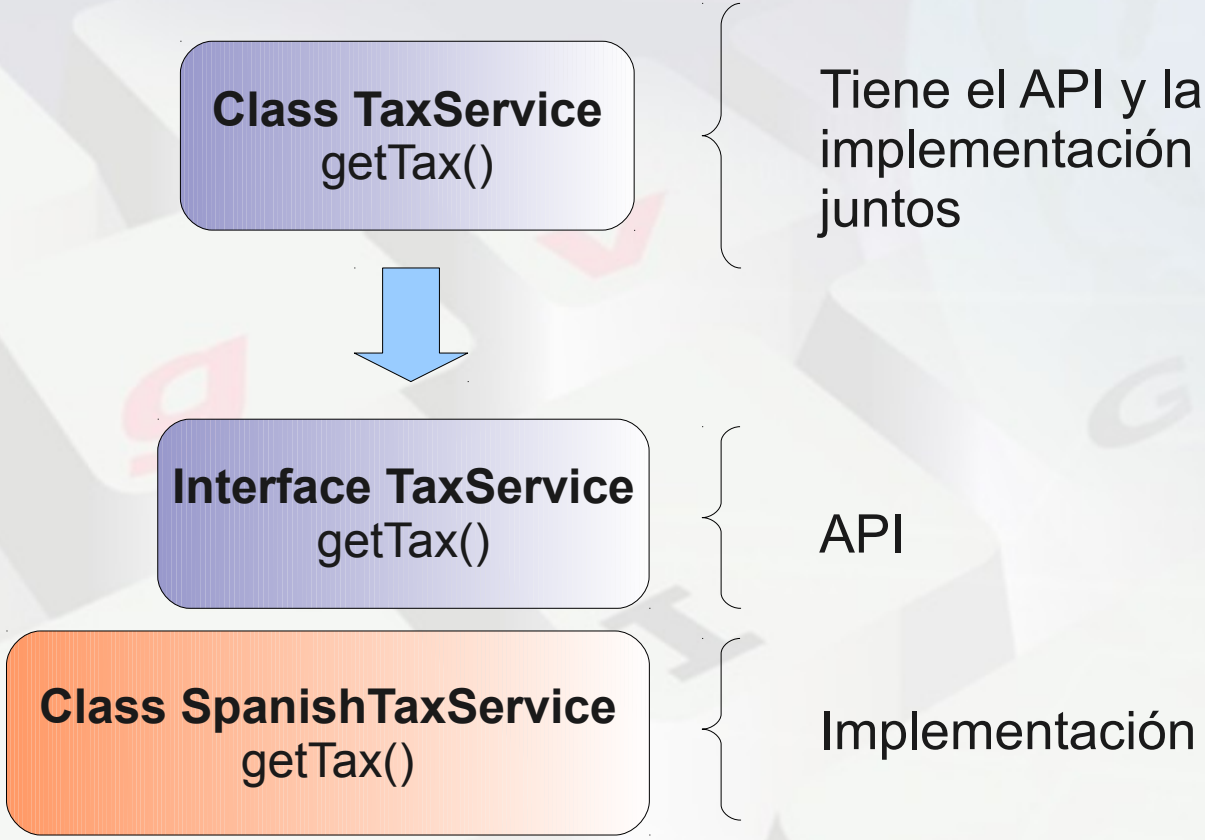
API, SPI e implementación

- API: Interfaz de Programación de Aplicaciones.
- Pongamos un ejemplo de un servicio:



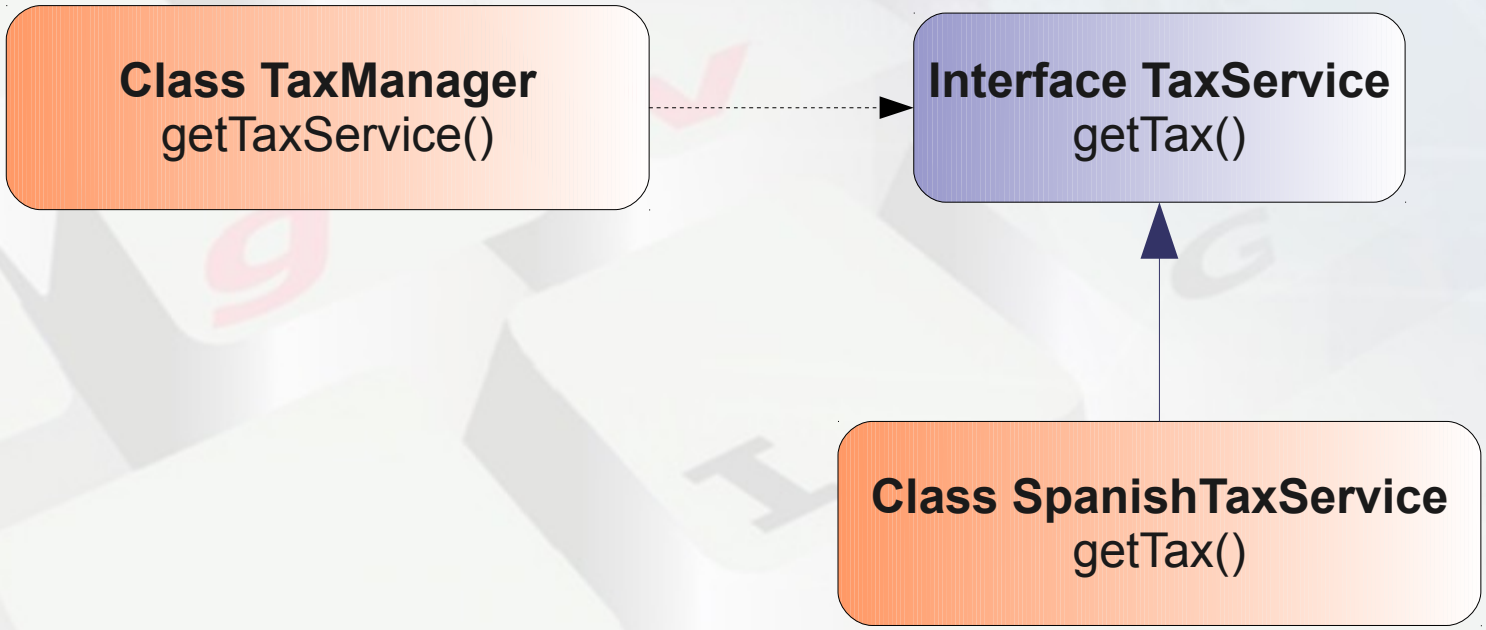
API, SPI e implementación

- Queremos extraer el API:



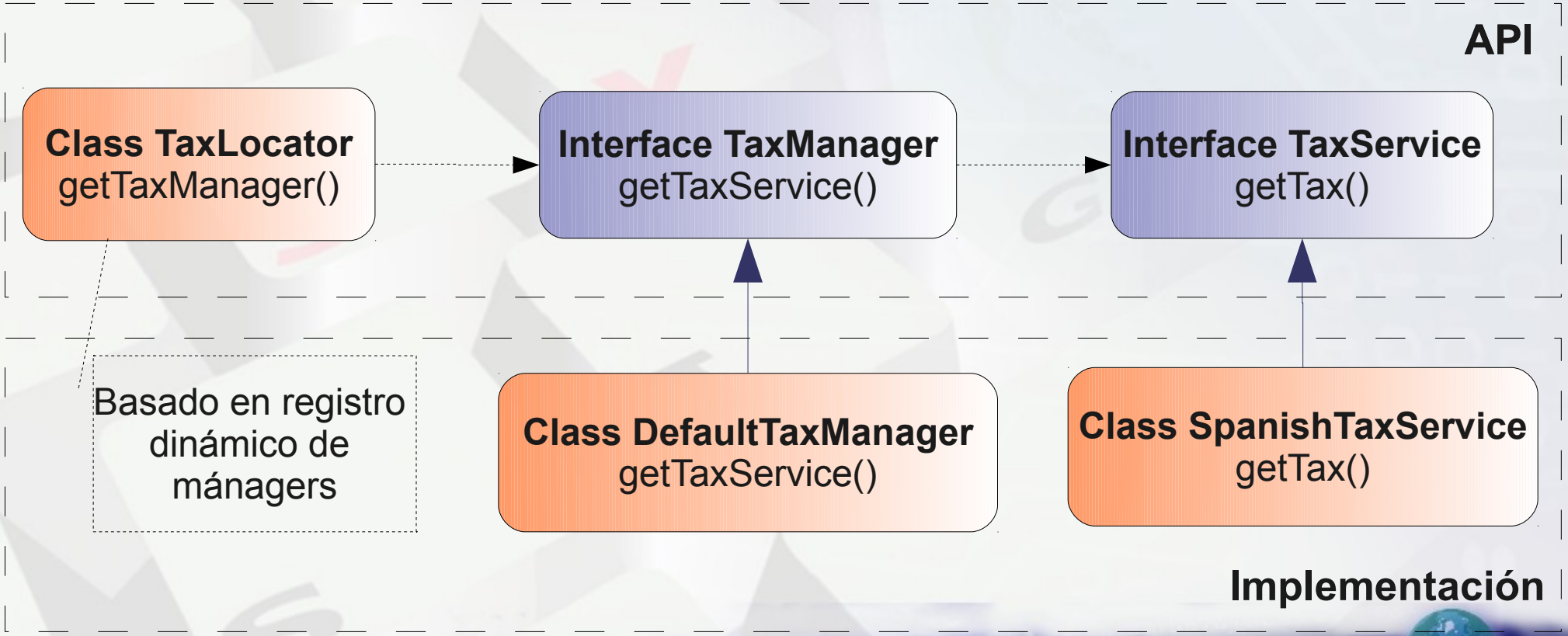
API, SPI e implementación

- ¿Cómo obtenemos una instancia del TaxService?
- Por ejemplo, a través del patrón *Factory*.



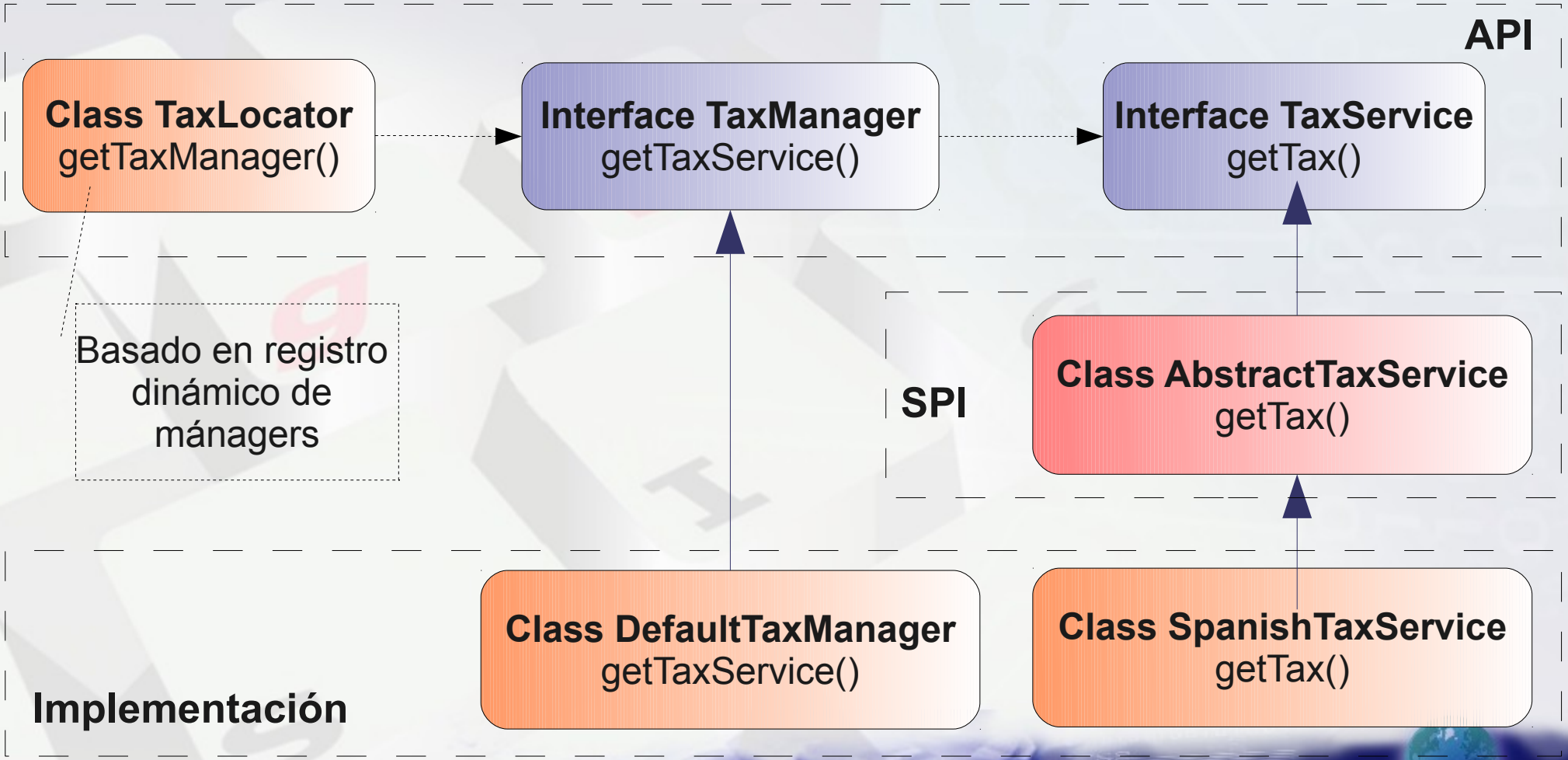
API, SPI e implementación

- Ahora dependemos de la implementación del TaxManager.
- Para evitarlo, empleamos el patrón *Service Locator*



API, SPI e implementación

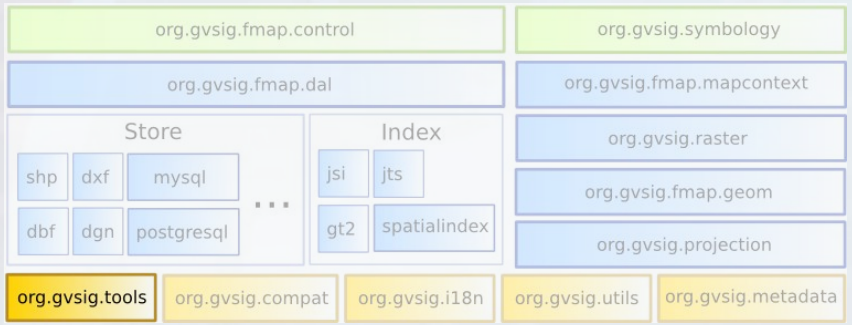
- Si introducimos alguna clase o interfaces para facilitar la implementación de TaxServices, aparece el SPI.



