

EXTENSIBILIDAD EXTREMA

La aplicación base es mínima; toda la funcionalidad geoespacial se inyecta dinámicamente como módulos independientes.

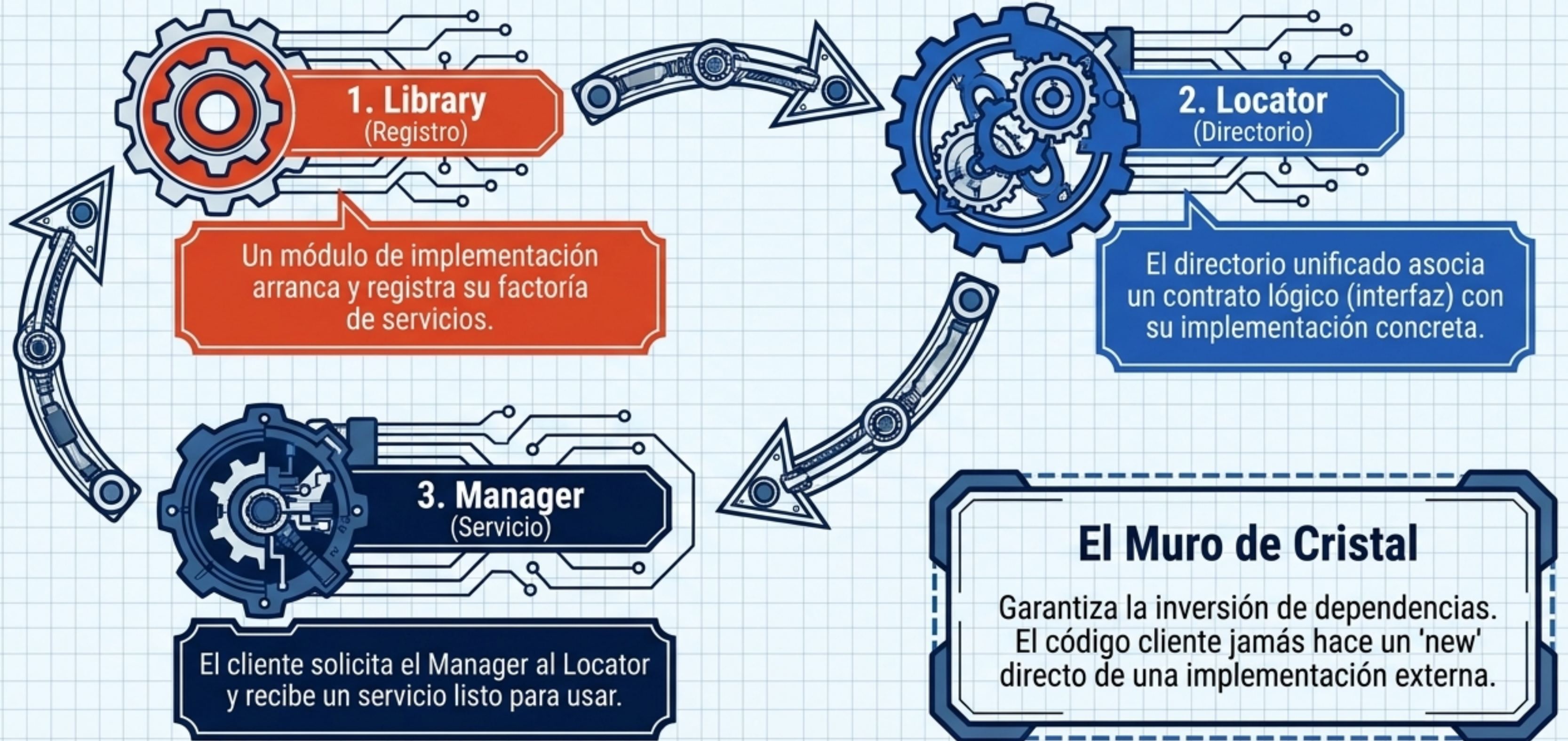
ORQUESTACIÓN ANDAMI

El framework central que descubre, carga y gestiona el ciclo de vida de cada extensión durante el arranque del sistema.

