



Modul 6: Voraussetzungen einer GDI
Vertiefende Dokumente | Stand: 24.01.2012

Architektur einer GDI: Service-oriented Architecture (SOA)

Zu den Hauptargumenten für eine Geodateninfrastruktur zählen unter anderem die Schlagworte Interoperabilität, Skalierbarkeit und Performanz. Die Anforderungen an eine GDI können nur in Form einer geeigneten Architektur, also einer bestimmten Konstellation beteiligter IT-Komponenten umgesetzt werden. Ganz konkret handelt es sich dabei um die sogenannte Dienste-orientierte Architektur (engl. Service-oriented Architecture SOA).

Um diese technische Voraussetzung zum Aufbau und zur effektiven Nutzung einer Geodateninfrastruktur verstehen zu können, müssen zunächst auch die Begriffe der verteilten Anwendung und der 3-Tier-Architektur erläutert werden.

Verteilte Anwendung

Bei einem verteilten System befinden sich die verschiedenen Hardware- und Softwarekomponenten auf unterschiedlichen, voneinander unabhängigen Rechnersystemen, die miteinander vernetzt sind. Durch den Austausch von Nachrichten können die Komponenten miteinander kommunizieren und somit ihre Aktionen untereinander koordinieren.

Das bekannteste Beispiel für ein verteiltes System ist das Internet. Durch Verwendung des Internets können die Nutzer unabhängig von ihrem momentanen Standort verschiedene Dienste und Anwendungen, die sich auf unterschiedlichen Rechnern/Servern befinden, nutzen.

Eine verteilte Anwendung nutzt die Technologie eines verteilten Systems, d.h. auch hier befinden sich die verschiedenen Komponenten (Hard- und Software) auf unterschiedlichen Computern. Über ein Rechnernetzwerk, wie beispielsweise einem Intranet oder dem Internet, können die Komponenten sowohl untereinander als auch mit einem Anwender kommunizieren.

Beispiele für verteilte Anwendungen sind Internetshops und Flugbuchungssysteme.

Das Ziel einer verteilten Anwendung ist die gemeinsame Nutzung von Ressourcen, wobei unter Ressourcen Hardware, Daten und Funktionalität zu verstehen sind. Diese Nutzung gemeinsamer Ressourcen führt zu erheblichen Kosteneinsparungen. Durch Anlegen redundanter Komponenten innerhalb einer verteilten Anwendung kann zusätzlich die Ausfallsicherheit der gesamten Anwendung erhöht werden. Fällt beispielsweise der Datenbankserver innerhalb einer verteilten Anwendung aus, so kann auf einfache Art und Weise eine Ersatzdatenbank angeschlossen werden. Dadurch sind verteilte Anwendungen sehr gut skalierbar, da einzelne Komponenten einfach ausgetauscht oder neue Komponenten hinzugefügt werden können. Durch Aufspaltung der Anwendung in einzelne Komponenten wird die Gesamtlast der Anwendung ebenfalls auf diese Komponenten verteilt. Auf diese Weise wird die Performance erhöht und es kommt zu schnelleren Antwortzeiten.

Folgende vier Komponenten entsprechen den grundlegenden Komponenten einer verteilten Anwendung:

- Client,
- Webserver,
- Applikationsserver,
- Datenbankserver.

Weiterhin können die Komponenten in die Gruppe der Clients und Server unterteilt werden.

