מיני פרויקט
רותם בן חמו

NEO

- רקע
- מכניזם קונצנזוס (Delegated Byzantine Fault).
- חוזים חכמים.
- מימוש מערכת רישום לקורסים.

רקע

בשנת 2014 הושקה מערכת בשם AntShares וביוני 2017 המערכת מותגה מחדש תחת השם Neo.

מלבד היותה מערכת מבוזרת כשאר מערכות הBlockchain, המטרות של Neo מאוגדות תחת הכותרת "כלכלה חכמה" (Smart Economy) **וכוללות נכסים דיגיטליים, זהויות דיגיטליות וחוזים חכמים.**

שילוב של נכסים וזהויות דיגיטליים מאפשר קישור בין נכסים במציאות לנכסים דיגיטליים (tokens). השימוש בחוזים החכמים מאפשר פעולות על הנכסים הדיגיטליים ושימושים שונים כגון אימות זהות דיגיטלית של מפעיל החוזה, ביצוע טרנזקציות של tokens, וכל מה שיבחר המתכנת של החוזה החכם לממש.

מנגנון קונצנזוס dBFT

במערכות Blockchain שונות יש מימושים שונים לצורך הגעה לקונצנזוס, המוכרות ביותר הן proof of work ו proof of stake.

Neo משתמשת במנגנון קונצנזוס Delegated Byzantine Fault Tolerant.

בdBFT ישנם נציגים (delegates) מבין המשתמשים (nodes) במערכת. מבין הנציגים נבחר דובר (speaker), והוא אחראי ליצור את הבלוק החדש בבלוקצ'יין. הspeaker מפיץ לכל הנציגים את הבלוק החדש והנציגים בודקים את נכונות הבלוק. אם לאחר זמן שידוע מראש לא התקבל רוב של 66% מבין הנציגים לטובת הבלוק החדש, נבחר speaker אחר שיצור את הבלוק החדש בעצמו, והתהליך חוזר על עצמו (הזמן שנמתין הפעם לקבלת רוב גדל אקספוננציאלית).

בהשוואה לPOS, המערכת של NEO פחות מבוזרת כיוון שלא כל הרשת לוקחת חלק ישיר בתהליך הקונצנזוס, אך תעבוד מהר יותר כתוצאה מכך.

יתרונות ע"פ POW: הקונצנזוס מתקבל לפני הוספת הבלוק, ולכן הבלוק הוא סופי מהרגע שבו נוצר (One Block Finality) -אין פיצולים. בנוסף, אין צורך בחישובים מסובכים ולכן זמן יצירת בלוק מהיר בהרבה ואין צורך בצריכת אנרגיה חריגה.

validate:
Is the data format consistent with the system • rules?
Is the transaction already on the blockchain?•
Are the contract scripts correctly executed?•
Does the transaction only contain a single • spend?(i.e. does the transaction avoid a double spend scenario?)

אלגוריתם הקונצנזוס בNEO:

1) Node כלשהו במערכת מפיץ מידע על עסקה שביצע לכל הרשת.

2) כל הConsensus Nodes מקבלים את המידע שהופץ ברשת ושומרים אותו.