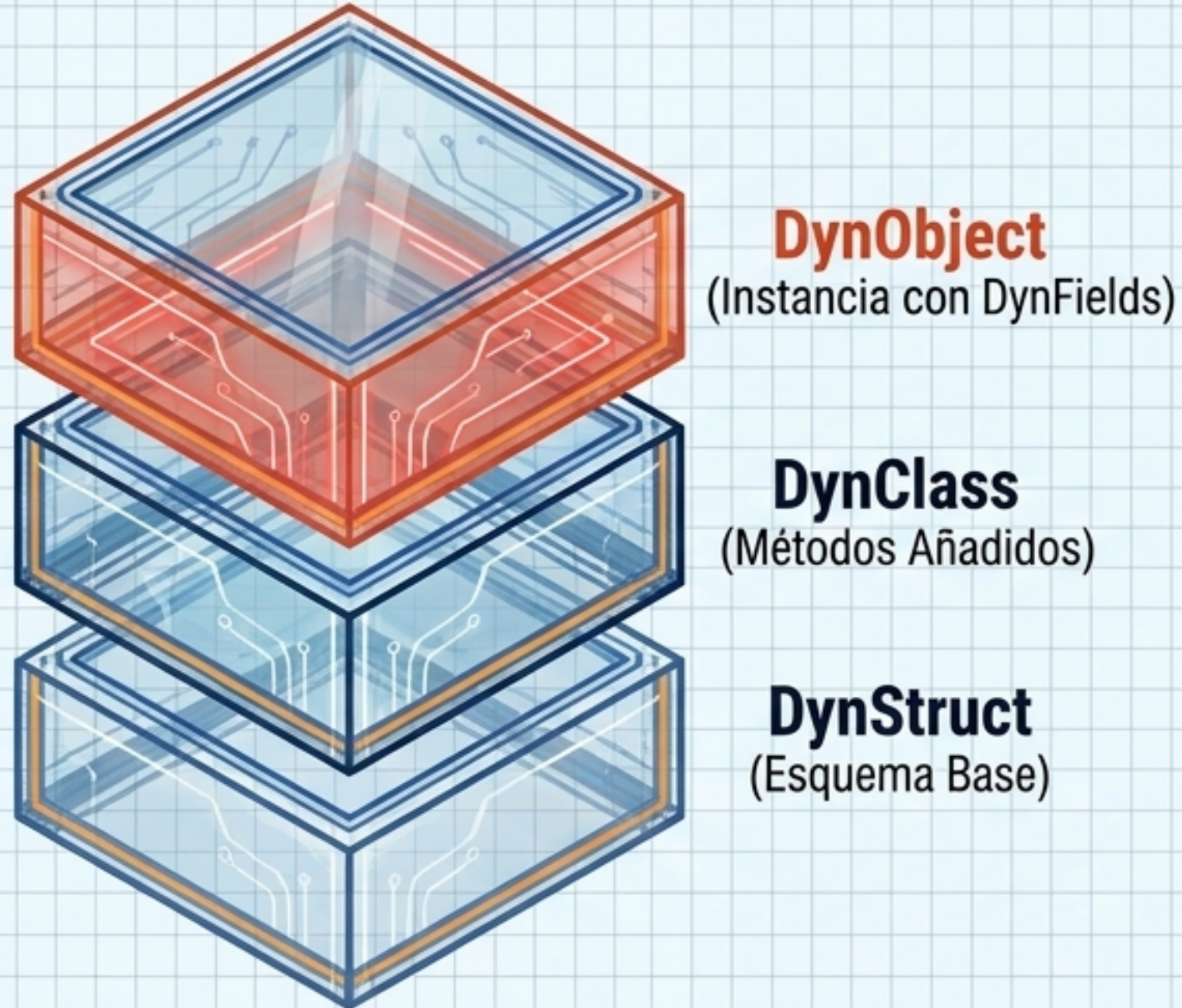
EL MANIFIESTO (CONFIG.XML)

El contrato XML declarativo que define menús, barras de herramientas y clases aportadas, sin necesidad de instanciación prematura.

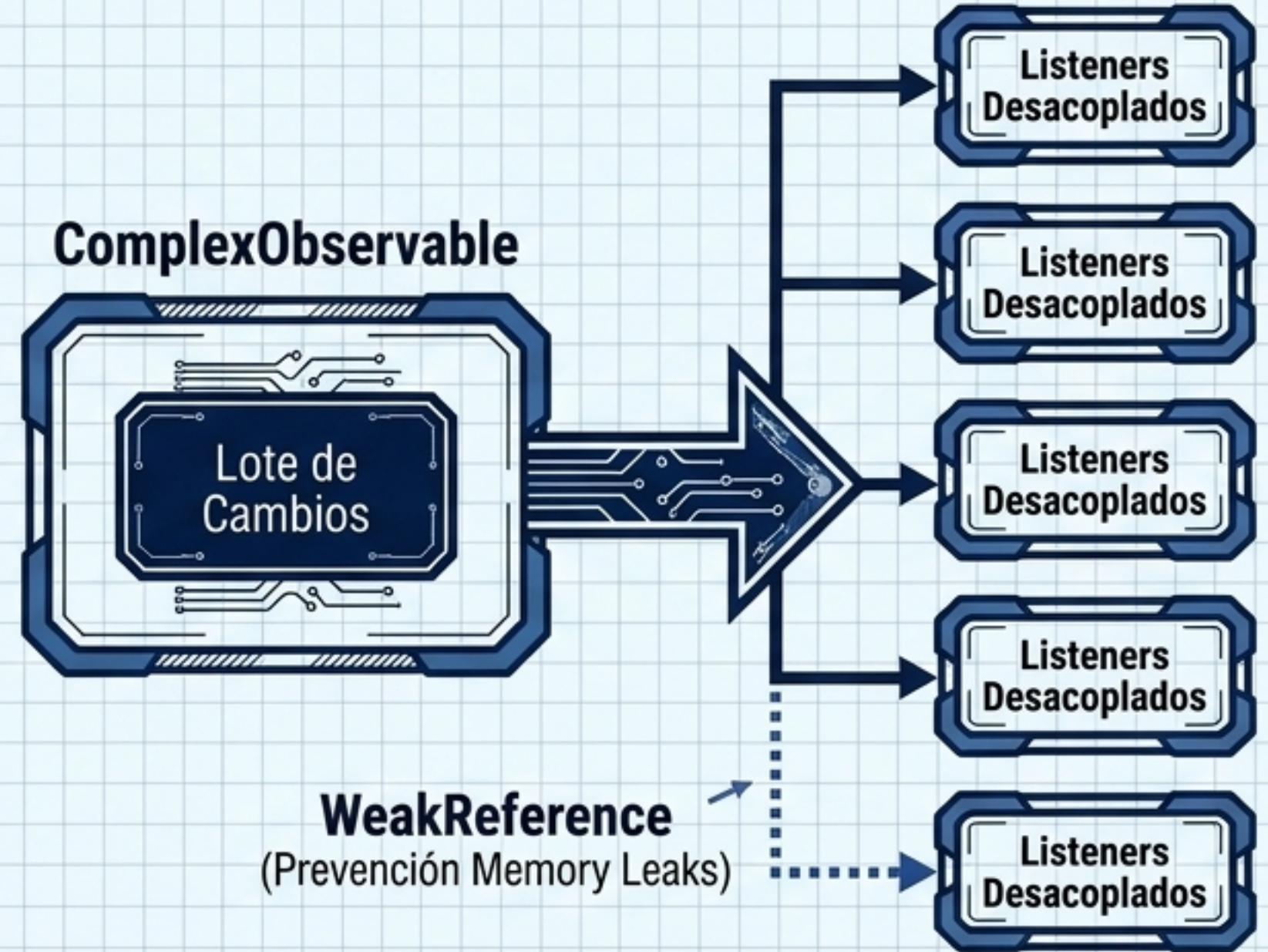
II. Inversión de Dependencias (org.gvsig.tools)



