



## WP4-A3. Funktionale Spezifikationen.



Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung -  
Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](https://creativecommons.org/licenses/by-sa/4.0/)

*„Finanziert durch die Europäische Union. Die geäußerten Ansichten und Meinungen sind jedoch ausschließlich die der Autoren und spiegeln nicht unbedingt die der Europäischen Union oder der Exekutivagentur Bildung, Audiovisuelles und Kultur (EACEA) wider. Weder die Europäische Union noch die EACEA können dafür haftbar gemacht werden.“*



Transilvania  
University  
of Brasov





## Inhalt

1.	EINLEITUNG .....	4
2.	SYSTEMÜBERSICHT UND ROLLEN .....	5
2.1.	RockChain-Konzept und wichtigste Trainingsszenarien .....	5
2.2.	Zielgruppe und Rollen.....	5
2.2.1.	Lernende (Teilnehmer).....	5
2.2.2.	Ausbilder/Moderatoren .....	6
2.2.3.	Technische Administratoren .....	6
2.3.	Bereitstellungs- und Laufzeitumgebung.....	7
2.3.1.	Geräte.....	7
2.3.2.	Konnektivität .....	7
2.3.3.	Umfang und Struktur der Sitzungen .....	7
3.	FUNKTIONELLE ARCHITEKTUR UND MODULE .....	8
3.1.	Hochrangige Client-Server-Architektur .....	8
3.2.	Frontend-Module .....	9
3.3.	Backend-Module .....	12
4.	DETAILLIERTE FUNKTIONELLE ANFORDERUNGEN .....	14
4.1.	Benutzerverwaltung und Authentifizierung.....	14
4.2.	Erstellung von Spielen, Beitritt und Lobby-Verwaltung .....	14
4.3.	Lebenszyklus von Spiel und Runde.....	15
4.4.	Markt, Ressourcen und Kreislaufwirtschaftslogik.....	16
4.5.	Herausforderungen beim Mining und Validierung von Lösungen .....	17
4.6.	Punktwertung, Belohnungen und Branchenauswahl.....	17
4.7.	Tracking, Protokollierung und Daten, die für die WP5-Bewertung benötigt werden .....	18
5.	SPIELABLÄUFE UND ZUSTANDSÜBERGÄNGE .....	19
5.1.	Gesamtlebenszyklus des Spiels.....	19
5.2.	Lebenszyklus einer Runde.....	20
5.3.	Spielerflüsse .....	22
5.4.	Synchronisationsregeln zwischen Clients .....	23
6.	NICHT-FUNKTIONELLE ANFORDERUNGEN.....	26



---

6.1. Leistung und Skalierbarkeit in typischen Berufsbildungs- /Erwachsenenbildungsumgebungen .....	26
6.2. Zuverlässigkeit und Fehlertoleranz unter Unterrichtsbedingungen .....	27
6.3. Sicherheit, Datenschutz und Datensicherheit .....	27
6.4. Benutzerfreundlichkeit und Zugänglichkeit für erwachsene Lernende mit heterogenen digitalen Fähigkeiten .....	28
7. SCHLUSSFOLGERUNGEN UND NÄCHSTE SCHRITTE .....	30

## 1. EINLEITUNG

Dieses Dokument präsentiert die Ergebnisse der Aktivität WP4.A3: Funktionale Spezifikationen des RockChain-E-Learning-Tools. Aufbauend auf der in WP4.A1 gelieferten Datenbank- und Backend-Architektur und der in WP4.A2 erstellten, verfeinerten, pilotierbaren Version des Tools konzentriert sich diese Aktivität darauf, in prägnanter und strukturierter Form zu beschreiben, was RockChain aus funktionaler Sicht leisten soll. Der Schwerpunkt liegt dabei auf dem Verhalten des Tools, wie es von Lernenden, Trainern und Moderatoren während realer Schulungen erlebt wird, und nicht auf der Implementierung auf Code-Ebene oder technischen Details auf niedriger Ebene.

Das Hauptziel von WP4.A3 ist es, eine klare Beschreibung der Funktionen, Module und Abläufe zu liefern, aus denen sich das RockChain-E-Learning-Tool nach der Verfeinerungsphase zusammensetzt. Dazu gehört, wie Benutzer authentifiziert und verwaltet werden, wie Spielsitzungen und -runden erstellt und koordiniert werden, wie sich Markt- und Mining-Mechanismen verhalten und wie Punktestände, Belohnungen und grundlegende Indikatoren sowohl für das Gameplay als auch für die Bewertung erstellt werden. Die Spezifikationen erfassen, wie diese Elemente in typischen Berufsbildungs- und Erwachsenenbildungsszenarien interagieren, mit kleinen Gruppen von Teilnehmern, oft über 45 Jahre alt und mit heterogenen digitalen Fähigkeiten, die von einem Trainer angeleitet werden, der das Spiel als Ausgangspunkt für Diskussionen über Kreislaufwirtschaft und Abfallmanagement nutzt.

In der Praxis dienen die funktionalen Spezifikationen als Brücke zwischen dem konzeptionellen und pädagogischen Design von RockChain und seiner technischen Umsetzung. Sie übersetzen die in früheren Arbeitspaketen identifizierten Anforderungen in beobachtbare Verhaltensweisen und Regeln, die als gemeinsame Referenz für Entwickler, Trainer und Entscheidungsträger dienen können: Was muss immer funktionieren, welche Informationen müssen auf dem Bildschirm verfügbar sein und unter welchen Bedingungen sollte das System von einem Zustand in einen anderen übergehen? Das Dokument bezieht sich auf die bestehende Architektur, die auf einem mobilen Client von React Native, Firebase-Diensten und einer Echtzeit-Kommunikationsschicht basiert, konzentriert sich jedoch weiterhin auf Rollen, Verantwortlichkeiten und das erwartete Systemverhalten. Auf diese Weise unterstützt WP4.A3 die Erstellung von Trainerrichtlinien in WP4.A4, die Paketierungs- und Bereitstellungsarbeiten in WP4.A5 und die Bewertungsaktivitäten in WP5 und stellt sicher, dass das RockChain-E-Learning-Tool über die Laufzeit des Projekts hinaus auf konsistente und reproduzierbare Weise gepflegt, wiederverwendet und angepasst werden kann.



## 2. SYSTEMÜBERSICHT UND ROLLEN

RockChain ist ein Multiplayer-Serious-Game, das erwachsenen Lernenden helfen soll, zu erkunden, wie Daten, Märkte und Validierungsmechanismen Strategien der Kreislaufwirtschaft im Werkstein- und Bausektor unterstützen können. In jeder Sitzung nimmt eine kleine Gruppe von Lernenden auf ihren Mobilgeräten an einem gemeinsamen Spiel teil und spielt eine oder mehrere Runden, in denen sie Produkte kaufen und umwandeln, Abfall erzeugen und reduzieren, Herausforderungen im Bergbau lösen und sehen, wie sich ihre Entscheidungen sowohl auf die wirtschaftlichen als auch auf die ökologischen Ergebnisse auswirken.

### 2.1. RockChain-Konzept und wichtigste Trainingsszenarien

Das Tool ist in erster Linie für den Einsatz in von Trainern geleiteten Sitzungen und nicht als eigenständige App gedacht. Typische Szenarien sind:

- Kurze interaktive Blöcke innerhalb eines Kurses oder Workshops (z. B. ein 45- bis 90-minütiger Abschnitt, in dem die Klasse ein oder zwei Spiele spielt und anschließend die Ergebnisse diskutiert).
- Demonstrationssitzungen, in denen ein Trainer RockChain verwendet, um Konzepte wie Wertschöpfungsketten, Abfallströme, Rückverfolgbarkeit und Anreize für Kreislaufpraktiken zu veranschaulichen.
- Gemischte Aktivitäten, bei denen die Lernenden zunächst theoretische Inputs erhalten (z. B. in WP3-Materialien oder anderen Modulen) und dann mit RockChain Entscheidungen ausprobieren und die Auswirkungen auf Abfall, Kosten und Belohnungen beobachten.

In allen Fällen soll RockChain nicht den Unterricht ersetzen, sondern als praktischer Motor für Erkundungen und Diskussionen dienen und eine gemeinsame Erfahrung bieten, die während der Nachbesprechung analysiert werden kann.

### 2.2. Zielgruppe und Rollen

Die funktionalen Spezifikationen berücksichtigen drei Haupttypen von Nutzern:

#### 2.2.1. Lernende (Teilnehmer)

- Erwachsene Teilnehmer an beruflicher Bildung und Erwachsenenbildung, viele von ihnen über 45 Jahre alt und mit unterschiedlichen digitalen Kompetenzen.
- Verwenden RockChain auf ihren eigenen Android- oder iOS-Geräten, um an einem Spiel teilzunehmen, während der Runden Entscheidungen zu treffen und deren Ergebnisse zu sehen.

- Interagieren nur mit dem mobilen Client: Einloggen, über einen Code an einem Spiel teilnehmen, zwischen den Bildschirmen „Markt“, „Mining“, „Recycling“, „Statistiken“ und „Profil“ navigieren sowie Feedback und Belohnungen erhalten.