- Im Allgemeinen handelt es sich bei einem Client um ein Programm, das auf die Ressourcen eines Servers zugreifen kann und mit diesem über ein Rechnernetzwerk kommuniziert. Ein Client kann in zwei unterschiedlichen Ausprägungen zum Einsatz kommen. Es wird hierbei zwischen einem Thin Client und einem Fat Client unterschieden. Bei einem Fat Client handelt es sich um eine spezielle Software, welche die Daten, die auf einem Server abgelegt sind, selbstständig verarbeitet. Im Gegensatz hierzu enthält ein Thin Client keine Anwendungslogik, sondern stellt eine Präsentationsschnittstelle oder grafische Benutzeroberfläche zur Präsentation der Daten für den Nutzer bereit. Die heute gängigste Variante eines Thin Clients ist ein einfacher Web-Browser wie beispielsweise der Microsoft Internet Explorer, Firefox von Mozilla oder Netscape Navigator.
- Für gewöhnlich ist ein Server ein Programm, welches auf eine Anfrage von einem Client wartet, diese bearbeitet und schließlich beantwortet. Generell können unterschiedliche Kategorien von Servern unterschieden werden. Im Folgenden sollen nur Webserver, Applikationsserver und Datenbankserver näher betrachtet werden.
 - > Ein Webserver stellt einem Client Informationen, z.B. Internetseiten, über das HTTP-Protokoll zur Verfügung. Aus diesem Grund werden Webserver oft als HTTP-Server bezeichnet. Die beiden heute bedeutendsten Webserver sind der Internet Information Server (IIS) von Microsoft und der Apache HTTP-Server von der Apache Software Foundation, wobei es sich bei dem zweiten Produkt um eine Open-Source-Lösung handelt.
 - > Ein Applikations- oder Anwendungsserver ist ein Server in einem Computernetzwerk auf dem spezielle Software-Applikationen ausgeführt werden. Des Weiteren stellt ein Applikationsserver für gängige Problemstellungen bereits integrierte Lösungen, wie Sicherheitslösungen oder Entwicklungswerkzeuge zur Verfügung. Auf dem Markt erhältliche Produkte sind beispielsweise BEA WebLogic, IBM WebSphere, Oracle und JBoss Application Server. Mit dem JBoss Application Server steht auch in dieser Kategorie ein Open-Source-Produkt zur Verfügung.
 - > Die letzte vorzustellende Gruppe der Server stellen die Datenbankserver dar. Ein Datenbankserver verwaltet die notwendigen Daten in einer verteilten Anwendung und stellt diese einem Client oder einer weiteren Anwendung zur Verfügung. Gängige Datenbanken sind Oracle, MySQL, Microsoft SQL Server und PostgreSQL. MySQL und PostgreSQL sind zwei im Bereich der Datenhaltungskomponenten weit verbreitete Open-Source-Produkte.

3-Tier-Architektur

Die vier Komponenten (Client, Webserver, Applikationsserver, Datenbankserver) einer verteilten Anwendung können auf unterschiedliche Art und Weise miteinander kombiniert und angeordnet werden. Für die möglichen Architekturen hat sich der Oberbegriff der N-Tier-Architekturen gebildet („Tier“ engl. für Schicht oder Ebene). Im Allgemeinen wird bei der N-Tier-Architektur zwischen den drei folgenden Schichten unterschieden:

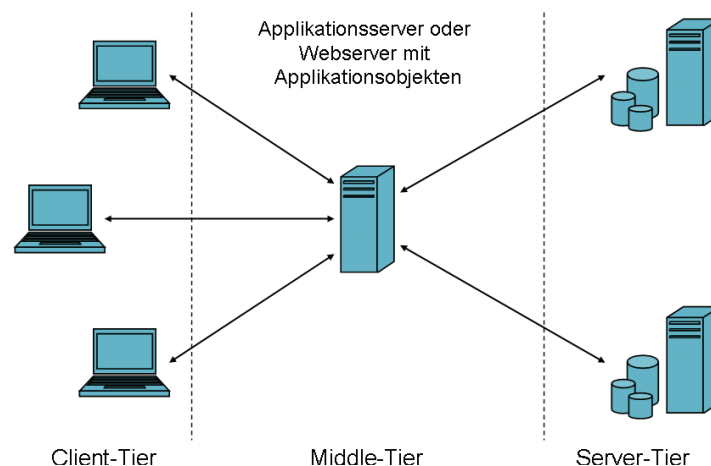
- Präsentationsschicht,
- Anwendungslogik,
- Datenhaltung.

Die Präsentationsschicht stellt die Schnittstelle zum Anwender dar. Sie entspricht im Grunde einer grafischen Benutzeroberfläche für einen Nutzer. Die Ebene mit der Anwendungslogik dient zur Bearbeitung und Beantwortung der Anfragen des Clients. Auf der dritten Schicht sind die eigentlichen Daten der Anwendung (in einer Datenbank) abgelegt.

Die Ebenen einer N-Tier-Architektur und die Komponenten einer verteilten Anwendung können unterschiedlich kombiniert werden. Hieraus ergeben sich die drei folgenden Architekturen:

- 2-Tier-Architektur,
- 3-Tier-Architektur,
- 4-Tier-Architektur.