II. Flexibilidad Dinámica: DynObjects y Observadores

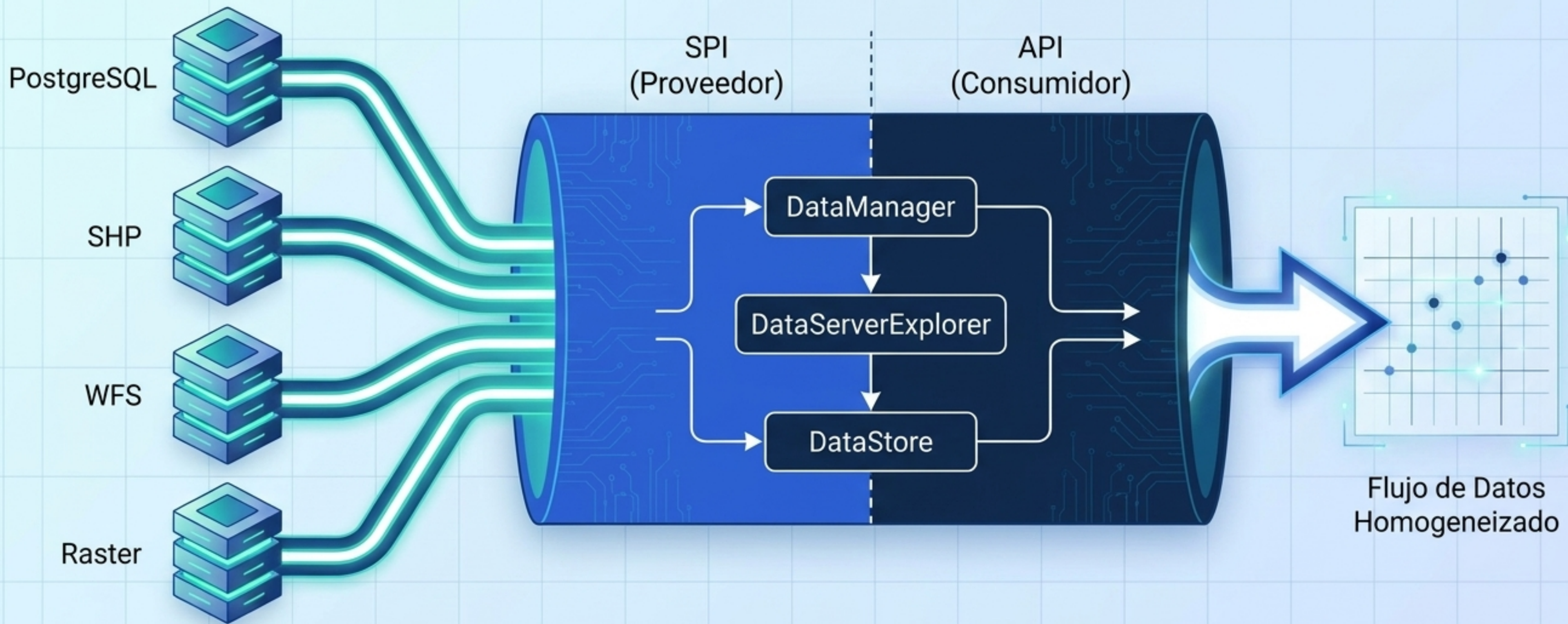


El motor que permite modelar dinámicamente cualquier tabla o capa vectorial cuyo esquema de datos se desconoce durante la compilación.

