[>>> Go to the english version](https://docs.google.com/document/d/1OOtW84KmgRC3ZqXgr-GxSytQ0mFemoiBBYVEwbg_Lmw/edit)

Dokumentation der Gradido-Plattform

Deutsche Version, Stand: 03.09.2020

[Allgemeine Betrachtungen](#_e4v1ghh8zn1p)

[Anforderungen](#_vuk6ceuxj4pm)

[Vermeidung multipler Schöpfung](#_wij2zkgtrflz)

[Open Source erst später](#_nf4rwry49flo)

[Architektur](#_yw3lvs1cd2cs)

[Herangehensweise](#_rgd2bceuzvs8)

[Schichten](#_p0f530w2t3t3)

[Schicht 1: Transaktions-Protokoll](#_brcy161optbs)

[Schicht 2: Blockchain](#_fve20pw1jkj8)

[Schicht 3: Benutzer-Konten](#_nyzv9dqczblc)

[Schicht 4: Benutzer-Oberfläche](#_8buv8zinvixp)

[Software-Komponenten](#_k1n0pnqjlehl)

[Community-Server (Group Server)](#_v5omkeev7iec)

[GDT-Server](#_4zbjeg6r4pmn)

[Login-Server](#_ur0kat1alr7u)

[Node-Server](#_79x5archfrxk)

[Abbildung Server-Verbindungen](#_ze6eim6mbb4l)

# 

# 

# Allgemeine Betrachtungen

## Anforderungen

* **Währungsplattform für das Gradido-Modell** – 3-fache Geldschöpfung 3x1000 GDD im Monat (Aktives Grundeinkommen, öffentlicher Haushalt, Ausgleichs- und Umweltfonds), 50% Vergänglichkeit im Jahr
* **Einfache Bedienung für jedermann** – Benutzung muss so einfach sein wie PayPal, dh. E-mail-Adresse bzw. Username. Deshalb müssen private Keys im Login Server verschlüsselt gespeichert sein.
* **Weltweite Skalierbarkeit** – Das System soll weltweit skalierbar sein, so dass es die ganze Welt für ihre Finanz-Transaktionen verwenden kann. Erreicht wird dies durch Unterteilung in Sub-Netze welche jeweils zu eine Gruppe Menschen z.B. einer Gemeinde gehören.
* **Sicherheit** – Das System muss den Sicherheitsanforderungen für Finanztransaktionen genügen.
* **Stufenweise Entwicklung in Milestones und Sprints** – das System läuft bereits als MVP1 auf einer kleinen zentralen Plattform. Im nächsten Schritt soll es mit eingeschränktem Funktionsumfang auf der DLT-Plattform Hedera Hashgraph laufen (MVP2). Schrittweise wird der gesamte Funktionsumfang entwickelt.

## Vermeidung multipler Schöpfung

Das Aktive Grundeinkommen wird durch Leistungen an die Gemeinschaft geschöpft. Dafür ist eine überschaubare Größe der Gemeinschaften (im Quellcode “Gruppen” genannt) erforderlich. Ein Gemeinschaftsmitglied muss seine geleisteten Stunden glaubhaft nachweisen. Da die Lebenszeit begrenzt ist, ist mehrfache Schöpfung beim aktiven Grundeinkommen weniger kritisch als beispielsweise bei einem bedingungslosen Grundeinkommen.

Wenn Gradido offizielle Währung wird, wird die Identität durch die Gemeindeämter sichergestellt.

Außerdem wird die Entwicklung der weltweit digitalen Identität von staatlicher und von privatwirtschaftlicher Seite vorangetrieben. Das ist nicht unsere Aufgabe.

## Open Source erst später

Zu einem späteren Zeitpunkt soll der Code offengelegt werden. Um die anfängliche Entwicklung so einfach und ungestört wie möglich zu gestalten, entwickeln wir zunächst in geschlossener Umgebung.

# Architektur

## Herangehensweise

* Aufbau in mehreren Schichten, ähnlich wie das [OSI-Modell](https://de.wikipedia.org/wiki/OSI-Modell)
* Nutzung von Public-Key Signatures für die Transaktionen ähnlich wie Bitcoin
* Nutzung von Hedera Hashgraph Consensus Service für die zeitliche Sortierung der Transaktionen
* Nutzung von google protobuf um die Transaktionen untereinander plattformunabhängig austauschen zu können
* Nutzung von C++ basierten Servern für die kritischen Systembereiche für maximale Sicherheit und Performance
* Nutzung von php und javascript für eine gute Benutzerfreundlichkeit und das schnellere Entwickeln neuer Komfort-Funktionen.

