

دانشکده مهندسی کامپیوتر

**پروژه مقطع کارشناسی مهندسي کامپیوتر**

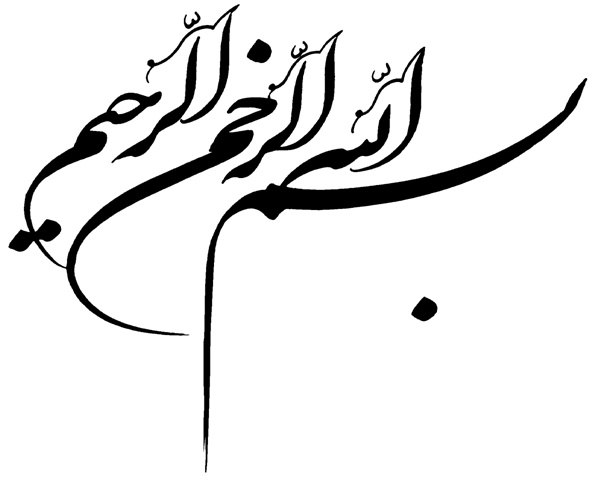
**بکارگیری قرارداد‌ هوشمند در آموزش آنلاین**

ایمان عباسی‌زاده

**استاد راهنما:**

دکتر علائیان

**شهریور ۱۴۰۱**



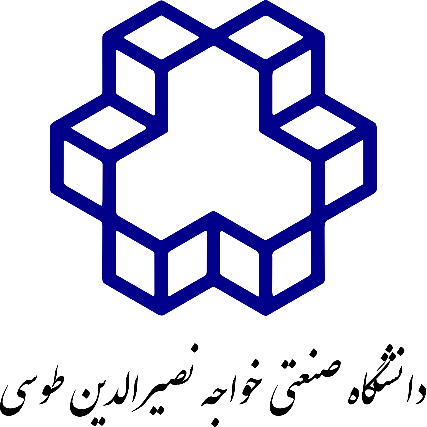
# تأييديّه هيات داوران

اعضاي هيئت داوران، نسخه نهائي پروژه خانم / آقاي: [نام دانشجو]

را با عنوان: [عنوان پروژه]

از نظر شکل و محتوي بررسي نموده و پذيرش آن را براي تکميل درجه کارشناسی تأييد مي‏کنند.

|  |  |  |  |
| --- | --- | --- | --- |
| **اعضاي هيئت داوران** | **نام و نام خانوادگي** | **رتبه علمي** | **امضاء** |
| 1. استاد راهنما |  |  |  |
| 1. استاد داور |  |  |  |

**دانشگاه صنعتی خواجه نصیرالدین طوسی**

|  |
| --- |
| **اظهارنامه دانشجو** |
| اینجانب ایمان عباسی‌زاده دانشجوی مقطع کارشناسی رشته مهندسی کامپیوتر گواهی می‌نمایم که مطالب ارائه شده در این پروژه با عنوان:  **بکارگیری قرارداد‌ هوشمند در آموزش آنلاین**  با راهنمایی استاد محترم دکتر علائیان توسط شخص اینجانب انجام شده است. صحت و اصالت مطالب نوشته شده در این پروژه تأیید می‌شود و در تدوین متن پروژه قالب مصوب دانشگاه را به طور کامل رعایت کرده‌ام.  امضاء دانشجو:  تاریخ: |

**حق طبع، نشر و مالکیت نتایج**

1- حق چاپ و تکثیر این پروژه متعلق به نویسنده و استاد راهنمای آن است. هرگونه تصویربرداری از کل یا بخشی از پروژه تنها با موافقت نویسنده یا استاد راهنما یا کتاب‌خانه دانشکده‌های مهندسی برق و کامپیوتر دانشگاه صنعتی خواجه نصیرالدین طوسی مجاز است.

2- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی خواجه نصیرالدین طوسی است و بدون اجازه کتبی دانشگاه قابل واگذاری به شخص ثالث نیست.

3- استفاده از اطلاعات و نتایج موجود پروژه بدون ذکر مرجع مجاز نیست.

###### تقديم به:

###### تشکر و قدرداني

###### چکيده

نیاز به آموزش آنلاین[[1]](#footnote-2) و غیر حضوری در سال های اخیر و به‌ویژه در دوره همه‌گیری کوید-۱۹[[2]](#footnote-3) تبدیل به یکی از نیازهای مهم و غیر قابل انکار شده، از طرفی ایجاد یک انگیزه موثر برای به پایان رساندن دوره[[3]](#footnote-4)های آنلاینی که شخص به صورت خودخواسته اخذ کرده و هیچ الزام بیرونی نظیر دانشگاه ، استاد و ... برای اتمام دوره‌های اخذ کرده ندارد و همچنین داشتن لیستی از این دوره ها به همراه تاریخ اخذ و موعد مقرر[[4]](#footnote-5) برای به پایان رساندن و از سوی دیگر تحقق هرچه کامل‌تر آرمان آموزش بدون تبیعض برای همه با در دسترس قرار دادن دوره‌های ترجمه شده به صورت رایگان و در آخر ایجاد یک ساز و کار مالی پایا برای تحقق اهداف ذکر شده، ایده ساخت پلتفرم[[5]](#footnote-6) آنلاین Smartlearn را به کمک تکنولوژی[[6]](#footnote-7) بلاکچین[[7]](#footnote-8) و قراردادهای‌‌هوشمند[[8]](#footnote-9) بوجود آورد.

کليد واژه: بلاکچین ، اتریوم[[9]](#footnote-10) ، آموزش آنلاین ، مشوق مالی[[10]](#footnote-11) ، جنگو[[11]](#footnote-12) ، **VueJS** ، متامسک[[12]](#footnote-13)، سالیدیتی[[13]](#footnote-14)

###### فهرست مطالب

عنوان صفحه

[تأييديّه هيات داوران ‌ه](#_Toc114087250)

[فهرست شکل‏ها ‌ه](#_Toc114087251)

[فهرست جدول‏ها ‌ز](#_Toc114087252)

[فصل 1- مقدمه 9](#_Toc114087253)

[1-1- انگیزه 9](#_Toc114087254)

[1-2- تبیین مسئله 10](#_Toc114087255)

[1-3- هدف 11](#_Toc114087256)

[1-4- نوآوری 11](#_Toc114087257)

[1-5- ساختار پایان نامه 12](#_Toc114087258)

[فصل 2- بلاکچین و قراردادهای هوشمند 14](#_Toc114087259)

[2-1- مقدمه 14](#_Toc114087260)

[2-2- دفتر کل حساب توزیع شده 16](#_Toc114087261)

[2-3- بلاکچین 17](#_Toc114087262)

[2-3-1- گره‌ها 17](#_Toc114087263)

[2-3-2- ساز و کار بلاکچین 19](#_Toc114087264)

[2-3-3- ویژگی‌ها 24](#_Toc114087265)

[2-4- اتریوم و قراردادهای هوشمند 24](#_Toc114087266)

[2-4-1- ساختار داده و انواع حساب در اتریوم 25](#_Toc114087267)

[2-4-2- ماشین مجازی اتریوم 31](#_Toc114087268)

[2-5- نتیجه‌گیری 35](#_Toc114087269)

[فصل 3- بلاکچین در آموزش آنلاین 37](#_Toc114087270)

[3-1- مقدمه 37](#_Toc114087271)

[3-2- پروژه‌های انجام گرفته 38](#_Toc114087272)

[3-3- نتیجه‌گیری 40](#_Toc114087273)

[فصل 4- تحلیل نیازمندی‌ها و طراحی 41](#_Toc114087274)

[4-1- مقدمه 41](#_Toc114087275)

[4-2- داستان‌های کاربر 42](#_Toc114087276)

[4-3- سند نیازمندی‌های نرم‌افزار 43](#_Toc114087277)

[4-3-1- تحلیل نیازمندی‌ها 48](#_Toc114087278)

[4-4- واسط کاربری 53](#_Toc114087279)

[فصل 5- ابزارها 54](#_Toc114087280)

[5-1- مقدمه 54](#_Toc114087281)

[5-2- ابزارهای توسعه 55](#_Toc114087282)

[5-2-1- سمت بلاکچین 55](#_Toc114087283)

[5-2-2- سمت کاربر 57](#_Toc114087284)

[5-2-3- سمت سرور 59](#_Toc114087285)

[5-3- پیاده‌سازی پروژه 59](#_Toc114087286)

[فصل 6- جمع‌بندی و نتیجه‌گیری 62](#_Toc114087287)

[6-1- مقدمه 62](#_Toc114087288)

[6-2- پیشنهادها 62](#_Toc114087289)

[فهرست مرجع‏ها 64](#_Toc114087290)

فهرست شکل‏ها

عنوان صفحه

[شکل(۲- 1) انواع سیستم‌های متمرکز، غیرمتمرکز، توزیع شده 15](#_Toc114070971)

[شکل(۲- 2) سلسله مراتب سیستم‌های توزیع شده و تکنولوژی‌های آن 16](#_Toc114070972)

[شکل(۲- 3) ساختار و ارتباطات گره‌های تشکیل‌دهنده شبکه بلاکچین 17](#_Toc114070973)

[شکل(۲- 4) انواع گره‌ها در شبکه بلاکچین 18](#_Toc114070974)

[شکل(۲- 5) ساختار درخت مرکل 19](#_Toc114070975)

[شکل(۲- 6) سازوکار مفهوم UTXO 20](#_Toc114070976)

[شکل(۲- 7) مراحل ارسال و تایید امضا دیجیتال 21](#_Toc114070977)

[شکل(۲- 8) انشعاب موقت و انتخاب طولانی‌ترین زنجیر ساخته شده 22](#_Toc114070978)

[شکل(۲- 9) سازوکار اثبات کار در بلاکچین 23](#_Toc114070979)

[شکل(۲- 10) روند قرارگیری تراکنش‌ها در بلوک 23](#_Toc114070980)

[شکل(۲- 11) بلاکچین اتریوم به عنوان ماشین‌ حالت توزیع شده 25](#_Toc114070981)

[شکل(۲- 12) انواع حساب‌ها در اتریوم و تراکنش بین آن‌ها 26](#_Toc114070982)

[شکل(۲- 13) روند تولید آدرس حساب‌های اتریوم 27](#_Toc114070983)

[شکل(۲- 14) ساختار داخلی حساب‌ها اتریوم 28](#_Toc114070984)

[شکل(۲- 15) مقایسه کد اسمبلی و بایت‌کد 28](#_Toc114070985)

[شکل(۲- 16) شمای کلی از یک وضعیت جهانی 29](#_Toc114070986)

[شکل(۲- 17) ساختار داخلی یک درخت پیشوندی وضعیت و مدل ذخیره‌سازی آن در بلوک اتریوم 29](#_Toc114070987)

[شکل(۲- 18) ساختار داخلی تراکنش‌ها در اتریوم 29](#_Toc114070988)

[شکل(۲- 19) شمایی از تراکنش ساخت قرارداد هوشمند 30](#_Toc114070989)

[شکل(۲- 20) شمایی از تراکنش با قرارداد و تغییر متغییرهای قرارداد 31](#_Toc114070990)

[شکل(۲- 21) تبدیل کد از زبان سطح بالا به بایت‌کد 32](#_Toc114070991)

[شکل(۲- 22) نمای داخلی و سازوکار EVM 32](#_Toc114070992)

[شکل(۲- 23) EVM به عنوان تغییردهنده وضعیت 33](#_Toc114070993)

[شکل(۲- 24) نمای کلی از جایگاه کد‌های قرارداد، EVM، نرم افزارهای اجرا کننده گره، سخت‌افزار 33](#_Toc114070994)

[شکل(۲- 25) نمای کاملی از ساختار کلی شبکه اتریوم 35](#_Toc114070995)

[شکل(۳- 1) دسته‌بندی مقالات انجام گرفته در زمینه به‌کارگیری بلاکچین در آموزش 38](#_Toc114071023)

[شکل(۳- 2) سازوکار پروژه Etherlearn 39](#_Toc114071024)

[شکل(۴- 1) ساختار نوشتن داستان کاربر 42](#_Toc114071184)

فهرست جدول‏ها

عنوان صفحه

[جدول(۲- 1) ویژگی‌های بلاکچین 24](#_Toc114087532)

[جدول(۴- 1) جدول سند نیازمندی سمت کاربر 44](#_Toc114087366)

[جدول(۴- 2) جدول سند نیازمندی سمت بلاکچین 46](#_Toc114087367)

[جدول(۴- 3) جدول سند نیازمندی سمت سرور 47](#_Toc114087368)

[جدول(۵- 1) ابزارها 54](#_Toc114087772)

# مقدمه

## انگیزه

با همه گیری کووید-۱۹ لزوم استفاده از دوره های آنلاین[[14]](#footnote-15) در جهان به مسئله جدی تبدیل شد و پلتفرم های مختلفی جهت رفع این نیاز ایجاد و یا پلتفرم های قبلی با جدیت بیشتری توسعه پیدا کردند.

پلتفرم smartlearn نیز در جهت رفع این نیاز و با ۳ انگیزه مهم جهت بهبود شرایط آموزش آنلاین بوجود آمد.

1. **تحقق آرمان آموزش بدون تبیعض برای همه :** وجود این مدل فکری برای هر سازمان، موسسه، پلتفرم ویا هر کسی که در زمینه آموزش فعالیت می‌کند یک اصل بسیار مهم باید باشد تا فردی به علت محدودیت‌هایی که خود عاملش نبوده از یادگیری و توسعه فردی باز نماند. در جهت همین خط فکری پلتفرم smartlearn سیاست های کلی خود را بر دو اصل رایگان بودن تمامی دوره‌ها و ترجمه دوره ها به زبان مقصد کاربر، در ابتدا به زبان فارسی، بنا نهاده و تحقق این دو اصل جز اساسی‌ترین انگیزه‌ها برای شکل‌گیری این پلتفرم می‌باشد.
2. **ایجاد انگیزه در کاربر[[15]](#footnote-16) برای تکمیل دوره های آنلاین اخذ کرده:** در شرایطی که فراگیر دوره آنلاین به صورت خودخواسته دوره ای را اخذ میکند و شخص یا سازمانی مانند استاد یا دانشگاه جهت نظارت خارجی و ایجاد الزام بیرونی برای اتمام و به سرانجام رساندن دوره اخذ شده وجود ندارد، لزوم وجود انگیزه مالی-تنبهی برای نتیجه‌گیری هرچه بهتر کاربر احساس می‌شود. پلتفرم smartlearn به کمک قرارداد هوشمندی که در این پلتفرم طراحی شده امکان تعیین موعدمقرر ‌زمانی برای انجام، تنبیه و تشویق مالی توسط خود کاربر برای دوره هایش، چه دورهای ارائه شده توسط پلتفرم و چه دوره های دیگر، را فراهم می‌کند.
3. **ناشناس[[16]](#footnote-17) بودن کاربران و حفظ حریم شخصی[[17]](#footnote-18) :** در دنیای امروز که اکثر شرکت‌ها و موسسات در تلاش برای کسب اطلاعات هرچه بیشتر کاربران خود در جهت استفاده های تبلیغاتی[[18]](#footnote-19)، جامعه شناسی[[19]](#footnote-20)، تحلیلی و ... می‌باشند، پلتفرم smartlearn در خلاف جهت این موج سعی بر حفظ هویت افراد و آزادی عمل کاربران خود با بهره‌گیری از تکنولوژی بلاکچین دارد. این جهت‌گیری از انگیزه های مهم شکل‌گیری این پلتفرم و همچنین از دلایل استفاده smartlearn از بلاکچین برای ذخیره‌سازی اطلاعات کاربران خود است.

## تبیین مسئله

پلتفرم smartlearn از نظر ساختاری به دو بخش کلی تقسیم بندی می‌شود :

1. ارائه دوره‌های آنلاین ترجمه شده به صورت رایگان و قابل دسترس برای تمامی کاربران
2. ارائه امکانی برای ثبت دوره‌ها، تعیین موعد مقرر زمانی و تنبیه و تشویق مالی و مدیریت دوره‌های برای انجام به کمک قرارداد هوشمند

بخش اول این پلتفرم به مانند سایت‌ها و بسترهای فعال در زمینه آموزش آنلاین اما با رویکرد تحقق آرمان آموزش بدون تبعیض، دوره‌هایی ترجمه شده به زبان مقصد کاربر و به صورت کاملا رایگان فراهم آورده که تمامی کاربران امکان بهره‌وری از این دوره‌ها را دارا می‌باشند.

بخش دوم مربوط به ارائه امکان مدیریت دوره‌های کاربران همچنین تعیین و تعریف خود انگیزشی[[20]](#footnote-21) از نوع مالی در جهت اتمام دوره‌های اخذ شده کاربر می‌باشد. به کمک قرارداد هوشمندی که در این بخش طراحی شده کاربر میتواند دوره‌های ارائه شده در سایت و یا هر دوره دیگری که اخذ نموده است را به لیست دوره‌های برای انجام خود با تعیین موعد مقرر زمانی و جایزه[[21]](#footnote-22) و انتخاب گزینه‌های اضافی موجود اضافه نماید و با در کنار هم داشتن تمامی دوره‌های اخذ کرده به صورت یکجا امکان مدیریت و برنامه ریزی دوره ها را در محیطی امن، غیر قابل تغییر و ناشناس داشته باشد.

این پلتفرم برای امتداد حیات خود و توانایی ارائه دوره ها به صورت ترجمه شده و رایگان نیاز به یک چرخه حیات مالی[[22]](#footnote-23) داشته که برای پاسخ به این نیاز یک سازوکار[[23]](#footnote-24) پایا[[24]](#footnote-25) با تعیین کارمزد برای اضافه کردن دوره در بخش دوم و استفاده از مجموع دریافتی ها از کاربران ویژه برای تعریف و ترجمه دوره ها در بخش اول طراحی شده که این چرخه عامل ارتباطی بین این دو بخش می‌باشد.

همچنین به جهت تشویق کاربران برای مشارکت در این چرخه مکانیزمی طراحی شده تا کاربران ویژه بتوانند در صورتی که مجموع مشارکت مالی آن‌ها به میزان از قبل تعیین شده‌ای (مبلغ کافی جهت انجام امور ترجمه و تولید دوره مورد نظر کاربر) رسید دوره ای را برای ترجمه از هر زبانی به زبان مورد تقاضای کاربر به سایت ارائه دهند که براساس تاریخ ارائه و گذشت زمان مورد نیاز جهت ترجمه و تولید دوره به صورت ترجمه شده و رایگان در اختیار تمامی کاربران قرار می‌گیرد.

## هدف

با خلق اولین رمزارز[[25]](#footnote-26) توسط ساتوشی ناکاموتو[[26]](#footnote-27) بر پایه فناوری بلاکچین در سال ۲۰۰۸ میلادی نگاه جدی‌تری به سیستم های غیرمتمرکز و بهره‌‌وری از این سیستم‌ها شد، تا در سال ۲۰۱۵ بلاکچین اتریوم توسط ویتالیک بوترین[[27]](#footnote-28) و کوین وود[[28]](#footnote-29) جهت توسعه و برطرف‌سازی کاستی‌هایی که در بیت‌کوین وجود داشت بوجود آمد، اتریوم توانست امکان ایجاد قراردادهای هوشمند را برای تمام پروژه‌ها عملی کرده و گام نوینی در جهت هوشمند‌سازی جهان بردارد.

در ادامه این روند پیشرو،  تولید پروژه‌های کاربردی جهت نهادینه کردن و استفاده هرچه بیشتر از سیستم‌های غیر متمرکز به صورت جدی‌تری دنبال شد، پلتفرم smartlearn نیز هم‌جهت با این روند ایجاد شد.

هدف اصلی خلق پلتفرم smartlearn استفاده از تکنولوژی نوین بلاکچین و قراردادهای هوشمند برای ایجاد یک برنامه کاربردی در فضای غیرمتمرکز بوده تا بتواند سهمی هرچند خیلی کوچک در ساخت و توسعه جهان هوشمند و آزاد و غیرمتمرکز آینده داشته باشد.

## نوآوری

پیاده‌سازی پلتفرم smartlearn به طور کلی از سه بخش فنی تشکیل شده که ابزارهای بکار گرفته شده در هر بخش از حیث سرعت، قدرت و به‌روز بودن در زمینه تخصصی خود پیشرو و نوآور به حساب می‌آیند :

1. **سمت سرور :** جهت ثبت و واکشی اطلاعات دوره های ارائه شده در سایت نیازمند یک پایگاه‌داده[[29]](#footnote-30) به شکل متمرکز و زبانی برای تعامل با این پایگاه‌داده[[30]](#footnote-31) می‌باشیم، برای پاسخ به این نیاز از زبان قدرتمند پایتون[[31]](#footnote-32) و چهارچوب[[32]](#footnote-33) جنگو استفاده شده است و پایگاه‌داده متمرکز این پلتفرم sqlite3 می‌باشد
2. **سمت بلاکچین :** نوآوری اصلی بکارگرفته شده در این پلتفرم بهری‌گیری از فناوری بلاکچین و استفاده از قراردادهای هوشمند می‌باشد، ثبت و واکشی اطلاعات دوره‌های برای انجام کاربران ویژه و میزان مشارکت مالی آنها، در بلاکچین اتریوم انجام می‌شود و قسمت منطق و اجرای دستورات از قبل نوشته شده و همچنین ارتباطات بین کاربر ویژه و سایت و برقراری تمامی شروطی و تفاهماتی که به مانند یک قرارداد فی مابین کاربر ویژه و سایت وجود دارد توسط قرارداد هوشمند طراحی شده با زبان سالیدیتی انجام می‌گیرد.
3. **سمت کاربر :** جهت نمایش اطلاعات ذخیره شده و برقراری تعامل پویا با کاربران نیاز به یک زبان برنامه نویسی سمت کاربر وجود دارد، در پاسخ به این نیاز پلتفرم smartlearn از زبان قدرتمند جاوااسکریپت[[33]](#footnote-34) و چهارچوب Vue و همچنین از کتابخانه‌ای به نام drizzle برای اتصال بین قراردادهوشمند و frontend استفاده نموده است.

## ساختار پایان نامه

روند ساختاری این پایان نامه در ادامه بدین شکل خواهد بود؛ ما بعد از ارائه یک چشم انداز کلی از انگیزه ها، نوآوری های بکار گرفته شده و هدف شکل‌گیری پروژه و همچنین بیان ساختار کلی آن در فصل اول و در مقدمه، قصد داریم در فصل دوم به تعریف بلاکچین و قراردادهای هوشمند به عنوان هسته اصلی و نوآورانه این پروژه بپردازیم تا با شناخت هرچه بیشتر این فناوری درک جامع‌تری نسبت به پتانسیل سیستم‌های غیرمتمرکز و جایگاه کاربردی آن در دنیای امروز کسب کنیم.