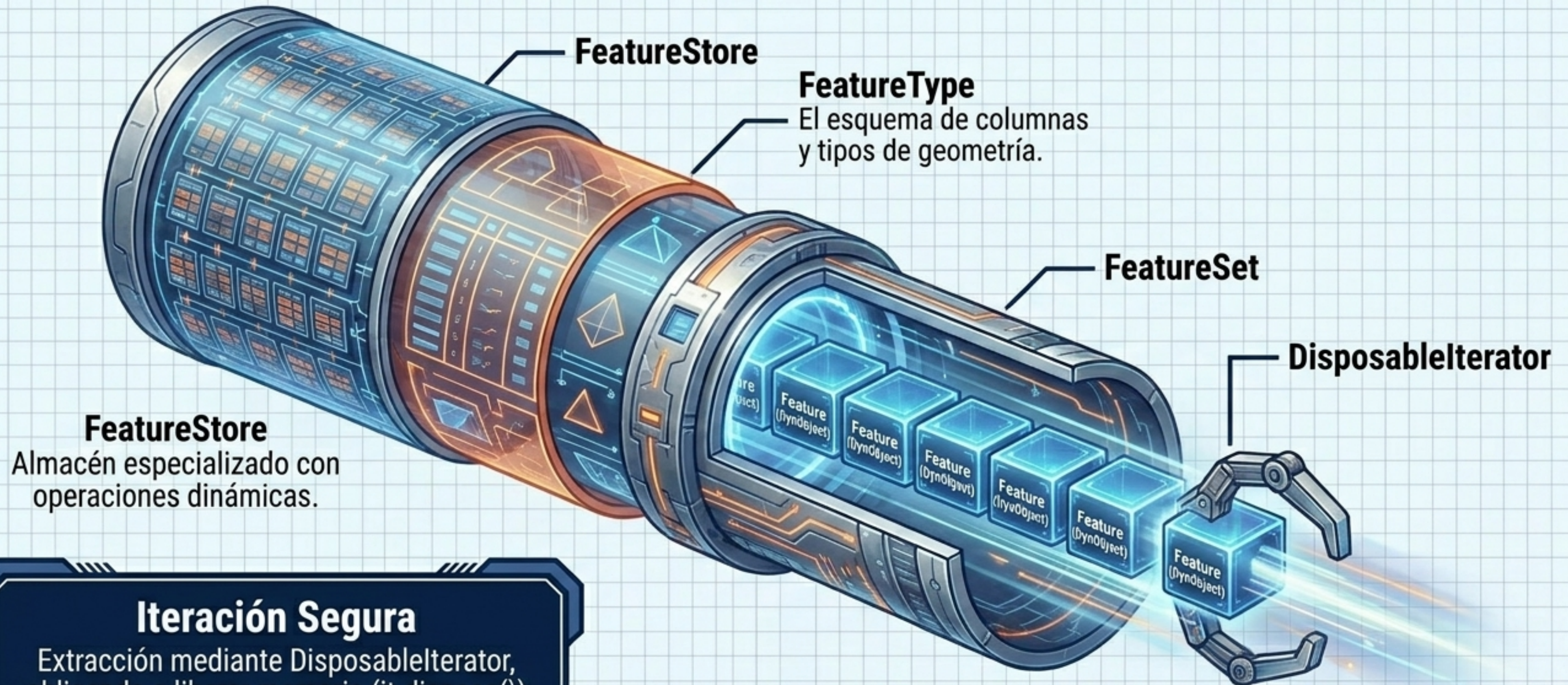


Agrupación estructurada de notificaciones masivas para optimizar el rendimiento de la interfaz y evitar fugas de memoria.

III. Capa de Acceso a Datos (DAL): Arquitectura Universal



III. Profundidad DAL: Procesamiento Vectorial



FeatureStore
Almacén especializado con operaciones dinámicas.

Iteración Segura
Extracción mediante DisposableIterator, obligando a liberar memoria (`it.dispose()`) tras el recorrido de grandes volúmenes.

IV. La Vista de Mapa: El Paradigma MVC

Vista / Controlador (MapControl)

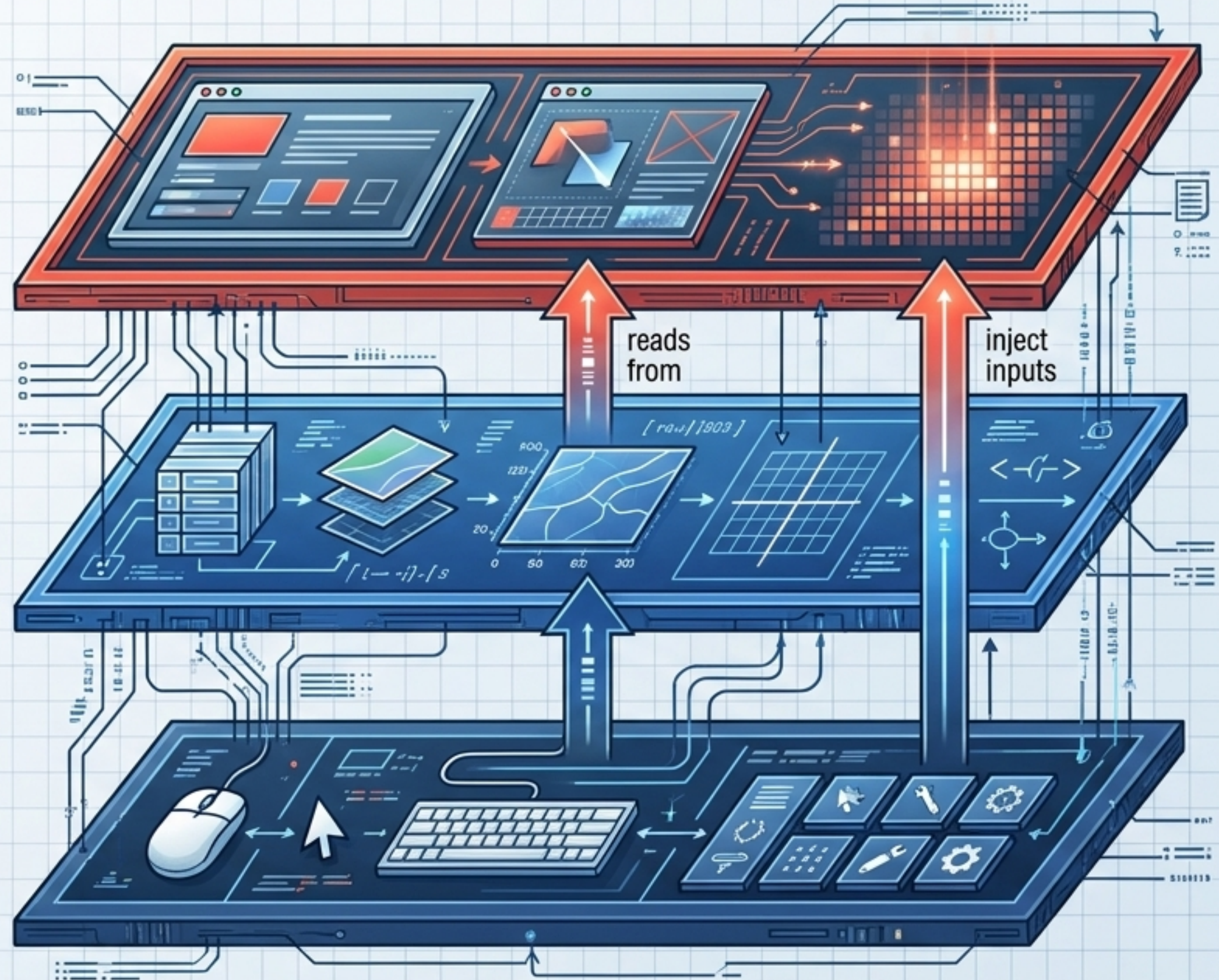
El lienzo visual (JComponent) que interactúa con Swing y renderiza los píxeles.

