Systemarchitektur

Analyse bestehender Software, Systemumgebung:

Vorgehen:

Um die Systemumgebung festzulegen, und vor allem die Software passgenau in die Softwarelandschaft des JRZ einfügen zu können, werden die bestehenden System wie zum Beispiel Smart Viz, die in Entstehung befindlichen Zugriffsysteme (BAC1 Gruppe - <Namen rein> und Importprogramme (BAC1 – Gruppe <Name rein>). Was war mit Opennes? In die Überlegungen werden weiters etwaige Kosten für Lizenzen und andererseits Sicherheitsaspekte einbezogen [referenz BSI: Grundschutzprofil].

Ergebnis:

* Hardware: für den Betrieb ist keine explizite Hardware vonnöten, vom JRZ wird eine virtuelle Maschine im Bladecenter zur Verfügung gestellt.
* Betriebssystem: auf Grund der größeren Erfahrung der Entwickler mit der Administration erfolgt die Implementierung auf einem Windows System (Windows Server 2012 R2),
* Programmiersprache: bestehende Software wurde im Umfeld des JRZ in Java implementiert, ebenso werden laufenden Projekte in Java programmiert, die Erfahrung der Umsetzenden reicht aus, um die Anforderungen umzusetzen. Im Sinne einer einfachen Übergabe, Weiterführung und Wartung fällt die Entscheidung, dieses Projekt in Java zu implementieren.
* Datenbank: neben unterschiedlichen RDBMS (Oracle Database Server, MySQL, ) wurden NoSQL Datenbankensysteme untersucht (MongoDB, Hadoop). Im Sinne der Integration in die bestehende Softwarelandschaft wird MySQL eingesetzt. Ziel des Projektes ist eine Integrationsdatenbank, was dazu führt dass sehr große Datenmengen verwaltet werden können müssen, und der Zugriff in akzeptabler Zeit erfolgt. MySQL unterstützt lokale (partitioning) und verteilte (sharding) Fragmentierung. Nach dem „Guide to Scaling Web Databases with MySQL Cluster“ [Referenz] kann der zu erwartenden Menge von Schreibzugriffen (~500k/d) durch Verteilung auf einzelne Nodes begegnet werden. Sollte ein Cluster mehrerer Nodes zu Performanceeinbußen führen können einem Cluster einfach weitere Nodes hinzugefügt werden, eine Anpassung der Applikation ist in diesem Fall nicht notwendig, über die Administration des MySQL Clusters wird das Partitionierung transparent für die Anwendung durchgeführt. Details dazu sind im Abschnitt „Datenbankzugriff“ erläutert.
* Datenbankdesigntool: untersucht wurden Oracle SQL Developer Data Modeler 4.1.5 und die MySQL Workbench 6.3.9. Für den graphischen Tabellenentwurf bieten beide Tools Unterstützung. Auf den Vorteil des logischen (ER) Designs, welches der Oracle Modeler unterstützt wird bewusst verzichtet, da der Aufwand die Überführung in ein Tabellenmodell zu machen geringer ist, als die Skriptdateien zur Anlage der Tabellen, Indizes, Einschränkungen bezüglich referentieller Integrität und die automatische Vergabe eindeutiger Schlüssel nachzubearbeiten. Da als Datenbanksystem MySQL eingesetzt wird, liegt es nahe das Designtool vom gleichen Hersteller einzusetzen und damit Kompatibilitätsprobleme zu vermeiden.
* Die Anbindung und die Veröffentlichung der Schnittstellen: <wir haben noch nicht festgelegt wie das API aufgerufen werden kann> Webservice? Datenstrukturen?

Literatur:

IT-Grundschutz-Profil für Open-Source-Software (GSProOSS): <https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Hilfsmittel/Extern/Diplomarbeiten/Erstellung_IT-Profil_Lefin.pdf?__blob=publicationFile&v=1>

MySQL: Guide to Scaling Web Databases with MySQL Cluster: <https://www.mysql.de/content/download/id/277/>