API, SPI e implementación

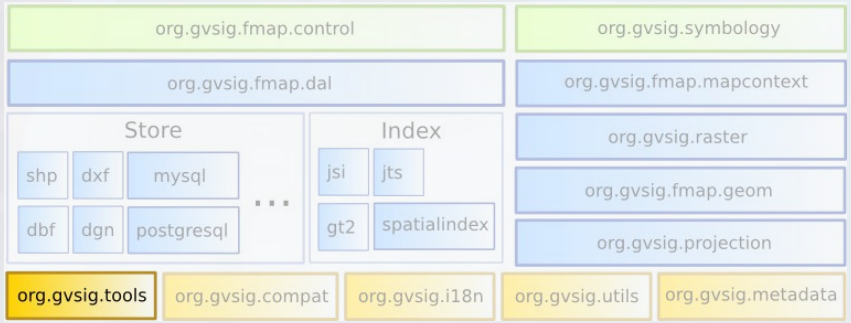
- Resumiendo, tendremos:
  - Un API formado por un juego de interfaces y un service locator.
    Sin dependencias del resto.
  - Una implementación de ese API.
    Deberíamos poder cambiarla sin afecta al API o el SPI.
  - Un posible SPI (Service Provider Interface).
    Depende sólo del API
- Este patrón se está empezando a aplicar en gvSIG.

org.gvsig.tools



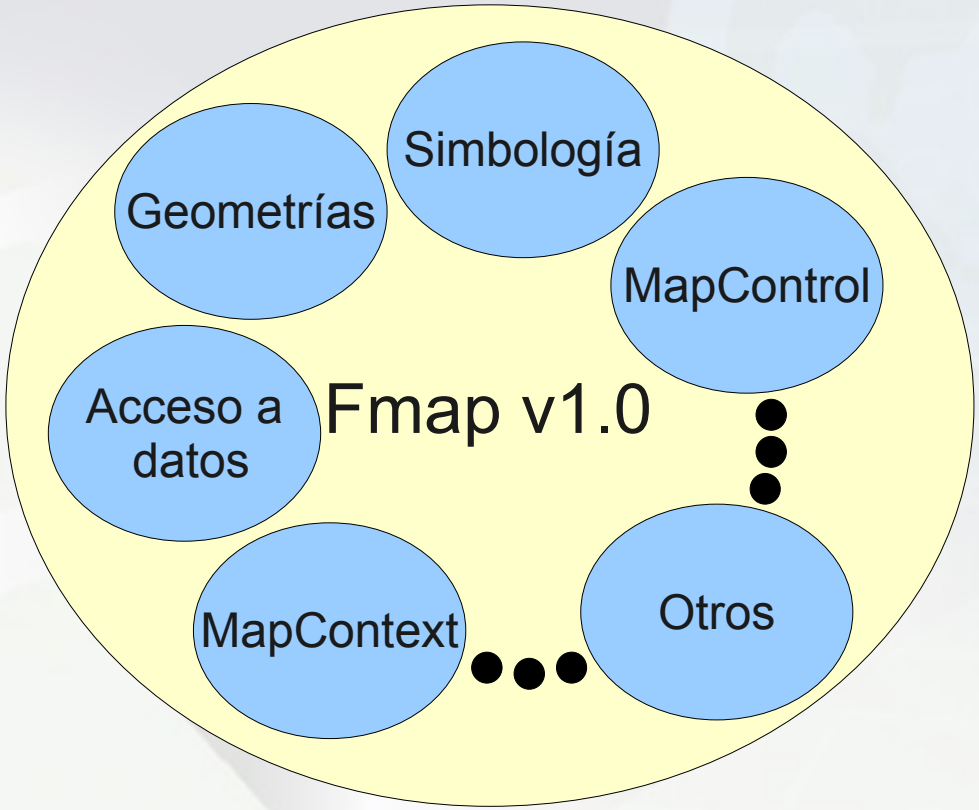
- Contiene dos tipos de herramientas:
  - Utilidades relacionadas con patrones de diseño
    - Locator: implementación del service locator
    - Library: inicialización automática de librerías
    - Service: implementación base del patrón API-SPI-Implementación.
    - Visitor: patrón visitor
    - Observers: patrón observer
  - APIs básicos