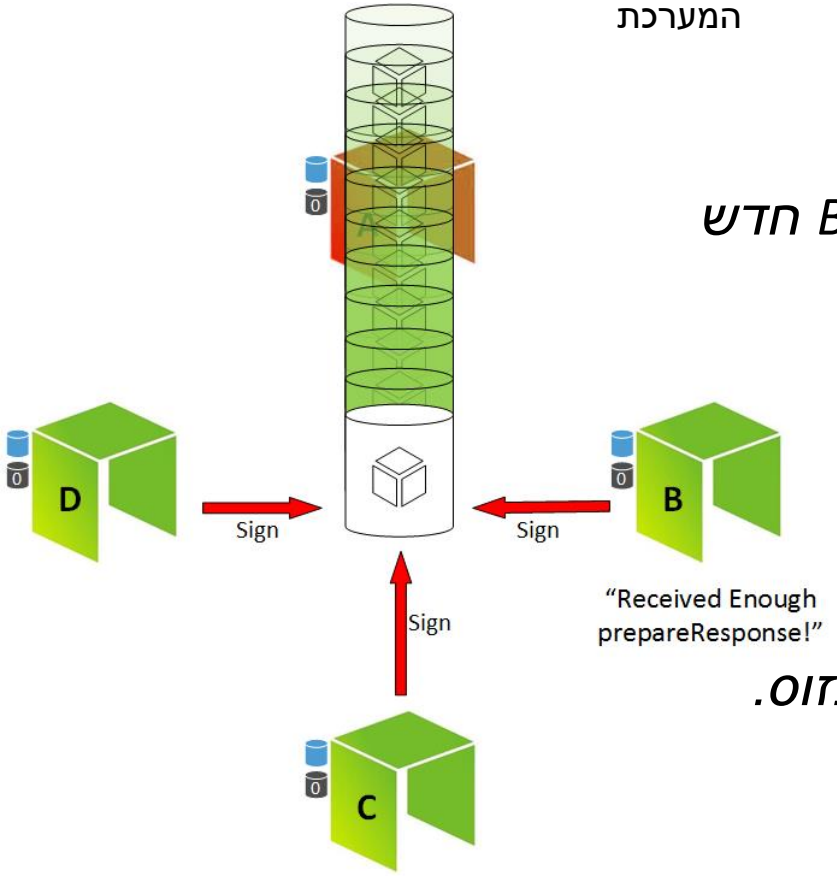
3) נבחר Speaker ע"פ נוסחה ידועה מראש, מבין הConsensus Nodes .

4) לאחר זמן קבוע מראש שמוקצה לכל בלוק, הSpeaker מפיץ הצעה- Block חדש למערכת.

5) כל Consensus Node מקבל את ההצעה, בודק אותה (validate) ומפיץ תגובה עם חתימתו.