در فصل سوم به بیان کارهایی که پیش از این در زمینه آموزش آنلاین با رویکرد استفاده از بلاکچین و قراردادهای هوشمند انجام گرفته می‌پردازیم.

فصل چهارم با بیان داستان‌های کاربر[[34]](#footnote-35) و نیازسنجی جدول نیازمندی‌های پروژه[[35]](#footnote-36) را ارائه می‌دهیم سپس توضیح جامعی از تحلیل و طراحی پروژه و جزئیات بکار رفته در آن خواهیم داشت همچنین نمایی از رابط کاربری پلتفرم smartlearn ارائه خواهیم دید.

فصل پنجم به توضیح کامل جزئیات ابزار‌های بکار رفته در پروژه، که در بخش نوآوری فصل اول به صورت تیتروار بیان شد، اختصاص دارد همچنین نحوه دانلود و نصب این ابزارها و نسخه استفاده شده از هریک در پروژه توضیح داده خواهد شد.

در فصل ششم و پایانی این پایان‌نامه جمع‌بندی، نتیجه‌گیری، پیشنهادات و ایده‌هایی که در آینده جهت توسعه پروژه در نظر داریم را ارائه خواهیم کرد.

# بلاکچین و قراردادهای هوشمند

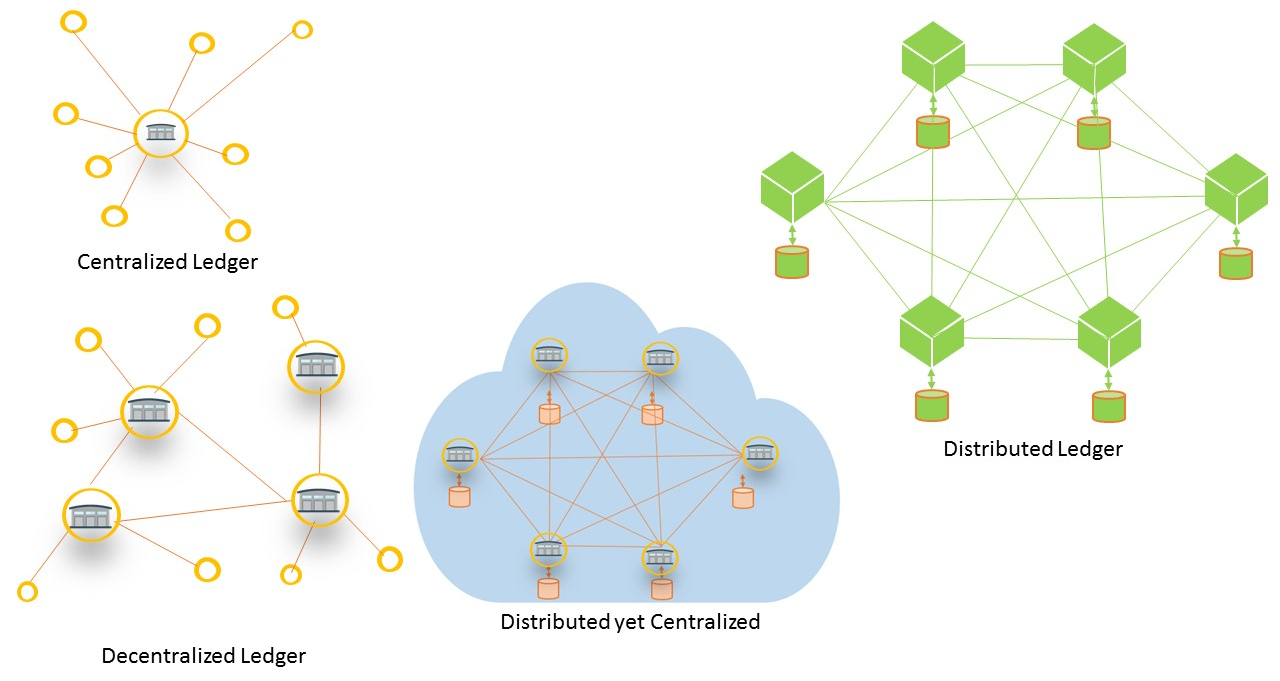
## مقدمه

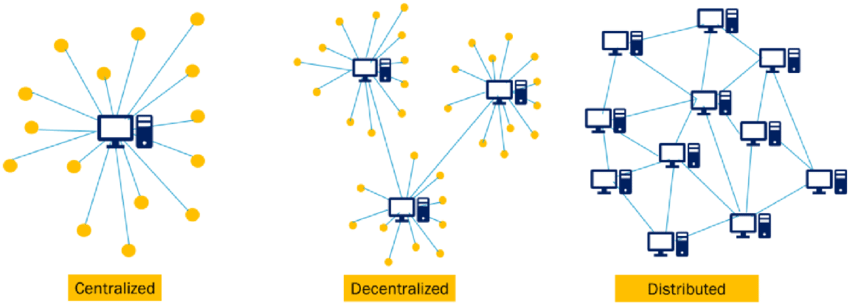
ثبت اطلاعات[[36]](#footnote-37) جهت دسترسی و ماندگاری اطلاعات از زمان آشنایی انسان با علم در حدود 3000 سال پیش از میلاد مسیح بر روی لوح های رسی/گلی انجام می‌گرفت [1]، با پیشرفت علم وسیله و شیوه ذخیره‌سازی[[37]](#footnote-38) داده[[38]](#footnote-39) تغییر و بهبود یافت تا با اختراع کامپیوتر در سال 1871 توسط چارلز بابیج[[39]](#footnote-40) مسیری جدید به روی انسان گشوده شد [2]، مسیری که دنیای امروز وام‌دار آن است و مسئله ذخیره‌سازی و واکشی[[40]](#footnote-41) داده را به یک مسئله بسیار جدی و مهم‌ تبدیل کرده است.

در طی این مسیر جدید و با پیشرفت تکنولوژی بخصوص با ظهور اینترنت به عنوان شبکه‌ای از کامپیوتر‌ها ذخیره‌سازی داده به شکل سیستمی درآمد و ساختاریافته دنبال شد، ذخیره‌سازی داده بر روی کامپیوترها در ابتدا به شکل سیستم های متمرکز[[41]](#footnote-42) به معنای کنترل اطلاعات به شکل واحد زیر نظر و حاکمیت یک شخص یا شرکت و همچنین مشترک یا توزیع نشده[[42]](#footnote-43) به معنای استقرار تمامی اجزای سیستم در یک مکان فیزیکی یکجا بود که این دو ویژگی متمرکز و توزیع نشده هر یک معضلاتی را در مسیر پیشرفت و گسترش تکنولوژی قرار میداد، از سویی متمرکز بودن یک سیستم کنترل داده ها را در اختیار کامل شخص یا شرکت قرار می‌دهد که این امر در دراز مدت عامل فساد و سوء استفاده از داده های ذخیره شده می‌شود همچنین به علت تحت کنترل بودن سیستم توسط افراد محدود باعث کندی پیشرفت تکنولوژی می‌شود و از سوی دیگر عدم توزیع سیستم، خطراتی از جمله خرابی و یا هک[[43]](#footnote-44) سرور مرکزی که باعث از دست رفت یا صدمه به داده‌ها می‌باشد را در پی خواهد داشت.

در ادامه روند پیشرفت سیستم های ذخیره‌سازی داده و در پی رفع معضلات بیان شده ایده‌ها و سیستم هایی بوجود آمدند؛ ارائه‌دهنده‌های متن باز[[44]](#footnote-45)ی مثل آپاچی[[45]](#footnote-46) نمونه ای از یک سیستم غیرمتمرکز[[46]](#footnote-47) ولی توزیع نشده است که به صورت غیرمتمرکز و با کمک جمیع افراد توسعه‌دهنده[[47]](#footnote-48) سراسر جهان توسعه داده می‌شود ولی همچنان معضلات یک سیستم توزیع نشده را به همراه دارند از سوی دیگر سیستم‌های خدمات دهنده ابری[[48]](#footnote-49) مانند کلودفلر[[49]](#footnote-50) توزیع شده[[50]](#footnote-51) بوده و دارای سرور[[51]](#footnote-52)های متعددی در اقصی نقاط جهان می‌باشند اما به صورت متمرکز بوده و کنترل مرکزی دارند.

در جهت فراهم شدن سیستمی غیرمتمرکز و توزیع شده رویکردی نوین با ظهور بیت‌کوین[[52]](#footnote-53) به عنوان اولین رمزارز[[53]](#footnote-54) جهان بر پایه تکنولوژی بلاکچین[[54]](#footnote-55) که یک نوع دفتر کل حساب توزیع شده[[55]](#footnote-56) است و صرفا جهت ثبت تراکنش‌های مالی استفاده می‌گردد شکل گرفت و با گسترش این رویکرد نوین، تکنولوژی‌های جدیدی جهت ذخیره‌سازی و واکشی اطلاعات بوجود آمدند.





شکل(۲- 1) انواع سیستم‌های متمرکز، غیرمتمرکز، توزیع شده

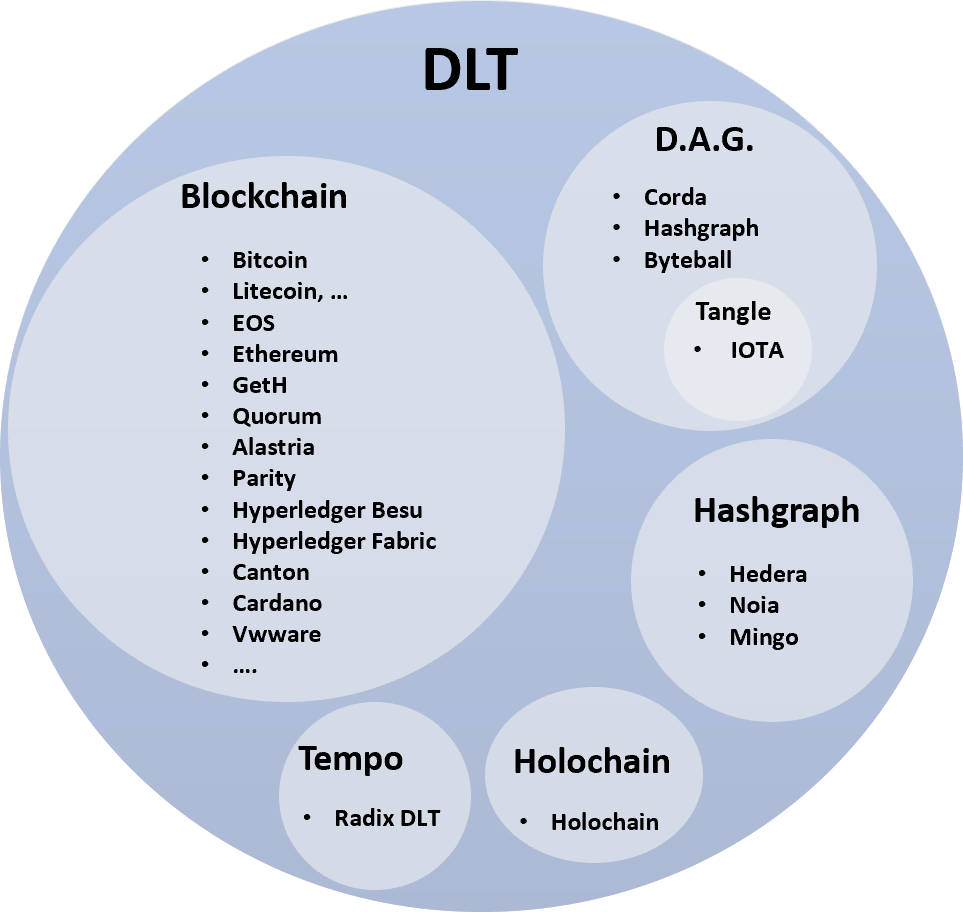
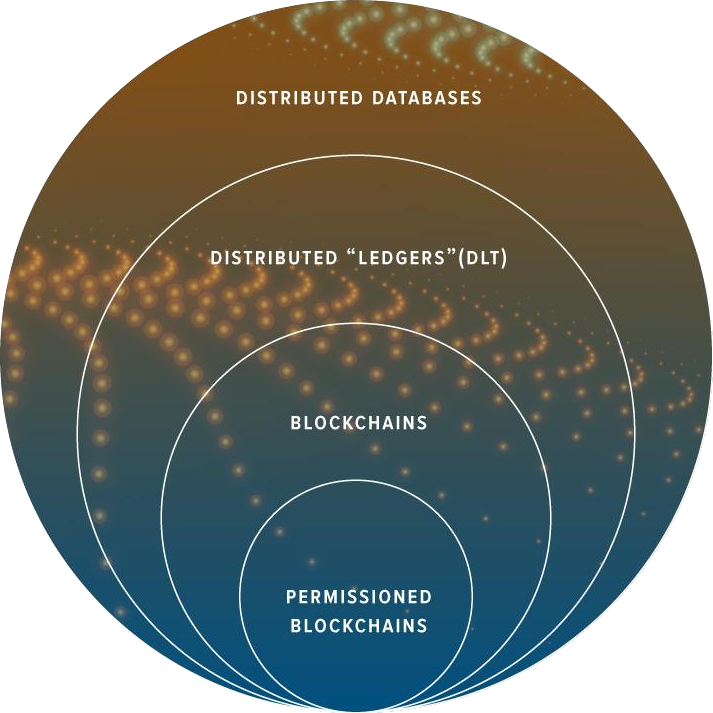
## دفتر کل حساب توزیع شده

دفتر کل حساب توزیع شده یک نمونه خاص از پایگاه داده توزیع شده است که تراکنش‌ها[[56]](#footnote-57) در آن به صورت توزیع شده ثبت می‌شوند، در یک دفتر کل حساب توزیع شده تمامی نودها[[57]](#footnote-58) باید یک نسخه یکسان از دفتر حساب داشته باشند و برای تغییر یا ثبت اطلاعات جدید، نودها باید روی حالت[[58]](#footnote-59) جدید توافق کنند.

تراکنش های ثبت شده روی دفتر کل حساب باید دارای سه ویژگی به شرح ذیل باشند :

1. **اتمیک[[59]](#footnote-60) :** به معنی انجام تمامی عملیات یک تراکنش است که درصورت انجام نشدن یک عملیات کل تراکنش انجام نشده تلقی می‌شود
2. **با ثبات[[60]](#footnote-61) :** به این معنا که تراکنش صورت گرفته ثبات کل جریان دفتر حساب را حفظ کند، به طور مثال در تراکنش انتقال پول در یک بانک کل بالانس[[61]](#footnote-62) مالی بانک ثابت می‌ماند
3. **پایا[[62]](#footnote-63) :** تراکتش در صورت انجام و قرارگیری در پایگاه داده باید به طور دائم وجود داشته باشد و قابل دسترس باشد

تکنولوژی دفتر کل حساب توزیع شده که به اختصارDLT [[63]](#footnote-64) نامیده می‌شود خود شامل انواع تکنولوژی‌های مختلفی است که شناخته شده‌ترین آن تکنولوژی بلاکچین است.



شکل(۲- 2) سلسله مراتب سیستم‌های توزیع شده و تکنولوژی‌های آن

## بلاکچین

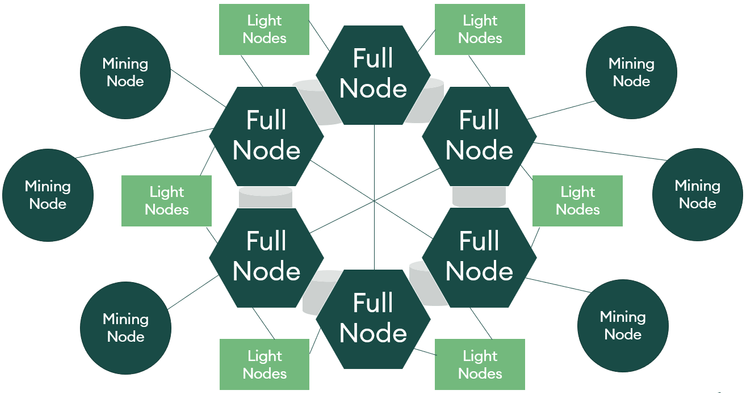
بلاکچین به عنوان مطرح‌ترین نوع دفتر کل حساب توزیع شده, زنجیره ای از بلوک‌[[64]](#footnote-65)ها است که هر بلوک حاوی تعدادی تراکنش و داده می‌باشد که به بلوک قبلی اشاره دارد، این زنجیره ادامه پیدا می‌یابد تا به اولین بلوک ایجاد شده به نام بلوک پیدایش[[65]](#footnote-66) برسد.

تعریف بلاکچین به شکل امروزی‌ که بیان شد متعلق به بعد از عرضه بیت‌کوین در سال ۲۰۰۸ است اما ایده اولیه شکل‌گیری بلاکچین در سال ۱۹۹۱ و با تلاش های استوارت هابر[[66]](#footnote-67) و اسکات استورنتا[[67]](#footnote-68) پایه‌ریزی شد و بلاکچین را به عنوان یک راه حل عملی محاسباتی برای زدن برچسب زمانی[[68]](#footnote-69) به اسناد دیجیتال معرفی کردند که این کار از برگرداندن زمان و دست‌کاری اسناد جلوگیری می‌کرد.

### گره‌ها

توزیع و غیرمتمرکز بودن شبکه بلاکچین به علت وجود گره‌ها در این شبکه است، به طور کلی هر شرکت کننده در شبکه بلاکچین یک گره است، أنواع مختلفی گره در شبکه وجود دارد که هر یک دارای ویژگی مخصوصی هستند و برای میزبانی[[69]](#footnote-70) یا اتصال به آن سخت‌افزار خاصی نیاز است. [3]

هر گره در شبکه بلاکچین مستقل بوده و برای خود تصمیم گیرنده است به این معنا که با داشتن یک نسخه کامل از کل شبکه می‌تواند در هر لحظه تصمیم بگیرد در صورت ایجاد تغییر در شبکه و به‌روزرسانی در الگوریتم‌ها از تغییرات پیروی کند یا نه و یا تصمیم بر ایجاد تغییر در شبکه بگیرد و یا حتی می‌تواند به صورت موقت یا دائم بلوک‌ای به زنجیره بلوک‌هایش اضافه نکند و از شبکه خارج شود که نتیجه این تصمیمات در صورت همکاری و موافقت دیگر گره‌های شبکه به ایجاد انشعاب[[70]](#footnote-71) و یا در حالت جدی‌تر به ایجاد یک بلاکچین جدید می‌انجامد.



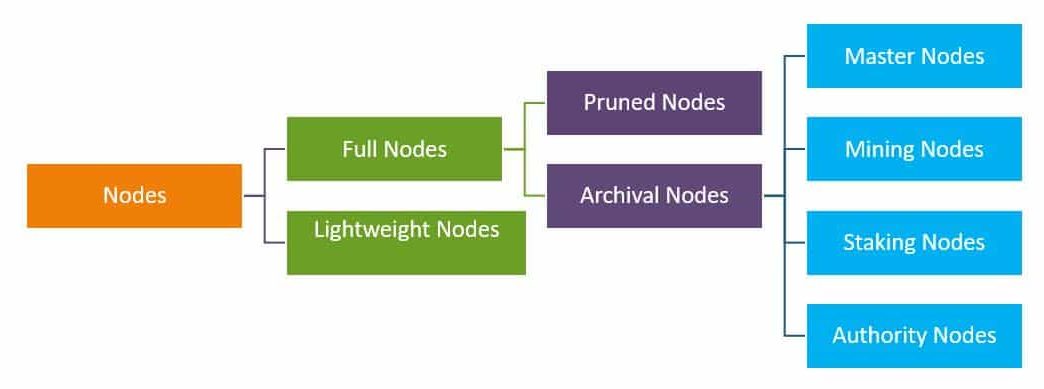
شکل(۲- 3) ساختار و ارتباطات گره‌های تشکیل‌دهنده شبکه بلاکچین

گره‌ها به دو دسته کلی تقسیم می‌شوند :

1. **گره‌های کامل[[71]](#footnote-72) :** گره‌های کامل یک کپی از بلاکچین را ذخیره می‌کنند که در نتیجه امنیت و صحت داده‌ها بر روی بلاکچین با اعتبارسنجی داده‌ها تضمین می‌شود، این دسته از گره خود به دو دسته گره‌های هرس شده[[72]](#footnote-73) و گره‌های بایگانی[[73]](#footnote-74) تقسیم‌بندی می‌شوند و تفاوت این دو گره در میزان ذخیره‌سازی آن‌ها است[[74]](#footnote-75)، گره‌های هرس شده ابتدا باید کل بلاکچین را از ابتدا طی کند و تمام بلوک‌های قبلی را تایید کند تا بتواند تراکنش‌های جدید را از نظر مقادیر ورودی، امضا فرستنده و موجودی کافی جهت انجام تراکنش اعتبارسنجی نماید.

* گره استخراج[[75]](#footnote-76) یک گره کامل از نوع گره بایگانی است که امکان تایید و استخراج بلوک از طریق اجماع اثبات کار[[76]](#footnote-77) دارد.

1. **گره‌های سبک وزن[[77]](#footnote-78) :** در گره سبک وزن فقط سربرگ بلوک‌ها ذخیره می‌شوند، این گره به امنیت شبکه کمکی نمی‌کند و برای استفاده کاربر بوجود آمده، این گره با تکیه بر یک نود کامل و پخش[[78]](#footnote-79) تراکنش در شبکه امکان تعامل کاربر با بلاکچین را نظیر ارسال، دریافت و امضا تراکنش‌ها برقرار می‌کند.



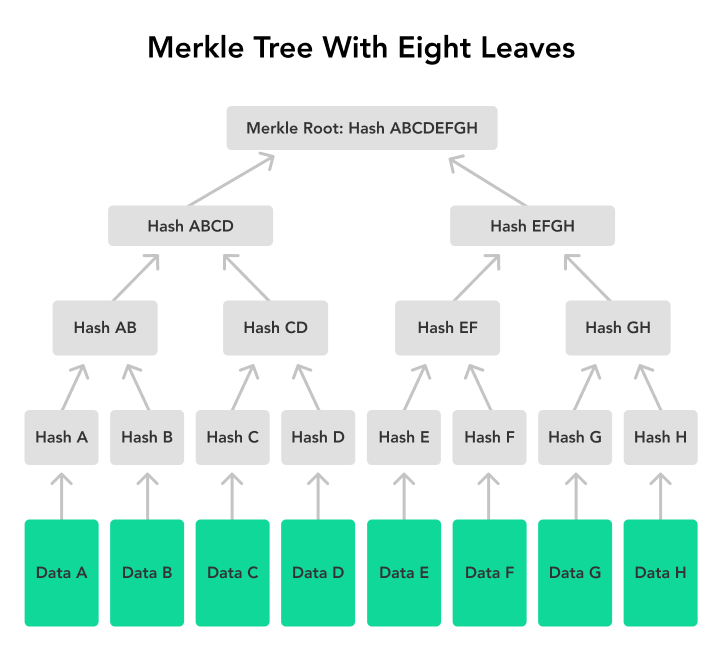
شکل(۲- 4) انواع گره‌ها در شبکه بلاکچین

### ساز و کار بلاکچین

در این بخش قصد داریم با بررسی بلاکچین بیت‌کوین به عنوان مهم‌ترین و معروف‌ترین بلاکچین، شناخت کامل‌تری نسبت به ساز و کار بلاکچین بدست آوریم.

