DIMAG-Architekturbeschreibung

Vorwort

Mit der nachfolgenden *Dokumentation der Architektur einer Webanwendung* wird eine Maßnahme des IT-Grundschutzes erfüllt[1].

Am 09. Oktober 2017 wurde die DIMAG-Software Release 3.0 dem DAN zur Verfügung gestellt. Die nachfolgenden Ausführungen basieren auf den vorliegenden Release Notes.

Der DIMAG-AFIS-Connector wird nachfolgend nicht beschrieben.

Umsetzung der IT-Grundschutz-Maßnahme M 5.169

Die IT-Grundschutz-Maßnahme *M 5.169 Systemarchitektur einer Webanwendung*[2] fordert u.a. eine mehrschichtige Architektur, eine Trennung nach Server-Rollen und eine mehrschichtige Netzwerkarchitektur.

DIMAG wird über diese Rollen strukturiert:

- Webanwendung basiert auf PHP inkl. SOAP-Service
- Datenbank basiert auf MySQL bzw. MariaDB
- IngestList-Service
- File-Service (Storage)
- Integrationscheck-Service
- SFTP-Service

[1] https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/_content/m/m02/m02486.html
[2] https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/_content/m/m05/m05169.html

DIMAG basiert auf einer mehrschichtigen Architektur:

- Webschicht und Anwendungsschicht
  DIMAG-PHP-Anwendung, SOAP, IngestList-Service, Integritätstest
  Das Kernmodul und der SOAP-Service bilden eine Komponente und können nicht getrennt werden. WebGUI (DIMAG-PHP) und SOAP verwenden den Ingestlist-Service. Die DIMAG-PHP-Anwendung nutzt den SOAP-Service-Code im Kernmodul-Backend.
- Datenschicht
  Datenbank, Storage, SFTP

Bei einer angestrebten Trennung nach Server-Rollen auf separaten Servern werden folgende BSI-Grundschutzforderungen an eine Architektur von Webanwendungen erfüllt:

- Eingeschränkte Konten für Serverprozesse der Systemkomponenten
- Mehrschichtige Netzwerkarchitektur
- Einsatz von Web Application Firewalls

Die Einlagerung von Digitalen Objekten erfolgt entweder über die WebGUI oder die SOAP-Schnittstelle. Der spätere Zugriff (Nutzung) auf die eingelagerten Digitalen Objekte erfolgt über ein für das DIMAG entwickeltes Web-Access-Tool.

Architekturdarstellung

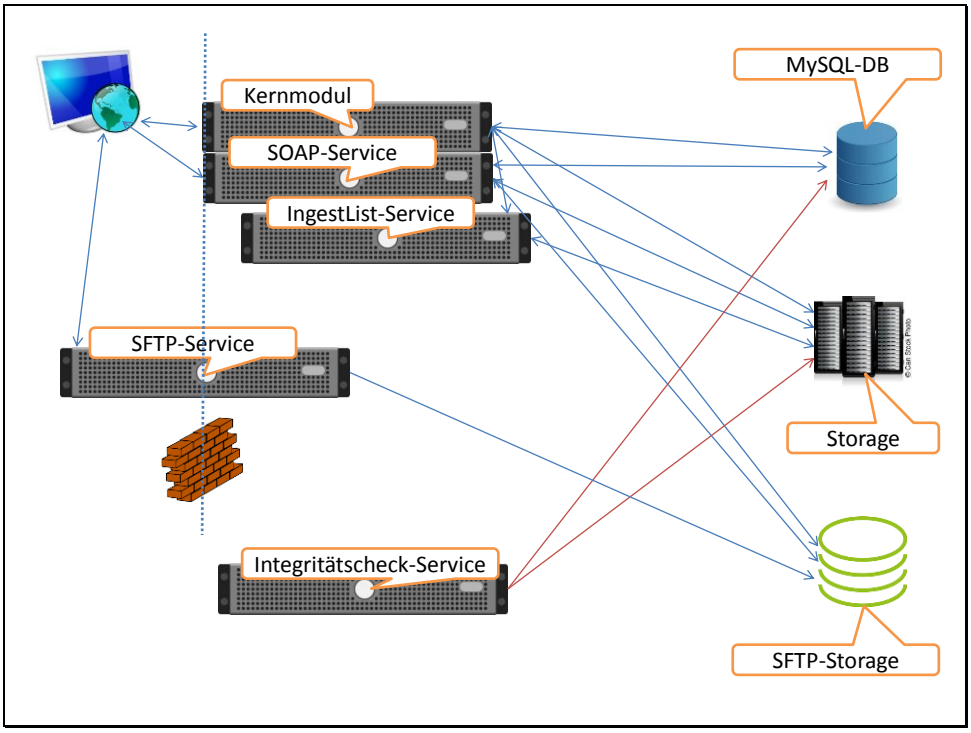

Abbildung 1 Grafische Darstellung aller Komponenten

Die nachfolgenden Übersichten beschreiben die in der vorherigen Grafik dargestellten Kommunikationsbeziehungen.