### 2.2.2. Ausbilder/Moderatoren

- Lehrer, Ausbilder oder Berater, die für die Durchführung der Lernaktivität verantwortlich sind.
- Verwenden Sie RockChain, um Sitzungen einzurichten und zu verwalten: Erstellen oder hosten Sie Spiele, helfen Sie den Lernenden beim Beitritt, entscheiden Sie, wann Runden beginnen und enden, und lenken Sie die Diskussion auf der Grundlage der Ergebnisse.
- Kann auch grundlegende Indikatoren (Punktestände, Abfallmengen, Gewinnerbranchen, ausgewählte Strategien) überprüfen, um die Nachbesprechung und Bewertung zu unterstützen.
- Sie benötigen keine tiefgreifenden technischen Kenntnisse; ihre Interaktion erfolgt in erster Linie über dieselbe Client-Schnittstelle, möglicherweise mit einer Projektionsansicht oder einem geteilten Bildschirm.

### 2.2.3. Technische Administratoren

- Mitarbeiter in Partnerinstitutionen oder technischen Zentren, die das Backend (Firebase-Projekt, Echtzeitserver, Builds der mobilen App) bereitstellen und warten.
- Stellen sicher, dass Authentifizierung, Datenbank, Netzwerkkonfiguration und Updates für die Institutionen, die RockChain nutzen, korrekt eingerichtet sind.
- Nehmen in der Regel nicht an den täglichen Schulungen teil, bieten jedoch Second-Level-Support, wenn neue Versionen eingeführt werden oder Änderungen an der Infrastruktur erforderlich sind.

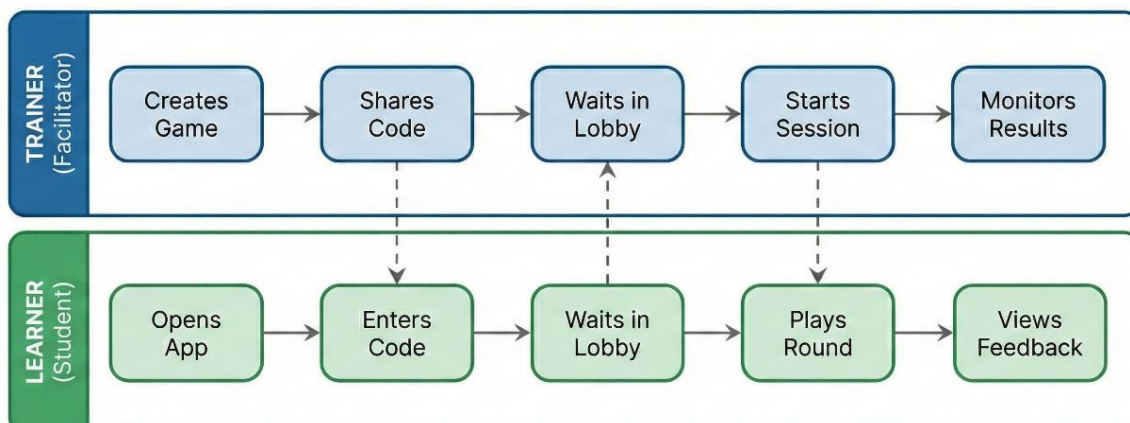


Abbildung1 : Geführte Sitzung durch den Trainer.

Das in diesem Dokument beschriebene funktionale Verhalten richtet sich hauptsächlich an Lernende und Trainer, während technische Administratoren als Enabler der Umgebung fungieren, in der das Tool ausgeführt wird.

## 2.3. Bereitstellungs- und Laufzeitumgebung

Funktional geht RockChain von einem gängigen Bereitstellungsmuster aus, das realistische Bedingungen in der beruflichen Bildung und Erwachsenenbildung widerspiegelt:

### 2.3.1. Geräte

- Die Lernenden verwenden Smartphones oder Tablets, auf denen die RockChain-App (Android oder iOS) läuft.
- Ausbilder können ebenfalls ein mobiles Gerät, ein Tablet oder einen Emulator auf einem Laptop verwenden und optional ihren Bildschirm auf die Anzeige im Klassenzimmer projizieren, um gemeinsame Ansichten (z. B. Warteraum, Ergebnisse am Ende einer Runde) anzuzeigen.

### 2.3.2. Konnektivität

- Die Sitzungen sollten mit einer stabilen Internetverbindung durchgeführt werden: in der Regel über das WLAN des Klassenzimmers oder institutionelle Netzwerke, aber auch mobile Daten sind möglich.
- Das System muss kurze Unterbrechungen (z. B. WLAN-Störungen, Ruhemodus des Geräts) tolerieren und es den Kunden ermöglichen, sich wieder zu verbinden und dem aktuellen Spiel beizutreten, ohne ihre Position zu verlieren.
- Eine vollständig offline Nutzung wird nicht unterstützt, da RockChain auf einem in der Cloud gehosteten Backend basiert.

### 2.3.3. Umfang und Struktur der Sitzungen

- Die Spiele sind für kleine Gruppen konzipiert (z. B. 3–10 Lernende pro Spiel), wobei bei Bedarf mehrere Spiele parallel in einer einzigen Klasse durchgeführt werden können.
- Die Ausbilder planen die Sitzungen so, dass genügend Zeit für Folgendes bleibt: Anmeldung und Einweisung, Spielen eines oder mehrerer Spiele und eine Nachbesprechung, in der die Ergebnisse des Spiels mit den Kursinhalten verknüpft werden.

Diese Annahmen zu Geräten, Konnektivität und Einsatz im Unterricht bilden den Rahmen für die in den nächsten Abschnitten beschriebenen funktionalen Anforderungen: Sie legen fest, was das System leisten muss, um reibungslose, von Trainern geleitete Sitzungen in typischen Berufsbildungs- und Erwachsenenbildungsumgebungen zu unterstützen.

### 3. FUNKTIONELLE ARCHITEKTUR UND MODULE

Das RockChain-E-Learning-Tool folgt einer Client-Server-Architektur, bei der eine mobile Anwendung die Benutzeroberfläche für Lernende und Trainer bereitstellt, während eine Reihe von Backend-Diensten die Authentifizierung, die Spielelogik, die Persistenz und die Echtzeitsynchronisation verwalten. WP4-A1 hat diese Architektur bereits technisch ausführlich beschrieben; WP4-A3 verwendet dieselbe Struktur und konzentriert sich auf die funktionalen Aufgaben jedes Teils: Was muss der Client leisten, was muss das Backend gewährleisten und wie arbeiten beide zusammen, um eine kohärente Lernerfahrung zu ermöglichen?

#### 3.1. Hochrangige Client-Server-Architektur

Auf hoher Ebene besteht RockChain aus drei zusammenarbeitenden Schichten:

- Ein mobiler React Native-Client (erstellt mit Expo), der auf Android- und iOS-Geräten läuft. Dieser Client ist die einzige Komponente, die direkt von Lernenden und Trainern verwendet wird. Er ist für die Benutzerinteraktion, die lokale Statusverwaltung während einer Sitzung und die visuelle Darstellung von Spielständen, Feedback und Ergebnissen zuständig.
- Ein Firebase-Backend, das Authentifizierung, persistenten Speicher und kontrollierte Geschäftslogik bereitstellt:
  - o Firebase Authentication identifiziert Benutzer und vergibt Anmeldedaten.
  - o Cloud Firestore speichert Spiele, Benutzer, Markt-Snapshots, Mining-Ereignisse und lernbezogene Daten, die über mehrere Sitzungen hinweg erhalten bleiben müssen.
  - o Firebase Functions implementiert serverseitige Vorgänge, die eine Validierung und einen vertrauenswürdigen Zugriff erfordern (z. B. das Erstellen von Spielen, das Beitreten zu Spielen, das Aktualisieren des Rundenstatus, das Berechnen von Belohnungen oder das Aktualisieren des Benutzerfortschritts).
- Ein autoritativer Echtzeitserver, der Runden und schnelle Interaktionen zwischen Spielern mithilfe von WebSockets koordiniert. Dieser Server verwaltet eine In-Memory-Ansicht jedes aktiven Spiels (Spieler, Inventare, Timer, Mining-Probleme) und sendet autoritative Ereignisse (Rundenbeginn, Rundenende, Marktaktualisierungen, Mining-Probleme) an alle verbundenen Clients.

Funktional versucht der mobile Client niemals, allein zu entscheiden, wie der Spielstatus sein sollte. Er reagiert auf autoritative Informationen, die vom Backend und vom

Echtzeitserver kommen, und sendet Benutzeraktionen als Anfragen (über HTTPS oder WebSockets), die der Server validiert und anwendet, bevor dauerhafte Änderungen vorgenommen werden.