بلوک‌های بلاکچین از دو بخش سربرگ و بدنه[[79]](#footnote-80) تشکیل می‌شوند:

* **بدنه :** بدنه هر بلوک شامل لیست تراکنش‌هایی است که هش[[80]](#footnote-81) هر یک با الگوریتم[[81]](#footnote-82) مخصوصی، در بیت‌کوین با الگوریتم SHA256، محاسبه شده و با ساختمان داده[[82]](#footnote-83) درخت مرکل[[83]](#footnote-84) ذخیره شده اند.

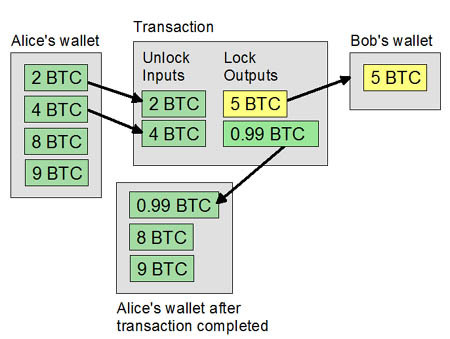


شکل(۲- 5) ساختار درخت مرکل

هر تراکنش، که در شبکه بیت‌کوین منظور تراکنش صرفا مالی است، شامل آدرس فرستنده، آدرس گیرنده، تعداد ساتوشی[[84]](#footnote-85) و کارمزد تراکنش[[85]](#footnote-86) است و جهت ارسال و تایید نیازمندی‌هایی دارد که بخشی از آن توسط کاربر ایجاد کننده تراکنش مرتفع می‌شود و بخش دیگر آن به کمک مفاهیم تعریف شده داخل شبکه انجام می‌گیرد.

مهترین نیازمندی جهت تایید تراکنش، محاسبه میزان موجودی ارسال کننده تراکنش است؛ بیت‌کوین موجودی آدرس کیف پول‌ها را در شبکه نگهداری نمی‌کند و از مفهومی به نام خروجی خرج نشده تراکنش[[86]](#footnote-87) [4] جهت تایید میزان موجودی کافی برای انجام تراکنش و همچنین برای جلوگیری از خرج مجدد[[87]](#footnote-88) و افزایش امنیت استفاده می‌کند.

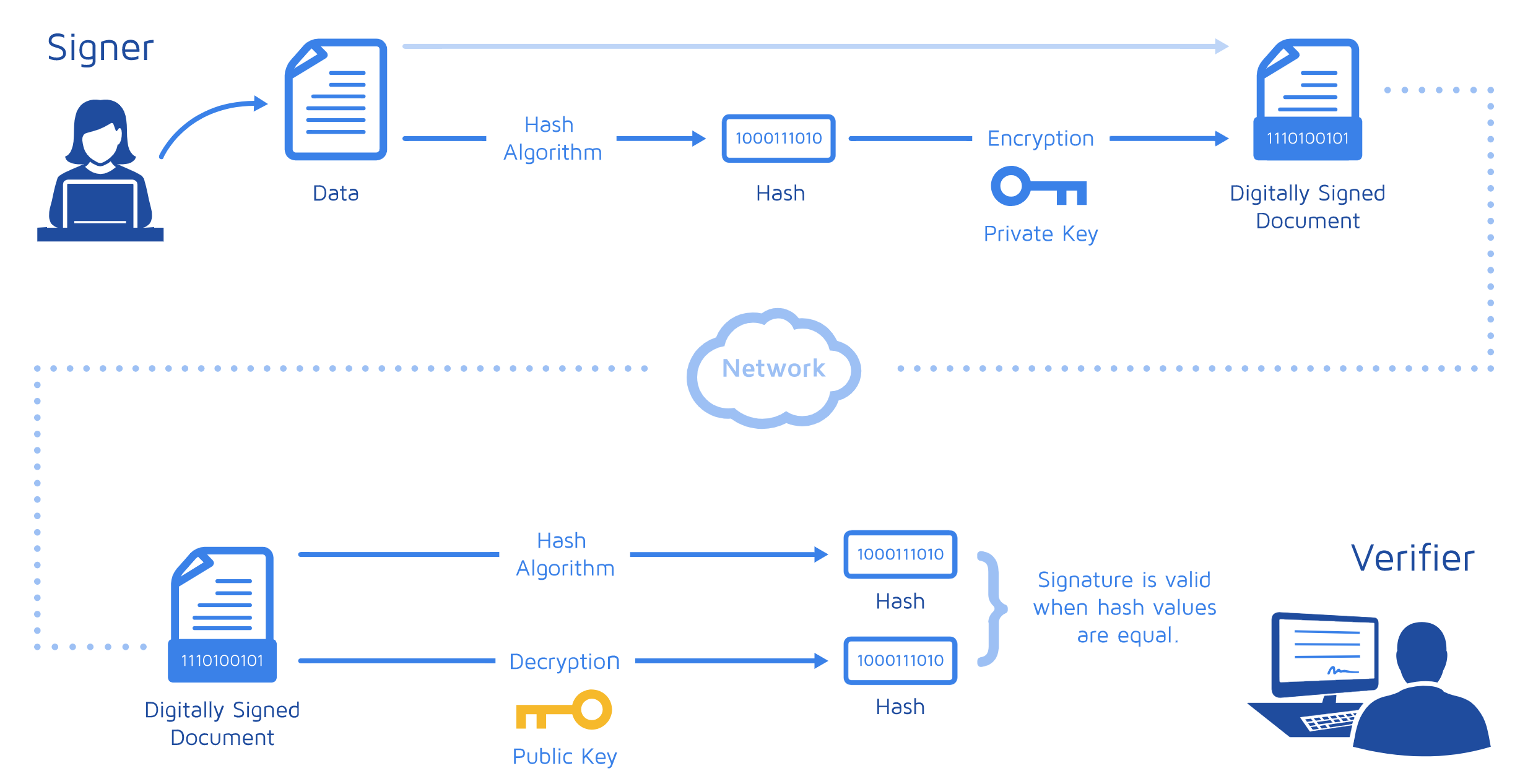
یک آدرس کیف پول از سه طریق موجودی بدست می‌آورد یا از طریق پاداشی است که از استخراج بلوک[[88]](#footnote-89) منتقل شده یا از آدرس دیگری انتقال صورت گرفته و یا به عنوان باقی مانده تراکنش به همان آدرس بازگردد، در هر صورت هر مقدار بیت‌کوین[[89]](#footnote-90) انتقال شده و موجود در آدرس کیف پول یک UTXO است، گره‌های شبکه زمان درخواست انجام یک تراکنش موجودی آدرس فرستنده تراکنش را بررسی می‌کنند تا مطمئن شوند UTXO کافی برای انجام تراکنش وجود دارد.



شکل(۲- 6) سازوکار مفهوم UTXO

همان‌طور که در شکل بالا مشاهده می‌کنید برای انجام این تراکنش 5 بیت‌کوین لازم است جهت تامین این میزان از دو UTXO حاوی 2 و 4 بیت‌کوین از آدرس کیف پول فرستنده استفاده شده که مجموعا 6 بیت‌کوین است که از این میزان 5 بیت‌کوین منتقل می‌شود و 0.01 کارمزد تراکنش کسر میشود و 0.99 بیت‌کوین به عنوان یک UTXO جدید به آدرس کیف پول فرستنده منتقل می‌شود.

نیازمندی مهم دیگر تایید ارسال تراکنش است که از مفهوم امضا دیجیتال[[90]](#footnote-91) استفاده می‌کند، امضا دیجیتال توسط رمزنگاری نامتقارن[[91]](#footnote-92) پیاده‌سازی شده و غیرقابل انکار[[92]](#footnote-93) است و گره‌های شبکه مسئولیت صحت‌سنجی[[93]](#footnote-94) به عهده دارند.



شکل(۲- 7) مراحل ارسال و تایید امضا دیجیتال

مطابق شکل بالا ابتدا هش تراکنش با کلید خصوص فرستنده تراکنش رمز می‌شود[[94]](#footnote-95) و سند رمز شده به همراه هش تراکنش برای صحت سنجی فرستاده می‌شود و گره‌های شبکه با استفاده از کلید عمومی سنده رمز شده را رمزگشایی[[95]](#footnote-96) کرده و هش بدست آمده را با هش تراکنش مقایسه می‌کنند که در صورت یکی بودن این دو امضا تایید می‌شود.

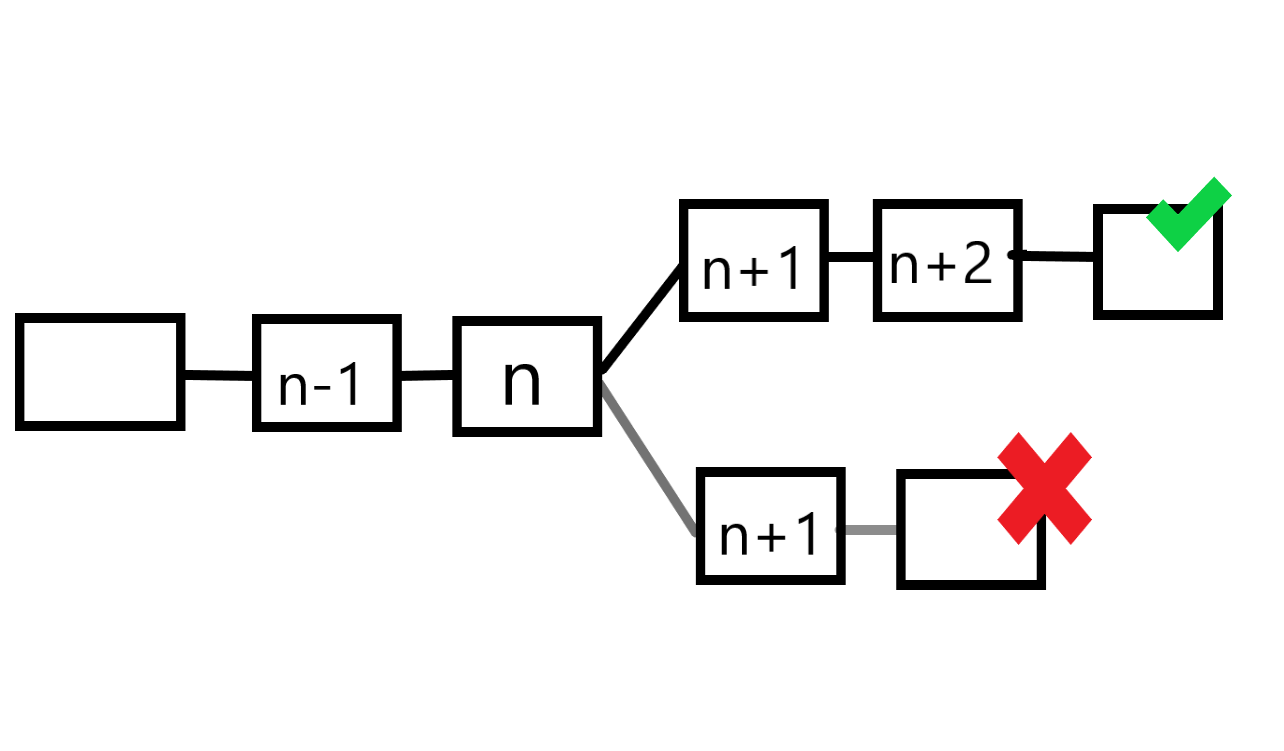
* **سربرگ :** سربرگ هر بلوک حاوی اطلاعاتی شامل سختی شبکه[[96]](#footnote-97)، عدد نانس[[97]](#footnote-98)، هش بلوک قبلی، هش خود بلوک، ریشه درخت مرکل تراکنش‌ها، مهر زمانی[[98]](#footnote-99)، نسخه بلاکچین[[99]](#footnote-100) و شماره بلوک است که در ادامه به توضیح دو مورد اول، سختی شبکه و عدد نانس می‌پردازیم اما قبل از آن باید شناختی نسبت به سازوکاری به نام اثبات کار[[100]](#footnote-101) پیدا کنیم.

همانطور که در قسمت گره‌ها بیان شد گره استخراج وظیفه تایید و استخراج بلوک برای قرارگیری در بلاکچین را دارد که با انجام این کار گره استخراج کننده پاداش دریافت می‌کند برای تعیین این که به کدام گره پاداش تعلق می‌گیرد نیاز به یک توافق یا اجماع میان گره‌ها داریم.

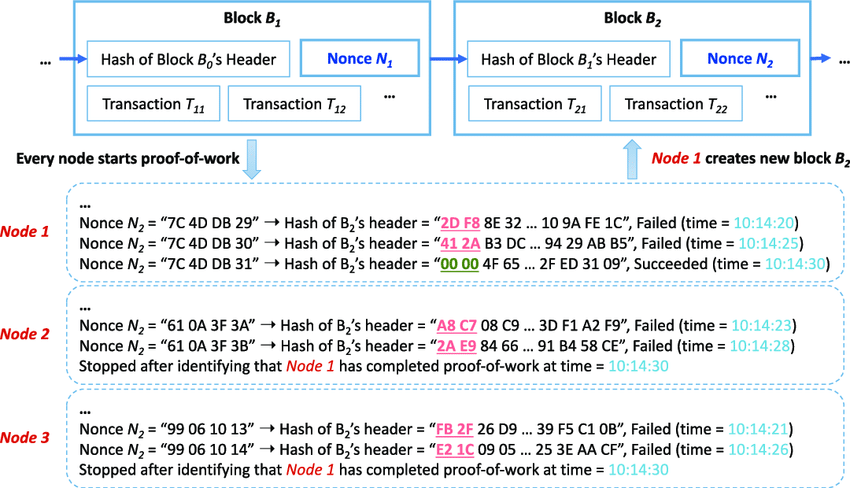
از آنجایی که ماهیت بلاکچین غیرمتمرکز و توزیع شده بوده و هر گره استقلال و آزادی عمل خود را دارد سازوکار اجماع[[101]](#footnote-102) بکارگرفته شده باید قابلیت تحمل خطای بیزانسی[[102]](#footnote-103) را نیز داشته باشد.

با توجه به مطالب بیان شده و نیاز به یک سازوکار اجماع، بلاکچین بیت‌کوین از سازوکار اثبات کار استفاده می‌کند و این شبکه را در مقابل خطای بیزانسی درصورتی که حداکثر یک سوم از گره‌ها غایب و یا خائن باشند مقاوم می‌کند.

سازوکار اثبات کار در سال ۱۹۹۳ و برای جلوگیری از ارسال هرزنامه‌ها[[103]](#footnote-104) با درگیر کردن پردازنده[[104]](#footnote-105) ارسال- کننده جهت هزینه‌بر کردن ارسال ایمیل[[105]](#footnote-106) بوجود آمد ولی مطرح شدن آن به واسطه بلاکچین انجام گرفت، در بلاکچین به این گونه است؛ شبکه بلاکچین تحت عنوان سختی شبکه تعیین میکند که هش بلوکی که مجاز است به بلاکچین اضافه شود باید شرایط خاصی داشته باشد، به طور مثال ۴ کاراکتر[[106]](#footnote-107) اول آن صفر باشد[[107]](#footnote-108)، سپس گره‌های استخراج برای اضافه کردن این بلوک باید عددی تحت عنوان عدد نانس حدس بزنند[[108]](#footnote-109) که هش بلوک طبق آن شرایط بدست بیاید، اولین گره‌ای که بتواند این عدد را به درستی بدست آورد آن بلوک را استخراج کرده و به بلاکچین اضافه می‌شود و همچنین پاداش به همراه کارمزدهای تراکنش‌های موجود در ان بلوک را بدست می‌اورد، در صورتی که دو گره همزمان دو گره دو بلوکی را استخراج که هر دو به بلوک قبلی اشاره می‌کنند، هر دو به بلاکچین اضافه می‌شوند که در اینجا یک انشعاب موقت[[109]](#footnote-110) رخ می‌دهد و آن بلوکی در شبکه میماند که بلوک‌های بیشتری به آن متصل شوند به عبارتی طولانی‌ترین زنجیر[[110]](#footnote-111) در شبکه ادامه پیدا می‌کند و به بلوک یا بلوک‌های در زنجیر ادامه نیافته به اصطلاح بلوک یتیم[[111]](#footnote-112) گفته می‌شود [5].

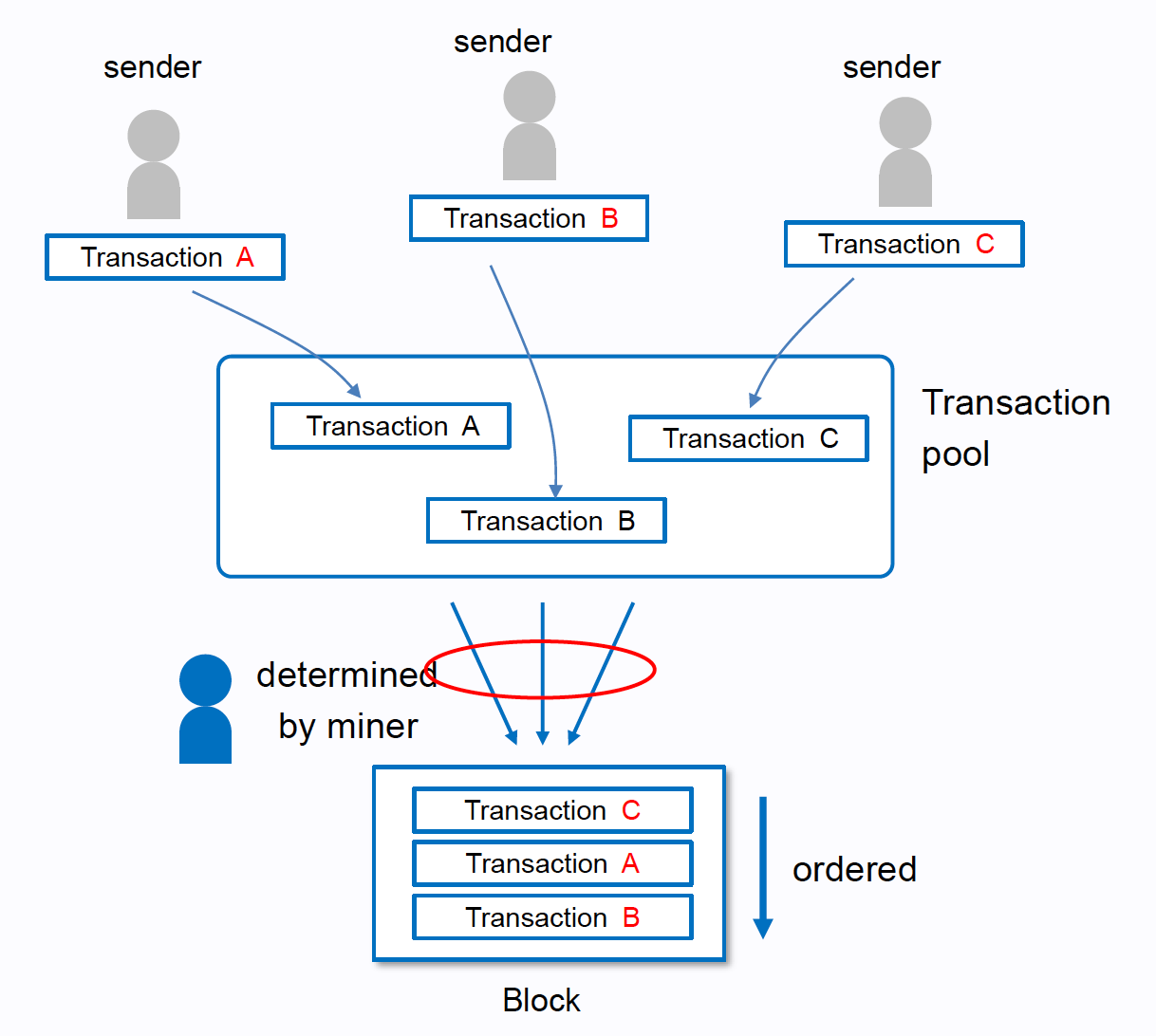
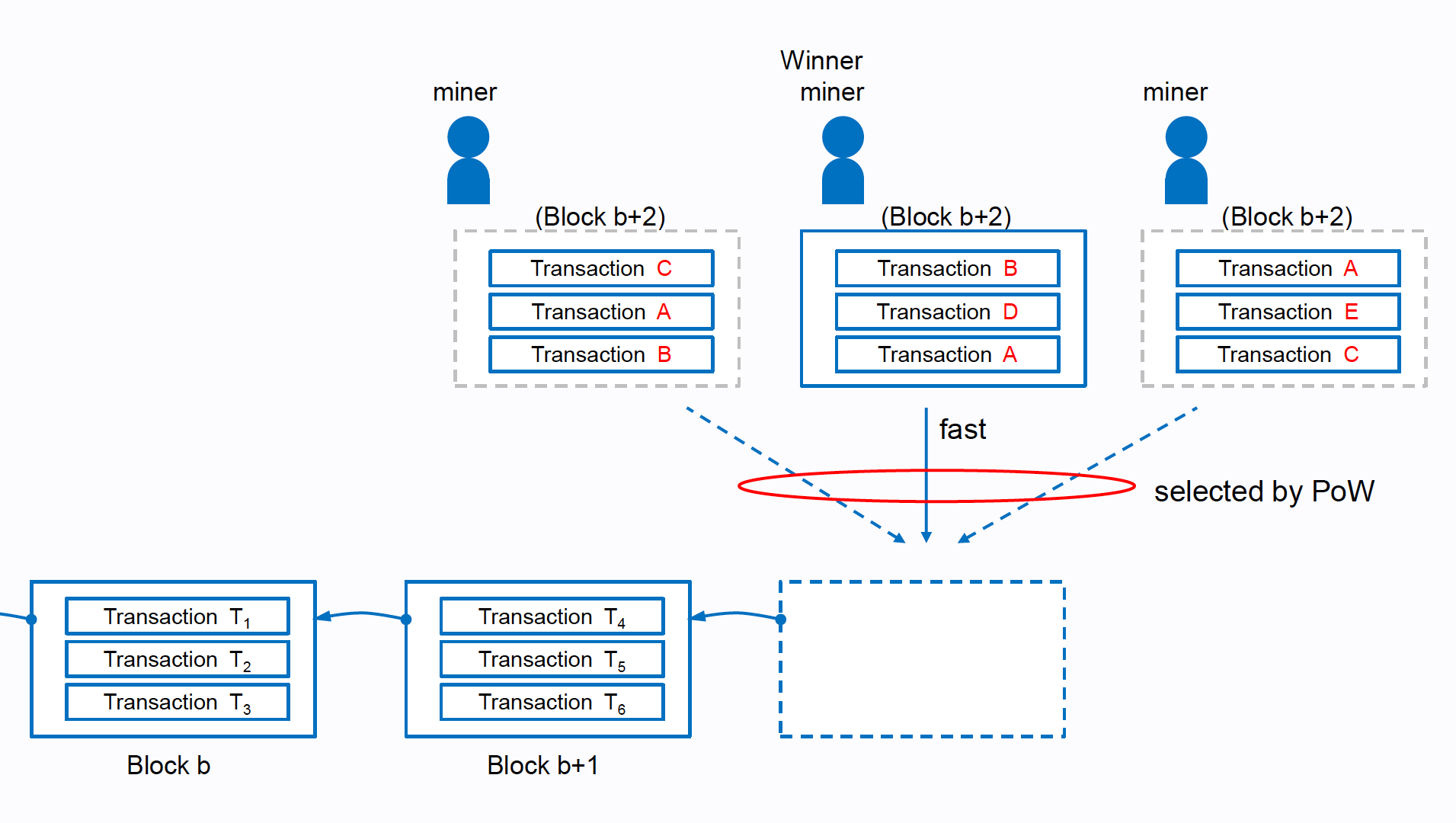


شکل(۲- 8) انشعاب موقت و انتخاب طولانی‌ترین زنجیر ساخته شده



شکل(۲- 9) سازوکار اثبات کار در بلاکچین

در هر بلوک که گره‌های استخراج سعی در اضافه کردن آن به بلاکچین دارند تعدادی تراکنش وجود دارد؛ این تراکنش‌ها بعد از ایجاد در جایی به نام استخر تراکنش‌ها[[112]](#footnote-113) قرار می‌گیرند و گره‌های استخراج از آنجا تراکنش‌ها را با توجه به میزان کارمزد هر تراکنش و مسائل دیگر به بلوکی که قصد اضافه کردن آن به بلاکچین را دارند اضافه می‌کنند.