Modelo (MapContext)

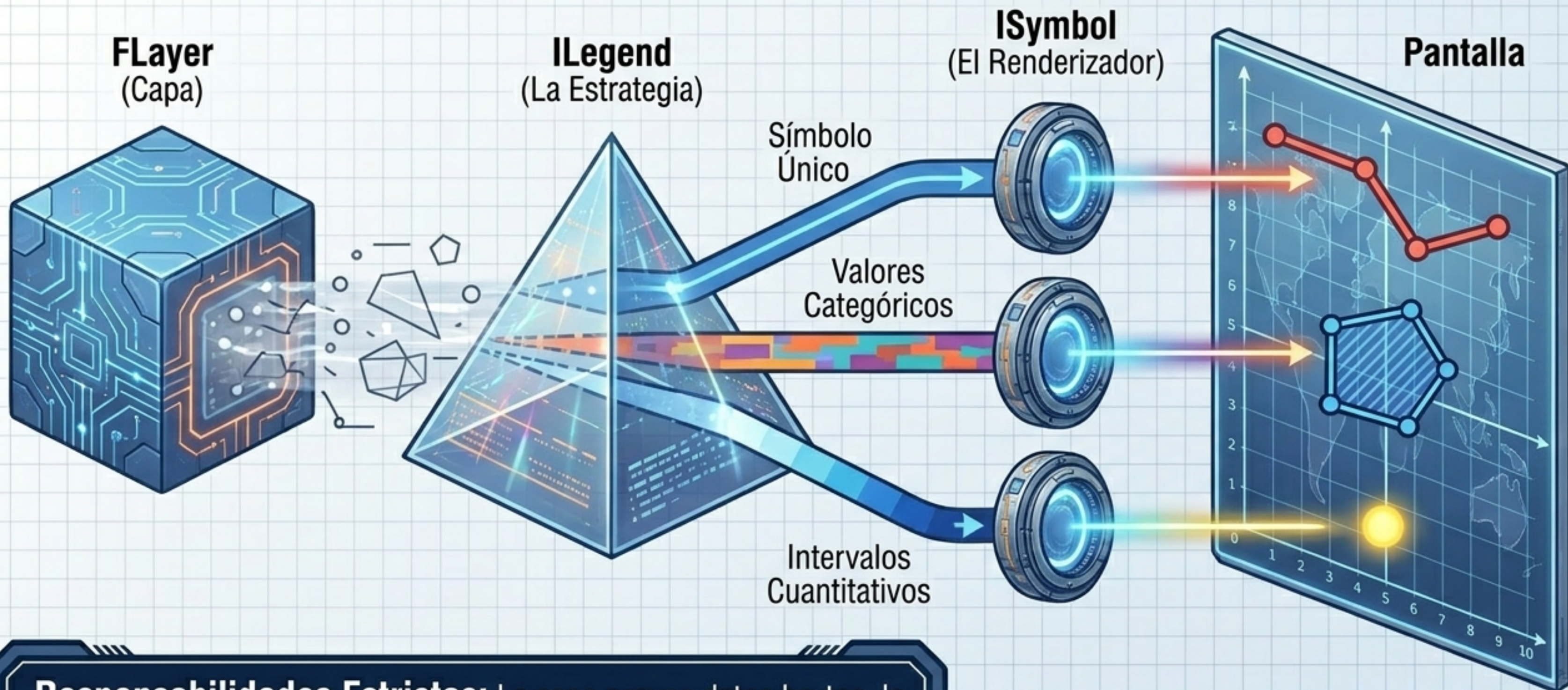
El contenedor del estado lógico. Almacena las capas geográficas (FLayer), la extensión espacial y la proyección actual. No dibuja nada físicamente.

Interactividad (Behaviors & ToolListeners)

Inyección de lógica mediante delegación. Separa los eventos del ratón del dibujo, permitiendo nuevas herramientas sin alterar el núcleo.



IV. Simbología: De los Datos al Píxel



Responsabilidades Estrictas: La capa provee datos brutos, la leyenda decide la clasificación, y el símbolo ejecuta el trazado.