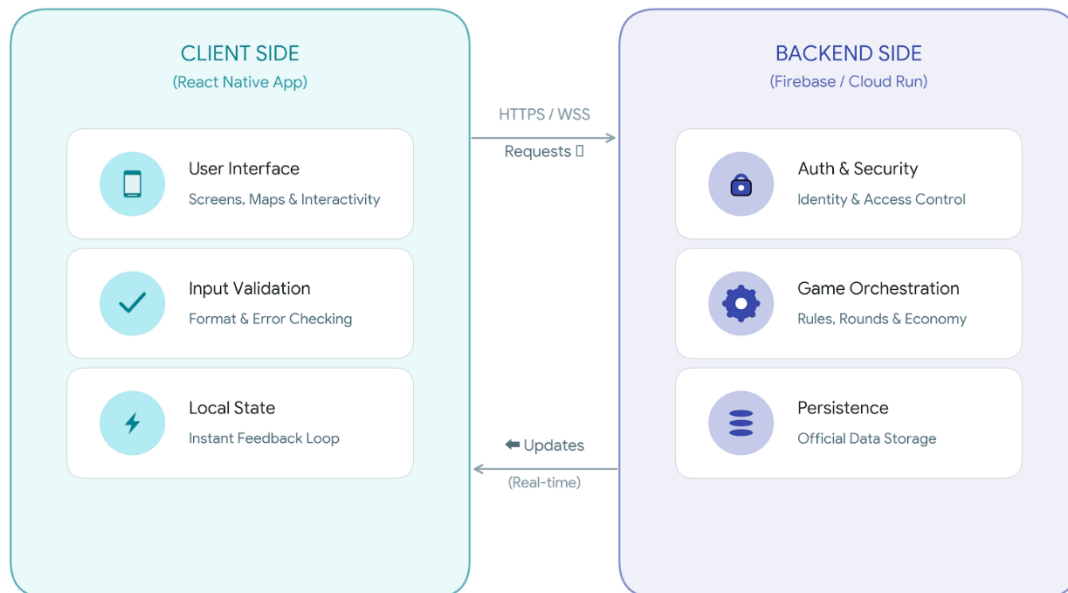


Abbildung2 : Architekturdiagramm

### 3.2. Frontend-Module

Auf der Client-Seite ist das Verhalten in eine Reihe von Modulen und Bildschirmen unterteilt, die alle durch gemeinsame Statuskontexte (für Spieldaten, Mining und Lernfortschritt) unterstützt werden:

**Authentifizierung und Onboarding (Anmeldung/Registrierung):** Behandelt die Anmeldung und Registrierung von Benutzern über Firebase Authentication, die Erstellung eines Basisprofils und erste Überprüfungen (z. B. ob der Benutzer ein Spiel hostet oder einem bestehenden Spiel beitrifft). Aus funktionaler Sicht muss dieses Modul den Benutzer immer dazu führen, entweder ein neues Spiel zu erstellen oder einen gültigen Spielcode einzugeben.

**Warteraum (Lobby):** Nach dem Erstellen oder Beitreten zu einem Spiel werden alle Spieler in den Warteraum geleitet.

- Zeigt an, wer beigetreten ist und welchen Status die Spieler haben (wartet/bereit).
- zeigt an, wer als Gastgeber oder Trainer fungiert
- ermöglicht es dem Gastgeber, das Spiel zu starten, wenn die Gruppe vollständig ist

- zeigt einen Countdown oder eine Meldung zum Start der ersten Runde an. Die Hauptfunktion besteht darin, die Gruppe vor Beginn des Spiels zusammenzuhalten.

GameTabs-Container und gemeinsamer Header: Sobald das Spiel begonnen hat, gelangt der Benutzer in den Hauptbereich des Spiels, der aus einer Reihe von Registerkarten (Markt, Bergbau, Recycling, Statistiken, Profil, Chat) besteht, die von einem gemeinsamen Header umgeben sind. Der Header zeigt die wichtigsten Indikatoren (Rundenanzahl, verbleibende Zeit, wichtige Ressourcen) in allen Registerkarten einheitlich an. Die Spezifikationen der Funktionen verlangen, dass diese Kopfzeile immer den maßgeblichen Status aus dem Backend widerspiegelt, damit sich die Lernenden bei ihren Entscheidungen darauf verlassen können.

Marktbildschirm: Zeigt die in der aktuellen Runde verfügbaren Produkte, ihre Preise, Auswirkungen auf den Abfall und andere Attribute an. Hier können die Spieler Gegenstände kaufen oder verkaufen, vorbehaltlich der vom Backend durchgesetzten Regeln. Funktional muss dieser Bildschirm:

- immer die Marktconfiguration der aktuellen Runde widerspiegeln
- ungültige Vorgänge verhindern (z. B. mehr RockCoins ausgeben als verfügbar sind)
- die lokale Ansicht bei erfolgreichen Transaktionen umgehend aktualisieren

MiningScreen: Zeigt Mining-Aufgaben (einfache arithmetische oder logische Aufgaben) an, die mit bestimmten Spielereignissen verknüpft sind, und ermöglicht es den Spielern, unter Zeitdruck Antworten einzureichen. Der Bildschirm muss:

- aktive Probleme und deren Zeitlimits anzeigen
- die Übermittlung deaktivieren, wenn ein Problem nicht mehr aktiv ist
- deutlich anzeigen, wenn ein Gewinner für einen Block ermittelt wurde

RecycleScreen: Ermöglicht es den Spielern, Teile ihres Inventars oder Abfalls in wertvollere Zustände umzuwandeln oder angesammelten Abfall gemäß den Spielregeln zu reduzieren. Seine funktionale Rolle besteht darin, zirkuläre Strategien sichtbar und umsetzbar zu machen und aktualisierte Abfall- und Ressourcenwerte in den Gesamtzustand des Spiels zurückzuführen.

StatsScreen: Bietet einen Überblick über die Leistung: Punktestände, Abfallmengen, Ranglisten und andere Indikatoren pro Runde oder pro Spiel. Es ist ein wichtiges Modul für die Nachbesprechung innerhalb der Sitzung und hilft sowohl Lernenden als auch Trainern, zu interpretieren, was passiert ist und welche Strategien effektiver waren.

ProfileScreen und Einstellungen: Ermöglicht es Benutzern, ihr Profil (Name, Avatar, Spracheinstellungen) anzuzeigen und anzupassen und eine Zusammenfassung ihrer bisherigen Aktivitäten oder Erfolge im aktuellen Spiel anzuzeigen. Es dient auch als Einstiegspunkt für grundlegende Barrierefreiheits- und Lokalisierungsoptionen.

ChatScreen und Zusatzkomponenten: Bietet einen einfachen Chat-Kanal, über den sich die Spieler während des Spiels abstimmen können. Zusatzkomponenten (wie ein Netzwerkstatus-Banner oder Tutorial-Overlays) informieren die Benutzer über Verbindungsprobleme, bieten kontextbezogene Hilfe und unterstützen eine reibungslosere Einarbeitung.

Zusammen definieren diese Module, was Lernende und Trainer zu jedem Zeitpunkt sehen und tun können. Die funktionalen Spezifikationen stellen sicher, dass jeder Bildschirm auf vorhersehbare Weise auf Änderungen im Backend-Status und auf Benutzeraktionen reagiert.