شکل(۲- 10) روند قرارگیری تراکنش‌ها در بلوک

سرعت اضافه شدن بلوک به بلاکچین در هر بلاکچینی متفاوت است به طور مثال در بیت‌کوین متوسط زمان اضافه شدن هر بلوک باید ۱۰ دقیقه باشد که برای تنظیم آن از سختی شبکه استفاده می‌شود، سختی شبکه با توجه به سرعت استخراج کنندگان هر چند مدت تغییر می‌کند که برای بیت کوین هر ۲۰۱۶ بلوک که ۲ هفته می‌شود[[113]](#footnote-114) بررسی می‌شود و سختی شبکه کم یا زیاد می‌شود.

### ویژگی‌ها

یک شبکه بلاکچین به علت غیرمتمرکز و توزیع شده بودن آن علاوه بر مزایای این نوع سیستم‌ها که امنیت و استقلال شبکه را به ارمغان می‌اورد به علت سازوکار ویژه ای که در بلاکچین تعبیه شده است ویژگی‌های کارآمد و خاصی را برای کاربرانش خواهد داشت، در زیر یک لیست ۱۰تایی از این ویژگی‌ها ارائه شده است :

جدول(۲- 1) ویژگی‌های بلاکچین

|  |  |
| --- | --- |
| ویژگی | توضیح |
| امنیت | به علت توزیع شده بودن امنیت اطلاعات و تراکنش‌ها حفظ می‌شود |
| تمرکز زدایی | شبکه متعلق به شرکت و یا شخص نبوده و هر یک از گره‌ها مجاز به عملکرد و تصمیم‌گیری مستقل هستند |
| شفافیت | شبکه به طور عمومی دسترسی همگانی دارد و می‌توان کل شبکه و اتفاقات داخلی آن را رصد کرد |
| غیرقابل تغییر بودن | تمامی اطلاعات و تراکنش‌ها در شبکه به شکلی ذخیره می‌شود که امکان تغییر هیچ‌یک وجود نداشته باشد |
| قابلیت ردیابی | می‌توان هر تراکنش و عملیات در شبکه را به واسطه شناسه مربوطه ردیابی و بررسی کرد |
| ناشناس بودن و حفظ حریم خصوصی | هر کاربر با شماره‌ی ادرس منحصر به فرد خود در شبکه فعال بوده و حریم خصوصی‌اش حفظ می‌شود |
| قابل اعتماد بودن | برای انجام هر نوع تراکنش در شبکه نیاز به واسطه‌ای جهت اعتبارسنجی نیست |
| مداوم و بدون وقفه | بصورت ۲۴/۷ شبکه به روز و کارا است |
| کاهش هزینه | به‌طور کلی بلاکچین می‌تواند هزینه‌های مربوط به حسابرسی/حساب‌داری، انجام معامله، ذخیره‌سازی و واکشی اطلاعات را کاهش دهد |
| کاهش مصرف کاغذ | استفاده از این شبکه می‌تواند به صورت چشمگیری مصرف کاغذ را کاهش دهد |

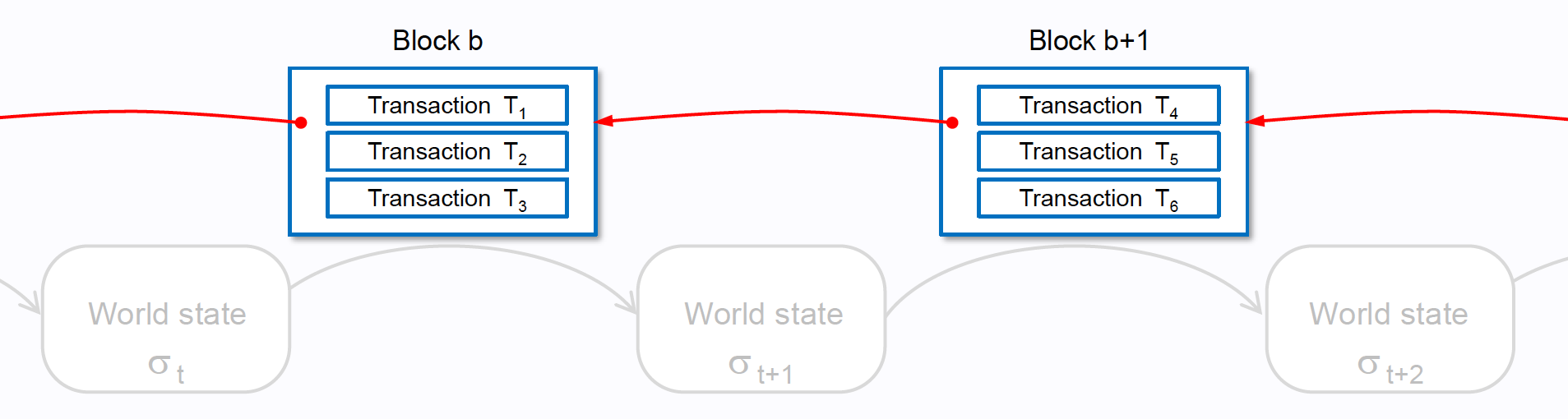
## اتریوم و قراردادهای هوشمند

آنچه تا اینجا بیان شد مربوط به بلاکچین بیت‌کوین بود که اکثر مطالب و منطق کلی آن بین تمام بلاکچین‌ها یکسان است، در این قسمت قصد داریم بلاکچین اتریوم را به عنوان مهمترین بلاکچین بعد از بیت‌کوین با قابلیت تعریف و اجرای قراردادهای هوشمند بررسی کنیم ولی قبل از آن یک تعریف جامع از بلاکچین اتریوم داشته باشیم به این شکل است که در اصل اتریوم تلفیق دو تکنولوژی بلاکچین و ماشین مجازی محاسباتی[[114]](#footnote-115) بوده که قادر به اجرای بدون وقفه[[115]](#footnote-116) و تغییرناپذیر[[116]](#footnote-117) هر برنامه کامپیوتری است [6].

### ساختار داده و انواع حساب در اتریوم

سازوکار بلاکچین اتریوم به طور کلی مانند سازوکار بلاکچین بیت‌کوین است ولی تفاوت‌های مهمی بین این دو بلاکچین وجود دارد که هر یک را متمایز از دیگری می‌کند یکی از این تمایزها استفاده اتریوم از ساختارداده‌ای از ترکیب دو ساختمان داده درخت مرکل و درخت پیشوندی[[117]](#footnote-118) و الگوریتم عملی برای بازیابی اطلاعات کدشده به صورت الفبای عددی[[118]](#footnote-119) است که به آن [[119]](#footnote-120)Merkle Patricia Trie می‌گویند و امکان جستجو و یافتن سریع اطلاعات را می‌دهد، پایگاه داده مورد استفاده اکثر گره‌های شبکه برای ذخیره‌سازی این ساختار داده پایگاه داده Level DB است که از نوع کلید/مقدار[[120]](#footnote-121) است [7] [8].

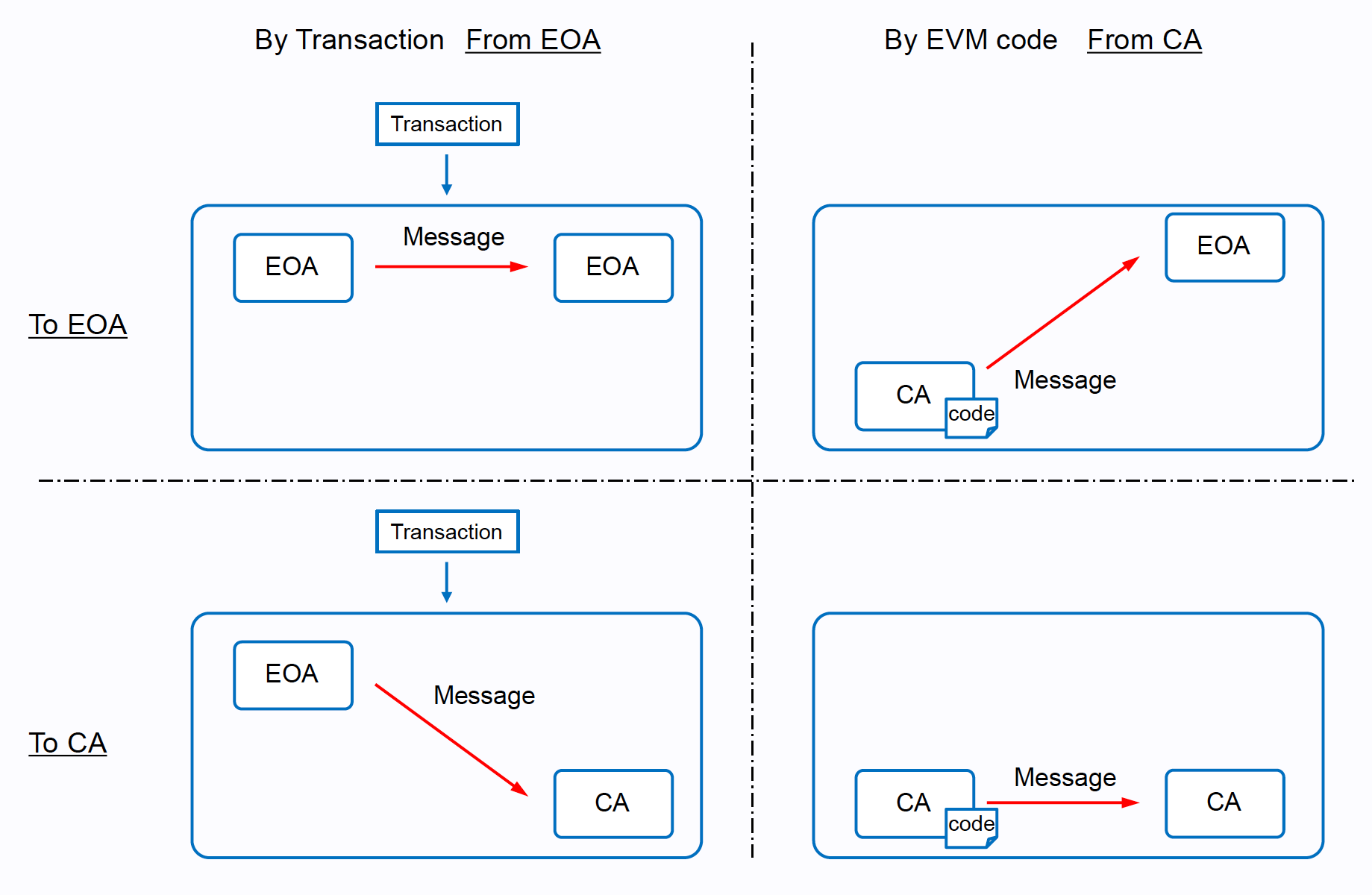
بلاکچین اتریوم بجای یک دفتر کل توزیع شده یک ماشین حالت توزیع شده[[121]](#footnote-122) است که این مدل ساختاری با استفاده از ساختارداده Merkle Patricia Trie امکان ایجاد قرارداد‌های هوشمند و ذخیره سازی حساب‌ها و موجودی هر حساب و اطلاعات مربوط به قرارداد هوشمند را می‌دهد، اگر از دید وضعیت به اتریوم نگاه کنیم می‌توان گفت که اتریوم زنجیره ای از وضعیت‌ها[[122]](#footnote-123) است.



شکل(۲- 11) بلاکچین اتریوم به عنوان ماشین‌ حالت توزیع شده

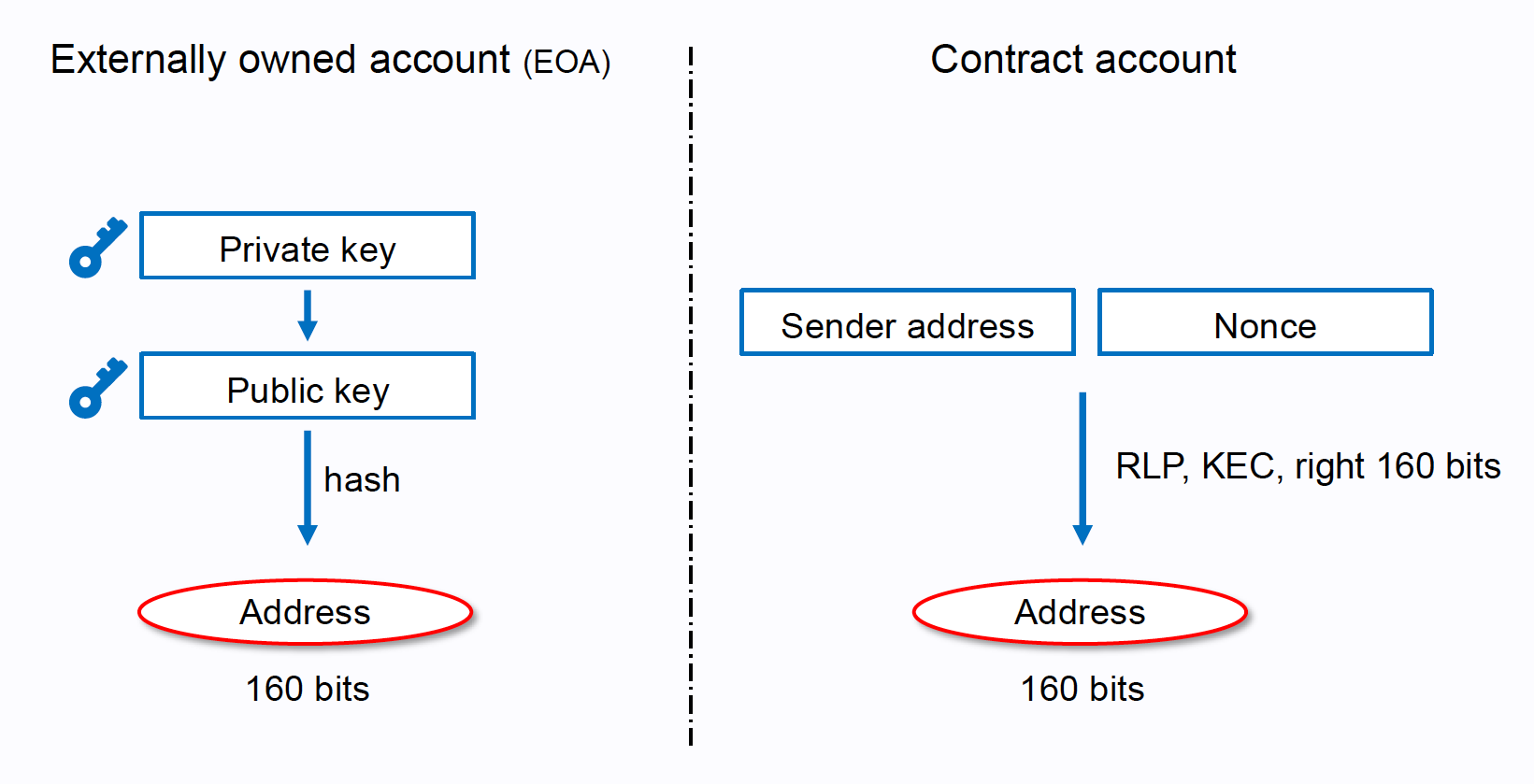
در بلاکچین اتریوم دو نوع آدرس حساب داریم :

1. **حساب‌های با مالکیت خارجی[[123]](#footnote-124) :** این نوع حساب‌ها از همان نوع آدرس کیف پول است که شامل کلید خصوصی جهت امضا تراکنش‌ها، تراکنش مالی ویا فعال کردن و تعامل با کد قرارداد ، می‌باشد. این حساب توسط کاربر مدیریت شده و می‌تواند به هر دو نوع حساب تراکنش ارسال کند و همچنین می‌توان از طریق این نوع حساب با قرار دادن آدرس صفر[[124]](#footnote-125) به عنوان آدرس گیرنده حساب‌های مبتنی بر قرارداد ایجاد کرد.
2. **حساب‌های مبتنی بر قرارداد[[125]](#footnote-126) :** این حساب دارای کلید خصوصی نبوده و توسط کد قرارداد هوشمند کنترل می‌شود، اجرا این نوع حساب توسط EOA و یا قرارداد دیگری با انجام می‌گیرد همچنین می‌تواند به هر دو نوع حساب تماس پیام[[126]](#footnote-127) ارسال کند.



شکل(۲- 12) انواع حساب‌ها در اتریوم و تراکنش بین آن‌ها

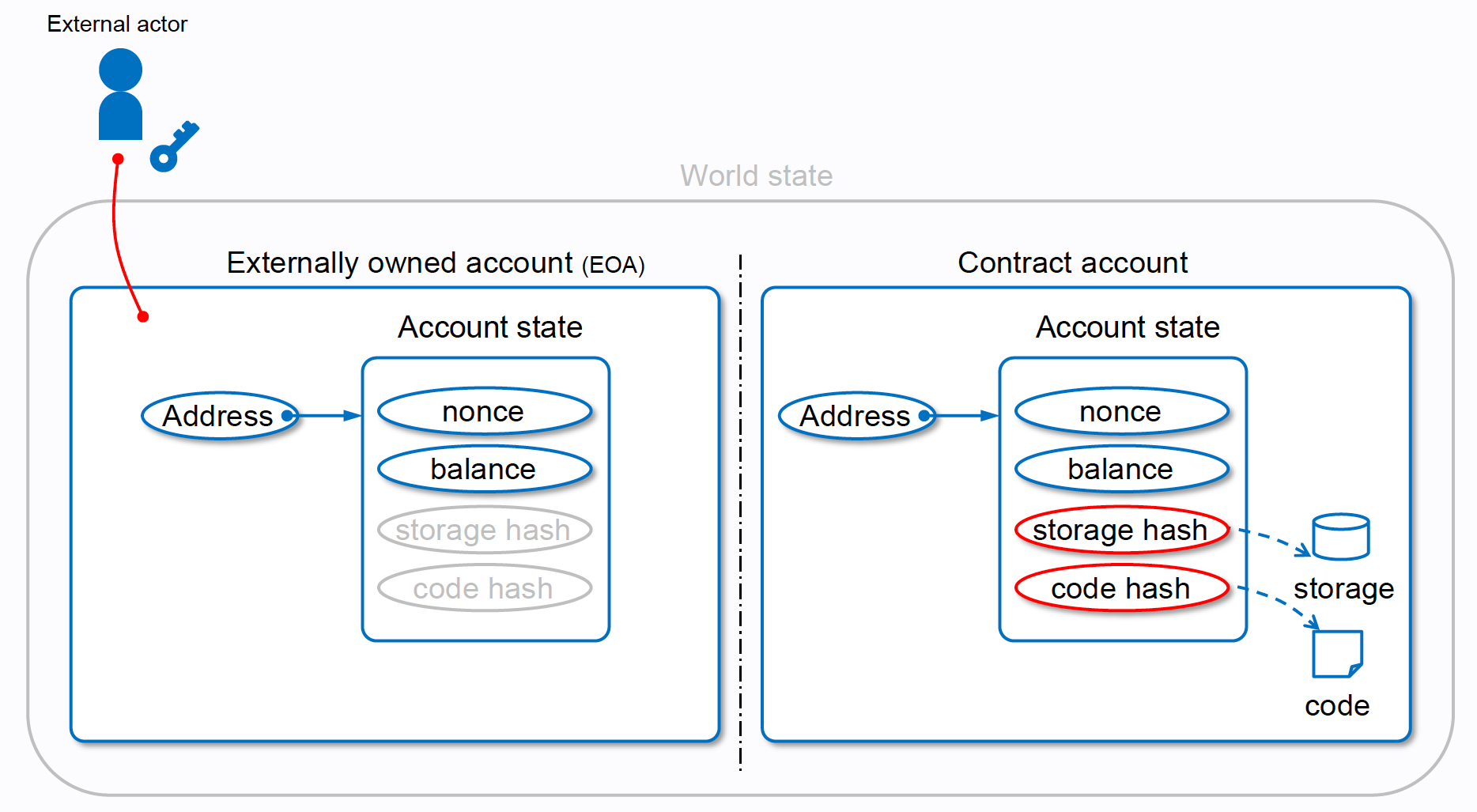
هر دو حساب دارای آدرس با طول ثابت ۲۰ بایت[[127]](#footnote-128) یا ۴۲ نیبل[[128]](#footnote-129) هستند ولی طریقه بدست آوردن هر یک متفاوت است، آدرس EOA ۲۰ بایت از هش کلید عمومی حساب است در اصل زمان درخواست ساخت کیف پول یک داده تصادفی[[129]](#footnote-130) تولید می‌شود سپس با الگوریتم مخصوص هر کیف پول کلید عمومی و خصوصی استخراج می‌شود و ۲۰ بایت از هش کلید عمومی به عنوان آدرس EOA در نظر گرفته می‌شود، آدرس حساب میتنی بر قرارداد از روی آدرس EOA سازنده آن و یک عدد نانس بدست می‌آید.