org.gvsig.tools

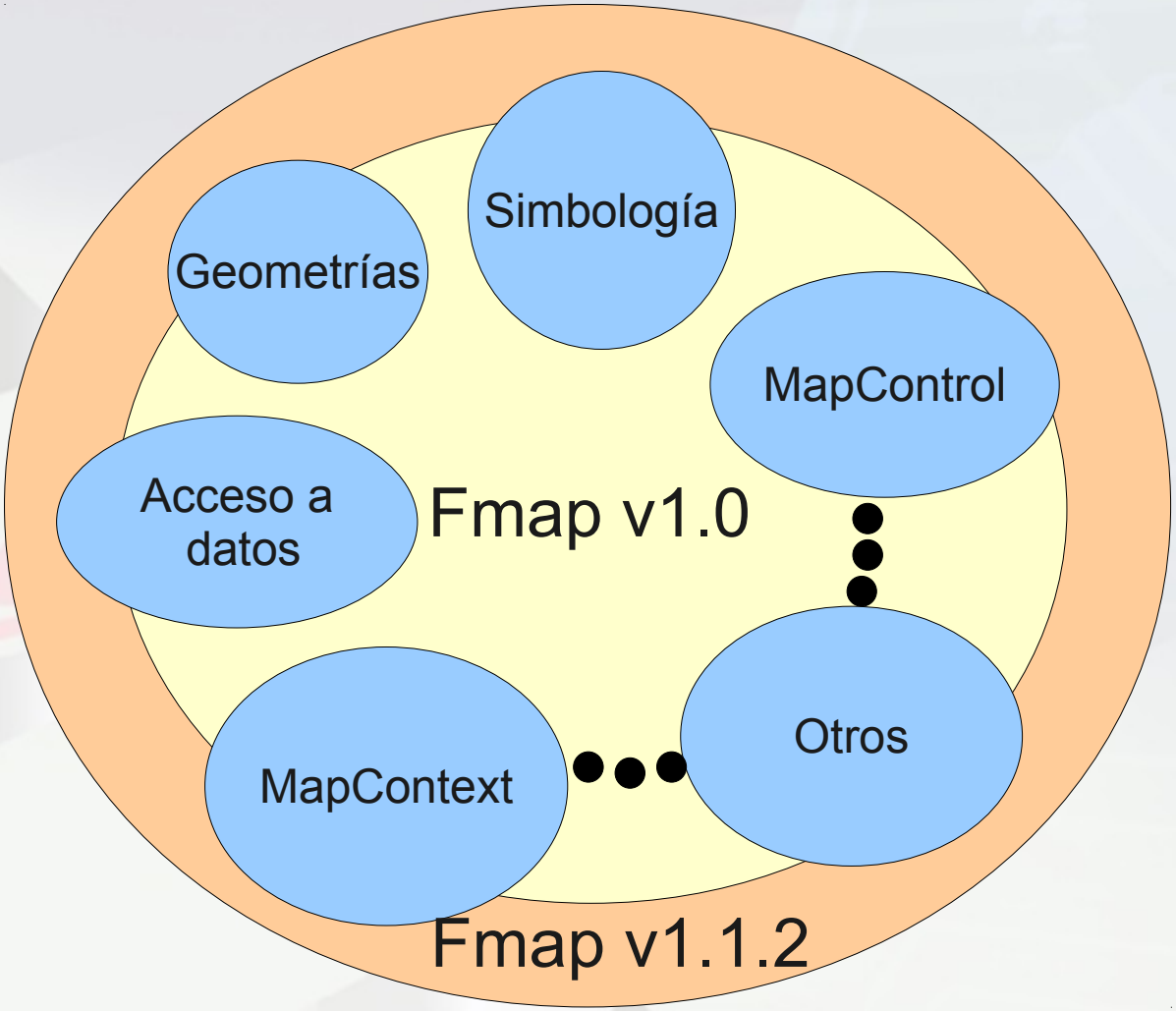


- Contiene dos tipos de herramientas:
  - Utilidades relacionadas con patrones de diseño
  - APIs básicos
    - Disposable: manejar de forma homogénea mecanismos para liberación de recursos.
    - Evaluator: forma homogénea de aportar evaluadores de código .
    - Registro de puntos de extensión.
    - Excepciones: jerarquía base de excepciones con soporte i18n.
    - Persistencia: nuevo mecanismo de persistencia de objetos.