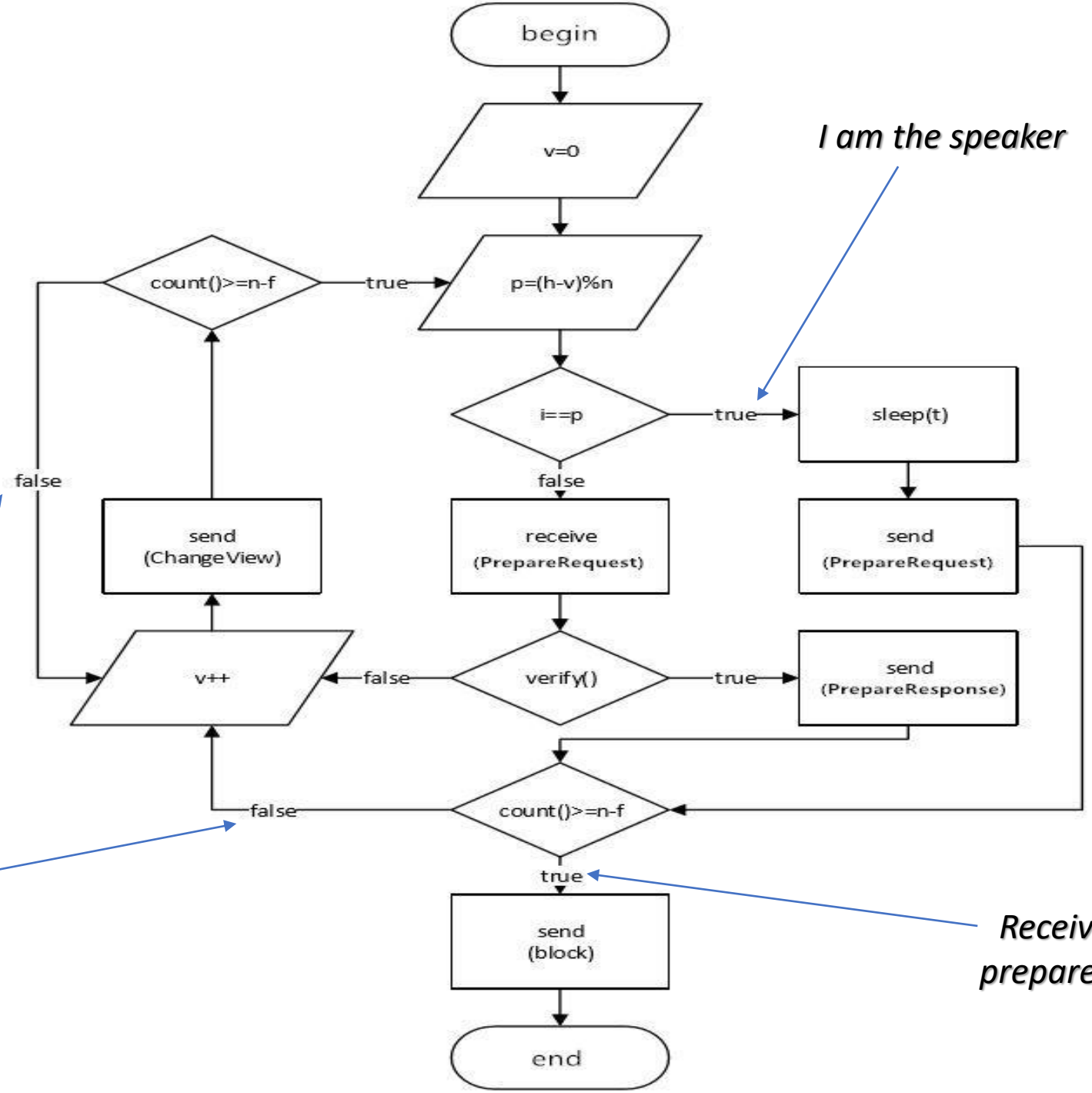
כעת יתקיים אחד מהשניים:
6.1) כל Consensus Node שמקבל לפחות 66% תגובות חיוביות מגיע לקונצנזוס.
6.2) עבר זמן שקבוע מראש ללא קונצנזוס ← מתחילים את התהליך מחדש.
הזמן הקבוע מראש שממתינים גדל אקספוננציאלית.

Nodes -Consensus Nodes
המשתתפים בתהליך הקונצנזוס של המערכת



Flow Chart

i- my node index
v- current view number
p- speaker index
t- time interval
h- block height



I am the speaker

False received after $2^{v+1}t$ seconds.

Received enough prepareResponses

הערות:

- המערכת תמיד שומרת על סנכרון ונציגים שלא מסונכרנים יחשבו כלא אמינים. נציג יכול להיות "לא אמין" גם מסיבות לא זדוניות.

- כמו שראינו בהרצאה הראשונה אפשר להוכיח שעם רוב של 66% ניתן להבטיח שלא נשיג קונצנזוס על מידע שקרי.
עם מעל 66% נציגים אמינים ניתן להבטיח הגעה לקונצנזוס ונכונות של המידע במערכת. בין 33% ל66% קונצנזוס לא בר השגה, ועם מתחת ל33% נציגים אמינים, הנציגים השקריים יכולים להשיג קונצנזוס ולרמות את המערכת.

