

POC 1:

Implementierung einer MySQL-Datenbank in einem NodeJS-Server und synchroner Datenaustausch zwischen Android-Anwendung und NodeJS-Server und anschließende Speicherung der Daten in einer MySQL-Datenbank

Exit:

1. Implementierung der MySQL-Datenbank in NodeJS ist erfolgt
2. Daten werden in der Datenbank fehlerfrei abgelegt
3. Daten sind über die Android-Anwendung fehlerfrei abrufbar

Fail:

1. Implementation der MySQL-Datenbank in NodeJS nicht möglich
2. Daten werden nicht oder fehlerhaft gespeichert
3. Abruf der gespeicherten Daten schlägt fehl

Fallback:

Implementierung einer MongoDB-Datenbank im NodeJS-Server

POC 2:

Implementation eines Publish-Subscribe -Nachrichtenservice mittels "RabbitMQ"

Exit:

Abonnieren und Publishen der folgend genannten Instanzen über beliebige Topics:

1. von mobilen Client publishen -> Nachricht wird an die Consumer übermittelt
2. mit mobilen Client subscriben -> Nachricht werden vom Client empfangen
3. mit NodeJS-Server subscriben und publishen -> Nachricht werden empfangen und übermittelt

Im RabbitMQ Manager sind die Consumer und die erstellten Queues aufgeführt.

Die übermittelten Nachrichten sind im Manager protokolliert.

Fail:

1. Der Nachrichtenaustausch zwischen o.g. Instanzen über "RabbitMQ" gelingt nicht.
2. Im "RabbitMQ" Manager sind die Consumer und die erstellten Queues nicht aufgeführt.
Die übermittelten Nachrichten sind im Manager nicht protokolliert.

Fallback:

Implementation eines Publish-Subscribe Nachrichtenservice mittels "Faye"

POC 3:

Prüfung von Arzneimittelinteraktionen, Kontraindikation mittels des “ifapWebservice”

Exit:

1. kostenfreier Zugriff auf Webservice vom Betreiber ifap gestattet
2. Zurückgelieferte Daten des Webservice können verwendet werden, um Kontraindikationen und Nebenwirkungen zu prüfen mittels den Daten der Patientenakte
3. Anbindung in den Node Server gelingt

Fail:

1. Zugriff auf Webservice nicht kostenfrei gestattet
2. Anbindung des Webservices an das MDKS (Mediaktations-System) misslingt

Fallback:

prototypische Simulation der Arzneimittelinteraktions- und Kontraindikationsprüfung auf Grundlage selbstverfasster Datensätze

POC 4:

Sichere Datenübermittlung vom Android-Client zum Node-Server durch SSL oder TLS

Exit:

1. Eine https Verbindung vom mobilen Client zum NodeJS-Server ist möglich.
2. Eine Abfrage vom Client mit dem https-Protokoll ist erfolgt.

Fail:

1. Eine https Verbindung vom mobilen Client zum NodeJS-Server ist nicht möglich.
2. Eine Abfrage vom Client mit dem https-Protokoll ist nicht erfolgt.

Fallback:

unverschlüsselte Datenübertragung innerhalb eines VPNs

POC 5:

Implementation eines “Failover-Clusters” (automatisierte Ausfallsicherung)

Exit:

Bei einem Ausfall des Hauptserverausfall werden Anfragen der Clients automatisch auf einen Ausweichserver umgeleitet und der Abruf ist trotzdem möglich.

Fail:

Der Abruf vom Client an den Server ist nicht mehr möglich, nachdem ein Server ausfällt.
Die Abfrage des Clients wird nicht auf einen Ausweichserver umgeleitet.

Fallback:

“Failback” (Zurücksetzen des Systems auf einen vorherigen Zustand)