V. Arquitectura GUI: El Muro de Cristal

Proyecto API (.api)

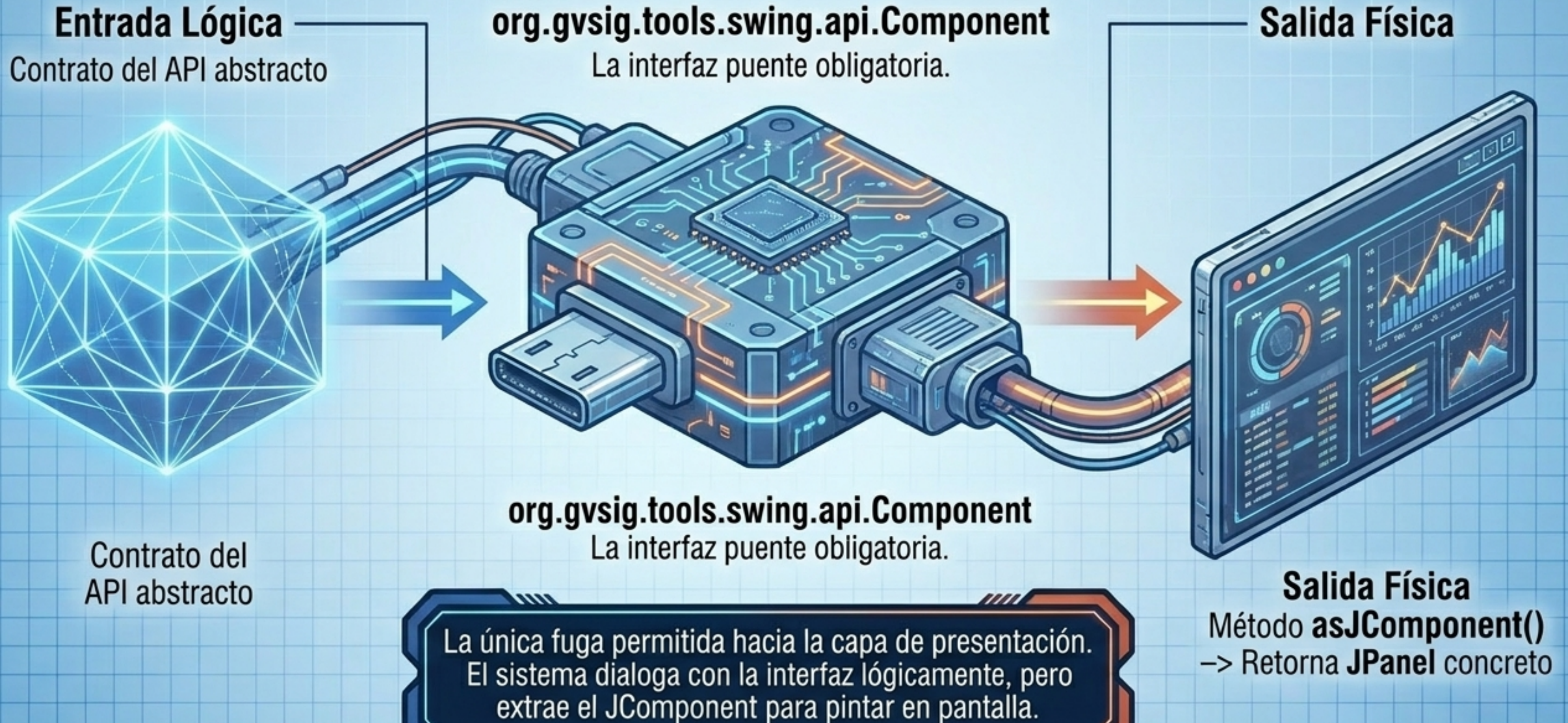
- Contiene exclusivamente interfaces Java puras.
- Define el contrato lógico (ej. `getPanelData()`).
- Totalmente agnóstico: prohibido incluir librerías gráficas, Swing o lógica pesada.



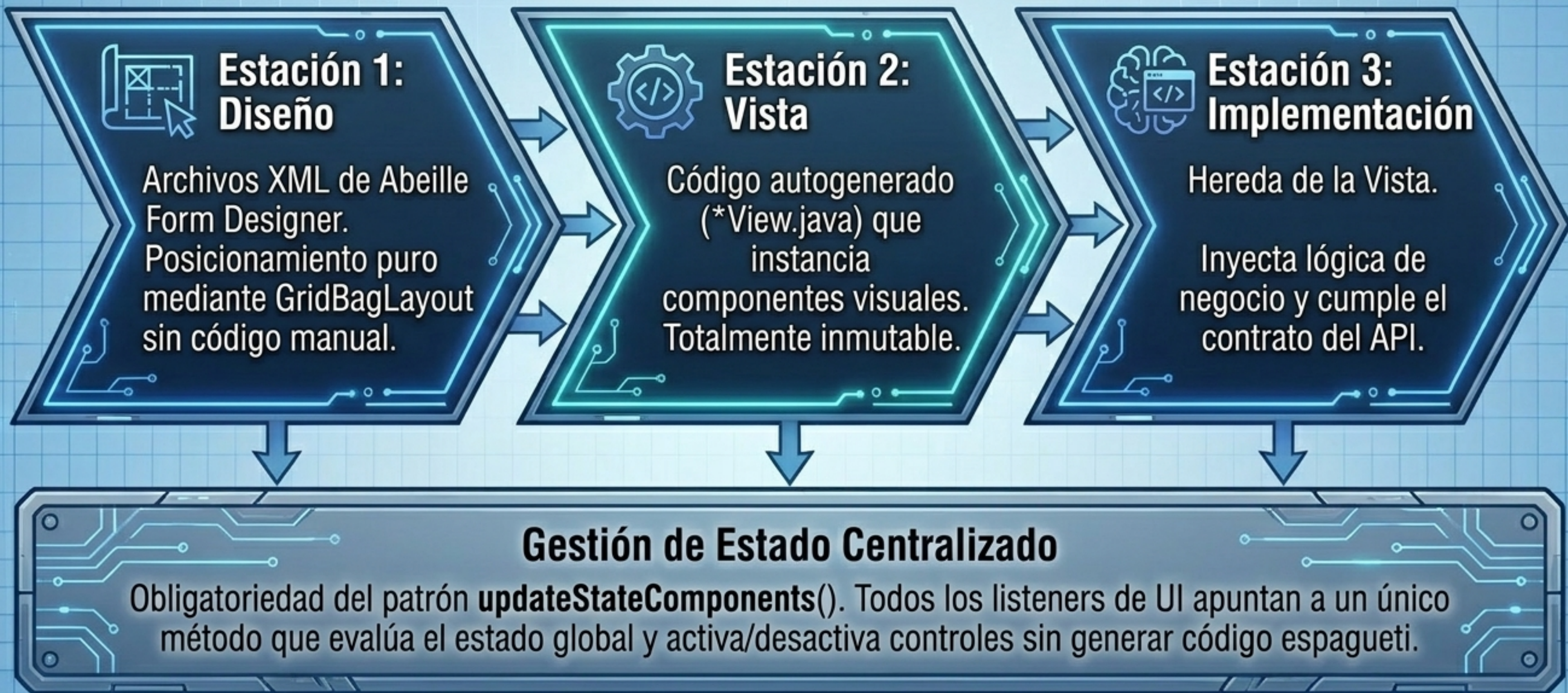
Proyecto Implementación (.impl)