נראה בקצרה שההנחה שיש פחות מ33% נציגים זדוניים מונעת פיצול במערכת (NEO docs):

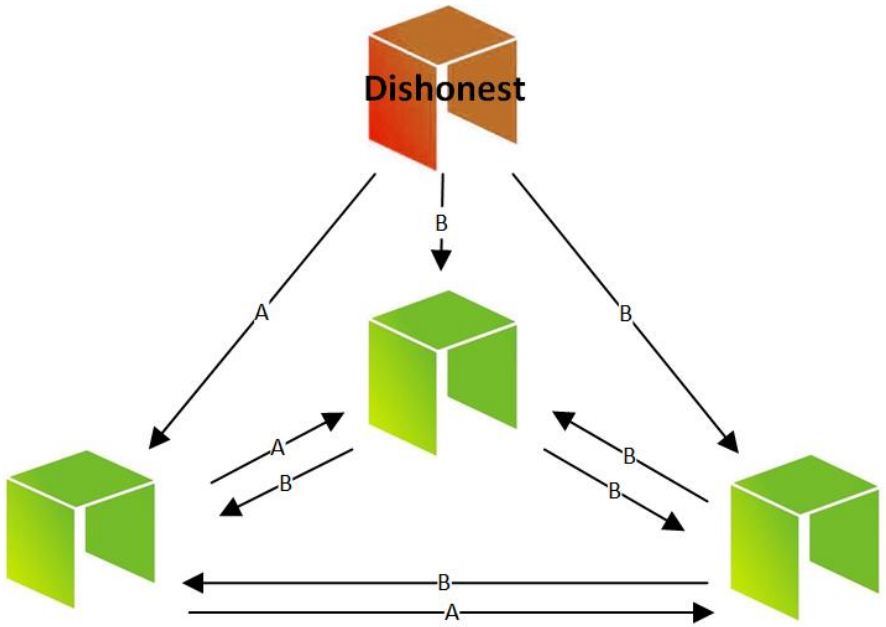
f = מספר הנציגים הזדוניים המקסימלי שהמערכת מאפשרת.
n= מספר הנציגים במערכת.

$$f = \left\lfloor \frac{(n-1)}{3} \right\rfloor$$

נניח שיש קבוצת נציגים זדונית F שרוצה לכפות על המערכת פיצול, כך שיתקבל קונצנזוס על מידע שקרי שהיא מפיצה. נסמן עוד שתי קבוצות זרות של נציגים אמינים, A,B, שאותם קבוצה F תרצה לפצל. (A U B U F=ALL CONSENSUS NODES).
נניח בשלילה שF מצליחה לפצל את קבוצה A וקבוצה B.
ע"מ ליצור קונצנזוס עם A נדרש $|A|+|F|\geq n-f$, וע"מ ליצור קונצנזוס עם B נדרש $|B|+|F|\geq n-f$. במקרה הכי גרוע, $|F|=f$, כלומר הכמות המקסימלית של נציגים זדוניים קיימת במערכת.
מכאן נקבל $|A|+|B|\geq 2n-4f$. כיוון ש $|A|+|B|=n-f$ נקבל $n-f\geq 2n-4f$ ← $n\leq 3f$ ← $n\leq n-1$ סתירה. לכן F לא תוכל להגיע לשני קונצנזוסים שונים עם קבוצה A ועם קבוצה B, כשמתקיים $|F|\leq f$.