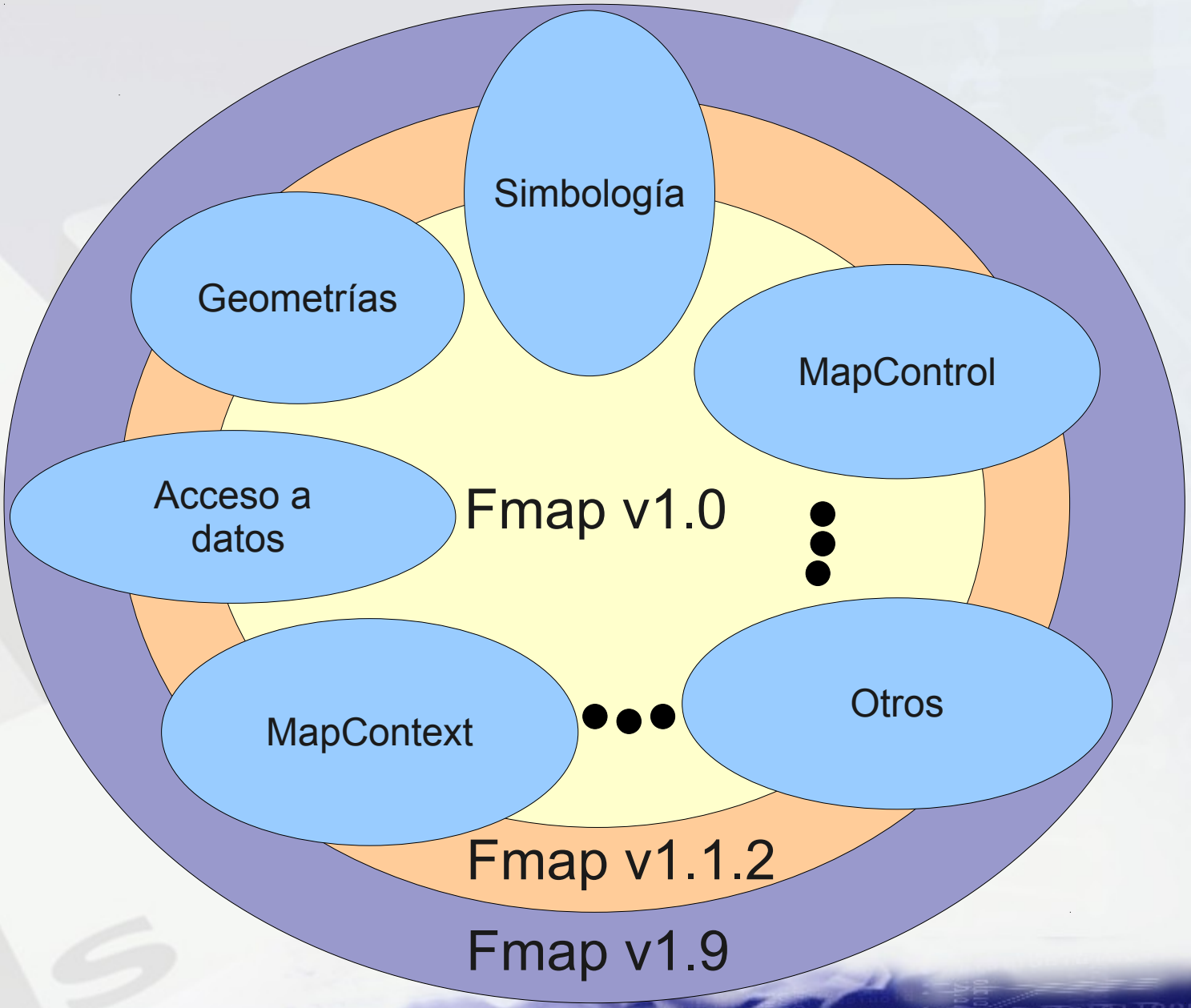
División de FMap en proyectos



División de FMap en proyectos



División de FMap en proyectos

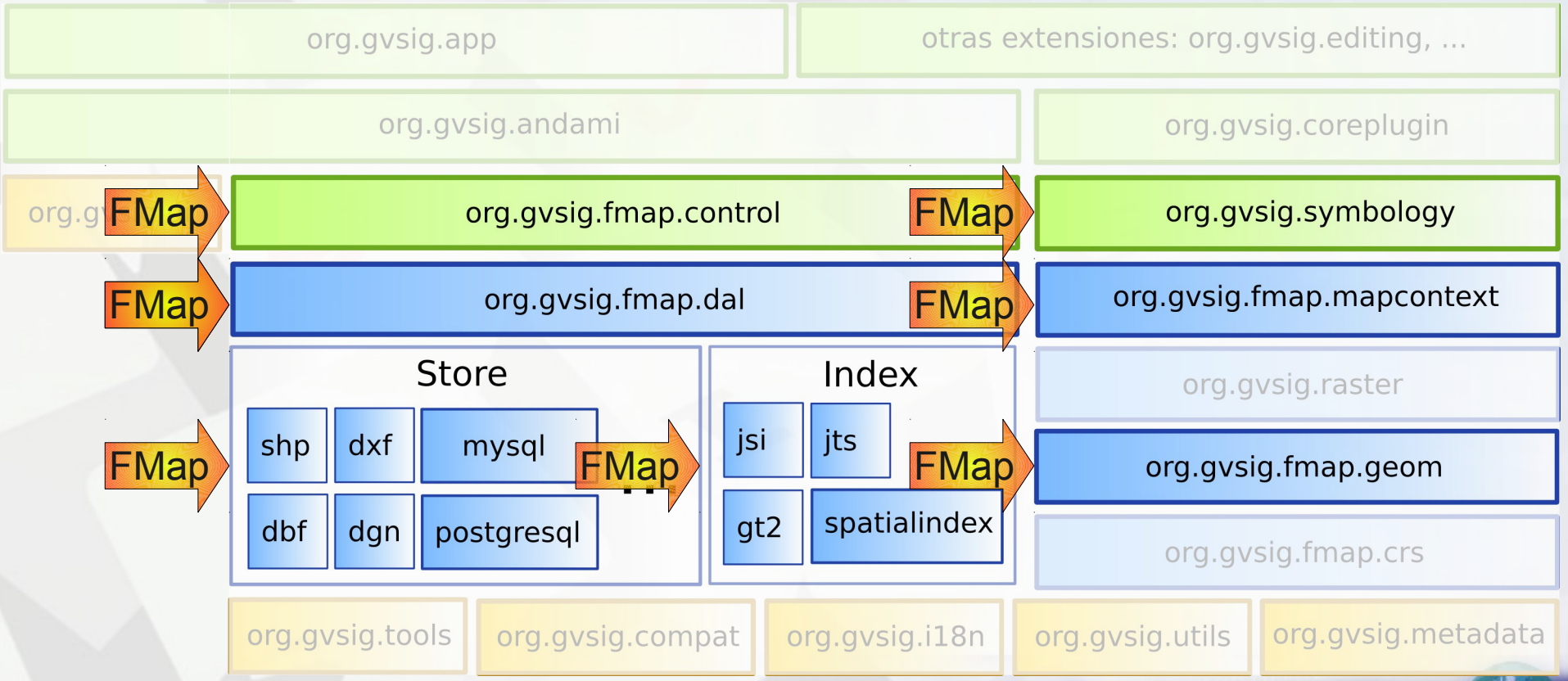


División de FMap en proyectos

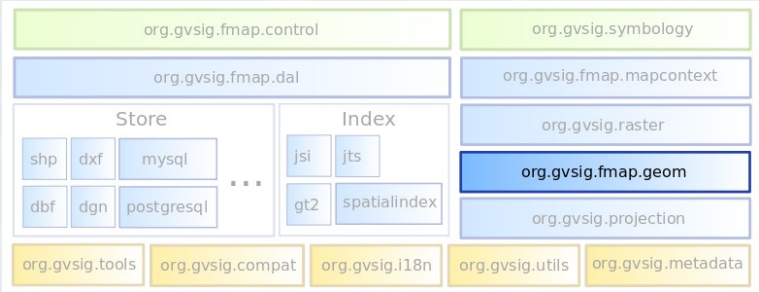
Interfaz de usuario y aplicación

Lógica Geo

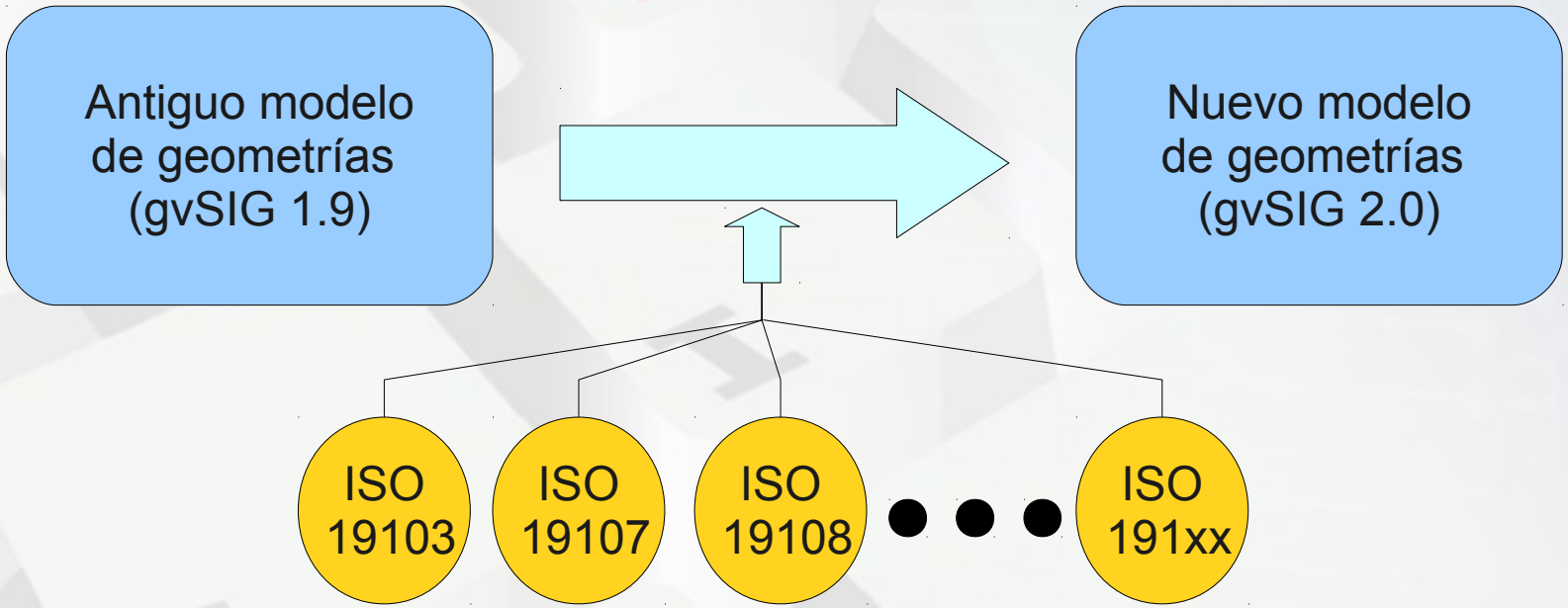
Utilidades y lógica NO Geo



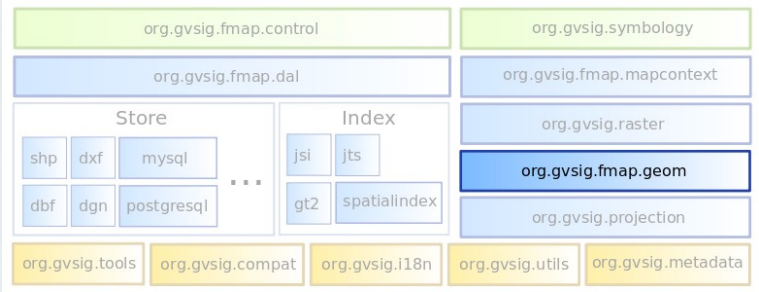
org.gvsig.fmap.geometry



- Creado a partir del viejo modelo de geometrías → minimiza el impacto sobre el código anterior.
- Aproximación al modelo de las ISO 191xx.



org.gvsig.fmap.geometry



- Separación en API e implementación.
- Existencia de un Locator que da acceso a un Manager como punto de entrada al API.