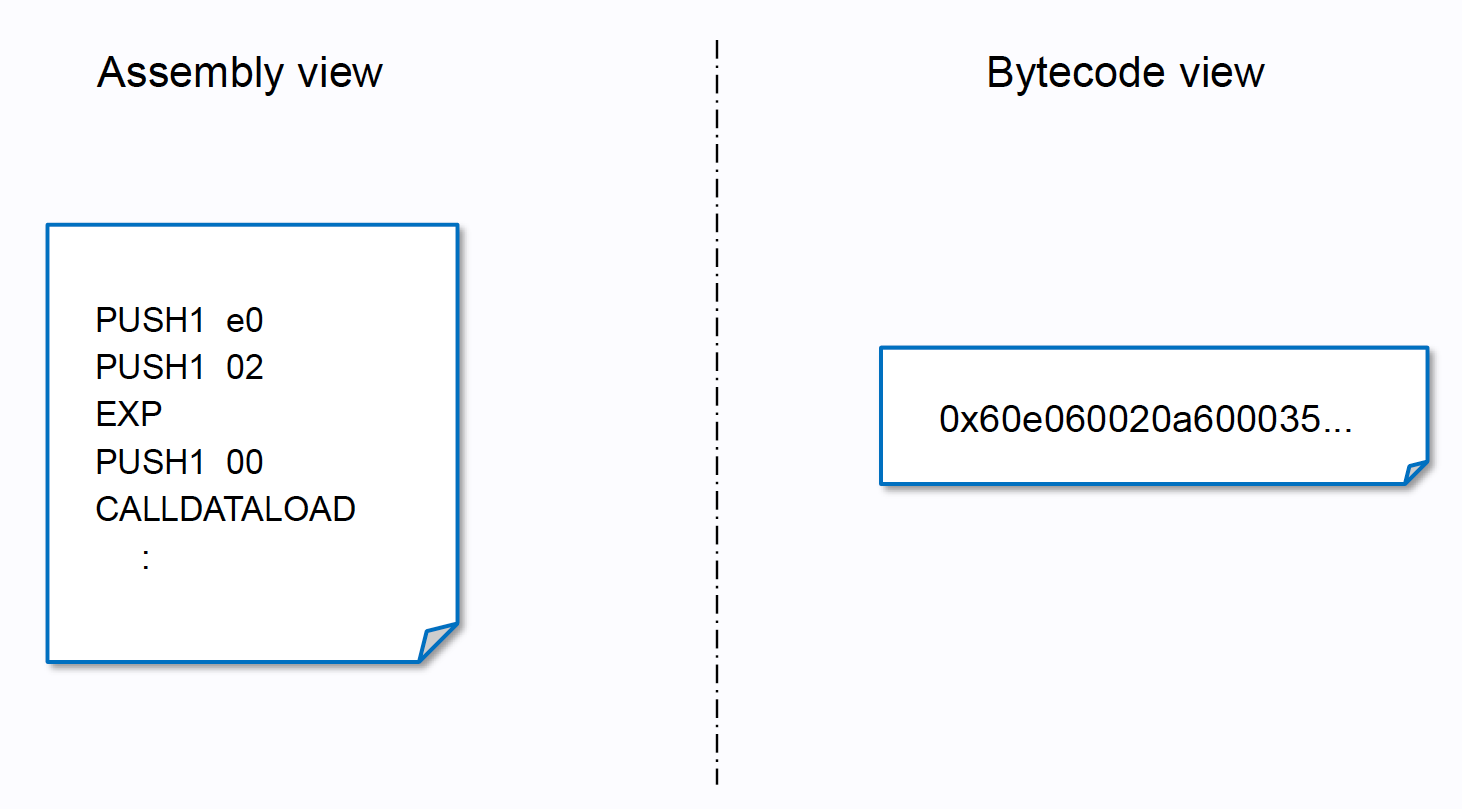
شکل(۲- 13) روند تولید آدرس حساب‌های اتریوم

هر دو نوع حساب از چهار قسمت تشکیل شده :

1. **عدد نانس :** عدد نانس در اینجا با عدد نانسی که گره‌های استخراج باید حدس بزنند متفاوت است، این عدد از صفر شروع شده و برای جلوگیری از خرج مجدد برای هر حساب تعریف شده اما برای هر حساب معنای متفاوتی دارد، این عدد برای یک EOA به این صورت است که به ازای هر تراکنش ارسالی یک واحد به آن اضافه می‌شود ولی برای حساب‌های مبتنی بر قرارداد به ازای هر تماس پیام به قرارداد دیگر یک واحد اضافه می‌گردد
2. **موجودی :** موجودی حساب را به wei[[130]](#footnote-131) نمایش می‌دهد
3. **هش ریشه انبارش[[131]](#footnote-132) :** اطلاعات قرارداد شامل ورودی ها و متغییر‌های وضعیت[[132]](#footnote-133) در این قسمت نگهداری می‌شوند و این قسمت برای EOAها خالی است
4. **هش کد[[133]](#footnote-134)‌ :** هش بایت کد[[134]](#footnote-135) ماشین مجازی اتریوم[[135]](#footnote-136) مربوط به قرارداد است و این قسمت برای EOAها خالی است



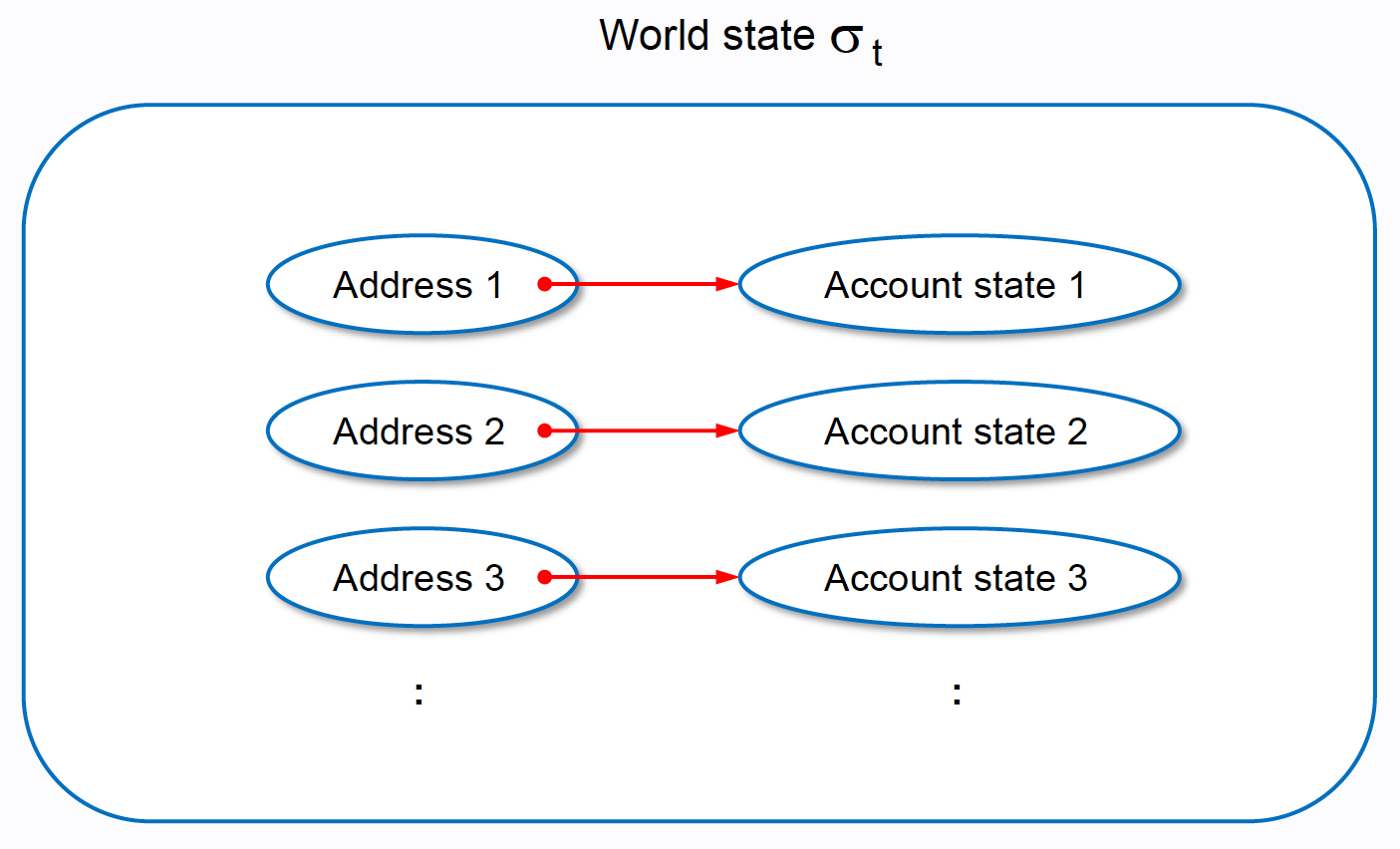
شکل(۲- 14) ساختار داخلی حساب‌ها اتریوم



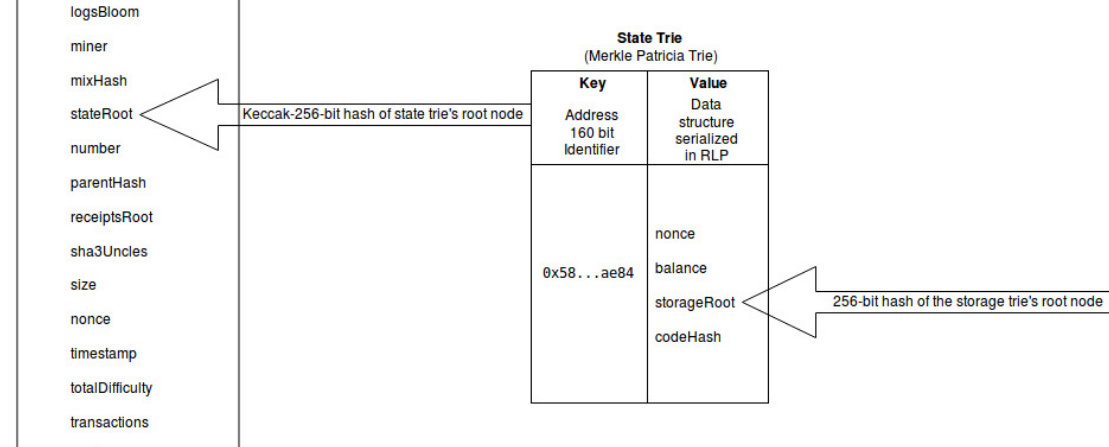
شکل(۲- 15) مقایسه کد اسمبلی و بایت‌کد

اجزای تشکیل دهنده بلوک در بلاکچین اتریوم نیز متفاوت‌تر، بیشتر و پیچیده‌تر از بلوک‌‌های بیت‌کوین است، سربرگ بلوک اتریوم علاوه بر موارد ذکر شده در عموم بلاکچین‌ها دارای چهار مورد مفهومی است:

1. **Uncles** **Hash :** اتریوم به استخراج کننده بلوک‌های یتیم جهت تشویق برای کمک به شبکه درصدی از پاداش و کارمزد‌های آن بلوک را می‌دهد و هش ستخراج کنندگان این بلوک‌ها را در سربرگ خود و لیستی از آن‌ها را در بلوک ثبت می‌کند.
2. **هش ریشه وضعیت جهانی[[136]](#footnote-137) :** وضعیت جهانی یکی از مهمترین مفاهیم در اتریوم است و همانطور که قبل بیان شد اتریوم را می‌توان زنجیره‌ای از وضعیت‌ها در نظر گرفت و حتی می‌توان گفت که پروتکل[[137]](#footnote-138) اتریوم صرفا به منظور حفظ عملکرد مداوم، بدون وقفه و تغییر ناپذیر این ماشین حالت[[138]](#footnote-139) ویژه وجود دارد [6]، منظور از وضعیت جهانی وضعیت و حالتی از کل شبکه است که تمام گره‌های سراسر شبکه یک نسخه یکسان از آن را دارند، وضعیت جهانی طبق ساختارداده Merkle Patricia Trie شکل می‌گیرد و هش ریشه آن در سربرگ بلوک ذخیره می‌شود، وضعیت جهانی به صورت کلید/مقدار در Merkle Patricia Trie ذخیره می‌شود که کلیدها حساب‌ها هستند و مقدار‌ها همان چهار قسمتی است که قبل بیان شد.

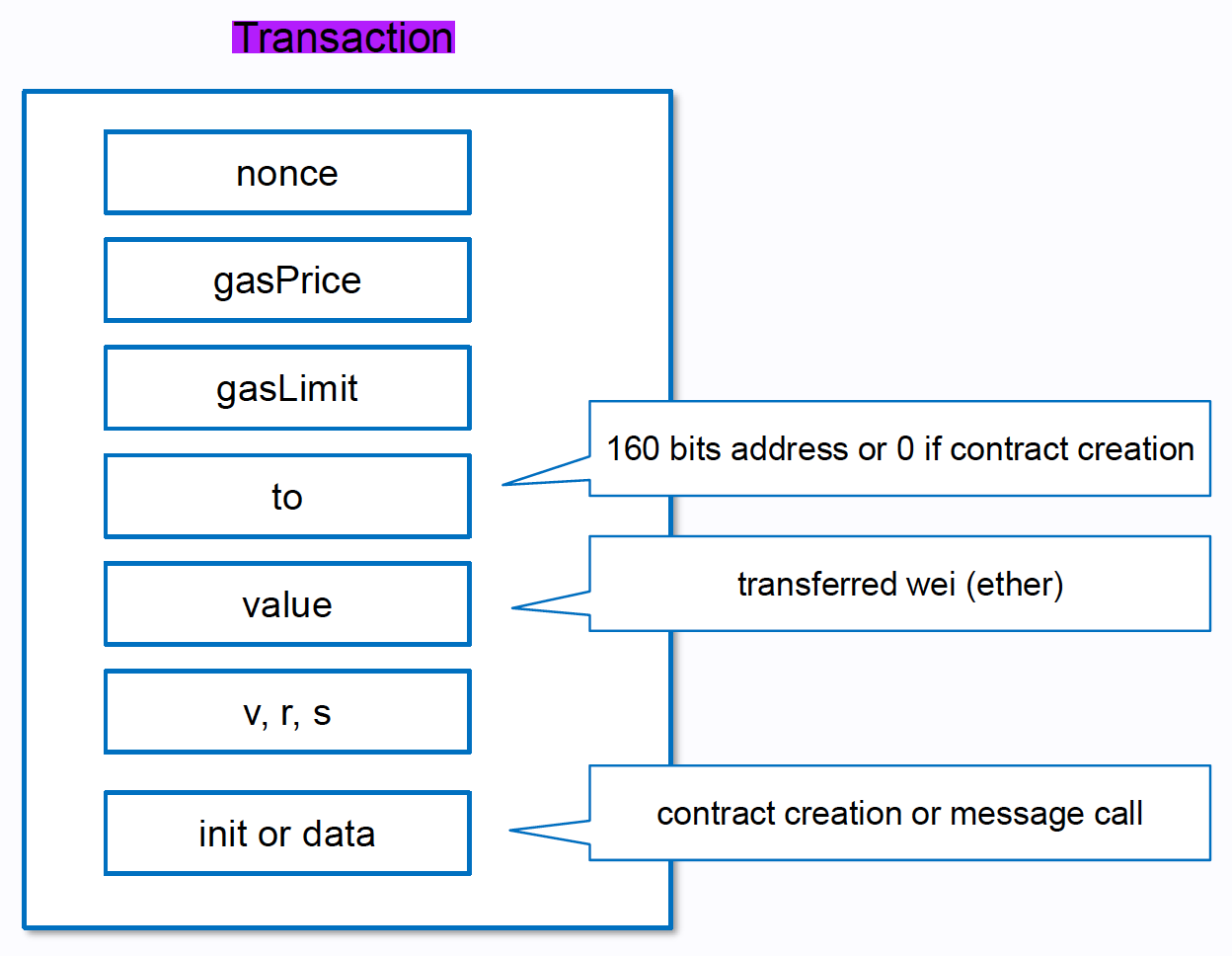


شکل(۲- 16) شمای کلی از یک وضعیت جهانی



شکل(۲- 17) ساختار داخلی یک درخت پیشوندی وضعیت و مدل ذخیره‌سازی آن در بلوک اتریوم

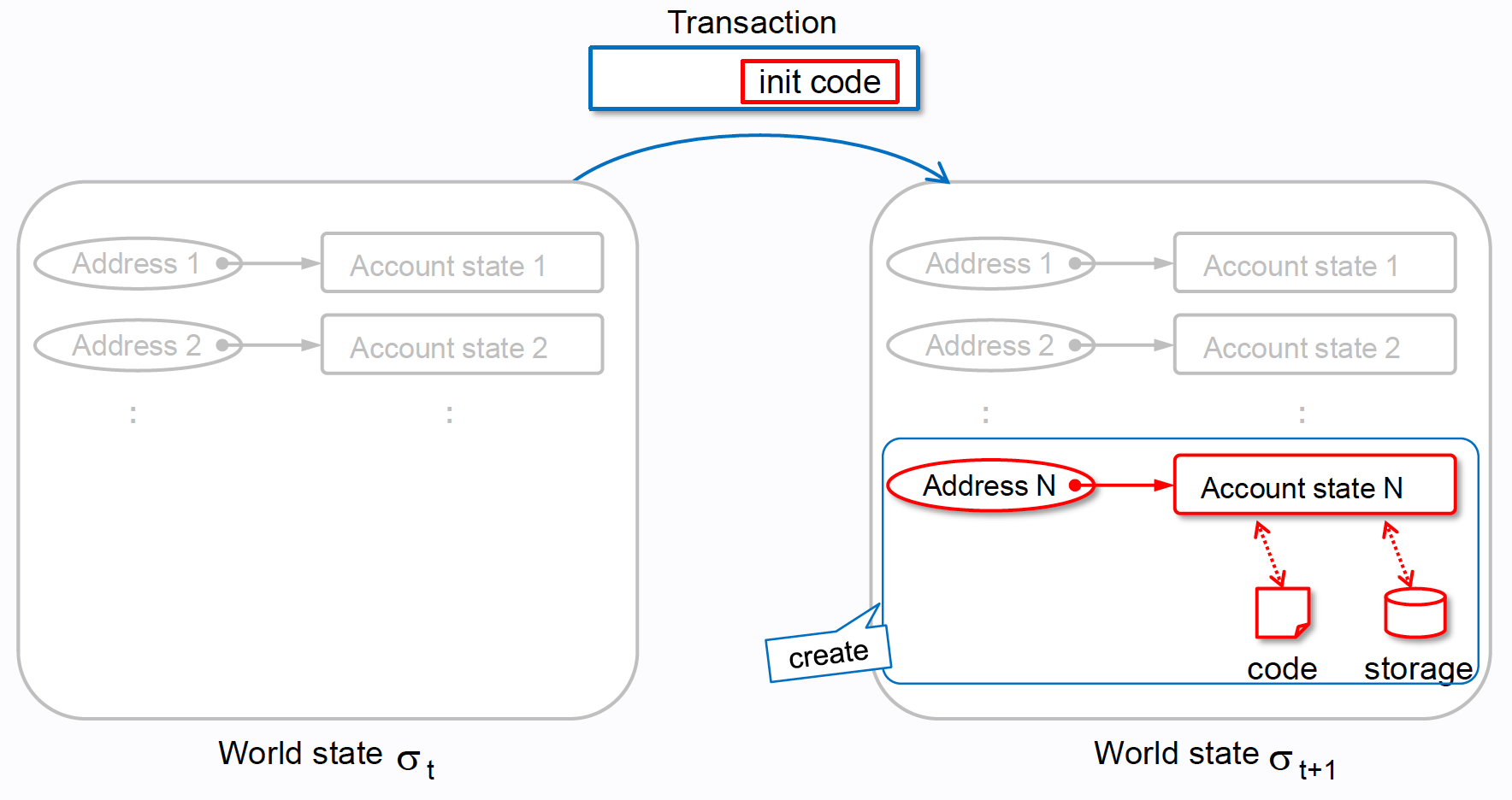
1. **هش ریشه تراکنش‌ها :** تراکنش**‌**ها در اتریوم نیز از ساختارداده Merkle Patricia Trie استفاده می‌کنند و هش ریشه تمام تراکنش‌ها در سربرگ بلوک ذخیره می‌شود، هر تراکنش ذخیره شده از بخش‌های یکسانی مانند شکل ذیل تشکیل می‌شود :