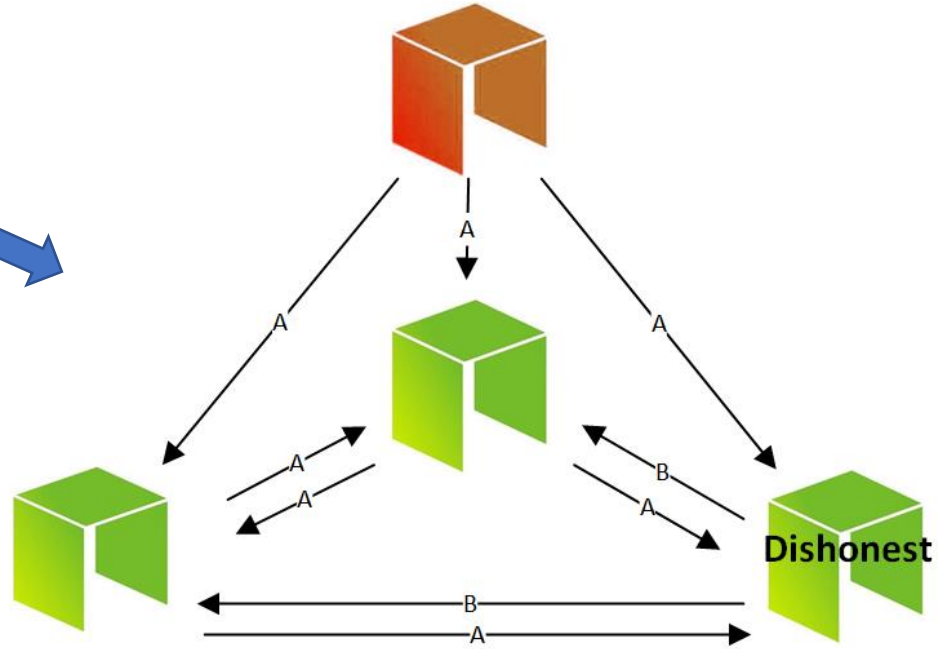
דוגמא למצב שבו הSpeaker שנבחר אינו אמין ומפיץ בלוק עם מידע כוזב

B הוא בלוק עם מידע כוזב ולכן הnode האמצעי והימני לא יאשרו אותו, ויקבעו ברוב של 66% מהנציגים שעל המערכת לבחור נציג חדש. אם הדובר היה שולח בלוק תקין ל2 מתוך השלושה, היה מתקבל רוב של 66% מבין הנציגים והמערכת הייתה מתקדמת לבלוק הבא.



דוגמא למצב שבו הבלוק הימני אינו אמין ומפיץ תשובה שקרית

כאן יש הסכמה של הnode האמצעי והשמאלי על הבלוק A שהופץ ע"י הדובר, ועל סמך הקונצנזוס בלוק A יתווסף לבלוקצ'יין. על סמך ההסכמה נוכל לומר שהדובר או הnode הימני לא אמינים (אין חשיבות למי מהם).



חוזים חכמים בNEO

- חוזים חכמים עונים על הצורך לביצוע עסקאות תחת תנאים בין צדדים שונים ללא גורם מתווך יחיד ועם רמת אמינות גבוהה.
- שימוש בבלוקצ'יין עונה על הצרכים הללו כיוון שהמערכת מבוזרת ומבטיחה את אמינות החוזה. **כל Node המשתתף בתהליך הקונצנזוס מריץ את החוזה החכם והבלוק החדש יכיל את המידע החדש שהתקבל כתוצאה מהרצת החוזה** (למשל- ביצוע טרנזקציה כתוצאה מהרצת החוזה).
- במערכת של NEO כל משתמש מחזיק בACCOUNT (ולא רק בכתובת כמו בביטקוין למשל). **ניתן לאמת את שיוך של חוזה מסוים לACCOUNT כלשהו ולהבטיח שהחוזה מופעל ממשתמש ספציפי**, ובנוסף ניתן לשייך נכסים למשתמשים שונים ולבצע העברות ופעולות כרצוננו.
- משתמש יכול ליצור **נכס גלובלי במערכת**. נכס גלובלי במערכת הוא מקביל למטבע NEO וניתן לבצע עליו פעולות ללא גישה ישירה לחוזה, אלא ע"י שימוש בארנקים המקוריים של NEO. בנוסף, לכל חוזה במערכת של NEO יש שטח אחסון פרטי משלו, אליו רק הוא יכול לגשת.
- **חוזה יכול לרוץ ולבצע שינויים בבלוקצ'יין פעם אחת בלבד בכל בלוק.**
- NEO מאפשרת (תאפשר בעתיד) לפתח את החוזים החכמים שלה במגוון רב של שפות.
- NEO מאפשרת ביצוע העברות של נכסים (tokens) בין מערכות בלוקצ'יין שונות, המקיימות מאפיינים מסוימים שקבועים בNEO.