Class Point2D | Class Polygon2D | ... | Class N

Registro de tipos por tipo/subtipo

Point2D | Polygon2D | ... | N

Creación de geometrías por tipo/subtipo

Volumen Point2D | Volumen Polygon2D | ... | Volumen N

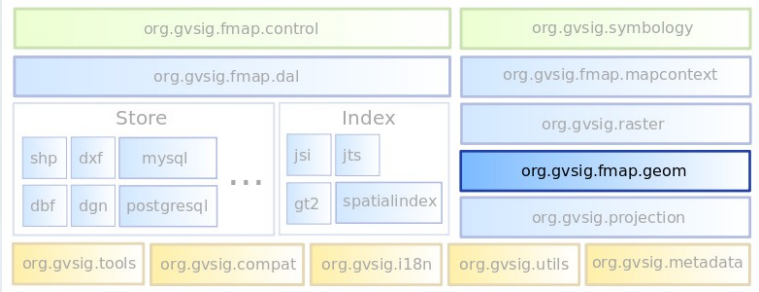
Registro de operaciones asociadas a tipo/subtipo

GeometryManager

Tipos: Punto, Línea Polígono, Círculo, Arco, etc.

Subtipos: 2D, 3D, 2DM, etc

org.gvsig.fmap.geometry



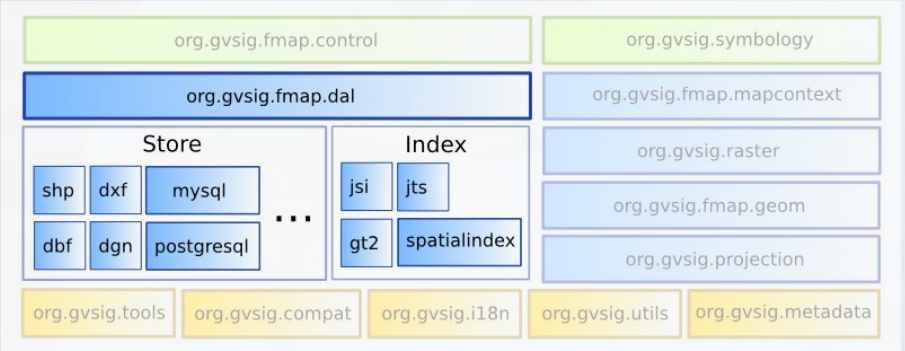
Documentación disponible del curso de desarrollo sobre gvSIG desktop 2.0 (julio 2009):

- Documentación de desarrollo:
  http://www.gvsig.org/web/docdev/gvsig_desktop_2_0/org-gvsig-fmap.geom/
- Presentación:
  http://gvsig-desktop.forge.osor.eu/downloads/pub/documents/learning/gvsig-courses/gvsig_des_2.x_d/geometrias.pdf
- Documentación adicional del curso:
  http://www.gvsig.org/web/docusr/learning/gvsig-courses/gvsig_des_2.x_d/pub/doc-entregar/nueva-api-de-geometrias
- Video de la presentación:
  http://gvsig-desktop.forge.osor.eu/downloads/pub/documents/learning/gvsig-courses/gvsig_des_2.x_d/videos/geom.mp4

Libreria de acceso a datos (DAL)

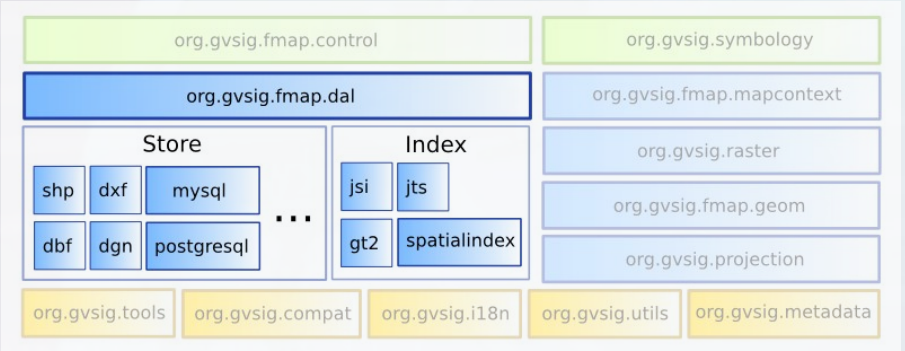
- Objetivos:

  Disponer de un API



- Con acceso a datos alfanuméricos y geometrías
- Definido y documentado
- Independiente de la implementación
- Independiente de la fuente de datos
- Predecible en entornos multi-hilo
- Usable en entornos JavaME

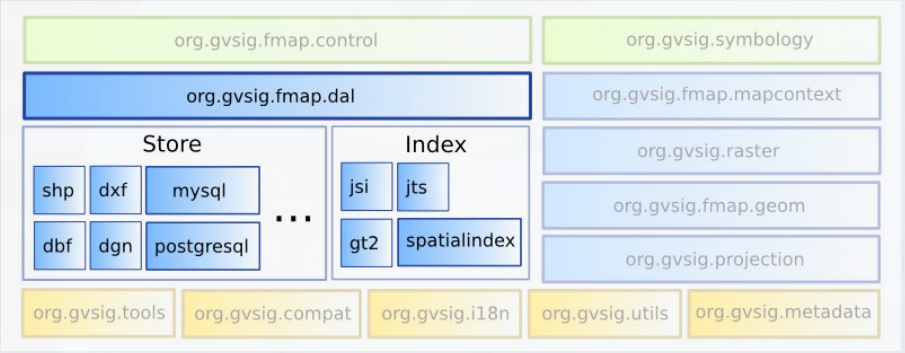
Libreria de acceso a datos (DAL)



Estado:

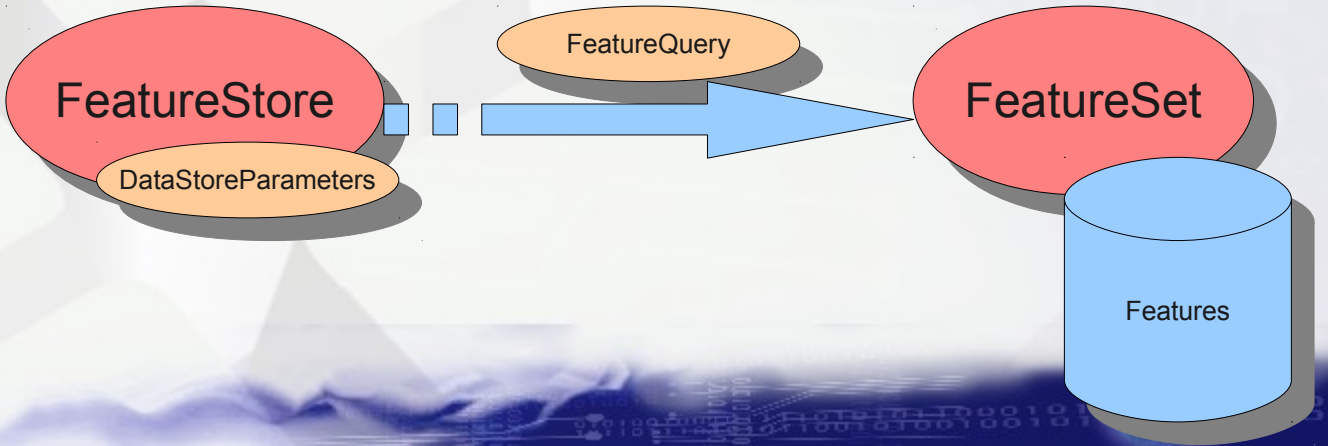
- Datos vectoriales.
  Soporte de acceso de lectura y escritura para los formatos mas comunes.
- Coverturas raster.
  Se esta trabajando en ello (2.1 como pronto).