- Contiene clases concretas (`*Impl.java`).
- Maneja eventos de usuario, hilos de fondo y estado técnico de Swing.
- Código físicamente oculto e inaccesible para el resto de los módulos.

V. El Puente Tecnológico (Component)



V. La Construcción: La Tripletta de UI



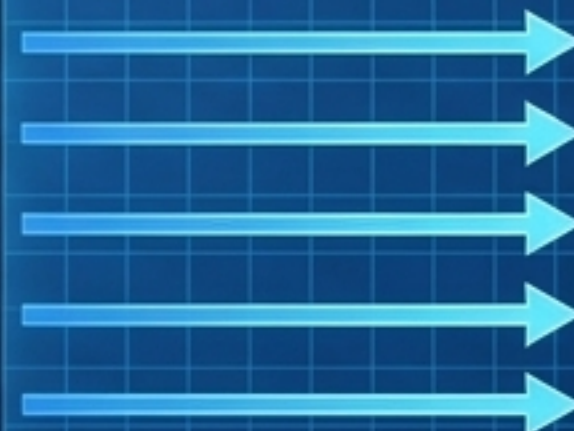
VI. Servicios Swing: Ventanas y Seguridad

WindowManager



Desacoplamiento absoluto entre el componente visual puro y el contenedor de ventana del sistema.

ThreadSafeDialogsManager



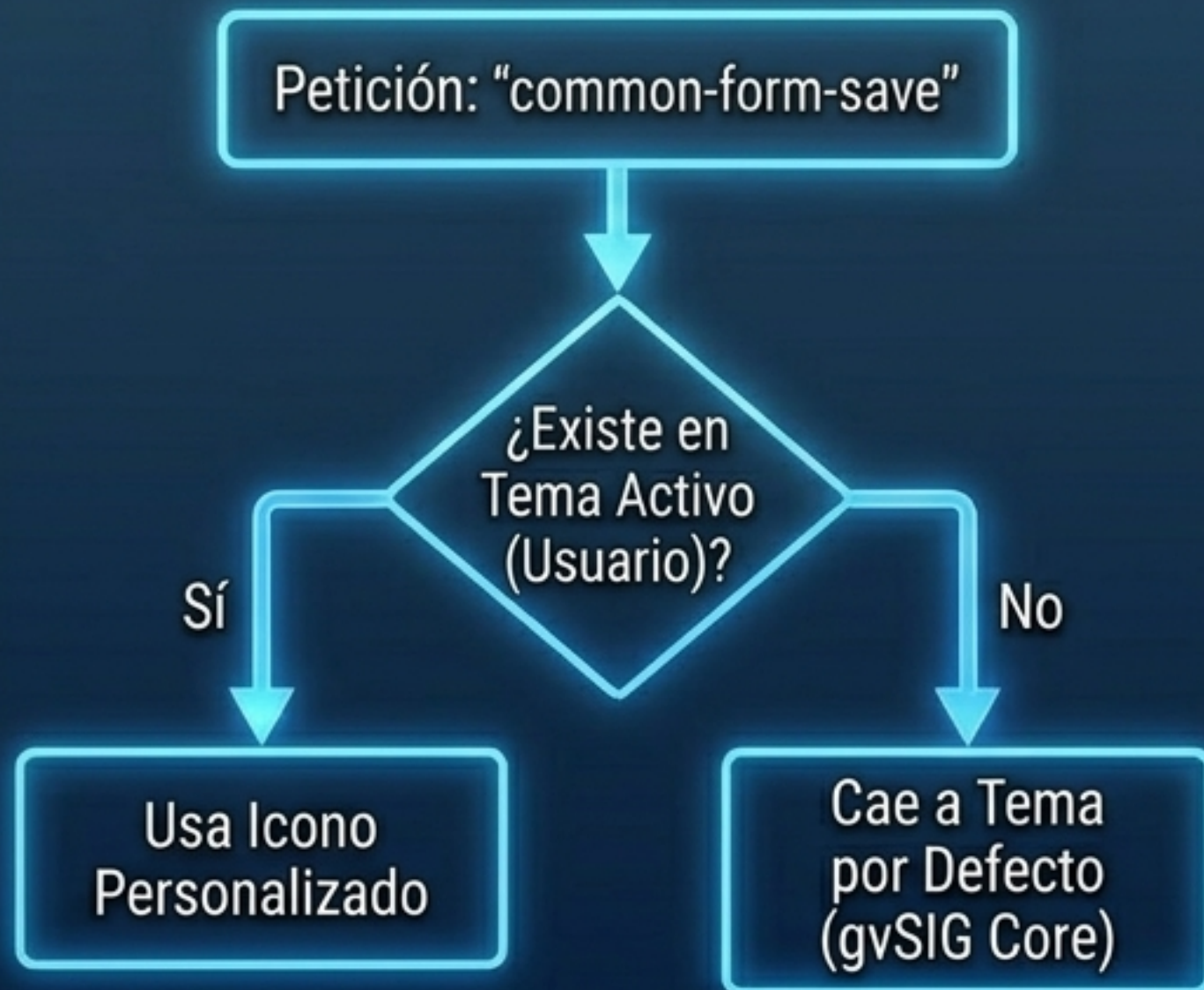
Event Dispatch
Thread (EDT)