### Schichten

#### Schicht 1: Transaktions-Protokoll

* definiert als google protocol buffers
* wird wie eine Blockchain gespeichert
* sollte keine persönlichen Daten enthalten
* orientiert am Hedera Hashgraph Transaktions-Format
* signiert mit ed25519 Signaturen

#### Schicht 2: Blockchain

* zeitliche Einordnung der Transaktion in eine Chain pro Gruppe, z.B. von Hedera
* überprüfen, speichern und öffentlich zur Verfügung stellen der Transaktionen
* gesichert vor nachträglicher Veränderung, Löschung und Verlust durch Serverausfall

#### Schicht 3: Benutzer-Konten

* verschlüsseltes Speichern der private-keys für die Nutzer
* nur mit der richtigen E-Mail- Passwort Kombination können die private Keys entschlüsselt werden
* Konto Wiederherstellung/Passwort ändern mit Hilfe einer Passphrase aus welcher der Private Key berechnet werden kann
* (Optional) Speichern der Passphrase des Benutzer verschlüsselt mit Server-Admin-Key(s) falls der Benutzer doch nicht seine Passphrase bei Konto Eröffnung notiert hat. Sicherheitsrisiko!
* Signieren der Nutzer-Transaktionen

#### Schicht 4: Benutzer-Oberfläche

* Cachen und Anzeigen der Transaktionen die zum Nutzer gehören
* Erstellen von Transaktionen
* Komfort-Funktionen wie Freundeslisten für schnelle Überweisungen

## Software-Komponenten

### Community-Server (Group Server)

Schicht 4 – Benutzer-Oberfläche

Wir streben ein dezentrales Netz von Communities an, von kleinen Communities (einige 100 User) bis zu sehr großen Communities. Um es den kleinen Communities leicht zu machen, soll der Community-Server auch auf einfachem Webspace laufen. Deshalb enthält er keine besonders sicherheitsrelevanten Funktionen und wird in CakePHP entwickelt. Die Community-Admins brauchen keine hohe Qualifikation.

Um die Transaktionen selber über Hedera zu schicken, würde der Community Server das php-Modul für grpc brauchen. Das würde es für die Benutzer erschweren, den Community Server auf einem einfachen Webspace zu installieren.

**Eigenschaften des Community-Servers:**

* CakePHP
* modern
* Bedienerfreundlich
* Kommuniziert viel mit dem Login-Server
* Eine Instanz pro Gruppe
* Kann auch auf einem einfache Webspace mit php und mysql gehostet werden

### GDT-Server

Für die Übergangsphase zur Projektfinanzierung (Schicht 4 – Benutzer-Oberfläche)

Mitglieds- und Förderbeiträge, die über Elopage oder Digistore eingezahlt wurden, werden an den GDT-Server übertragen und dort mit einem Vielfachen an GradidoTransform (GDT) honoriert. GDT unterliegen nicht der Vergänglichkeit und können zu einem späteren Zeitpunkt in GDD umgewandelt werden.

**Eigenschaften des GDT-Servers:**

* CakePHP
* Anbindung an Elopage und Digistore
* Verwaltet GDT für alle die das Gradido-Projekt unterstützen
* Kommuniziert mit dem Community-Server über json-ajax-requests
* Zur Zeit nur eine Instanz
* Sollte später im voll etablierten System keine Rolle mehr spielen

### 

### Login-Server

Schicht 3, Benutzer-Konten und Schicht 2, Blockchain

Mehrere Communities können auf einen Login-Server zugreifen. Neben dem Login erfüllt er auch andere komplexe und sicherheitsrelevante Funktionen, z.B.

* Speicherung der Private Keys
* Signieren und Routen der Transaktionen ist Hauptfunktionen vom Login-Server.

Aus Gründen der Sicherheit und Performance wird der Login-Server in C++ entwickelt.

**Eigenschaften des Login-Servers:**

* C++ mit Poco
* sicherheitskritisch
* Kommuniziert via json-ajax-request mit dem Community-Server
* Kommuniziert via jsonrpc-request mit dem Node-Server
* Maximal ein Login-Server pro Gruppe
* kann mehrere Gruppe verwalten
* Hosting bedarf ssh-Zugriff also ein VPS oder Root-Server
* Server-Admin muss verlässlich sein und sollte etwas von Server-Sicherheit verstehen
* Sendet signierte Transaktionen ans Hedera-Netzwerk

### Node-Server

Schicht 2, Blockchain

* C++ mit Poco
* performance-kritisch
* Ein Node Server kann viele Gruppe verwalten
* speichert die Blockchain jeder Gruppe in einem separaten Ordner
* Anbindung ans Hedera-Netzwerk als Mirror-Node
* Erhält Transaktionen vom Hedera-Netzwerk
* Stellt Transaktionen-Daten bereit für Login- und Community-Server
* Es sollten mehrere pro Gruppe existieren falls mal einer ausfällt.
* Sollte – ausreichend Hardware vorausgesetzt – in der Lage sein für ein Supernode eingesetzt zu werden, welcher alle Transaktionen der Welt erhält, prüft und speichert (zurzeit etwa 40 Transaktionen pro ms, ca. 650 GByte pro Tag)

### Abbildung Server-Verbindungen