شکل(۲- 18) ساختار داخلی تراکنش‌ها در اتریوم

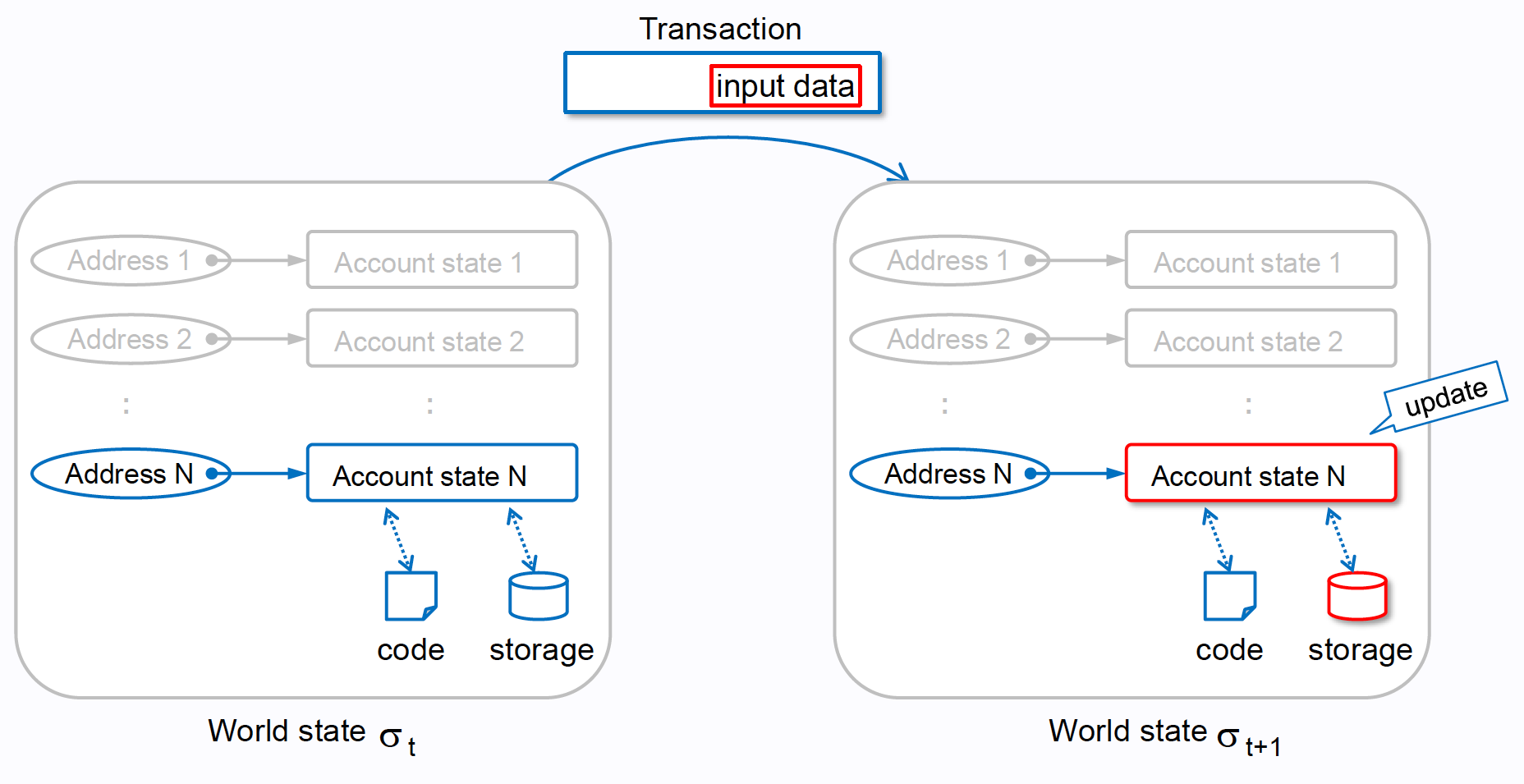
طبق شکل بالا هر تراکنش از اجزای مهمی تشکیل شده که در ادامه به توضیحی از هر یک خواهیم پرداخت

* **Nonce :** عدد نانس در اینجا هم معنی نانس در حساب‌ها است و مشخص می‌کند که حساب ارسال‌ کننده قبلا چند تراکنش موفق و تایید شده ارسال کرده است
* **گاز[[139]](#footnote-140) :** گاز همان کارمزد تراکنش است که در تمام انواع شبکه‌های بلاکچین وجود دارد ولی در شبکه اتریوم به علت وجود قراردادهای هوشمند ساختار پیچیده‌تری دارد، مفهوم کارمزد در اتریوم علاوه بر ایجاد امنیت و ثبات در شبکه از نوشتن کدهای معیوب مانند حلقه بی‌نهایت[[140]](#footnote-141) برای آسیب به شبکه و کدنویسی بد جلوگیری می‌کند، این کار به این شکل است که در اتریوم دو مفهوم قیمت گاز[[141]](#footnote-142) و محدودیت گاز[[142]](#footnote-143) تعریف می‌شود همچنین این رویکرد باعث می‌شود تا درآمد استخراج‌کنندگان فارغ از تغییرات شدید قیمت اتر ثبات داشته باشد
* **v, r,** **s :** این مقادیر برای امضا رمزنگاری مورد استفاده قرار می‌گیرد جهت تعیین فرستنده تراکنش
* **init or data :** در صورتی که یک حساب EOA تراکنشی مبنی بر ساخت یک قرارداد هوشمند ارسال کند مقدارcode init که کد سازنده[[143]](#footnote-144) کد قرارداد در این بخش نوشته می‌شود و یک قرارداد در وضعیت جهانی ایجاد می‌شود



شکل(۲- 19) شمایی از تراکنش ساخت قرارداد هوشمند

در صورتی که حساب EOA یا حساب قرارداد ورودی به قراردادی داشته باشد مقادیر آن در data ذخیره می‌شود و انبارش وضعیت جهانی را تغییر می‌دهد که این به‌روزرسانی طبق کد قرارداد یا همان کد ماشین مجازی اتریوم، که در ادامه به آن می‌پردازیم، توسط ماشین مجازی اتریوم انجام می‌شود.



شکل(۲- 20) شمایی از تراکنش با قرارداد و تغییر متغییرهای قرارداد

1. **هش ریشه رسید تراکنش‌ها :** در این قسمت رسید تراکنش‌های موفق و به طور کامل انجام شده طی همان ساختارداده مطرح شده در اتریوم ذخیره شده و هش ریشه آن در سربرگ بلوک ثبت می‌شود.

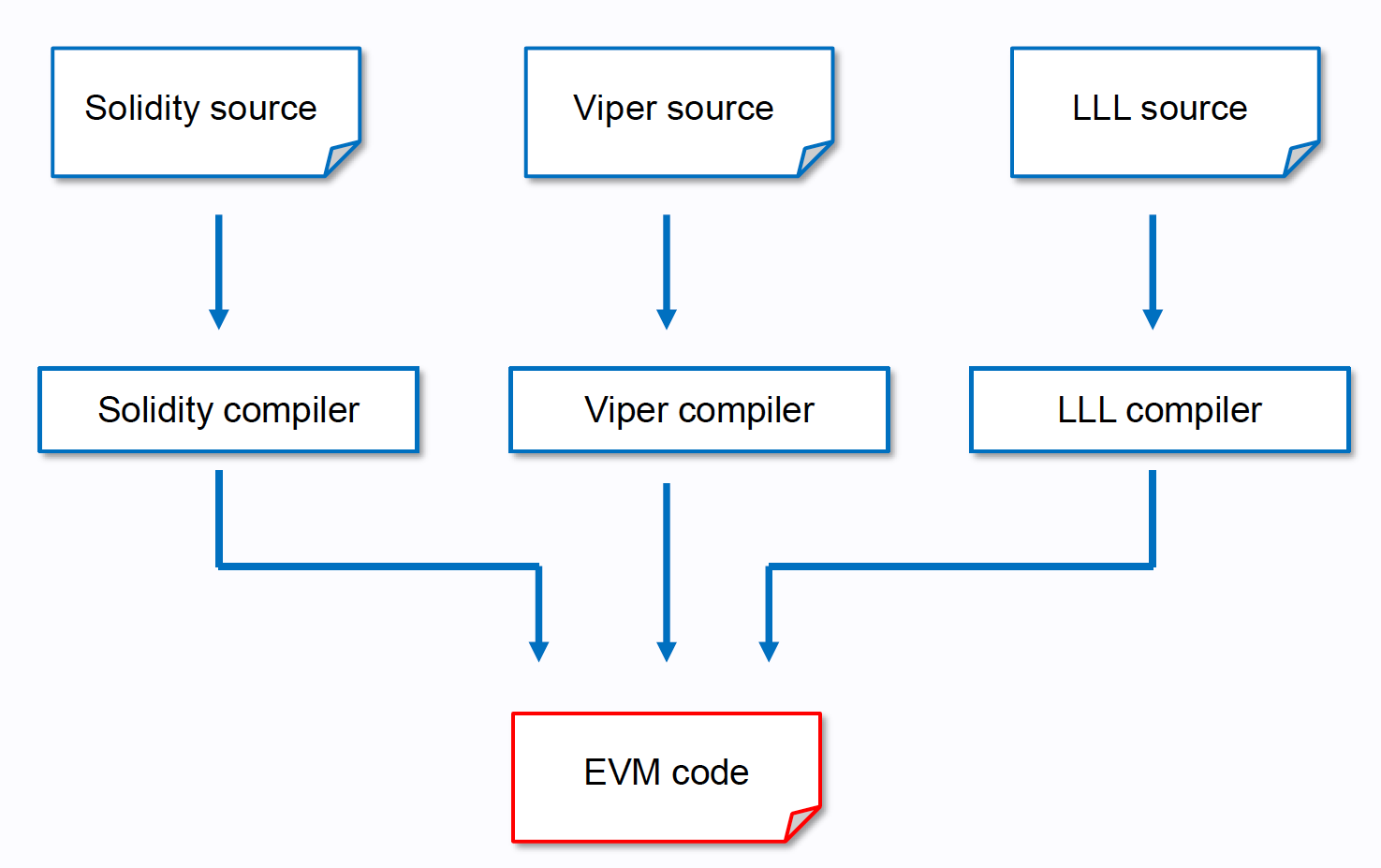
### ماشین مجازی اتریوم

قرارداد هوشمند یک قرارداد خوداجرا شونده[[144]](#footnote-145) است که شرایط قرارداد طرفین در یک کد برنامه نویسی شده و در سراسر شبکه بلاکچین غیرمتمرکز و توزیع شده قرار می‌گیرد، قراردادهای هوشمند اجازه می‌دهند تا معاملات و توافقات قابل اعتماد بین طرف‌های ناهمگون و ناشناس بدون نیاز به یک مرجع مرکزی، سیستم قانونی یا مکانیزم اجرایی خارجی انجام شود.

تعریف قرارداد هوشمند به این شکل با کمک بلاکچین اتریوم و ماشین مجازی آن تحقق یافت اگرچه ایده اولیه قرارداد‌های هوشمند توسط دانشمند کامپیوتر و رمزنگاری، نیک سابو[[145]](#footnote-146) در سال ۱۹۹۴ مطرح شد اما در آن زمان فضای مناسب برای تحقق این ایده‌ وجود نداشت.

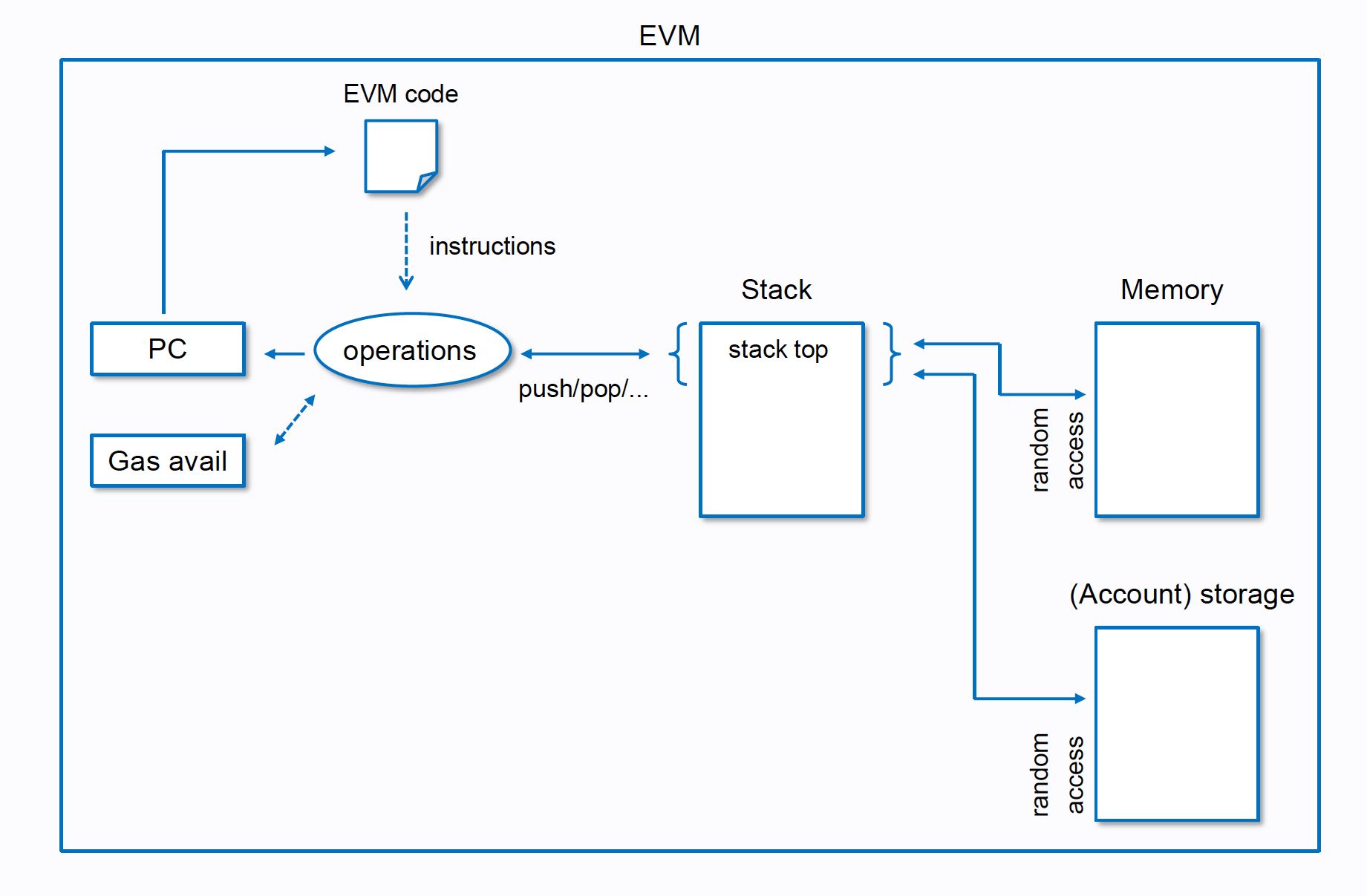
کد قراردادهای هوشمند اتریوم توسط ماشین مجازی اتریوم روی همه گره‌های موجود در بلاکچبن اتریوم اجرا می شوند که کار اجرای کد به عنوان بخشی از وظیفه گره‌ها برای تایید بلاک های جدید محسوب می شود، به همین علت کد قرارداد هوشمند را کد EVM نیز می‌نامند و از این رو شناخت ماشین مجازی اتریوم بسیار مهم است.

کد قرارداد به بایت‌کد بوده و نوشتن آن به این صورت برای برنامه‌ نویسان تقریبا غیر ممکن است به همین جهت زبان‌های برنامه نویسی سطح بالایی مثل سالیدیتی، وایپر[[146]](#footnote-147)، LLL [[147]](#footnote-148) و ... بوجود امدند.



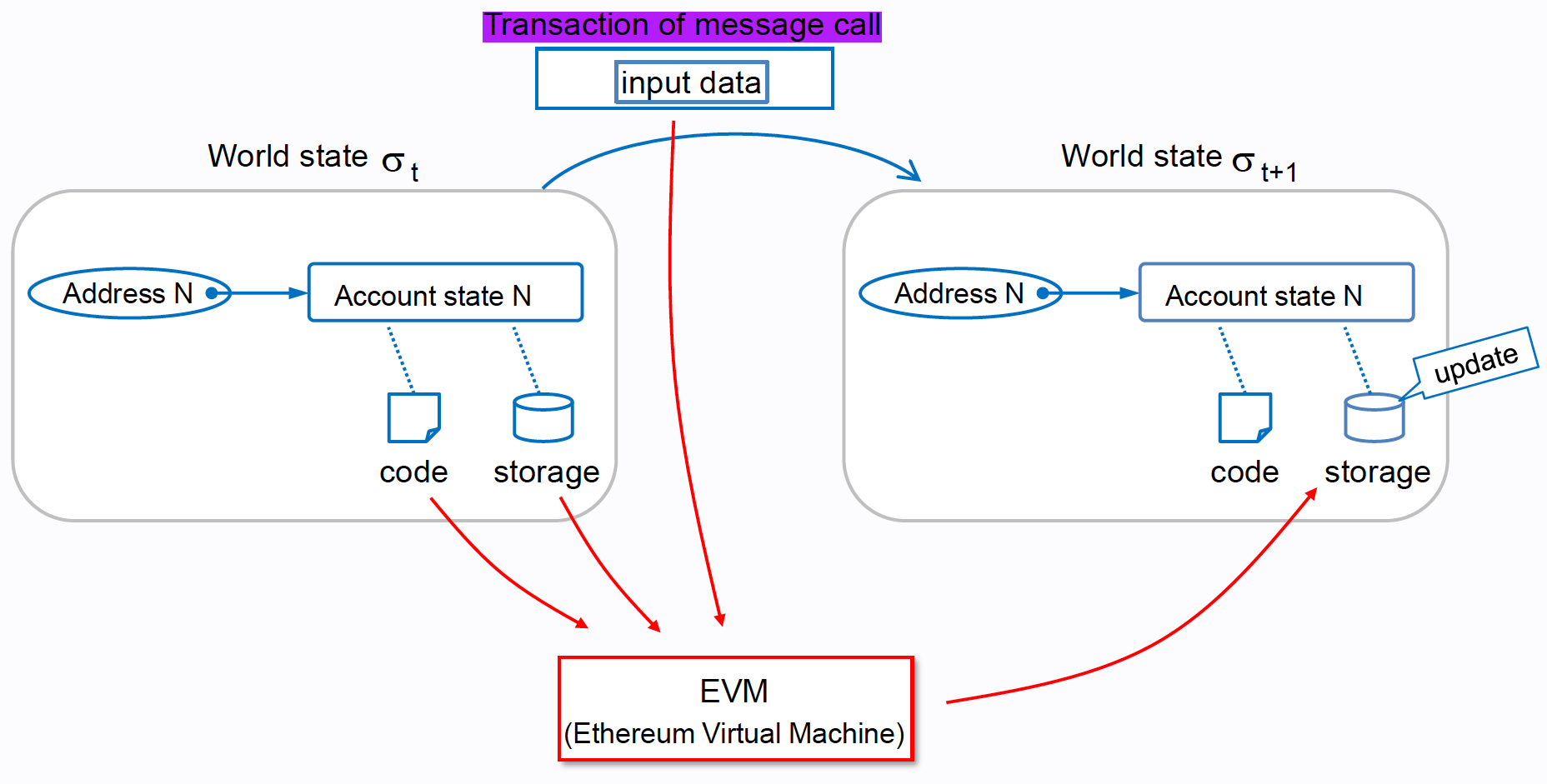
شکل(۲- 21) تبدیل کد از زبان سطح بالا به بایت‌کد

بایت‌‌کد قرارداد کامپایل[[148]](#footnote-149) شده به صورت تعدادی از آپ‌کد[[149]](#footnote-150)های EVM مانند XOR، AND، ADD، SUB و ... اجرا می شود، از این رو ماشین مجازی اتریوم یک ماشین تورینگ کامل[[150]](#footnote-151) بوده و می‌تواند عملکردی مانند یک کامپیوتر داشته باشد به همین جهت و به واسطه وجود EVM در اتریوم، این بلاکچین را یک کامپیوتر جهانی به حساب می‌آورند



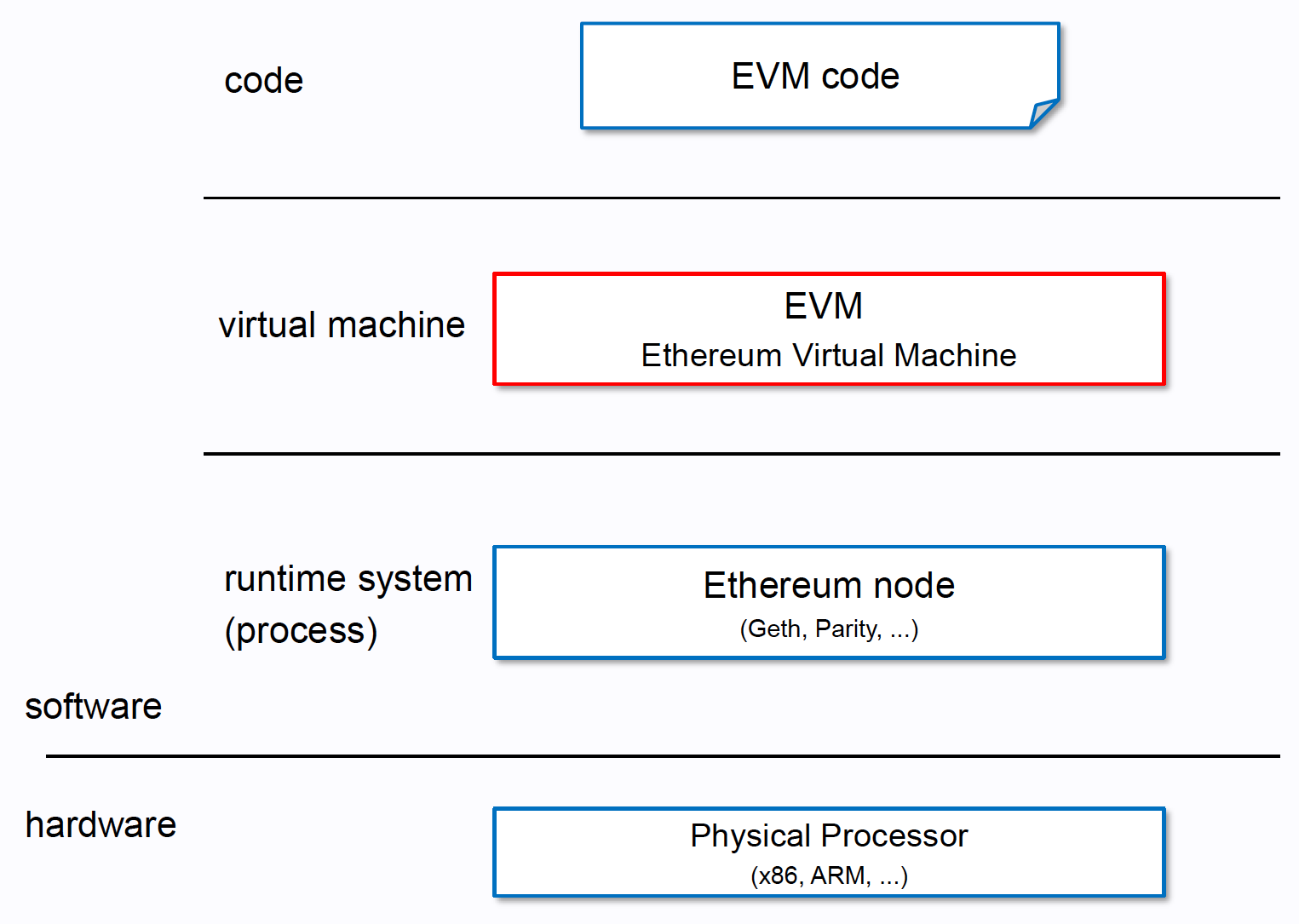
شکل(۲- 22) نمای داخلی و سازوکار EVM

وظیفه اصلی ماشین مجازی اتریوم تغییر از یک حالت جهانی به یک حالت جهانی است که این کار طبق کد‌های قرارداد است که بر روی آن نوشته شده و EVM آن‌ها را اجرا می‌کند، در اصل EVM مثل یک تابع انتقال حالت قطعی[[151]](#footnote-152) نامید که وضعیت جهانی معتبر قبلی را به همراه مجموعه ای جدید از تراکنش‌های معتبر را به عنوان ورودی تابع می‌گیرد و حالت جهانی معتبر جدید را به عنوان خروجی تولید می‌کند.