ThreadSafeDialogsManager

Evita el bloqueo de la UI inyectando operaciones y diálogos de forma segura en el hilo principal gráfico.

VI. Servicios Swing: Iconografía y Controladores

IconThemeManager (Cascada de Fallback)



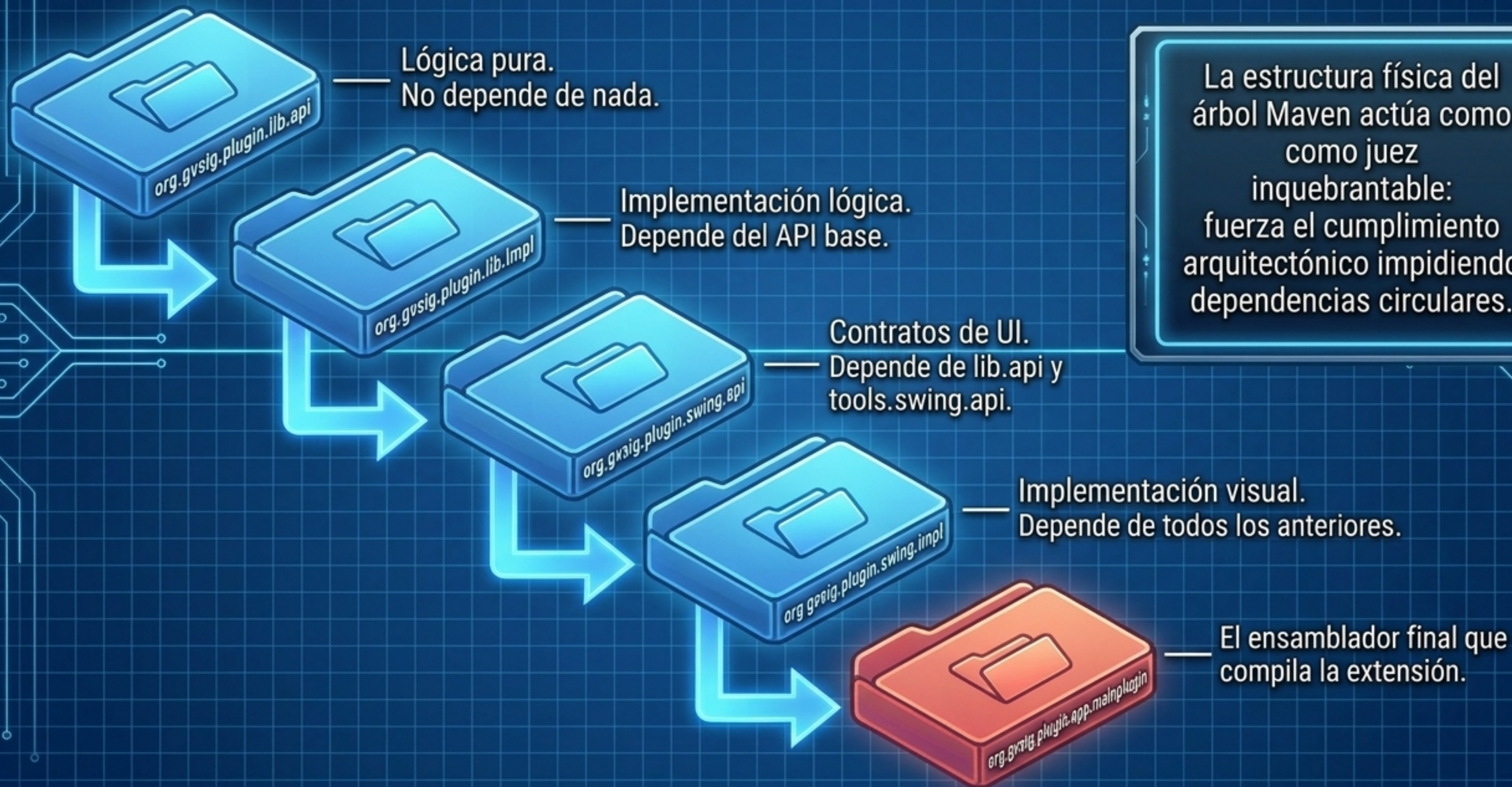
Resolución dinámica para temas visuales.

Framework de Pickers



Inyección de controladores complejos (Pickers) en contenedores mudos, resolviendo diálogos, concurrencia y persistencia de memoria sin código extra.

VII. Ecosistema de Desarrollo: Estructura Maven



VII. El Ciclo de Vida: El Despliegue Final



La culminación del plano estratificado: capas en aislamiento funcional que garantizan un SIG robusto, predecible y altamente escalable.