מימוש מערכת רישום לקורסים ע"י חוזה

- <u>השימוש בBlockchain:</u>

- כל סטודנט מחזיק בNODE ברשת.

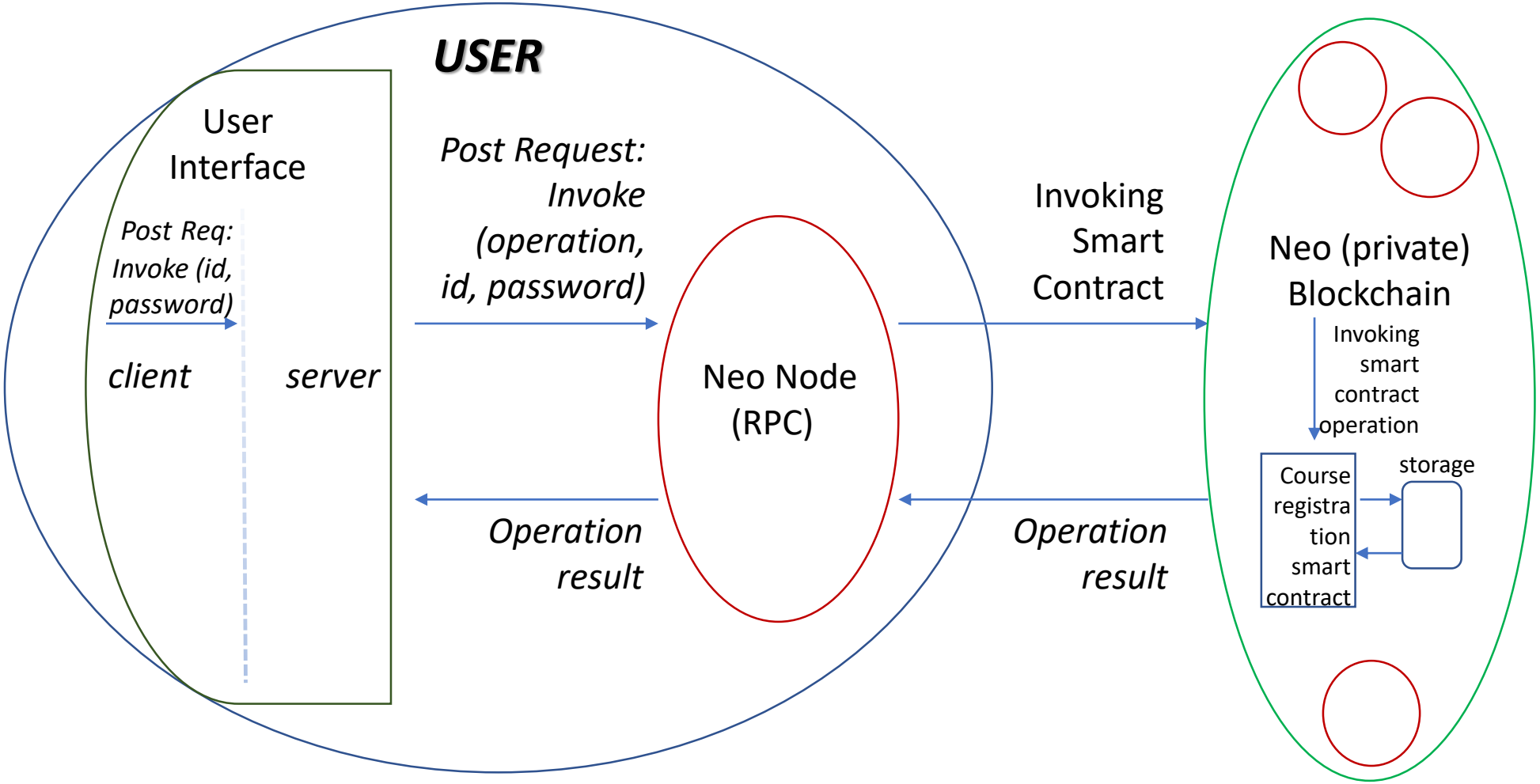
- אחסון טבלת הרישום לקורסים בבלוקצ'יין.