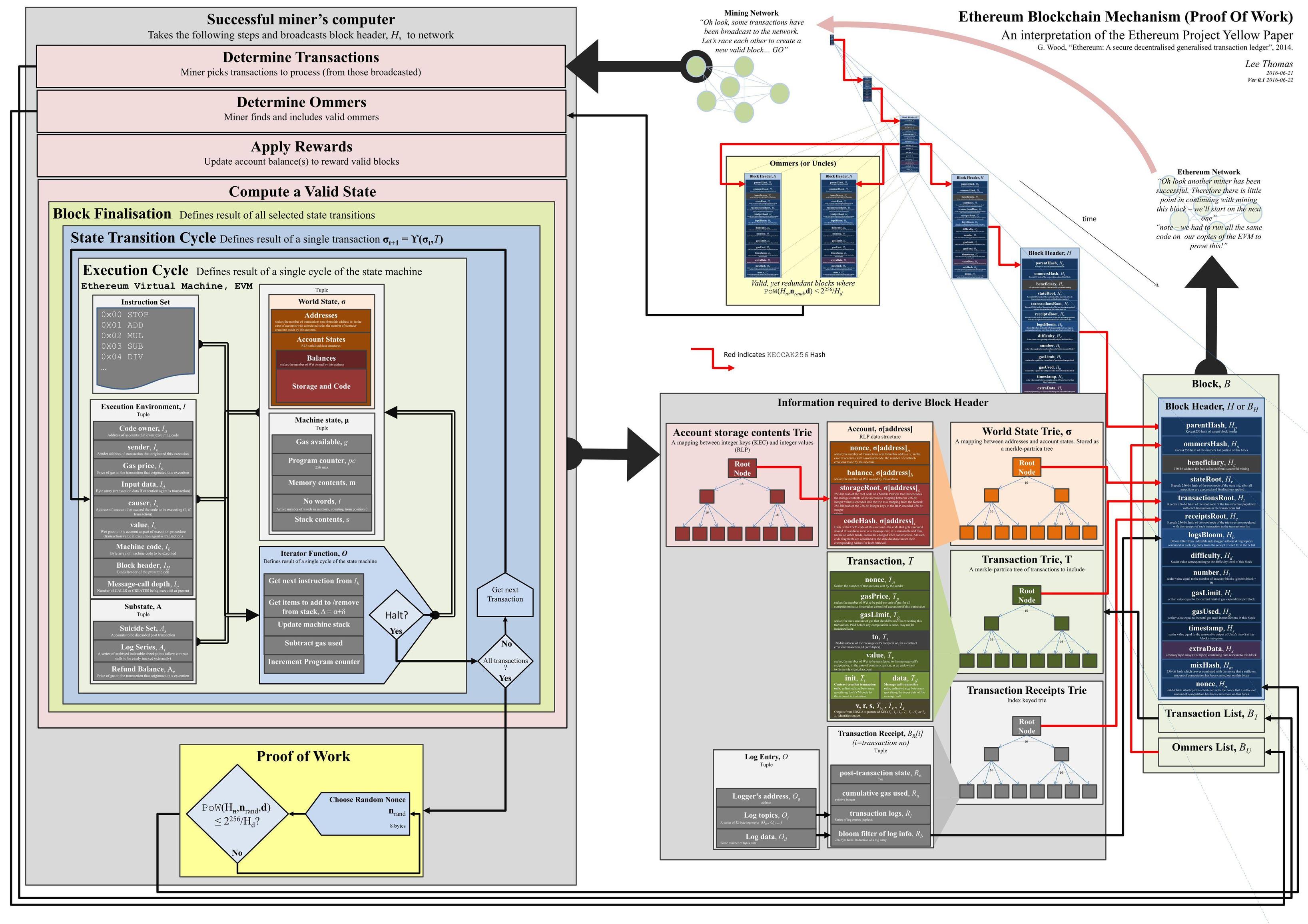
شکل(۲- 23) EVM به عنوان تغییردهنده وضعیت

از نظر ساختاری وضعیت کدهای قرارداد و EVM به شکل ذیل بوده و گره‌های شبکه موظف به اجرا کردن نرم‌افزار مخصوص جهت پیاده سازی EVM هستند.



شکل(۲- 24) نمای کلی از جایگاه کد‌های قرارداد، EVM، نرم افزارهای اجرا کننده گره، سخت‌افزار

در پایان این بخش قصد داریم یک نمای کاملی از کل شبکه بلاکچین اتریوم با توجه به مطالب بیان شده در شکل ذیل داشته باشیم :



شکل(۲- 25) نمای کاملی از ساختار کلی شبکه اتریوم

## نتیجه‌گیری

# بلاکچین در آموزش آنلاین

## مقدمه

از ابتدای حضور انسان در این کره خاکی همواره نیاز به آموزش از نیازهای حیاتی بشر بوده، گونه انسان از ابتدای وجودی خود به واسطه بهروی از مغزش و ساخت ابزارالات توانسته گونه خود را حفظ نماید و تسلط و شناختش بر جهان را گسترش دهد.

مطالعات بر روی انسان نشان می‌دهد که این گونه در حدود ۳.۳ میلیون سال قبل و در دوران قبل از تاریخ، دوره ای قبل از توانایی مکتوب کردن تاریخ، موضوع آموزش وجود داشته.

آموزش؛ این عضو جدایی‌ناپذیر از انسان در هر دوره از سیر تکاملی بشر نقش مهم و کلیدی داشته و در نقاط عطف تاریخی همگام با اختراعات نوآورانه و جدید بشر خود را به روز کرده است. [9]

بهره‌وری از تکنولوژی روز هر دوره برای آموزش در ابتدا با هدف بهبود و تسهیل فرآیند آموزش بوده و از نظر تاریخی از دوره غارنشینی انسان با استفاده از ابزار اولیه مانند نقاشی روی دیواره‌های غار شروع شده [10] رفته رفته در هر دوره با ظهور تکنولوژی جدید، فرآیند آموزش به‌روزتر و کارآمدتر شده و با رشد انسان به واسطه مجموعه آموزش‌ها، هدف و آرمانی جدید در کنار هدف تسهیل آموزش تعریف شد؛ آرمان آموزش برای همه فارغ از موقعیت جغرافیایی، وضعیت مالی و حتی زبان دوره آموزشی آرمانی است که در قرن اخیر و با درک این موضوع که انسان با دانش قوی‌تر است و دنیای قدرتمند نیازمند انسان‌های قوی است شکل گرفت.

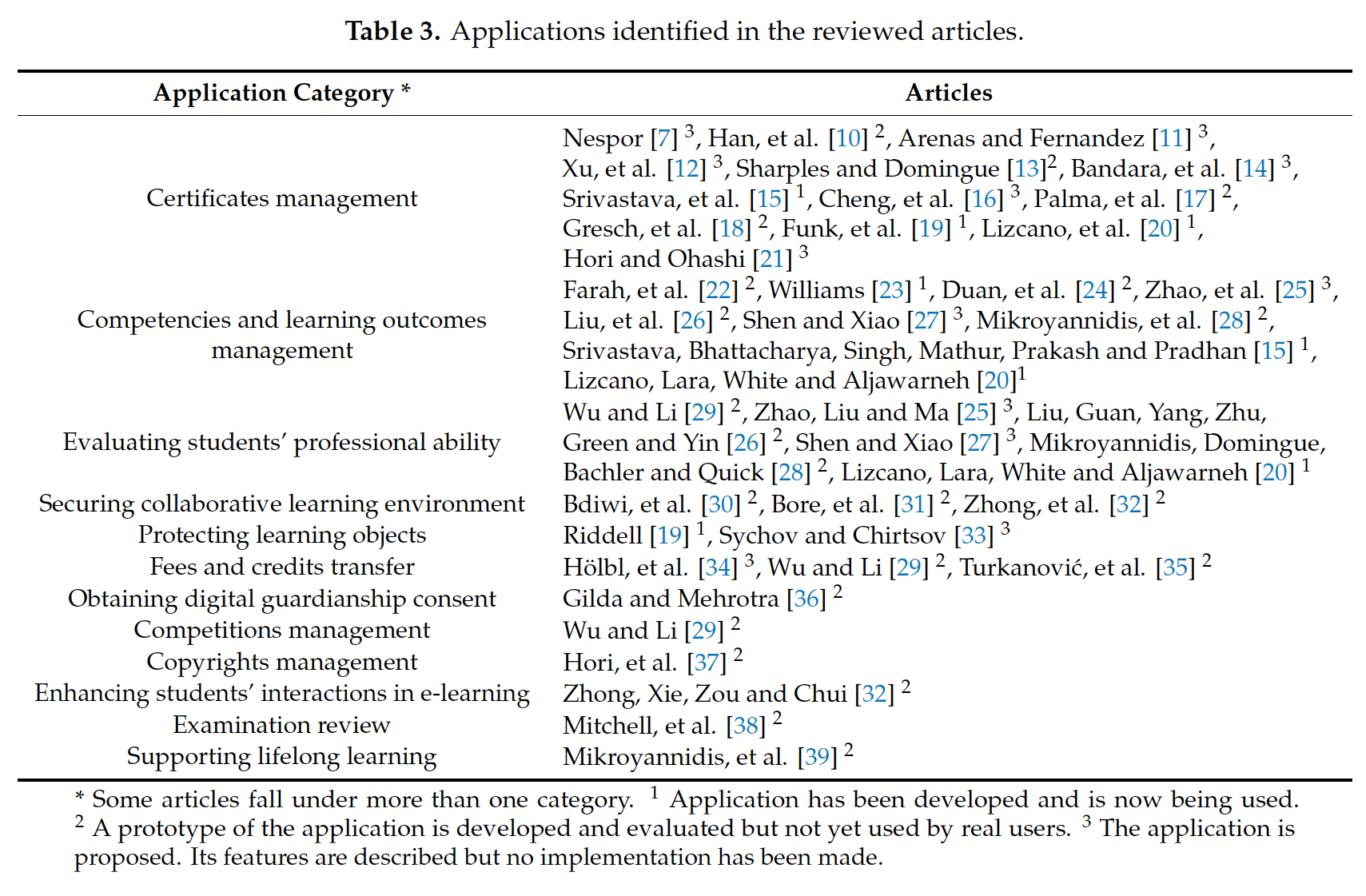
در حال حاضر جدیدترین تکنولوژی پیش روی انسان تکنولوژی بلاکچین بوده و افراد مختلف در سراسر جهان سعی دارند با مطرح کردن ایده‌های نوآورانه جهت استفاده از پتانسیل‌های این تکنولوژی در زمینه آموزش و به خصوص آموزش آنلاین اهداف مطرح شده را تحقق بخشند.

یک نگاه کلی و جامع بر روی پروژه‌های انجام‌ گرفته در زمینه آموزش و بر پایه تکنولوژی بلاکچین می‌تواند به افراد فعال در حوزه آموزش جهت نیازسنجی، ایده‌پردازی و تحقق هرچه بهتر اهداف ذکر شده کمک شایانی کند و همچنین می‌تواند با توسعه فضای کاری بلاکچین وکاربردی کردن هرچه بیشتر این تکنولوژی بستر لازم جهت همه‌گیری و پذیرش حداکثری آن را فراهم آورد، در همین راستا مقاله‌های مختلفی را مورد بررسی قرار دادیم که در ادامه به بیان آن‌ها خواهیم پرداخت.

## پروژه‌های انجام گرفته

با بررسی مقاله‌های مختلف به یک مقاله جامع [11] جهت بیان کاملی از پروژه‌های انجام گرفته رسیدیم، این مقاله از نوع مقاله‌های مروری[[152]](#footnote-153) بوده و با بررسی تمام مقالات منتشر شده مرتبط در زمینه به کارگیری بلاکچین در آموزش تا سال ۲۰۱۹ به تعداد ۲۵۳۶ و غرابالگری سخت‌گیرانه بر اساس کاربردی بودن به ۳۱ مقاله مهم رسیده.

مقالات بیان شده در این مقاله مروری، به ۱۲ دسته اصلی تقسیم بندی شده‌اند و همچنین از این نظر که ایده مطرح شده دارای پروژه پیاده‌سازی شده می‌باشد و یا صرفا یک برنامه پیشنهادی است و حتی اینکه پروژه انجام گرفته توسط کاربران واقعی مورد استفاده است یا نه مورد بررسی قرار گرفته‌اند



شکل(۳- 1) دسته‌بندی مقالات انجام گرفته در زمینه به‌کارگیری بلاکچین در آموزش

همانطور که در شکل بالا مشخص است و تعداد مقاله‌های انتشار یافته نشان می‌دهد عموم مقاله‌های انجام شده با رویکرد استفاده از ویژگی اصلی بلاکچین یعنی غیرقابل تغییر بودن اطلاعات ثبت شده است و تمرکز بر روی ارائه راهکاری عملی، به صرفه و کارآمد برای ثبت سابقه تحصیلی، أنواع مدارک تحصیلی، ریزنمرات و اطلاعات مربوط به عملکرد فراگیر آموزش بوده است. نتیجه این تلاش‌ها می‌تواند در شفافیت و عدم تقلب مدارک، سهولت دسترسی به مدارک برای کارفرما و یا خود فراگیر و همچنین کمک به صرفه جویی در مصرف کاغذ کمک کننده باشد.

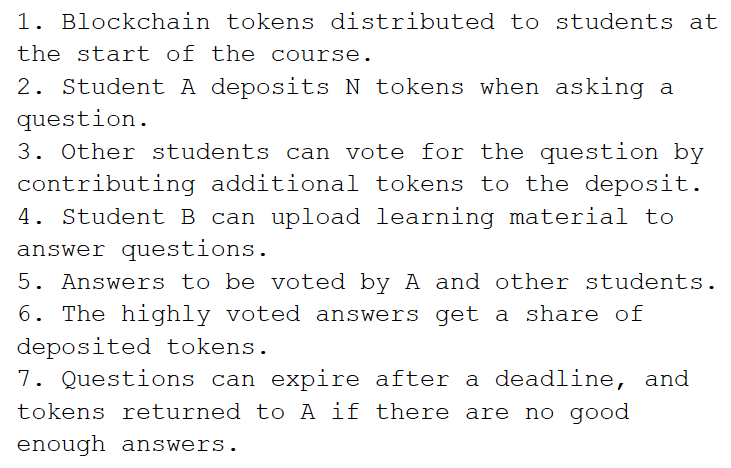
در بین مقالات بررسی شده اکثر مقالات مربوط به سال ۲۰۱۸ بوده و مورد پوشش و بررسی مقاله جامع ما بوده است اما در این بین یک مقاله جدید [12] و با یک پروژه خلاقانه وجود دارد که انتشار آن مربوط به سال ۲۰۲۲ است و در مقاله مروری جامع بیان شده ما وجود ندارد که قصد داریم به بیان این پروژه بپردازیم.

به طور کلی در سال‌های اخیر سعی شده کاربرد‌های خلاقانه ای جهت استفاده از بلاکچین در آموزش مطرح شود که یکی از این پروژه‌ها، که زمان نگارش این پایان‌نامه جدید است، پروژه ای به نام Etherlearn است.

ایده کلی این پروژه ارائه یک سیستم آموزشی همتا[[153]](#footnote-154) است که دانش‌آموزان و فراگیران بتوانند در این سیستم غیرمتمرکز سوالات و منابع و موارد آموزشی را به اشتراک بگذارند، این رویکرد که در تضاد با آموزش سنتی و متمرکز که یک مرجع واحد تحت عنوان استاد و یا موسسه مسئولیت ارائه موارد آموزشی را داشته و پاسخ‌گویی و پوشش مطالب آموزشی نیز بر عهده عموما استاد بوده سعی در راه اندازی سیستم آموزشی نوین است تا فضای مناسبی را برای فراگیران آموزشی بوجود آورد.

نیازمندی‌های این فضا برای ارائه یک سیستم آموزشی غیر متمرکز و مبتنی بر فراگیر آموزشی به این شکل است که باید برای جلب مشارکت فراگیران یک انگیزه قوی ایجاد شود که بهترین نوع آن انگیزه مالی است و همچنین فضا به شکلی باشد که مشارکت افراد به شکل ناشناس باشد تا فراگیری از ترس اشتباه کردن و یا مورد تمسخر قرار گرفتن از ارائه پاسخ و مشارکت صرف نظر نکند. برآورد کردند این نیازمندی ‌ها استفاده از بلاکچین را ایجاب می‌کند و تا قبل از این تکنولوژی چنین امکانی وجود نداشته است.

سازوکار این پروژه برای فراهم کردن چنین سیستم آموزشی توسط قرارداد هوشمند طراحی شده انجام می‌گیرد و از ۷ مرحله که در شکل ذیل آورده شده تشکیل شده است.



شکل(۳- 2) سازوکار پروژه Etherlearn

## نتیجه‌گیری

# تحلیل نیازمندی‌ها و طراحی

## مقدمه

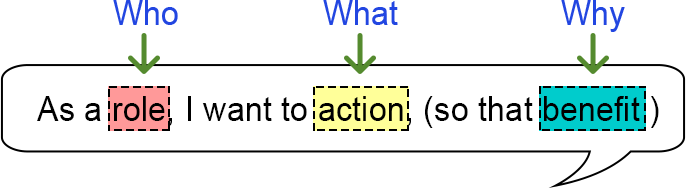
پروژه طراحی و ساخته شده پیش روی شما که با عنوان پلتفرم smartlearn در سراسر این پایان‌نامه شناخته می‌شود حاصل ماه‌ها تلاش شبانه‌روزی انفرادی بوده و سعی داشته هم‌جهت با هدف والای آموزش که تحقق آرمان آموزش بدون تبعیض برای همه می‌باشد توسعه یابد، در این فصل از پایان‌نامه در ابتدا با بیان خواست کاربران در قالب داستان‌های کاربر به نیازسنجی یک فرد استفاده کننده از دوره های آنلاین آموزشی پرداختیم و در ادامه با تبیین سند نیازمندی‌های این پلتفرم، نیازمندی‌های سیستم جهت پیاده سازی پروژه و نیازهای کاربران به شکل جدول در سه بخش سمت کاربر، سمت سرور و سمت بلاکچین ارائه میشود که هر یک از نیازمندی های مطرح شده جزء به جزء مورد بررسی قرار میگیرد در انتها نمای کلی از پروژه و صفحات پیاده شده را به عنوان واسط کاربری خواهیم داشت.

## داستان‌های کاربر

در قدم‌های ابتدایی شروع توسعه هر نرم افزاری نیازسنجی مهمترین بخش می‌باشد تا بر اساس هدف و انگیزه‌های شکل‌گیری یک سیستم نیازهای استفاده کنندگان حوزه مورد توسعه شناسایی و باتوجه به این نیازها توسعه نرم‌ افزار انجام گیرد.

روش نیازسنجی کاربران در رویکرد توسعه نرم افزار چابک[[154]](#footnote-155)، داستان‌های کاربر می‌باشد تا از زبان کاربران سندی جهت نیازسنجی برای ارائه به تیم فنی فراهم شود.

قالب اصلی داستان‌های کاربر به شکل ذیل می‌باشد:



شکل(۴- 1) ساختار نوشتن داستان کاربر

* من به عنوان 《 یک شخص 》، میخواهم 《کاری را انجام دهم》[نیازی دارم]، (تا《به هدفی برسم》[کاری برایم آسان شود])[[155]](#footnote-156)

در ادامه می‌توان داستان‌های کاربری که منجر به پیاده‌سازی پلتفرم smartlearn شد را مشاهده کرد:

* من به عنوان یک فرد می‌خواهم هیچ اطلاعات شخصی برای استفاده از دوره‌های آموزی آنلاین و دیگر خدماتی که در این حوزه ارائه می‌شود را ارائه نکنم و به صورت ناشناس فعالیت کنم
* من به عنوان یک فرد می‌خواهم دوره‌های آموزشی رایگان و ترجمه شده به زبان خودم را در دسترس داشته باشم
* من به عنوان یک فرد می‌خواهم دسترسی آسان به دوره‌های آموزشی آنلاین داشته باشم
* من به عنوان یک فرد می‌خواهم بتوانم برای دوره‌های آنلاینی که اخذ کرده ام محدودیت زمانی و جایزه قرار دهم تا انگیزه ام برای تمام کردن تمام دوره‌هایم در زمان مقرر بیشتر شود
* من به عنوان یک فرد می‌خواهم لیستی از تمام دوره‌های آنلاینی که اخذ کرده ام را داشته باشم
* من به عنوان یک فرد می‌خواهم بتوانم تنبیه مالی در صورت به پایان نرساندن دوره‌ای که اخذ کرده ام داشته باشم تا الزام و فشار بیشتری برای تمام کردن دوره در زمان معین داشته باشم
* من به عنوان یک فرد می‌خواهم بتوانم درصورت دلخواه فرد دیگری را با انگیزه مالی تشویق به اتمام دوره آموزشی اش در زمان مقرر کنم
* من به عنوان یک فرد می‌خواهم بتواتم دوره‌هایی را جهت ترجمه به زبان خودم درخواست تولید دهم
* من به عنوان یک فرد می‌خواهم تراکنش های مالی‌ام بر بستر بلاکچین و با رمز ارز[[156]](#footnote-157) اتر[[157]](#footnote-158) انجام گیرد تا تراکنش هایم مورد تحریم یا هر نوع محدودیت دیگری قرار نگیرد
* من به عنوان یک فرد می‌خواهم بتوانم درصورت دلخواه حمایت مالی از سایت ارائه دهنده خدمات آموزشی داشته باشم

## سند نیازمندی‌های نرم‌افزار

در ادامه سند نیازمندی‌های پلتفرم smartlearn را به صورت جدول خواهیم داشت ولی قبل از آن لازم است تعریف شفافی از کاربران این پلتفرم ارائه شود؛ کاربران این پلتفرم به دو دسته تقسیم می‌شوند:

**۱. کاربران ویژه :** کاربری که کیف پول رمزارزی[[158]](#footnote-159)متامسک داشته و کیف پولش حاوی مقداری اتر جهت انجام تراکنش بوده و همچنین کاربر افزونه متامسک در مرورگرهای مورد پوشش متامسک[[159]](#footnote-160) داشته باشد و اتصال کیف پول به سایت انجام گرفته باشد.