Übersicht der Kommunikationsbeziehungen zwischen Komponenten

- Nutzer – Kernmodul (https)
- Nutzer – SOAP-Service (https)
- Nutzer – SFTP (sftp)

- Kernmodul - Nutzer
- Kernmodul – Datenbank (MySQL/MariaDB)
- Kernmodul – SFTP-Storage
- Kernmodul - Storage
- Kernmodul – IngestList (Socket)

- SOAP-Service - Nutzer
- SOAP-Service - Datenbank (MySQL/MariaDB)
- SOAP-Service – Storage

- SOAP-Service – SFTP-Storage

- IngestList-Service – Kernmodul (Socket)
- IngestList-Service – Storage (read only)

- SFTP-Service – Nutzer (SFTP)
- SFTP-Service – SFTP-Storage

- Integritätscheck-Service – Storage (read only, log)[3]
- Integritätscheck-Service – Datenbank (MySQL/MariaDB)

- SOAP – AFIS[4]-Connector (noch nicht freigegeben)

Übersicht der Storage- und Datenbank Zugriffsrechte

- Ein Database-Account für das Kernmodul (PHP-Anwendung und SOAP-Service)
- Der Integritätscheck-Service nutzt per default denselben Datenbank-Account. Nach einer Konfigurationsanpassung könnte ein separater Datenbank-Account genutzt werden.[5]

Sonstige Zugriffsrechte

- Webserver: Zugriffsrechte auf Storage und SFTP-Storage
- Ingestlist-Service: Read Only Zugriffsrechte auf Storage und evtl. SFTP-Storage
- Integritätscheck-Service: Read Only Zugriffsrecht auf Storage
- Nutzer: Write Only Zugriffsrecht auf SFTP-Storage

Ingest

Der Ingest beinhalt die Übernahme der Originaldaten, dem SIP (Submission Information Package[6]), und die Aufbereitung als AIP (Archival Information Package[6]) für die Aufnahme ins Digitale Magazin. Die eigentliche Aufnahme und Einlagerung im Digitalien Magazin DIMAG erfolgt über die WebGUI oder die SOAP-Schnittstelle unter Einbeziehung des SFTP-Services. Dabei gibt es drei mögliche Vorgehensweisen. Erstens: Bei der Übernahme per

[3] Der Storage wird read only genutzt, aber das Log File wird im Storage gespeichert (Filename: md5.txt).
[4] Archivische Fachinformationssysteme.
5 Konfigurationsanpassung in Abstimmung mit den DIMAG-Entwicklern.
[6] https://de.wikipedia.org/wiki/OAIS

WebGUI wird ein aufbereites AIP in das DIMAG übernommen und unverändert eingelagert. Zweitens: Das im Hessischen Hauptstaatsarchiv entwickelte „IngestTool“ bereitet SIPs auf und kopiert die erzeugten AIPs per SFTP in das DIMAG und führt anschließend per SOAP-Schnittstelle die Einlagerung ins DIMAG durch. „IngestTool“ wurde für den Masseningest entwickelt. Dabei können unterschiedliche „Ingest-Vorlagen“ entwickelt und abgearbeitet werden. Drittens: Das im Landesarchiv Baden-Württemberg entwickelte „IngestList“ bereitet ein SIP als AIP auf und übergibt es dem DIMAG per SFTP. Anschließend wird es über die WebGUI im DIMAG eingelagert.

Access

Die Generaldirektion der Staatl. Archive Bayerns entwickelt gegenwärtig ein „Web-Access-Tool“, mit dem AIP unter Berücksichtigung der Anforderungen und Zugriffsrechte zu DIP (Dissemination Information Packages[6]) aufbereitet und bereitgestellt werden.

Varianten der Mandantenfähigkeit

DIMAG ist de facto ein mandantenfähiges digitales Magazin. Diese Mandantenfähigkeit wird durch die mehrfache Bereitstellung (redundante DIMAG Installation) der Webanwendung auf einem Webserver ermöglicht. Dabei wird der Quellcode je Mandant vorgehalten und notwendige Konfigurationsanpassungen werden innerhalb dieser Struktur durchgeführt. Der gesamte Quellcode umfasst ca. 50 MB. Bei dieser Methode können Storage und Datenbank bei jedem Mandanten klar voneinander abgegrenzt werden. Es wird das höchstmögliche Sicherheitsniveau erreicht. Diese Variante kann bereits mit dem Release 3.0 umgesetzt werden. Die DIMAG-Entwickler streben an, den unveränderlichen Quellcode auf einem Webserver nur einmal vorzuhalten und die speziellen Konfigurationsanpassungen je Mandant auszulagern (voraussichtlich DIMAG 3.2 – Termin noch nicht festgelegt).