Die 3-Tier-Architektur besteht aus drei Ebenen:

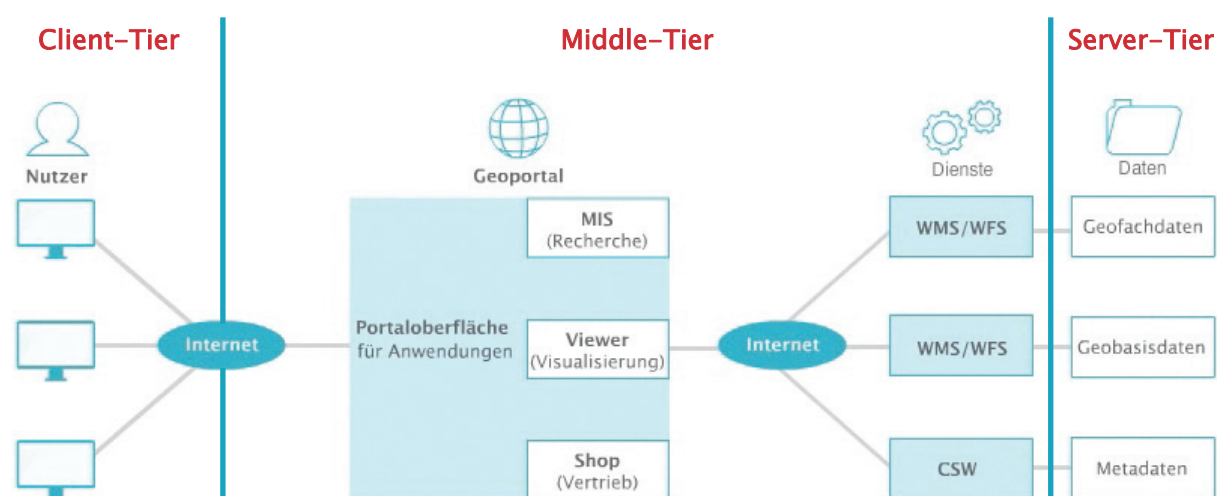


Auf dem Client-Tier befindet sich lediglich die Präsentationsschicht, so dass Thin Clients, d.h. Web-Browser eingesetzt werden können. Das Middle-Tier enthält die Anwendungslogik und besteht entweder aus einem Applikationsserver oder einem Webserver, der mit Applikationsobjekten umgehen kann. Das Server-Tier enthält letztlich nur noch die eigentlichen Daten. Die 3-Tier-Architektur ist heute die gängigste Variante der N-Tier-Architektur.

Eine einfache Internet-Anwendung mit Browser, Webserver und Datenbank ist ein Beispiel für die 3-Tier-Architektur. Die 3-Tier-Architektur ist sehr gut skalierbar, so dass neue Komponenten einfach hinzugefügt oder Komponenten ausgetauscht werden können. Die Gesamtlast der Anwendung wird auf weitere Komponenten verteilt, so dass die Antwortzeiten beschleunigt werden können.

Bei einer Geodateninfrastruktur handelt es sich um eine verteilte Anwendung, die der 3-Tier-Architektur zugeordnet werden kann.

Im Folgenden sollen die einzelnen Komponenten der GDI den verschiedenen Schichten der Architektur zugeordnet und erläutert werden.



(Quelle: J. Knab, 2005, Hessisches Landesamt für Bodenmanagement und Geoinformation – modifiziert)

Die Abbildung liefert einen Überblick über die wichtigsten Komponenten und deren Einordnung in das Client-, Middle- und Server-Tier.

- Das Client-Tier oder die Präsentationsschicht befindet sich auf der Seite der Nutzer.
- Das Middle-Tier umfasst die Anwendungslogik einer verteilten Anwendung. Bezogen auf die GDI sind dies die Komponenten des Geoportals, d.h. die verschiedenen Geoanwendungen, und die Dienste.
- Das Server-Tier beinhaltet die Datenbanken, welche sämtliche Daten der GDI, wie Geobasisdaten, Geofachdaten und Metadaten, beinhalten.

Nun sollen die einzelnen Schichten detaillierter betrachtet werden:

- Wie bereits beschrieben, kann im Allgemeinen bei der Client-Komponente zwischen einem Thin und einem Fat Client unterschieden werden. Im Rahmen der GDI können beide Clientvarianten zum Einsatz kommen. Bei einem Thin Client handelt es sich meist um einen Web-Browser, mit dem Daten präsentiert werden. Innerhalb der GDI kann ein Thin Client beispielsweise zur Recherche in Metadaten oder zum Visualisieren eines Datendienstes in einem Kartenviewer (z.B. Hessen-Viewer) verwendet werden. Als Fat Client kann beispielsweise ein herkömmliches Desktop-GIS zum Visualisieren eines WMS oder WFS gemeinsam mit den eigenen Daten eingesetzt werden.
- In einer verteilten Anwendung enthält das Middle-Tier die Anwendungslogik. Übertragen auf die GDI sind dies die Komponenten des Geoportals, d.h. die verschiedenen und Anwendungen, und die Geodatendienste. Die wichtigsten Anwendungen dieses Geoportals sind ein Metadateninformationssystem (MIS) zur