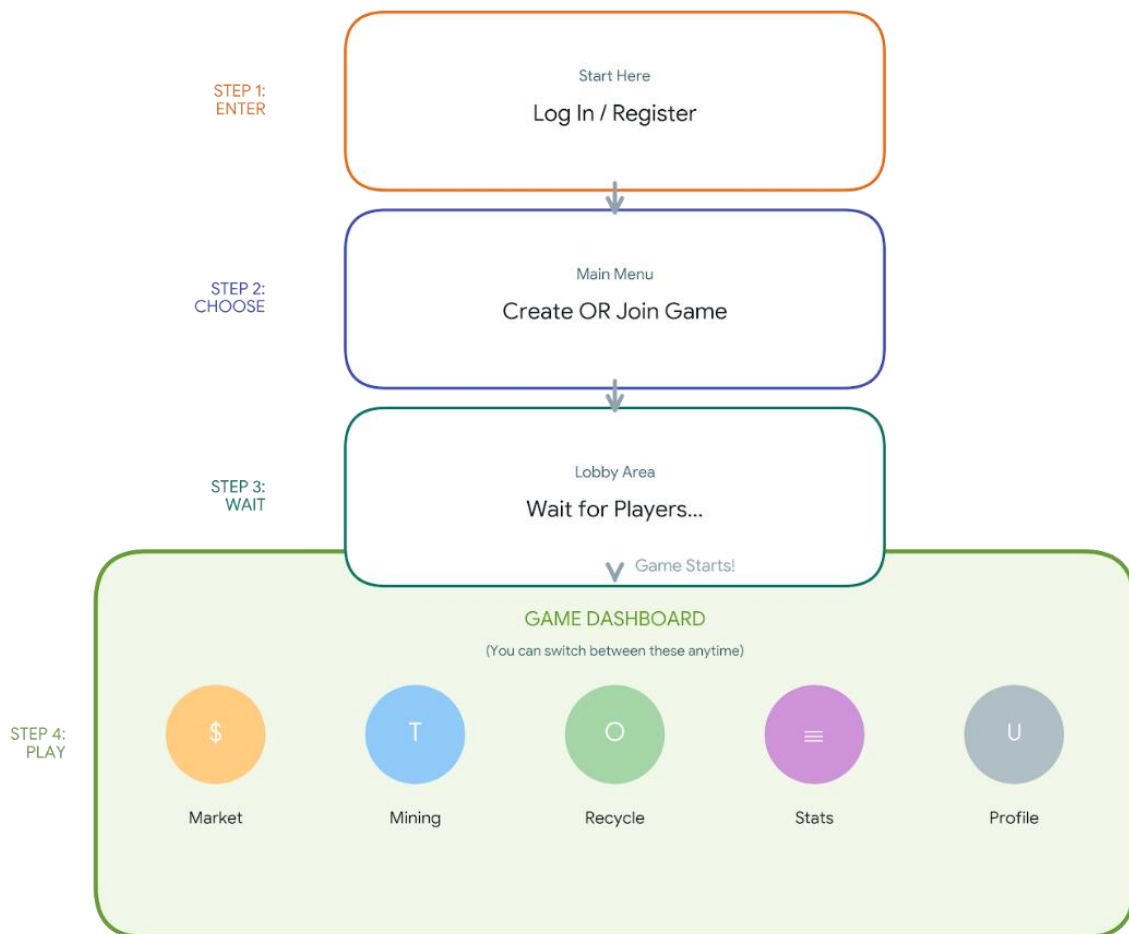


Abbildung3 : Diagramm der Frontend-Module

### 3.3. Backend-Module

Auf der Backend-Seite sind die Funktionen in mehrere logische Module unterteilt, die über Firebase-Funktionen, Firestore-Regeln und den Echtzeitserver implementiert sind:

- Benutzer- und Profilverwaltung: Behandelt die Registrierung, Authentifizierung und Speicherung grundlegender Benutzerinformationen in Firestore. Funktional muss dieses Modul Folgendes sicherstellen:
  - Jeder authentifizierte Benutzer über eine eindeutige und stabile Kennung verfügt.
  - Profilaktualisierungen validiert und konsistent sind.
  - Und nur autorisierte Benutzer können auf ihre eigenen Daten zugreifen oder diese ändern.
- Spiel- und Rundenkoordination: Deckt den Lebenszyklus von Spielen und Runden ab: Erstellen und Löschen von Spielen, Beitreten und Verlassen, Starten und Beenden von Runden sowie die Koordination zur Bestimmung des Spielendes. Dieses Modul definiert die gültigen Zustandsübergänge und garantiert, dass:
  - Spielcodes eindeutig und auflösbar sind.
  - Nicht mehr als die zulässige Anzahl von Spielern an einem Spiel teilnehmen.
  - Statusänderungen (Warten, läuft, Rundenende, beendet) für alle Spieler konsistent angewendet werden.
- Markt- und Ressourcenlogik: Verwaltet die Generierung von Marktconfigurationen pro Runde, Preis- und Abfallregeln sowie die Validierung von Transaktionen. Aus funktionaler Sicht stellt dieses Modul sicher, dass:
  - Der Markt jeder Runde vordefinierten Regeln und Zufallsbeschränkungen folgt.
  - Die Bestände und Salden der Spieler werden atomar und konsistent aktualisiert.
  - Die resultierenden Daten können zur Analyse zurückverfolgt werden (z. B. durch Snapshots oder Protokolle).
- Mining- und Validierungslogik: Generiert Mining-Probleme, zeichnet Antworten auf und ermittelt Gewinner gemäß den Spielregeln. Dieses Modul muss:
  - sicherstellen, dass Probleme mit einem bestimmten Spielkontext verknüpft sind.



- Versuche auf eine Weise aufzeichnen, die überprüft werden kann.
- Bei der Vergabe von Belohnungen faire Kriterien anwenden (z. B. erste richtige Antwort).
- Echtzeit-Koordinationsschicht: Implementiert den WebSocket-basierten Server, der den aktiven Spielstatus im Speicher hält und maßgebliche Ereignisse überträgt. Zu seinen funktionalen Aufgaben gehören:
  - Alle verbundenen Spieler hinsichtlich Rundenzustand, Timern und wichtigen Aktualisierungen synchron halten.
  - Behandlung von Verbindungsabbrüchen und Wiederverbindungen, damit Spieler nach Möglichkeit wieder an laufenden Runden teilnehmen können.
  - Interaktion mit Firestore und Funktionen an genau definierten Punkten (z. B. beim Speichern der Ergebnisse am Ende einer Runde).
- Lernfortschritt und Protokollierung: Zeichnet relevante Ereignisse und Zusammenfassungen auf, die für die pädagogische Überwachung und spätere Bewertung in WP5 erforderlich sind. Dieses Modul definiert, welche Aktionen und Ergebnisse (z. B. geförderte Blöcke, reduzierter Abfall, Wahl der Branche) als Lernindikatoren gespeichert werden und wie sie auf Benutzer-, Spiel- oder Kohortenebene abgerufen und aggregiert werden können.

Durch die klare Zuordnung dieser Verantwortlichkeiten zu Frontend- und Backend-Modulen stellt die funktionale Architektur sicher, dass RockChain sich auf vorhersehbare, testbare und erweiterbare Weise verhält. Der nächste Abschnitt baut auf dieser Struktur auf, um die zentralen funktionalen Anforderungen genauer zu definieren.

## 4. DETAILIERTE FUNKTIONELLE ANFORDERUNGEN

Die folgenden Anforderungen beschreiben, was das RockChain-E-Learning-Tool aus Sicht der Lernenden, der Ausbilder und des Ausbildungskontexts leisten muss. Sie bauen auf der oben skizzierten Architektur auf, werden jedoch eher in Form von beobachtbarem Verhalten und Systemregeln als in Form von Implementierungsdetails ausgedrückt.

### 4.1. Benutzerverwaltung und Authentifizierung

Das System muss Benutzer über mehrere Sitzungen hinweg konsistent identifizieren und ihre grundlegenden Informationen und ihre Teilnahme am Spiel in einer stabilen Form speichern.

Funktional bedeutet dies, dass das System:

- Benutzer müssen sich vor dem Zugriff auf spielbezogene Funktionen authentifizieren.
- Für jeden authentifizierten Benutzer eine eindeutige, stabile Benutzerkennung und ein Basisprofil (z. B. Anzeigename, Sprachpräferenz) erstellen und pflegen.
- Benutzern die Möglichkeit geben, ausgewählte Profildfelder zu aktualisieren, wobei die Verknüpfungen zu früheren Spielen und Lerndaten erhalten bleiben.
- den Zugriff so einschränken, dass jeder Benutzer nur sein eigenes Profil und seinen Status im Spiel lesen oder ändern kann, mit Ausnahme von Informationen, die ausdrücklich innerhalb eines Spiels geteilt werden (z. B. öffentliche Ranglisten).

### 4.2. Erstellung von Spielen, Beitritt und Lobby-Verwaltung

Das System muss Trainern oder designierten Hosts die Erstellung von Spielen und anderen Benutzern die einfache Teilnahme daran ermöglichen und sicherstellen, dass alle Teilnehmer vor Spielbeginn korrekt registriert sind.

Das System muss:

- Benutzern ermöglichen, ein neues Spiel zu erstellen und automatisch einen eindeutigen Spielcode und eine eindeutige Kennung zuzuweisen.
- sicherstellen, dass ein neu erstelltes Spiel mit Standardparametern (Rundenanzahl, Timer, maximale Spieleranzahl, Anfangsressourcen) initialisiert wird.

- Benutzern ermöglichen, einem bestehenden Spiel beizutreten, indem sie einen gültigen Spielcode eingeben, mit klarer Rückmeldung, wenn der Code falsch ist oder das Spiel nicht beigetreten werden kann.
- Beschränkungen für die Anzahl der Spieler pro Spiel durchsetzen und verhindern, dass Spieler zu spät beitreten, sobald das Spiel einen vereinbarten Punkt überschritten hat.
- Stellen Sie einen Warteraum (Lobby) bereit, in dem alle teilnehmenden Spieler und ihr Bereitschaftsstatus angezeigt werden und in dem der Gastgeber das Spiel starten kann, sobald alle Bedingungen erfüllt sind.

### 4.3. Lebenszyklus von Spiel und Runde

Sobald ein Spiel erstellt wurde, muss das System dessen Lebenszyklus auf kontrollierte Weise verwalten und zwischen klar definierten Zuständen wechseln, die auf allen Clients konsistent widergespiegelt werden.

Das System muss:

- Für jedes Spiel einen eindeutigen Spielstatus (z. B. Wartet, läuft, beendet) und einen Rundenstatus (z. B. Nicht gestartet, aktiv, wird ausgewertet, abgeschlossen) pflegen.
- Nur gültige Übergänge zwischen diesen Zuständen zulassen, die durch autorisierte Aktionen ausgelöst werden (z. B. Start des Spiels durch den Gastgeber, Ablauf des Rudentimers, explizites Spielende).
- Jede Runde durch Übertragung der erforderlichen Informationen (Rundennummer, verbleibende Zeit, anfängliche Markt- und Mining-Konfiguration) an alle Spieler starten.
- Beenden Sie jede Runde auf der Grundlage verbindlicher Bedingungen (Timer, Abschluss der erforderlichen Schritte) und stellen Sie sicher, dass alle Clients einen eindeutigen Hinweis darauf erhalten, dass das Spiel beendet ist.
- Markieren Sie das Spiel als beendet, wenn die konfigurierte Anzahl von Runden abgeschlossen ist oder wenn der Host die Sitzung ausdrücklich beendet, und verhindern Sie danach weitere Aktionen in diesem Spiel.