Eine Pseudo-Mandantenfähigkeit[7] kann alternativ durch **ein** DIMAG je Server erreicht werden, wobei je Mandant ein Nutzer innerhalb einer DIMAG-Installation eingerichtet wird. Eine Abgrenzung der Datenschicht innerhalb der Datenbank ist dabei nicht möglich, eine Abgrenzung des Storage je Mandant ist nicht möglich. Anforderungen des BSI-Grundschutzes werden bei dieser Variante nur unzureichend erfüllt.

[7] Dieser Begriff wurde nur zur Abgrenzung gegenüber einem echten mandatenfähigen Systems gewählt.

Nachfolgend werden Unterschiede der Architekturen beider Mandantenansätze beschrieben, wobei die Rollentrennung als wesentliche BSI-Grundschutzanforderung berücksichtigt wird.

- Die Konfigurationsanpassungen beschränken sich auf eine übersichtliche Anzahl von Dateien (siehe nächstes Kapitel).
- Wenn je Mandant ein DIMAG genutzt wird (redundante DIMAG-Installation),
    - dann besteht kein Migrationsbedarf beim Release 3.2. Nach Einschätzung der Entwickler werden dann nur die Quellcode- und Konfigurationsdateien ausgetauscht bzw. geändert.
    - dann kann für den Storage und die Datenbank ein standardisiertes Rechtekonzept angewendet werden.
    - dann kann jederzeit ein Mandant auf einen anderen Server „verschoben" werden.
    - dann können Services für jeden Mandanten konfiguriert, gegeneinander abgegrenzt und standardisierte Rechtekonzepte angewendet werden, ohne zusätzliche Server aufbauen zu müssen.
    - dann bestehen keine logischen und technischen[8] Abhängigkeiten zwischen den einzelnen Mandanten.

Fazit: Die Umsetzung der Variante „redundante DIMAG Installation" empfiehlt sich bereits mit dem Release 3.0.

Umsetzung der Mandantenfähigkeit bei Umsetzung der Maßnahmen M 5.169

Nachfolgend werden alle Konfigurationsmöglichkeiten des DIMAG benannt. Die Aufzählung illustriert, mit welch geringem Aufwand mehrere DIMAG-Installationen auf einem Server aufgebaut werden können.

- Kernmodul
  .htaccess (Verweis auf die config.inc.php, PHP-Error-Log)
  config/db.inc.php (Datenbank-Verbindungskonfiguration)
  config/config.inc.php[9] (Storage, SOAP-WSDL, , Integritäts-Log)
  config/setup.inc.php (Pfadangaben für IngestList, Storage, SFTP)
  config/options.xml (PHP-GUI-Konfiguration – kann auch über DIMAG-Oberfläche angepasst werden)

[8] Bis auf den gemeinsame Anwendungscode. - Das XML-Verzeichnis muss in den Storage verlagert werden (hängt von Database-Fields ab) – Anpassung Konfiguration notwwendig (config/config.inc.php).
[9] Diese Datei sollte nicht verändert werden.

grafik/* (DIMAG-Images[10])

setup/*.sql (Datenbank- Inititialiserungsskripte unverändert einmalig ausführen)

afis-config.inc.php (zukünftig Konfiguration der AFIS-Integration)

- SOAP

  _soap/clients/generic/WebServices.wsdl (generische DIMAG-SOAP-WSDL-Datei)
- Es müssen XSD-Dateien einmalig je DIMAG generiert werden.

  sudo -u apache _bat/updateXSD.php.sh[11]
- Die initiale Tektonik wird über ein Skript geladen.

  sudo -u apache _bat/loadxml.php.sh tektonik.xml

Sonstige Hinweise:

- Der SOAP-Service kann nicht ausgelagert werden.
- Das Integritätscheck– Skript befindet sich im Verzeichnis bat/icheck*.sh.
- Der Integritätscheck wird per cronjob aktiviert.
- Der SFTP- Storage wird nicht automatisch bereinigt (cronjob-Skript notwendig).
- ImageMagick wird für die PDF/A-Generierung benötigt.
- Je DIMAG-Installation wird maximal eine AFIS-Connector verfügbar sein.

10 Es sollte nur das Logo angepasst werden.
11 Einmalig zum Installationszeitpunkt und bei jeder Änderung in DB:fields. Bei einer Tabellenänderung müssen XSDs neu generiert werden – Abstimmung mit den Entwicklern notwendig.