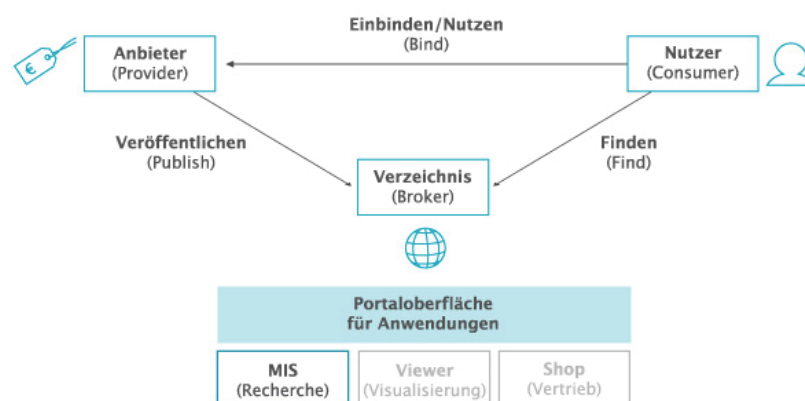
Recherche in den Metadaten, ein Viewer zur Visualisierung von verschiedenen Datendiensten (z.B. WMS oder WFS) und ein Shopsystem zum Vertrieb und zur Abrechnung der verschiedenen Geodaten-Produkte. Im Rahmen der GDI-Süd Hessen zählen die beiden Leitprojekte ebenfalls zu den Anwendungen.

- Die Datenhaltungsschicht befindet sich in der 3-Tier-Architektur auf dem Server-Tier. Auch innerhalb der GDI sind hier die notwendigen Daten in den entsprechenden Datenbanken oder als Dateien (z.B. Shapefiles) aufzufinden. Diese Datenhaltungskomponente befindet sich in der Regel bei der für die Daten zuständigen Stelle.

Service-oriented Architecture (SOA)

Das Prinzip der Service-orientierten Architektur (SOA) setzt auf der 3-Tier-Architektur auf.

Dabei stellen die Anbieter (Service Provider) ihre webbasierten Dienste plattformübergreifend zur Verfügung und publizieren diese in einem öffentlichen Verzeichnis (Service Broker), also in einem Metadateninformationssystem eines Geoportals. Die Verbraucher (Service Consumer) sind dadurch in der Lage, die Dienste in diesem Verzeichnis zu finden, um sie letztendlich nutzen zu können – also in ein Web-GIS (z.B. ein Online-Viewer als Bestandteil eines Geoportals) oder ein eigenes (Desktop-) GIS einbinden zu können. Dieses Prinzip bezeichnet man als Publish-Find-Bind-Muster.



Ein wichtiges Merkmal ist, dass hierbei über standardisierte Schnittstellen auf autonome Dienste zugegriffen wird.

Durch die Umsetzung des SOA-Architekturmusters ergeben sich einige Vorteile:

- Die Dienste sind Plattform-neutral, d.h. ihre Nutzung ist unabhängig von der Art des Nutzer-Systems möglich.
- Die Dienste verschiedener Anbieter sind aufgrund der Einhaltung von Standards einheitlich und interoperabel nutzbar.
- Vorhandene Geodaten werden innerhalb einer GDI gemeinsam genutzt, wodurch eine redundante Datenerhebung und -haltung vermieden wird.
- Dadurch und durch die Benutzung weiterer zur Verfügung gestellter Komponenten können erhebliche Kosteneinsparungen erreicht werden.

- Die Architektur führt zu einer Unabhängigkeit von der Technologie „hinter“ den Diensten. Es spielt beispielsweise keine Rolle, welche Software ein Datenanbieter zur Datenhaltung oder zur Publizierung als Dienst verwendet, da die Daten standardisiert nach außen gegeben werden.
- Die lose Kopplung der Komponenten einer GDI führt zu voneinander unabhängigen Diensten.
- Neue Komponenten können sehr einfach in die Infrastruktur integriert werden, wodurch ein problemloses und rasches Wachstum einer GDI gewährleistet wird.
- Die bereits erwähnte Ausfallsicherheit wird bei entsprechender Konstellation der Anwendungen gewährleistet.