#### 4.4. Markt, Ressourcen und Kreislaufwirtschaftslogik

Die Markt- und Ressourcenmechanismen von RockChain müssen den beabsichtigten Lernschwerpunkt auf Abfall, Wert und Kreislaufstrategien widerspiegeln.

Das System muss:

- Für jede Runde eine Reihe verfügbarer Produkte mit zugehörigen Attributen (Typ, Material, Qualität, Preis, Auswirkungen auf den Abfall) gemäß vordefinierten Regeln generieren.
- Jedem Spieler eine Marktansicht präsentieren, die den maßgeblichen Stand für diese Runde widerspiegelt, und diese aktualisieren, wenn Produkte gekauft oder verkauft werden.
- Alle Transaktionen anhand der aktuellen Ressourcen des Spielers (z. B. Rockcoins, Inventar) validieren und Aktionen verhindern, die gegen Beschränkungen verstoßen würden.
- Aktualisieren Sie die Bestände, Salden und Abfallmengen der Spieler nach jeder gültigen Transaktion konsistent.
- Stellen Sie sicher, dass die Marktkonfiguration und die daraus resultierenden Zustände anschließend für Analysezwecke rekonstruiert werden können (z. B. durch gespeicherte Momentaufnahmen oder abgeleitete Daten).

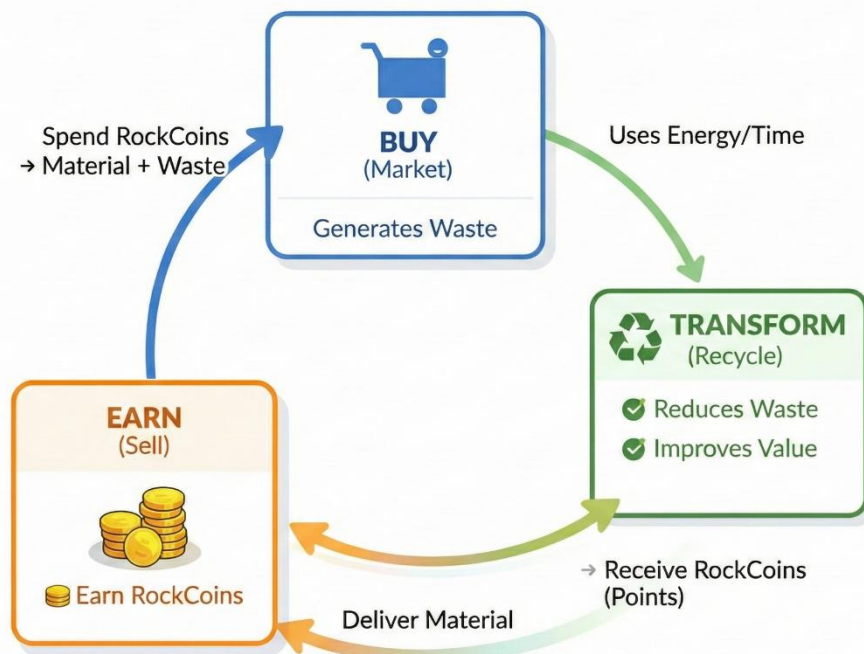


Abbildung4 : Kreislaufwirtschaft in Rockchain.

## 4.5. Herausforderungen beim Mining und Validierung von Lösungen

Mining-Events bieten zeitlich begrenzte Herausforderungen, die Entscheidungen und Validierungsmechanismen mit der Blockchain-inspirierten Erzählung des Spiels verbinden.

Das System soll:

- Mining-Probleme generieren, die eindeutig mit einem Spiel und einem Rundenkontext verbunden sind, und diese nur verfügbar machen, solange sie aktiv sind.
- Alle relevanten Spieler benachrichtigen, wenn ein neues Mining-Problem verfügbar wird, und angeben, wie lange es offen bleibt.
- Die Antworten der Spieler akzeptieren und aufzeichnen, solange das Problem aktiv ist, und verspätete Einsendungen ablehnen, sobald es geschlossen oder gelöst wurde.
- Die Gewinner nach klar definierten Regeln (z. B. erste richtige Antwort) ermitteln und sicherstellen, dass pro Problem nur ein Gewinner erfasst wird.
- den Spielstatus aktualisieren (z. B. Mining-Belohnungen, Verlauf gelöster Probleme) und alle Spieler auf transparente Weise über das Ergebnis informieren.

## 4.6. Punktwertung, Belohnungen und Branchenauswahl

Die Bewertung und die Belohnungen müssen den Lernenden helfen, die Auswirkungen ihrer Entscheidungen sowohl auf die wirtschaftlichen Ergebnisse als auch auf die Abfall-/Kreislaufindikatoren zu verstehen.

Das System muss:

- Für jeden Spieler eine aktuelle Punktzahl und wichtige Indikatoren wie angesammelte Abfälle und Ressourcen führen.
- den Spielern ermöglichen, eine Branche oder Strategie auszuwählen oder zu bestätigen, sofern relevant, und diese Informationen bei der Berechnung der Belohnungen am Ende jeder Runde zu verwenden.
- die Belohnungen pro Runde auf der Grundlage transparenter Regeln berechnen, die Marktverhalten, Bergbau-Ergebnisse und Entscheidungen zur Kreislaufwirtschaft kombinieren.

- diese Belohnungen atomar auf den Status der Spieler anwenden (Aktualisierung von Punkteständen, Abfallmengen und Beständen) und eine Zusammenfassung der Rundenergebnisse pro Spieler speichern.
- Präsentieren Sie Zusammenfassungen am Ende jeder Runde und am Ende des Spiels, aus denen klar hervorgeht, wie die Punktestände und Indikatoren zustande gekommen sind, um die Nachbesprechung und Diskussion zu unterstützen.

#### 4.7. Tracking, Protokollierung und Daten, die für die WP5-Bewertung benötigt werden

Schließlich muss das System unter Einhaltung der Datenschutzgrundsätze genügend Informationen aufzeichnen, um die pädagogische Bewertung in WP5 und zukünftige Analysen zu unterstützen.

Das System muss:

- wichtige Ereignisse und Zustände aufzeichnen, die für die Lernanalyse relevant sind, wie z. B. gekaufte und verkaufte Produkte, erzeugter und reduzierter Abfall, Bergbauversuche und -erfolge sowie die Wahl der Branche oder Strategie.
- diese Aufzeichnungen mit pseudonymen Benutzerkennungen und Spielkennungen verknüpfen, damit Muster auf Spieler-, Spiel- und Kohortenebene analysiert werden können.
- eine Möglichkeit bieten, diese Informationen (z. B. durch spezielle Sammlungen oder Momentaufnahmen) für Partner, die für die Bewertung und Berichterstattung zuständig sind, zu extrahieren oder zu aggregieren.
- Technische Ereignisse (z. B. Verbindungsabbrüche, Wiederverbindungen, Fehler) so protokollieren, dass Anomalien im Spielverlauf interpretiert und kontinuierliche Verbesserungen unterstützt werden können.
- Beachten Sie die im Projekt vereinbarten Richtlinien zur Datenminimierung und -aufbewahrung und stellen Sie sicher, dass Protokolle und Lerndaten nur so lange gespeichert werden, wie es für die Bewertung und Berichterstattung erforderlich ist.

Diese zentralen funktionalen Anforderungen definieren, was RockChain bei jeder Bereitstellung zuverlässig leisten muss, und dienen als gemeinsame Referenz für die detaillierteren Abläufe und nicht-funktionalen Einschränkungen, die in den folgenden Abschnitten beschrieben werden.

## 5. SPIELABLÄUFE UND ZUSTANDSÜBERGÄNGE

Das funktionale Verhalten von RockChain basiert auf einer kleinen Anzahl wiederkehrender Abläufe, die sich in jeder Sitzung wiederholen. Diese Abläufe bestimmen, wie ein Spiel beginnt, wie die Runden verlaufen, was passiert, wenn die Zeit abgelaufen ist, und wie das System reagiert, wenn Spieler die Verbindung trennen und wiederherstellen. Sie bilden das Rückgrat, das Trainer und Lernende während der Nutzung erleben, und müssen über alle Einsätze hinweg vorhersehbar bleiben.

### 5.1. Gesamtlebenszyklus des Spiels

Aus funktionaler Sicht folgt jedes Spiel einem gemeinsamen Lebenszyklus:

#### **Spielaufbau**

- Ein neues Spiel wird beim Einloggen erstellt, wobei ein Spielcode vergeben wird.
- Das Spiel wird mit Standardparametern (maximale Spieleranzahl, Anzahl der Runden, Rundendauer, Startressourcen) initialisiert und als „wartend“ markiert.