- ביצוע שאילתות והפעלת פונקציות שונות על ידי חוזה חכם:

- רישום, מחיקה, קבלת סטטוסים שונים, בדיקת מקום פנוי וביצוע עסקת החלפה- קורס תמורת קורס.

- הזדהות סטודנט עם סיסמא, הגבלת פעולות על סטודנטים, ספירת נק"ז.

1- סטודנט מקים שרת RPC node.
2- סטודנט מעביר דרך הממשק משתמש בקשה לשרת שמתקשר עם הRPC node שעל מחשב הסטודנט.
3- הNODE שולח בקשת INVOKE עם הפרמטרים המתאימים לשרת הבלוקצ'יין ע"מ להריץ את החוזה.
4- הNODE מחזיר לסטודנט RESPONSE.
4- במידה ויש צורך בעדכון הבלוקצ'יין, הNODE שולח RAW TRANSACTION עם הTX שהתקבל והבלוקצ'יין יעודכן כרצוי.
5- הNODE מעביר את התגובות של הבלוקצ'יין חזרה לסטודנט.



יתרונות השימוש בBlockchain כמערכת הרישום

- אחסון בטוח ומבוזר של טבלת הרישום על רשת הBlockchain.
- הפעלת פונקציות שונות באופן מבוזר ע"י החוזה החכם, ובכך עדכון טבלת הרישום באופן מבוזר גם כן.
- אלו ימנעו שינויים ועסקאות לא הוגנות כתוצאה מתקלות ברשת או פעולות זדוניות.

חסרונות

- **מערכת איטית שלא מגיבה מיד לכל עדכון. בנוסף עדכונים באחסון של החוזה אפשריים פעם אחת בכל בלוק.**
- המערכת של Neo לא שלמה ועדיין בפיתוח, לכן פיתוח החוזה החכם לא נוח במיוחד ועדיין לא נגיש לשפות תכנות רבות. (כלים מסוימים עובדים רק במערכות הפעלה מסוימות וכו').

נקודות נוספות

- *על מנת שהמערכת תהיה מבוזרת באופן מספק, כל סטודנט מחזיק NODE של הבלוקצ'יין. הקוד של הממשק משתמש כולו נגיש למשתמש. פעולות מסוימות כמו אתחול הטבלה שמורות לשרתי האוניברסיטה ויוכלו להתבצע רק בזמן שקבוע מראש ע"י החוזה.*
- *ניתן להשתמש ברשת הגלובלית של NEO אך נעדיף להקים רשת פרטית.*

- ***לסיכום,***

לדעתי שימוש בטכנולוגית החוזים החכמים של NEO אינה מתאימה למימוש של מערכת רישום לקורסים באוניברסיטה. הסיבה העיקרית היא העדכונים הלא מידיים של המערכת, דבר שמערכת רישום תחרותית דורשת.

המימוש של המערכת הוא אמין ולא מטעה סטודנטים: מי שמקבל אינדיקציה לפעולה חיובית- הפעולה אכן התבצעה עבורו.

