**Protokoll, 4.11.2016**

Anwesende: Eduard Hirsch, Fabian Knirsch, Armin Veichtlbauer, Christopher Wieland, Maximilian Unterrainer

Papers und Software gibts unter www.en-trust.at

grundlegender Aufbau: Christian DB - API || RBAC

Usecases: Analyse und Entscheidung für repräsentative Usecases fürs für API Funktionen

Priorisierung:

1. Datenmodell: sofern die zu implementierte Software der BAC Gruppe verwendet werden bleibt Zeit um Alternativen zu Christian DB zu untersuchen (Stern Schema, KeyVAlue,Hives, Hadoop HDFS,....)

2. Authentifizierung und RBAC: BAC Gruppe schreibt RBAC für Personen und Komponenten, Authentifizierung über eine PKI oder Benutzername/Passwort

3. API definieren (REST/XML, Einzelsatz, Aggregation), Hauptaugenmerk ist die flexible Verwendung des API.

4. Testanwendung: eventuell Visualisierungsprojekt zur Validierung verwenden, oder eigenes Validierungsszenario überlegen

im Anschluss mit Manfred Wurz und Christian Peuker über die Verwendung von Smart-Viz zu Validierung:

stellt sich auf den ersten Blick schwierig dar. Eine Verwendung (Ersetzen des JDBC/JPA Zugriffs durch eine Schicht die das API verwendet).

OpenNES Gruppe:

Clients (Geräte oder User) greifen zuerst auf den LDAP Server zu und fragen ab, in welche Gruppe sich der Client befindet und damit welche Berechtigungen sie haben und daraus agiert man dann mit der DB??

Verbindung zwischen Client und Server ist mit einem Zertifikat gesichert. (Jeder Client muss anfangs mit einem Public Zertifikat versehen werden für spätere Kommunikation)

Berechtigungsgruppen (in unserem Fall Universitäten, Endkunden, Stromversorger) werden vordefiniert und dann an Clients vergeben sprich der Client wird anfangs zu einer oder mehreren Gruppen hinzugefügt, worauf später abgefragt wird.