#### **Teilnahme- und Lobby-Phase**

- Die Lernenden authentifizieren sich und geben den Spielcode ein, um derselben Sitzung beizutreten.
- Das System registriert jeden Spieler im Spiel, zeigt ihn im Warteraum an und verfolgt seinen Bereitschaftsstatus.
- Der Gastgeber überwacht, wer beigetreten ist, und entscheidet, wann die Gruppe vollständig ist.

#### **Spiel läuft**

- Wenn der Gastgeber das Spiel startet, ändert sich der Status zu „im Gange“ und die erste Runde wird initialisiert.
- Die Spieler werden in den Spielbereich (Registerkarten für Markt, Bergbau, Recycling, Statistiken usw.) geführt, während eine gemeinsame Kopfzeile die aktuelle Runde und die verbleibende Zeit anzeigt.
- Das Spiel verläuft über eine feste Anzahl von Runden, die jeweils dem unten beschriebenen Rundenzyklus folgen.

#### **Ende des Spiels und Nachbesprechung**

- Nach der konfigurierten Anzahl von Runden oder wenn der Gastgeber beschließt, das Spiel vorzeitig zu beenden, markiert das System das Spiel als beendet.
- Eine abschließende Zusammenfassung zeigt die Punktestände, Abfallindikatoren und wichtigsten Ergebnisse für alle Spieler an.
- Trainer können diese Zusammenfassung zusammen mit den gespeicherten Daten als Grundlage für Diskussionen und Reflexionen verwenden.

Sobald ein Spiel beendet ist, sind in dieser Sitzung keine weiteren Aktionen mehr möglich; die Lernenden können das Spiel verlassen oder einem neuen Spiel beitreten, wenn der Trainer beschließt, ein weiteres Spiel zu starten.

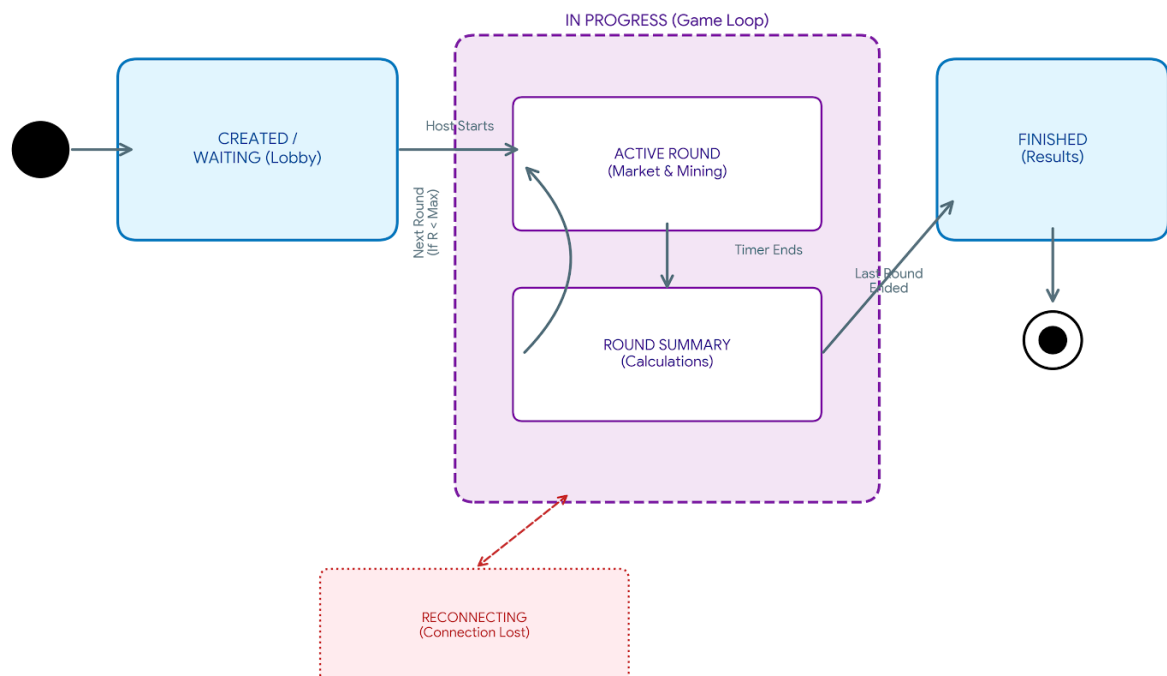


Abbildung5 : Zustandsdiagramm

## 5.2. Lebenszyklus einer Runde

Innerhalb des gesamten Spiels hat jede Runde ihren eigenen internen Lebenszyklus. Das System verwaltet diesen Lebenszyklus so, dass alle Spieler gleichzeitig die gleichen Phasen durchlaufen:

### Rundeninitialisierung

- Zu Beginn einer Runde definiert das Backend den maßgeblichen Zustand: Rundennummer, Endzeit, Marktkonfiguration und etwaige anfängliche Mining-Kontexte.



- Diese Informationen werden an alle verbundenen Clients übertragen, die ihre Ansichten entsprechend aktualisieren. Der Header startet den Countdown.

### Aktive Spielphase

- Während des aktiven Spiels können die Spieler:
  - Mit dem Markt interagieren (Produkte kaufen/verkaufen).
  - Mining-Probleme lösen, sofern verfügbar.
  - Nutzen Sie Recycling-Optionen.
  - Konsultieren Sie Statistiken oder chatten Sie.
- Alle Aktionen werden als Anfragen an das Backend gesendet; nur validierte Vorgänge werden angewendet und anderen Spielern angezeigt.

### Auflösungsphase

- Wenn der maßgebliche Timer abläuft oder wenn das System entscheidet, dass alle notwendigen Bedingungen erfüllt sind, geht die Runde in einen Lösungszustand über.
- Neue Aktionen werden blockiert oder eingeschränkt. Das Backend berechnet:
  - Gewinner und Belohnungen beim Mining.
  - Branchenbezogene Boni.
  - Änderungen bei Punkteständen, Beständen und Abfallmengen.
- Ein konsistenter Satz von Ergebnissen wird in den persistenten Speicher geschrieben und an alle Clients übertragen.

### Feedback am Ende der Runde

- Die Spieler sehen eine Zusammenfassung am Ende der Runde: ihre aktualisierten Ressourcen, Abfallmengen, erzielten Belohnungen und alle Gewinnerbranchen.
- Trainer können diese Ansicht nutzen, um Strategien zu kommentieren und sie mit Konzepten der Kreislaufwirtschaft zu verknüpfen, bevor sie die nächste Runde beginnen.

### Übergang zur nächsten Runde

- Wenn noch weitere Runden verbleiben, löst der Gastgeber oder das System die nächste Runde aus.
- Der Zyklus aus Initialisierung, aktivem Spiel, Auswertung und Feedback wiederholt sich, bis das Spiel beendet ist.

Dieser Rundenzyklus stellt sicher, dass alle Lernenden in jeder Gruppe einen klaren Rhythmus erleben: *vorbereiten* → *handeln* → *Konsequenzen sehen* → *reflektieren*.

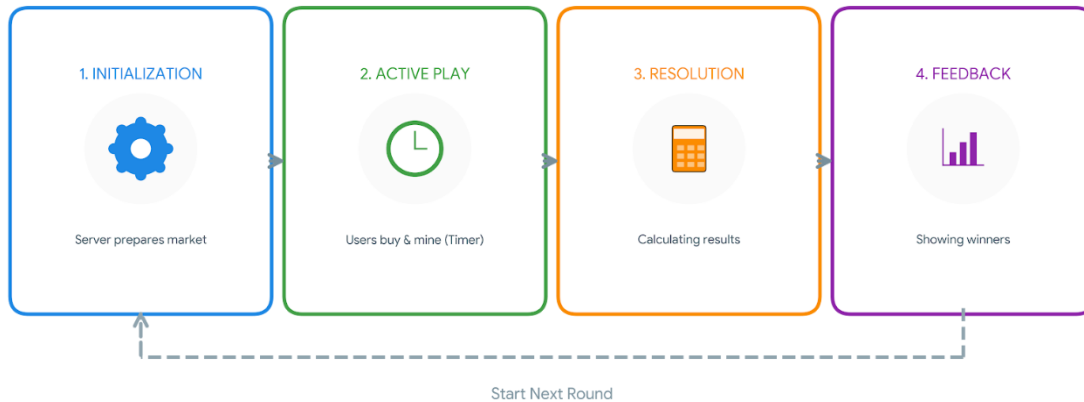


Abbildung6 : Rundenzyklus.

### 5.3. Spielerflüsse

In realen Trainingseinheiten kann es vorkommen, dass Teilnehmer zu spät kommen, vorübergehend die Verbindung verlieren oder versehentlich die App schließen. RockChain definiert Funktionsregeln, um diese Situationen reibungslos zu bewältigen:

#### Teilnahme vor Beginn

- Spieler können dem Spiel frei beitreten, solange es sich im Wartezustand befindet.
- Das System zeigt verspätete Teilnehmer in der Lobby an und aktualisiert die Liste für alle.
- Sobald der Gastgeber das Spiel startet, werden neue Teilnehmer gemäß den vereinbarten Regeln entweder verhindert oder eingeschränkt (z. B. keine Teilnahme nach Beginn der ersten Runde).