Libreria de acceso a datos (DAL)

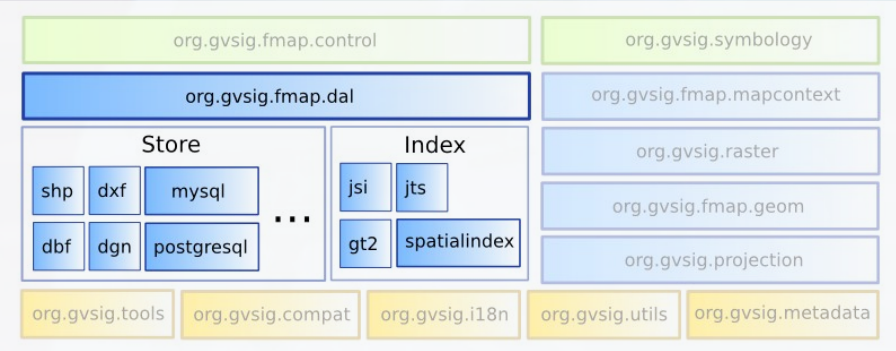


Piezas clave en DAL:

- FeatureStore y DataStoreParameters
  Representa a un almacen de features, una tabla, shape, dxf, ...
- FeatureSet
  Representa un conjunto de feature de un almacen.
- FeatureQuery
  Especifica una consulta, filtro u ordenacion a realizar sobre las features de un almacen.
- Feature
  Representa una feature dentro del almacen.



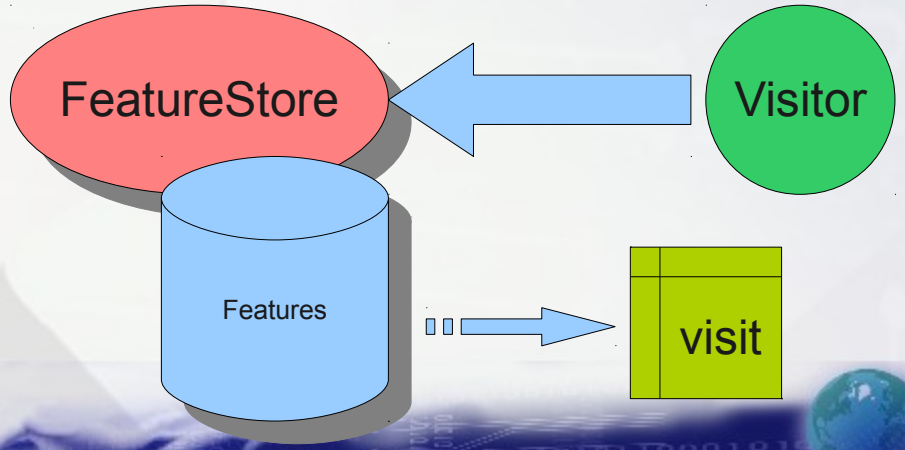
Libreria de acceso a datos (DAL)



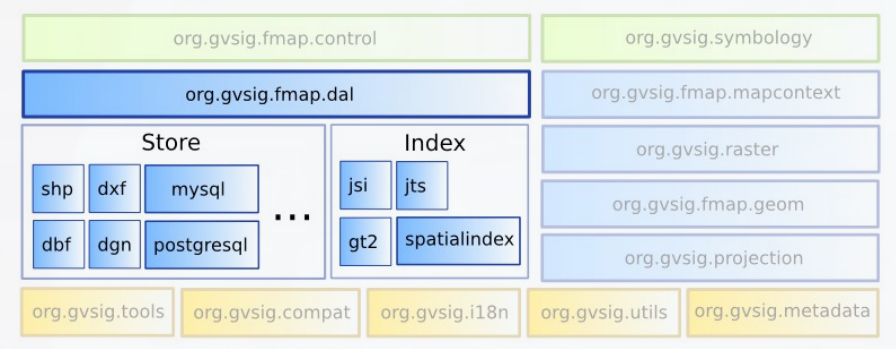
- Piezas clave en DAL
  Un ejemplo:

```
manager = DALLocator.getDataManager();
params = manager.createStoreParameters("Shape");
params.setDynValue("shpfilename","data/prueba.shp");
store = (FeatureStore)manager.createStore(params);

store.accept( new Visitor() {
    public void visit(Object obj) {
      Feature feature = (Feature)obj;
      System.out.println(feature.getString("NOMBRE"));
    }
  }
);
store.dispose();
```



Libreria de acceso a datos (DAL)



- Donde conseguir mas información sobre DAL
  - En la web www.gvsig.org

    http://www.gvsig.org/web/docdev/gvsig_desktop_2_0/org.gvsig.fmap.dal
  - En el sitio web maven del proyecto de DAL estan disponibles los javadocs

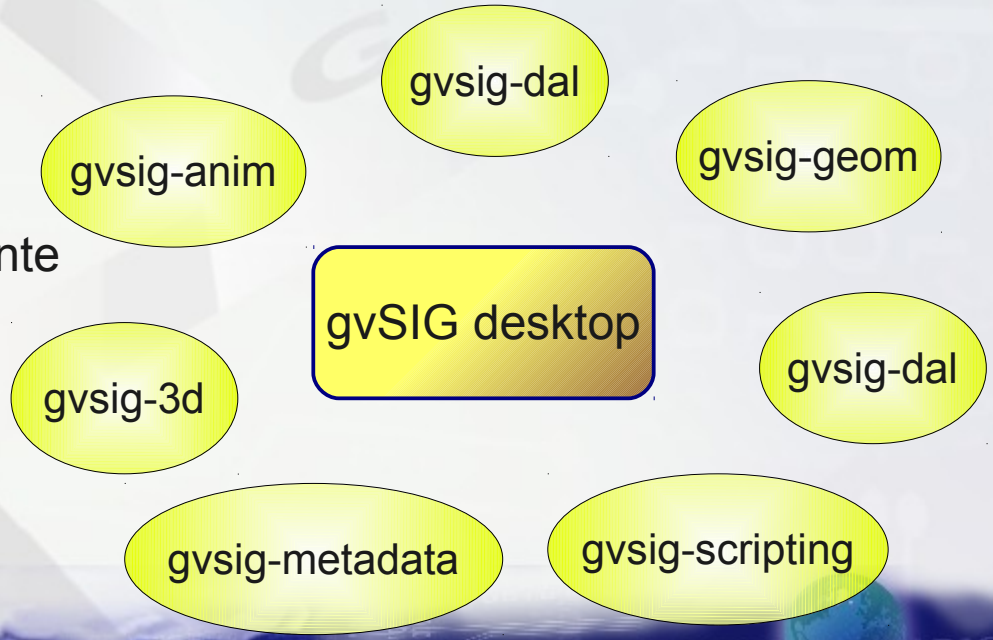
    http://gvsig-desktop.forge.osor.eu/downloads/pub/projects/gvSIG-desktop/docs/reference/org.gvsig.fmap.dal/2.0.0
  - En la documentacion del curso sobre desarrollo de la 2.0 impartido en julio.