המערכת יכולה להתאים לרישום ארוך טווח שאינו תחרותי ועמוס, הדורש רמת אבטחה גבוהה המסופקת ע"י הבלוקצ'יין.

http://docs.neo.org/

נקודות טכניות

- **פעולת invoke מה RPCNode מסמלצת הרצה של החוזה- מתאימה לשליפת מידע רצוי מאחסון של חוזה. בפועל ניתן להריץ חוזה פעם אחת בכל time interval, ולכן בקריאה שלא דורשת שינויי באחסון החוזה אנו מבצעים סימולציה, ובפעולה שכן דורשת שינויים נשתמש ב raw transaction המריצה את החוזה בכל הconsensus nodes, ומבצעת שינויים כרצוננו.** *במידה והחוזה כבר רץ ב time interval הנוכחי, המערכת מיידעת אותנו, ונודיע למשתמש לנסות שוב בעוד מספר שניות.*
- **סנכרון הבלוקצ'יין- מתבצע מתוך הseedlist. אצלנו הסנכרון הוא מתוך הדוקרים וברשת המקורית יש רשימה ארוכה של seedlist שמהם מתבצע הסנכרון של הבלוקצ'יין.**
- **כל NODE שמגיע לקונצנזוס מפרסם את הבלוק, וכל NODE שמקבל את הבלוק שהתקבל כקונצנזוס זונח את תהליך הקונצנזוס הישן ועובר לתהליך הקונצנזוס הבא.**
- **יש רשימה של Bookkeepers בפרוטוקול של הNODE. בנוסף ניתן להוסיף ע"י שימוש בפונקציה Validator.Register. כל אחד יכול להיות Bookkeeper, אבל כדי להיבחר הוא צריך לקבל מספר הצבעות עבורו (כל אחד יכול להצביע לbookkeeper).** *הbookkeepers מתוגמלים על ידי המערכת על השתתפותם בתהליך הקונצנזוס. בנוסף, מצביעיו של הbookkeeper שהשתתף בתהליך קונצנזוס תקין מתוגמלים גם הם.*

- בדיקה האם יש מספיק אישורים לבלוק בקוד: שיטה checkSignatures(). המשתנה M הוא בעצם 66% מהCONSENSUS NODES. יוצר בלוק חדש וחותם עליו בשם כל מי שאישר אותו. בסוף שולח את הבלוק ומסמן את הסטטוס של הבלוק כblockSent. קוראים לבדיקה למשל כשמקבלים prepareResponse.
- ספירת החתימות בקוד- צריך להוסיף לטרנזקציה שרוצים לאשר חתימה. מתבצע בaddSignature שקורא לadd. מתבצע כאשר מקבלים prepare request, שם צריך לעשות verify. גם כאשר מוסיפים טרנזקציה (דרך signAndRelay).
- בקובץ RPCserver.cs אפשר לראות שinvoke לא מריץ את החוזה אצל כל הרשת, אלא רק בVM שלנו. קשה להניח שהבלוקים שלנו לא תואמים למציאות כי בכל 15 שניות יש עדכון של בלוק והנכונות של הבלוקצ'יין נבדקת, והNODE לא יוכל להסתנכרן יותר, אלא אם הוא תואם את בלוקצ'יין התקין. קריאה להרצה אמיתית בבלוקצ'יין היא ע"י raw ובקוד דרך הפונקציה relay. בכל בלוקצ'יין יש מופע של localnode ולו יש רשימת remoteNodes, להם ניתן לבצע relay (ניתן לראות בקובץ localNode.cs).

https://github.com/neo-project/neo/tree/b1cf41d60576e13be772e3b180722f50cd6d64bc/neo/Consensus

- בstorage של החוזה יש רשומה לכל סטודנט, רשומה לכל קורס ורשומה לפרטי קורס (כמות, קיבולת, נק"ז).
- החוזה יכול להיות- נגיש לכל סטודנט, פרטי הסטודנטים (סיסמאות) לא חשופות לסטודנט- האוניברסיטה היא היחידה שיכולה להפעיל את פונקציית addStudent.
- סוג נוסף של חוזה- lock contract: ניתן להעביר לכתובת של החוזה מטבעות, ובכל פעם שנרצה להעביר כסף מהכתובת של החוזה הקוד ירוץ והעסקה תתבצע רק אם הקוד מחזיר true. ערך ההחזרה של החוזה צריך להיות בונליאני.