#### Wiedereintritt während eines laufenden Spiels

- Wenn ein Spieler vorübergehend die Verbindung verliert oder die App schließt:
  - o Bei der erneuten Verbindung und Anmeldung verwendet das System seine Identität, um ihn nach Möglichkeit wieder mit dem aktuellen Spiel zu verbinden.

- Er wird zum entsprechenden Bildschirm und zur entsprechenden Runde zurückgebracht, wobei der aktuelle maßgebliche Status angewendet wird.
- Wenn eine erneute Verbindung nicht mehr möglich ist (z. B. weil das Spiel beendet ist), informiert das System den Benutzer darüber und bietet ihm an, zum Hauptmenü zurückzukehren.

#### **Verhalten bei Verbindungsabbrüchen**

- Während der Offline-Zeit kann der Benutzer keine Aktionen ausführen, die den Spielstatus verändern.
- Die Benutzeroberfläche zeigt möglicherweise den Status „eingeschränkt“ oder „Warten auf erneute Verbindung“ an. Sobald die Verbindung wiederhergestellt ist, wird der aktuelle Status abgerufen und die Ansicht aktualisiert.

#### **Freiwilliges Beenden**

- Ein Spieler, der das Spiel absichtlich verlässt (z. B. über eine Beenden-Option), wird als abwesend markiert.
- Seine bisherigen Beiträge bleiben im Spielverlauf erhalten, aber er nimmt nicht mehr an nachfolgenden Aktionen oder Belohnungen teil, es sei denn, er tritt gemäß den vom Trainer festgelegten Regeln wieder bei.

Diese Abläufe sollen Störungen im Unterricht reduzieren, wo kurze Verbindungsprobleme oder Geräteprobleme häufig vorkommen, und gleichzeitig den Spielstatus für alle Teilnehmer konsistent und fair halten.

## **5.4. Synchronisationsregeln zwischen Clients**

Da RockChain ein Echtzeit-Multiplayer-Tool ist, besteht ein zentraler Aspekt seines funktionalen Verhaltens darin, alle Clients auf eine einzige, maßgebliche Ansicht des Spiels abzustimmen. Die wichtigsten Synchronisationsregeln lauten:

#### **Maßgeblicher Backend-Status**

- Das Backend, nicht einzelne Geräte, entscheidet:
  - Welches Spiel und welche Runde aktiv sind.
  - Wann Runden beginnen und enden.
  - Welche Markt- und Mining-Konfigurationen gelten.
  - Wie die aktuellen Punktestände und Indikatoren aussehen.

- Die Clients hören diesen Status ab und passen ihre lokalen Ansichten entsprechend an.

### Gemeinsame Zeitreferenz

- Runden-Countdowns und zeitlich begrenzte Herausforderungen verwenden eine gemeinsame Referenzzeit, die vom Backend bereitgestellt wird.
- Clients können die verbleibende Zeit mit geringfügigen lokalen Anpassungen anzeigen, aber die Entscheidung, eine Runde zu beenden oder ein Mining-Problem zu schließen, richtet sich immer nach dem maßgeblichen Timer und nicht nach der Uhr des Geräts.

### Konsistente Aktualisierungen

- Wenn eine gültige Aktion durchgeführt wird (Kauf, Verkauf, erfolgreiches Mining, Recyclingvorgang), führt das Backend Folgendes durch:
  - Aktualisiert den Spielstatus.
  - speichert relevante Änderungen.
  - sendet den aktualisierten Status oder das Ereignis an alle Spieler.
- Die Clients aktualisieren ihre Schnittstellen erst nach Erhalt einer Bestätigung, um sicherzustellen, dass alle das gleiche Ergebnis sehen.

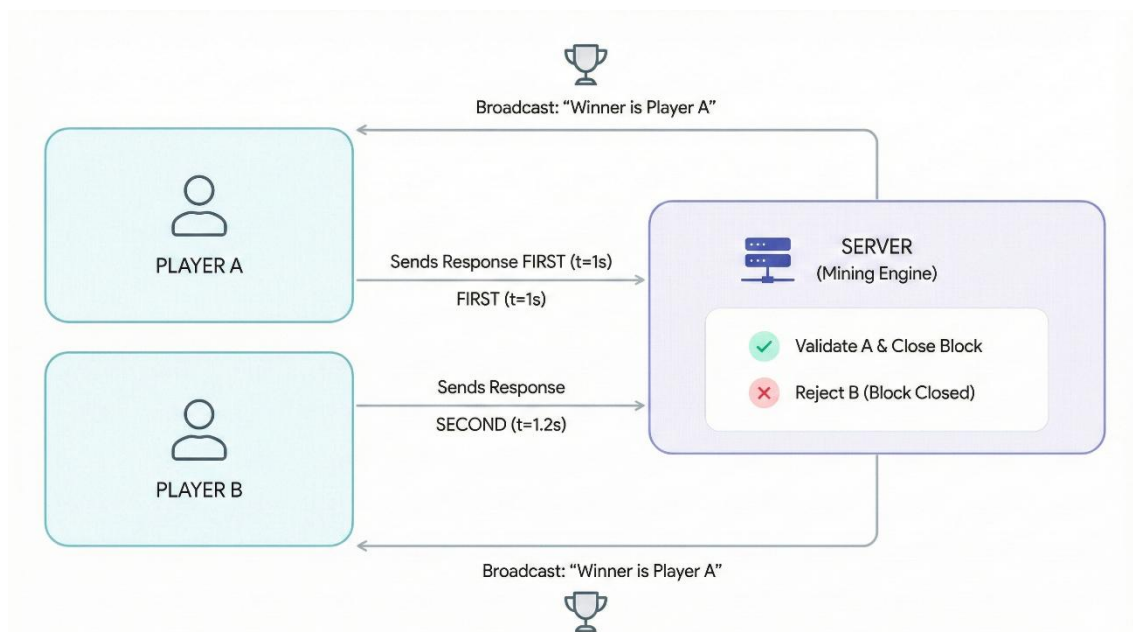


Abbildung 7: Mining-Antwort und Verbindung mit dem Server.

### Reibungslose Handhabung verspäteter oder widersprüchlicher Aktionen



- Aktionen, die nach Ende einer Runde oder nach Lösung eines Problems eintreffen, werden abgelehnt, wobei der Benutzer eine klare Rückmeldung erhält.
- Wenn zwei Aktionen miteinander in Konflikt stehen würden (z. B. zwei Spieler, die um dieselbe Mining-Belohnung wetteifern), wendet das Backend deterministische Regeln an (z. B. „erste korrekte Antwort“) und benachrichtigt alle Clients über das Endergebnis.

Diese Synchronisationsregeln stellen sicher, dass sich RockChain auch bei Netzwerkschwankungen wie ein einziges gemeinsames Spiel verhält und nicht wie eine Sammlung unabhängiger lokaler Simulationen. Für Trainer und Lernende bedeutet dies ein einheitliches Erlebnis, bei dem die Runden gemeinsam beginnen und enden, die Ergebnisse von allen Geräten bestätigt werden und die Diskussion nach jeder Runde auf einer stabilen und vertrauenswürdigen Darstellung des Geschehens basieren kann.

## 6. NICHT-FUNKTIONELLE ANFORDERUNGEN

Über die funktionalen Anforderungen an RockChain hinaus muss das Tool auch eine Reihe von nicht-funktionalen Anforderungen erfüllen, damit es in realen Berufsbildungs- und Erwachsenenbildungsumgebungen zuverlässig eingesetzt werden kann. Diese Anforderungen betreffen die Leistung, Zuverlässigkeit, Sicherheit und den Datenschutz sowie die Benutzerfreundlichkeit des Tools für erwachsene Lernende mit unterschiedlichen digitalen Kompetenzen.

### 6.1. Leistung und Skalierbarkeit in typischen Berufsbildungs-/Erwachsenenbildungsumgebungen

RockChain ist nicht für massive Online-Spiele gedacht, sondern für kleine bis mittelgroße Gruppen in Klassenzimmern oder Werkstätten. In diesem Zusammenhang muss das System:

- mindestens 2–5 gleichzeitige Spieler pro Spiel unterstützen, mit der Möglichkeit, mehrere Spiele parallel innerhalb einer einzigen Einrichtung auszuführen, ohne dass es zu einer merklichen Verschlechterung der Reaktionszeiten kommt.
- sicherstellen, dass wichtige Echtzeit-Interaktionen (Rundenstart, Marktaktualisierungen, Mining-Ereignisse) auf Client-Geräten innerhalb einer kurzen und vorhersehbaren Verzögerung (in der Größenordnung von Sekunden, nicht Zehntelsekunden) unter normalen Verbindungsbedingungen im Klassenzimmer angezeigt werden.
- Sorgen Sie für eine akzeptable Leistung, wenn mehrere Runden nacheinander gespielt werden, ohne dass zwischen den Spielen Neustarts oder manuelle Eingriffe erforderlich sind.
- Vermeiden Sie einen übermäßigen Verbrauch von Geräte-Ressourcen (CPU, Speicher, Akku), damit die App auf typischen Smartphones und Tablets der Mittelklasse, die von erwachsenen Lernenden verwendet werden, reibungslos läuft.