    http://www.gvsig.org/web/docusr/learning/gvsig-courses/gvsig_des_2.x

Nuevos proyectos www.osor.eu

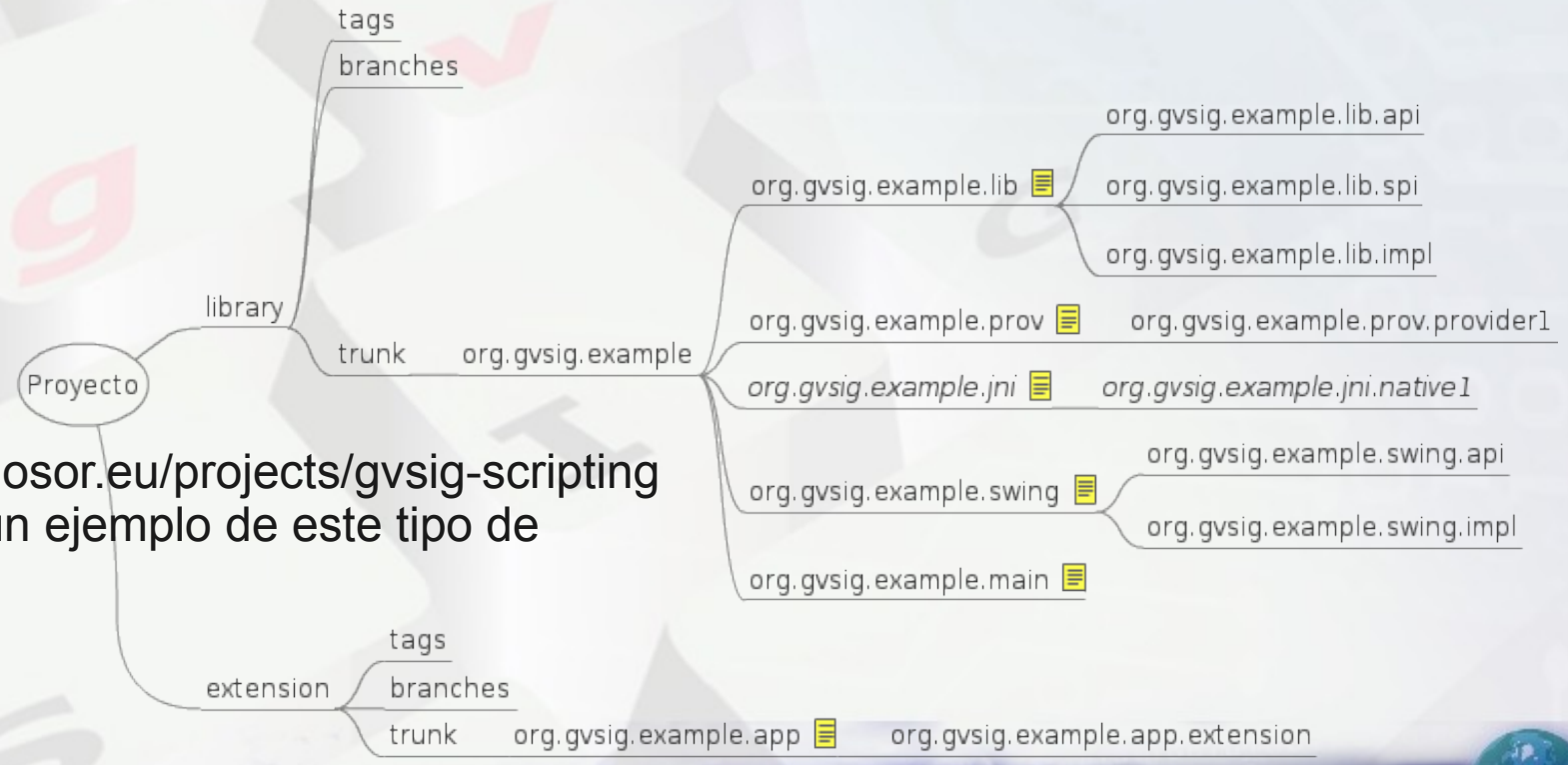
- gvSIG desktop, un proyecto en la forja de OSOR.
- Reducir el número de extensiones que forma gvSIG desktop.
- Cada proyecto o extensión como proyectos independientes en OSOR.

- Repositorio de fuentes propio e independiente
- Gestión de tickets por proyecto
- Ciclo de vida independiente de gvSIG.



Nuevos proyectos
Estructura de un proyecto

- Los nuevos proyectos utilizarán la estructura de maven orientada a multimódulo.
- Se separara el API de la implementación, tanto en la parte de lógica como en el interfaz de usuario.

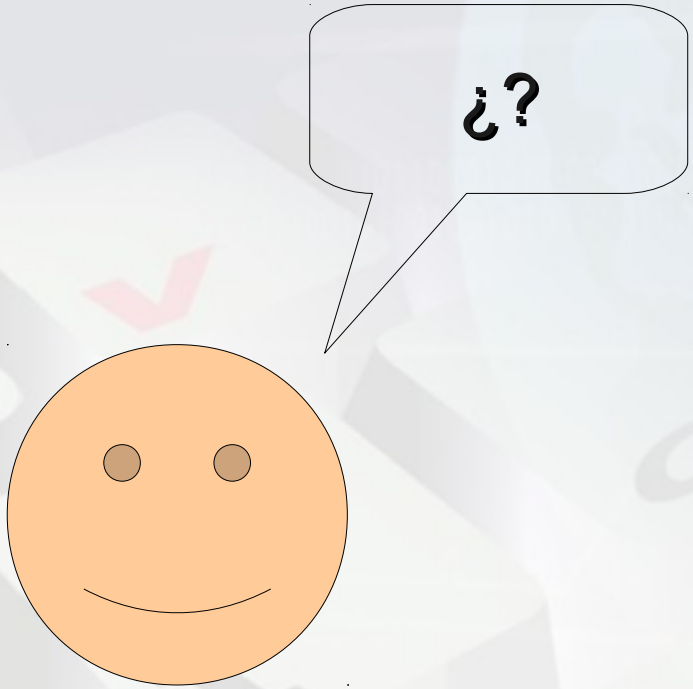


En http://forge.osor.eu/projects/gvsig-scripting podemos ver un ejemplo de este tipo de arquitectura.

Otras cosas que han cambiado

- Migración al API de logging de SLF4J
- Librería de compatibilidad Java SE – Java ME CDC
- Separación entre API e implementación en:
  - proyectos
  - documentos
- Primeros pasos de extracción del API de simbología
- Nueva implementación del documento tabla

¿Dudas?



gvSIG. Geographic Information System of the Valencian Government