Diese Leistungserwartungen spiegeln die realistische Nutzung in VET/ADU-Kursen wider und stellen sicher, dass technische Einschränkungen die Lernerfahrung nicht beeinträchtigen.

## 6.2. Zuverlässigkeit und Fehlertoleranz unter Unterrichtsbedingungen

In realen Sitzungen sind vorübergehende Netzwerkstörungen, der Ruhezustand von Geräten oder kurze Verbindungsunterbrechungen häufig. RockChain muss robust genug sein, um diese Situationen zu bewältigen, ohne die Aktivität zu beeinträchtigen. Das System muss:

- kurze Verbindungsunterbrechungen tolerieren, indem es Clients ermöglicht, sich wieder zu verbinden und mit dem aktuellen Spiel und der aktuellen Runde zu synchronisieren, wann immer dies technisch möglich ist.
- Verhindern von Datenkorruption oder inkonsistenten Zuständen, wenn während kritischer Vorgänge (z. B. Berechnungen am Ende einer Runde, Vergabe von Belohnungen) Verbindungsabbrüche auftreten.
- Abstürze oder nicht behebbare Fehler im mobilen Client bei normaler Nutzung vermeiden und klare Meldungen ausgeben, wenn doch einmal ein kritischer Fehler auftritt.
- sicherstellen, dass Trainer eine geplante Sitzung (ein oder mehrere Spiele) abschließen können, ohne die Infrastruktur neu starten zu müssen, selbst wenn einzelne Spieler vorübergehende technische Probleme haben.

Das Ziel ist es, dass Zuverlässigkeitsprobleme kein Hindernis für Trainer darstellen, die während ihres Unterrichts möglicherweise nur über begrenzte Zeit und technischen Support verfügen.

## 6.3. Sicherheit, Datenschutz und Datensicherheit

Die Anforderungen an Sicherheit, Datenschutz und Datensicherheit ergänzen die bereits in WP4.A1 definierten technischen Sicherheitsvorkehrungen für das Backend und die Datenbank. Aus funktionaler Sicht muss das System:

- sicherstellen, dass nur authentifizierte Benutzer auf Spielsitzungen zugreifen können und dass Spielcodes in der Praxis nicht erratbar sind.
- den Zugriff auf persönliche und spielbezogene Daten so einschränken, dass jeder Benutzer nur Folgendes sehen kann:
  - o sein eigenes Profil und seinen Status im Spiel
  - o Informationen, die ausdrücklich innerhalb eines Spiels geteilt werden (z. B. Punktestände und Ranglisten, die für alle Spieler in diesem Spiel sichtbar sind).

- nur die für die Durchführung von Sitzungen und die Bewertung von Lernergebnissen erforderlichen Mindestdaten speichern und verarbeiten, wobei nach Möglichkeit pseudonyme Identifikatoren verwendet werden.
- Unterstützen Sie die Verpflichtungen des Projekts gemäß den Datenschutzbestimmungen (z. B. DSGVO), darunter:
  - o Die Möglichkeit, Anträge auf Zugriff, Berichtigung oder Löschung personenbezogener Daten zu erfüllen.
  - o Die Anwendung vereinbarter Aufbewahrungsfristen für Protokolle und Lerndaten.
  - o Sichere Handhabung von Anmeldedaten und Konfigurationsgeheimnissen.
- Schützen Sie die Kommunikation zwischen Client und Backend (z. B. durch die Verwendung verschlüsselter Kanäle) und wenden Sie geeignete Sicherheitsvorkehrungen an, um unbefugten Zugriff oder Manipulationen des Spielstatus zu verhindern.

Diese Anforderungen stellen sicher, dass RockChain in Bildungseinrichtungen eingesetzt werden kann, ohne die Privatsphäre der Lernenden oder die Richtlinien der Einrichtung zu beeinträchtigen.

## 6.4. Benutzerfreundlichkeit und Zugänglichkeit für erwachsene Lernende mit heterogenen digitalen Fähigkeiten

Schließlich muss das Tool für erwachsene Lernende, von denen viele über 45 Jahre alt sind und möglicherweise nicht mit mobilen Spielen oder komplexen Benutzeroberflächen vertraut sind, nutzbar sein. Das System muss

- eine einfache und konsistente Navigationsstruktur mit klaren Bezeichnungen und minimalen Schritten zwischen der Anmeldung, dem Beitritt zu einem Spiel und dem Spielstart aufweisen.
- eine leicht verständliche Sprache und visuelle Elemente verwenden, unnötige Fachbegriffe vermeiden und bei Bedarf kurze Erklärungen bereitstellen (z. B. für Timer, Belohnungen oder Branchenauswahl).
- wichtige Informationen (verbleibende Zeit, Rundenummer, Kernressourcen) an einer festen Stelle auf dem Bildschirm sichtbar halten, damit die Lernenden während der Entscheidungsfindung nicht danach suchen müssen.



- die Verwendung mehrerer Sprachen unterstützen, indem Trainer und Lernende ihre bevorzugte Sprache aus den im Projekt angebotenen Sprachen auswählen können und indem sichergestellt wird, dass wichtige Meldungen und Rückmeldungen in diesen Sprachen verfügbar sind.
- Verhalten Sie sich vorhersehbar, wenn Lernende Fehler machen oder länger brauchen, um zu handeln, geben Sie klares Feedback und vermeiden Sie plötzliche Änderungen, die Verwirrung oder Ängste auslösen könnten.

Durch die Erfüllung dieser Anforderungen an Benutzerfreundlichkeit und Barrierefreiheit kann RockChain in Kurse mit unterschiedlichen erwachsenen Zielgruppen integriert werden, wodurch der Bedarf an ständiger technischer Unterstützung reduziert wird und die Kursleiter sich auf die Moderation und Diskussion konzentrieren können, anstatt sich mit der Fehlerbehebung des Tools zu beschäftigen.



## 7. SCHLUSSFOLGERUNGEN UND NÄCHSTE SCHRITTE

WP4.A3 hat das E-Learning-Tool RockChain zu einem klar definierten funktionalen System konsolidiert, das direkt auf der in WP4.A1 gelieferten Datenebene und Backend-Architektur sowie den in WP4.A2 vorgenommenen Verfeinerungen aufbaut. Während sich frühere Aktivitäten darauf konzentrierten, RockChain technisch robust und pädagogisch nutzbar zu machen, erfasst dieses Ergebnis, was das Tool in der Praxis leisten soll: wie Benutzer authentifiziert und in Spielen organisiert werden, wie Runden ablaufen, wie sich Märkte und Mining-Herausforderungen verhalten, wie Punktestände und Indikatoren erstellt werden und welche Ereignisse zur Unterstützung der Bewertung aufgezeichnet werden müssen. Auf diese Weise werden die Lernziele des Projekts in eine Reihe von beobachtbaren Verhaltensweisen und Regeln übersetzt, die konsistent umgesetzt, getestet und wiederverwendet werden können.

Diese Funktionsspezifikationen dienen nun als gemeinsamer Bezugspunkt für technische und pädagogische Partner. Für WP4-A4 bilden sie die Grundlage für Trainerrichtlinien und Moderationsmaterialien: Schritt-für-Schritt-Beschreibungen der Spielabläufe, Zustände und Bildschirme, die Trainer bei der Vorbereitung und Durchführung von Sitzungen sehen werden. Für WP4-A5 reduzieren sie die Unsicherheit bei der endgültigen Verpackung und Bereitstellung, da das erwartete Verhalten des Tools stabil und dokumentiert ist, sodass sich die Partner auf die Fertigstellung von Assets, Übersetzungen und Build-Konfigurationen konzentrieren können, ohne die Kernmechanik in Frage zu stellen. Dieselben Spezifikationen klären auch, welche Daten und Indikatoren für die Analyse verfügbar sind, und geben direkt Auskunft darüber, wie WP5 seine Bewertungsinstrumente und Datenerfassungsstrategien planen kann.

Mit Blick auf die Zukunft soll das in WP4.A3 definierte Verhalten robust genug für den Pilotbetrieb und flexibel genug für zukünftige Anpassungen sein. Die Pilotprojekte in WP5 werden diese Spezifikationen unter realen Bedingungen in verschiedenen Ländern, Institutionen und mit unterschiedlichen Lernendenprofilen testen und Belege dafür liefern, wie gut das funktionale Design das Lernen über Kreislaufwirtschaft und Abfall im Ziergesteinssektor unterstützt. Das Feedback aus diesen Pilotprojekten kann zu schrittweisen Anpassungen führen, aber die hier beschriebenen Kernabläufe und Anforderungen werden voraussichtlich als Referenzrahmen bestehen bleiben. Auf diese Weise markiert WP4.A3 den Übergang vom Entwurf und der Verfeinerung von RockChain zu dessen systematischer Einführung und Nutzung, wodurch sichergestellt wird, dass das Tool gewartet, repliziert und bei Bedarf über die Laufzeit des Projekts hinaus erweitert werden kann.