**۲. کاربر معمولی :** کاربر معمولی که ما در این پایان‌نامه جهت سهولت به اسم کاربر از آن یاد می‌کنیم به هر فردی که به اینترنت دسترسی داشته و به پلتفرم smartlearn مراجعه می‌کند اطلاق می‌شود.

جدول(۴- 1) جدول سند نیازمندی سمت کاربر

|  |  |  |
| --- | --- | --- |
|  |  | Client Side |
| کد ویژگی | عنوان ویژگی | توضیحات |
| C01 | مشاهده صفحه خانه | کاربر می‌تواند صفحه خانه را مشاهده کند |
| C02 | مشاهده صفحه دوره‌های ارائه شده | کاربر می‌تواند لیست دوره‌های تهیه شده سایت را در صفحه مربوطه مشاهده کند |
| C03 | مشاهده جزئیات دوره‌های ارائه شده در سایت | کاربر می‌تواند جزئیات مربوط به هر دوره را مشاهده نماید |
| C04 | مشاهده ویدئو‌های دوره‌های ارائه شده | کاربر می‌تواند ویدئو‌ دوره‌ ارائه شده را از طریق youtube مشاهده نماید |
| C05 | اتصال کیف پول کاربر به سایت | از طریق افزونه Metamask اتصال کیف پول کاربر به سایت انجام می‌شود و کاربر به کاربر ویژه تغییر می‌یابد |
| C06 | امکان ارسال نظر و پیشنهاد | کاربر می‌تواند به شکل کاملا ناشناس پیشنهادات خود را جهت بهبود سایت ارسال کند |
| C07 | مشاهده لیست دوره‌های برای انجام | کاربر ویژه می‌تواند لیستی از دروسی که برای انجام ثبت کرده را با جزئیات هر درس مشاهده نماید |
| C11 | اضافه کردن دوره‌ به لیست دوره‌های برای انجام | کاربر ویژه می‌تواند دوره های اخذ کرده خود را؛ چه این دوره از دوره‌های ارائه شده سایت باشد و چه هر دوره دیگری که اخذ نموده است را با پر کردن موارد مربوطه به لیست دوره‌های برای انجام خود اضافه نماید |
| C12 | امکان اضافه کردن دوره‌های ارائه شده به لیست دوره‌های برای انجام | کاربر ویژه‌ می‌تواند از قسمت جزئیات هر دوره ارائه شده در سایت دوره انتخاب کرده را به لیست دوره‌های برای انجام اضافه کند |
| C13 | امکان تعیین تشویق مالی برای دیگر افراد | کاربر ویژه می‌تواند آدرس کیف پول فرد دیگری و یا حتی آدرس دیگر کیف پول خودش را به عنوان ادرس مقصد جهت تراکنش پرداخت جایزه قرار دهد |
| C14 | امکان تعیین تنبیه مالی دوره‌های در حال انجام | کاربر ویژه‌ می‌تواند تعیین نماید در صورت عدم تکمیل دوره ثبت شده برای انجام در زمان مقرر شده، میزان اتر از قبل تعیین شده به عنوان جایزه، جهت تنبیه به آدرس کیف پول سایت منتقل شود و به همان میزان به مشارکت کلی کاربر ویژه اضافه شود |
| C15 | تکمیل دوره برای انجام | کاربر ویژه‌ای که دوره ای برای انجام دارد می‌تواند دوره برای انجامش را به حالت تکمیل شده تغییر دهد |
| C16 | مشاهده زمان دقیق کامل شدن دوره برای انجام | کاربر ویژه‌ای که دوره برای انجامش را به حالت کامل شده تغییر می‌دهد می‌تواند زمان دقیق کامل شدنش را مشاهده نماید |
| C17 | کشتن دوره برای انجام | کاربر ویژه‌ای که دوره ای برای انجام دارد می‌تواند دوره برای انجامش را به حالت کشته شده که به معنای عدم تکمیل دوره است تغییر دهد |
| C18 | حذف دوره برای انجام | کاربر ویژه‌ای که دوره ای برای انجام دارد می‌تواند دوره برای انجامش را در صورتی که تکمیل شده و یا kill شده باشد به طور کامل حذف نماید |
| C19 | مشاهده میزان کل مشارکت | کاربر ویژه‌ می‌تواند میزان کل مشارکتش با سایت را به اتر مشاهده نماید |
| C20 | مشاهده تعداد دفعات پیشنهاد دوره | کاربر ویژه‌ می‌تواند تعداد دفعاتی را که میتواند دوره‌ای را برای ترجمه پیشنهاد دهد را مشاهده کند |
|  | امکان حمایت مالی سایت | کاربر ویژه می‌تواند به میزان دلخواه با انتقال اتر به آدرس کیف پول سایت حمایت مالی انجام دهد و به همان میزان به مشارکت کلی کاربر ویژه اضافه شود |
| C21 | امکان درخواست یک دوره برای ترجمه | کاربر ویژه‌ می‌تواند در صورتی که مجاز باشد با پر کردن موارد مربوطه دوره‌ای را برای ترجمه درخواست دهد |
| C22 | برقراری ارتباط با قرارداد هوشمند |  |
| C23 | برقراری ارتباط با پایگاه داده |  |
| C24 | بررسی و محاسبه به روز وضعیت و شرایط مربوط به هر دوره برای انجام |  |
| C25 | بررسی موارد مجاز ورودی‌ها |  |
| C26 | دریافت ورودی‌ها |  |

جدول(۴- 2) جدول سند نیازمندی سمت بلاکچین

|  |  |  |
| --- | --- | --- |
|  |  | Blockchain Side |
| کد ویژگی | عنوان ویژگی | توضیحات |
| B01 | Push کردن اطلاعات دوره‌های برای انجام |  |
| B02 | واکشی آرایه ای از دوره‌های برای انجام |  |
| B03 | حذف (pop) کردن دوره‌ای از آرایه دوره‌های برای انجام |  |
| B04 | اعمال تغییر وضعیت دوره‌های برای انجام |  |
| B05 | محاسبه تعداد دفعات مجاز برای پیشنهاد دوره | تعداد دفعات مجاز هر کاربر ویژه |
| B06 | محاسبه میزان کل مشارکت | میزان کل مشارکت کاربر ویژه بر حسب wei محاسبه می‌شود |
| B07 | انجام تراکنش‌ها | انتقال اتر از یک آدرس به آدرس دیگری |
| B08 | واکشی میزان کل مشارکت |  |
| B09 | واکشی تعداد دفعات مجاز برای پیشنهاد دوره |  |
| B10 | ثبت تعداد دفعات مجاز برای پیشنهاد دوره |  |
| B11 | ثبت میزان کل مشارکت |  |
| B12 | محاسبه و کسر کارمزد |  |
| B13 | اعمال محاسبات و شروط دوره‌های برای انجام |  |
| B14 | بررسی زمان مجاز برای تعیین ضرب الاجل دوره های برای انجام |  |
| B15 | چک کردن انقضا ضرب الاجل دوره های برای انجام |  |
| B16 | بررسی امکان حذف |  |
| B17 | بررسی حداقل میزان جایزه |  |

جدول(۴- 3) جدول سند نیازمندی سمت سرور

|  |  |  |
| --- | --- | --- |
|  |  | Server Side |
| کد ویژگی | عنوان ویژگی | توضیحات |
| S01 | ثبت اطلاعات دوره‌های تهیه و آماده شده | اطلاعات دوره‌های تهیه و آماده شده توسط ادمین در پایگاه داده متمرکز ذخیره و ثبت می‌شود |
| S02 | واکشی اطلاعات دوره‌های سایت | اطلاعات دوره‌های تهیه و آماده شده جهت نمایش به کاربر از پایگاه داده متمرکز واکشی می‌شود |
| S03 | ثبت اطلاعات دوره‌های پیشنهادی | اطلاعات دوره‌های پیشنهادی از طرف کاربر ویژه در پایگاه داده متمرکز ذخیره و ثبت می‌شود |

### تحلیل نیازمندی‌ها

* **کد ویژگی :** C01 **|** **بازیگر :** کاربر

**عنوان نیازمندی :** مشاهده صفحه خانه

**توضیح نیازمندی :** این صفحه، صفحه ورودی هر کاربر می‌باشد و شامل دکمه ورود به صفحه لیست دوره‌های ارائه شده و دکمه ورود به صفحه اضافه کردن دوره به لیست دوره‌های برای انجام می‌باشد

**توضیح فنی به همراه تکه کد :** کد‌های این قسمت فنی نبوده و صرفا طراحی می‌باشد

* **کد ویژگی :** C02 **|** **بازیگران :** کاربر - پایگاه داده متمرکز (سرور)

**عنوان نیازمندی :** مشاهده صفحه دوره‌های ارائه شده

**توضیح نیازمندی :** کاربر با کلیک بر روی دکمه[[160]](#footnote-161) Courses واقع در سربرگ[[161]](#footnote-162) و یا دکمه Explore courses در صفحه خانه وارد صفحه دوره‌های ارائه شده می‌شود و لیست دوره‌های تهیه شده سایت مشاهده می‌کند.

**توضیح فنی به همراه تکه کد :** با استفاده از کتابخانه axios اطلاعات تمام دوره‌های تهیه شده از پایگاه داده متمرکز (سرور) جهت نمایش به کاربر واکشی می‌شود و این اطلاعات به صورت یک آرایه[[162]](#footnote-163) به سمت frontend فرستاده می‌شود

|  |
| --- |
| courses: []  async created() {  try {  const response = await axios.get(`http://127.0.0.1:8000/api/v1/courses`);  this.courses = response.data;  } catch (e) {  this.errors.push(e);  }  } |

* **کد ویژگی :** C03 **|** **بازیگران :** کاربر - پایگاه داده متمرکز (سرور)

**عنوان نیازمندی :** مشاهده جزئیات دوره‌های ارائه شده در سایت

**توضیح نیازمندی :** کاربر با کلیک بر روی دکمه GO در باکس[[163]](#footnote-164) مربوط به هر دوره ارائه شده در سایت وارد صفحه جزئیات آن دوره می‌شود که این جزئیات شامل { نام– توضیحات– ویدئو- مدت زمان مقرر اتمام– زبان ترجمه شده– اضافه کردن دوره به لیست دوره‌های برای انجام } می‌باشد.

**توضیح فنی به همراه تکه کد :** با استفاده از کتابخانه axios اطلاعات جزئیات مربوط به دوره انتخاب شده از پایگاه داده متمرکز (سرور) جهت نمایش به کاربر واکشی می‌شود و این اطلاعات به صورت یک شی[[164]](#footnote-165) به سمت frontend فرستاده می‌شود

|  |
| --- |
| course: {}  async created() {  try {  const response = await axios.get( `http://127.0.0.1:8000/api/v1/courses/${this.courseId}`  );  this.course = response.data;  } catch (e) {  this.errors.push(e);  } |

هر دوره از روی آیدی منحصر به فردش شناخته می‌شود و با کد زیر آیدی مربوط به دوره‌ای که کاربر انتخاب کرده جهت واکشی اطلاعات استخراج و ذخیره می‌شود

|  |
| --- |
| courseId: this.$route.params.id |

* **کد ویژگی :** C04 **|** **بازیگران :** کاربر - پایگاه داده متمرکز (سرور)

**عنوان نیازمندی :** مشاهده ویدئو‌های دوره‌های ارائه شده

**توضیح نیازمندی :** کاربر می‌تواند در صفحه جزئیات دوره انتخابی‌اش ویدئو دوره را از طریق [youtube](https://www.youtube.com/) مشاهده نماید.

**توضیح فنی به همراه تکه کد :** آدرس ویدئو ذخیره در یوتوب به عنوان ویدئو در پایگاه داده ذخیره می‌شود و با کد زیر به کاربر نمایش داده می‌شود

|  |
| --- |
| <b-embed type="iframe" width="640" height="360" :src="`https://www.youtube.com/embed/${course.video}`"  allowfullscreen></b-embed> |

* **کد ویژگی :** C05 **|** **بازیگر :** کاربر

**عنوان نیازمندی :** اتصال کیف پول کاربر به سایت

**توضیح نیازمندی :** به محض ورود کاربر به سایت و نصب داشتن افزونه متامسک در مرورگرهای مورد پوشش به صورت خودکار صفحه متامسک برای اتصال کیف‌پول رمزارزی کاربر ظاهر می‌شود و کار می‌تواند با برقراری این اتصال از تمامی امکانات سایت استفاده نماید، در صورت عدم برقراری اتصال پیامی به شکل ذیل برای کاربر نمایش داده می‌شود :

**Is [MetaMask](https://metamask.io/" \t "_blank) extension installed on your browser?**

**Is proper network selected in MetaMask?**

**(Rinkeby Test Network or Localhost on port 8545)**

**توضیح فنی به همراه تکه کد :** با استفاده از چهارچوب Drizzle و متد[[165]](#footnote-166) isDrizzleInitialize تعریف شده در آن اتصال کیف پول کاربر بررسی می‌شود

|  |
| --- |
| <div v-if="isDrizzleInitialized">  import { mapGetters } from "vuex";  ...mapGetters("drizzle", ["isDrizzleInitialized"]) |

نیاز است جهت استفاده از چهارچوب Drizzle پیکربندی اولیه‌ای انجام شود که این کار با ایجاد یک فایل جدید به نام drizzleOptions.js در پوشه Frontend و با کد ذیل انجام می‌گیرد.

|  |
| --- |
| import SmartLearn from './contracts/SmartLearn.json'  const options = {  web3: {block: false,fallback: {  type: 'ws',  url: 'ws://127.0.0.1:9545'}},  contracts: [SmartLearn],events: {  SmartLearn: ['LogCoursesUpdated']},  polls: {accounts: 15000}}  export default options |

چهارچوب Drizzle درون کدهای خود از کتابخانه Web3.js استفاده می‌کند که به واسطه این کتابخانه و دستور ذیل به محض ورود کاربر به سایت جهت اتصال کیف پول کاربر و همچنین زمان‌های انجام تراکنش جهت تایید تراکنش صفحه متامسک باز می‌شود.

|  |
| --- |
| window.ethereum.enable(); |

* **کد ویژگی :** C06 **|** **بازیگر :** کاربر

**عنوان نیازمندی :** امکان ارسال نظر و پیشنهاد

**توضیح نیازمندی :** در قسمت سربرگ سایت دکمه Any Offer وجود دارد که کاربر می‌تواند با کلیک کردن آن پیشنهادات خود را به شکل کاملا ناشناس ارسال کند

**توضیح فنی به همراه تکه کد :** با استفاده از کتابخانه EmailJS ارسال ایمیل انجام می‌گیرد.

|  |
| --- |
| sendEmail() { if (this.message.length >= 10)  {emailjs.sendForm("service\_9sjxu8w",  "template\_cut3mh7", this.$refs.form,  "Hmqsuup3nMwM\_91eK");}} |

* **کد ویژگی :** C07 - C08 - C09 **|** **بازیگر :** کاربر ویژه - بلاکچین

**عنوان نیازمندی :** اضافه کردن دوره‌ به لیست دوره‌های برای انجام - امکان تعیین تشویق مالی برای دیگر افراد - امکان تعیین تنبیه مالی دوره‌های در حال انجام

**توضیح نیازمندی :** کاربر ویژه می‌تواند با کلیک بر روی دکمهdo Add to واقع در سربرگ سایت وارد صفحه اضافه کردن دوره‌ به لیست دوره‌های برای انجام شود سپس با وارد کردن اطلاعاتی شامل {نام دوره - مقدار اتر برای جایزه (که این مقدار باید شامل ۰.۱ اتر به عنوان کارمزد باشد) - تاریخ و زمان دقیق موعد انجام دوره} و همچنین در صورت نیاز، کاربر ویژه می‌تواند به صورت امکان اضافی[[166]](#footnote-167) با انتخاب گزینه punishMe خود را تنبیه نماید و یا با قراردادن آدرس کیف پول فرد دیگری یا دیگر آدرس کیف پول خودش به عنوان ادرس مقصد، آدرس واریز جایزه را تغییر دهد.

**توضیح فنی به همراه تکه کد :**

* **کد ویژگی :** C07 **|** **بازیگر :** کاربر

**عنوان نیازمندی :**

**توضیح نیازمندی :**

**توضیح فنی به همراه تکه کد :**

* **کد ویژگی :** C07 **|** **بازیگر :** کاربر

**عنوان نیازمندی :**

**توضیح نیازمندی :**

**توضیح فنی به همراه تکه کد :**

* **کد ویژگی :** C07 **|** **بازیگر :** کاربر

**عنوان نیازمندی :**

**توضیح نیازمندی :**

**توضیح فنی به همراه تکه کد :**

* **کد ویژگی :** C07 **|** **بازیگر :** کاربر

**عنوان نیازمندی :**

**توضیح نیازمندی :**

**توضیح فنی به همراه تکه کد :**

* **کد ویژگی :** C07 **|** **بازیگر :** کاربر

**عنوان نیازمندی :**

**توضیح نیازمندی :**

**توضیح فنی به همراه تکه کد :**

* **کد ویژگی :** C07 **|** **بازیگر :** کاربر

**عنوان نیازمندی :**

**توضیح نیازمندی :**

**توضیح فنی به همراه تکه کد :**

## واسط کاربری

# ابزارها

## مقدمه

در این فصل قصد داریم شناخت کاملی نسبت به ابزار‌های لازم برای توسعه یک اپلیکیشن غیرمتمرکز[[167]](#footnote-168) کسب کنیم و همچنین به طور کامل مراحل لازم جهت اجرا این پروژه مرور کنیم تا بتوانیم در پایان پلتفرم smartlearn پیاده‌سازی و اجرا نماییم.

جدول ذیل یک لیست کاملی از تمام ابزار بکار گرفته شده در این پروژه به عنوان یک Dapp بوده که به تفکیک سه بخش سمت کاربر، سمت سرور و سمت بلاکچین همراه نسخه مورد استفاده آن ارائه شده.

جدول(۵- 1) ابزارها

|  |  |  |
| --- | --- | --- |
| سمت کاربر | Node JS | 16.15.0 |
| npm | 7.6.0 |
| Vue | 2.6.10 |
| Vuex | 3.1.1 |
| Vue-router | 3.5.4 |
| axios | 0.27.2 |
| moment | 2.29.3 |
| Bootstrap | 5.1.1 |
| Bootstrap-vue | 2.21.2 |
| EmailJS | 3.6.2 |
| Drizzle store | 1.5.3 |
| Drizzle vue plugin | 0.1.0 |
| Metamask | - |
| سمت سرور | Python | 3.10.4 |
| pip | 22.1.1 |
| Django | 4.0.4 |
| Django Rest Framework | 3.13.1 |
| Django-CORS-Headers | 3.12.0 |
| Sqlite3 | - |
| سمت بلاکچین | Truffle | 5.5.13 |
| Ganache | 7.1.0 |
| Solidity | 0.5.16 |
| Truffle HD Wallet | 2.0.8 |
| OpenZeppelin/contracts | 4.3.1 |
| Infura | - |

## ابزارهای توسعه

به این علت که تمرکز اصلی این پایان‌نامه بر روی بلاکچین بوده و هدف شناخت بهتر نسبت به این تکنولوژی است بنابراین به ابزارهای این بخش به طور کامل خواهیم پرداخت و به توضیح کوتاهی از دیگر ابزارها اکتفا می‌کنیم.

### سمت بلاکچین

جهت توسعه یک قرارداد هوشمند می‌توان از ۳ نوع شبکه استفاده کرد:

1. **شبکه اصلی[[168]](#footnote-169) :** منظور همان شبکه بلاکچین اتریوم است که استفاده از این شبکه به علت هزینه‌بر بودن و این مورد که قرارداد ثبت شده در شبکه غیرقابل تغییر است و در صورت نیاز به تغییر به هر نوع آن امکان تغییر وجود ندارد در فاز توسعه یک قرارداد هوشمند عموما از این شبکه استفاده نمی‌کنند و در زمان عرضه و استفاده عمومی با اعمال تمام تغییرات مورد نیاز و بررسی‌های امنیتی قرارداد را در شبکه اصلی قرار می‌دهند[[169]](#footnote-170).
2. **شبکه تست[[170]](#footnote-171) :** این شبکه از نظر ساختاری کاملا مشابه شبکه اصلی است و دارای استخراج کننده در سراسر جهان می‌باشد اما تفاوت اصلی آن با شبکه اصلی استفاده از رمزارزی از نظر مالی بی‌ارزش و رایگان است در نتیجه تعامل و قراردادن قرارداد در این شبکه هزینه‌بر نیست، از این شبکه در فاز توسعه قراردادهای هوشمند استفاده می‌شود ولی در توسعه‌های اولیه به علت زمان‌بر بودن تعامل با این شبکه‌ها عموما استفاده نمی‌شود و نیاز به یک شبکه رایگان و سریع احساس می‌شود.,Rinkeby ,Ropsten ,Kovan Goerli نمونه ای از نوع شبکه هستند.
3. **شبکه محلی[[171]](#footnote-172) :** شبکه‌های محلی همان شبکه رایگان و سریع است که در فاز توسعه اولیه قرارداد‌های هوشمند مورد استفاده قرار می‌گیرد، این نوع شبکه‌ها عملا بلاکچین نیستند ولی تمام وظایف بلاکچین را در یک محیط محلی مانند کامپیوتر شخصی انجام می‌دهند و شبکه ای شبیه‌سازی شده از بلاکچین به توسعه‌دهندگان ارائه می‌دهند.

Ganache نمونه ای از این نوع شبکه است، این ابزار همان‌طور که بیان شد یک شبکه محلی بوده و شبیه‌سازی کاملی از بلاکچین اتریوم در کامپیوتر شخصی می‌سازد، نصب این ابزار به سادگی توسط nmp انجام می‌گیرد و بعد از اجرای آن ۱۰ حساب از نوع EOA با موجودی هریک ۱۰۰ اتر تستی در اختیار توسعه‌دهندگان قرار می‌دهد.

برای اتصال به شبکه اصلی و یا شبکه تست نیاز به یک گره کامل است، این گره را می‌توانیم خودمان با استفاده از ابزاری به نام Geth [[172]](#footnote-173) اجرا و پیاده‌سازی کنیم که این‌کار علاوه بر هزینه بر بودن از نظر سخت‌افزاری و پهنای‌باند[[173]](#footnote-174) مناسب بسیار زمان‌بر بوده و هفته‌ها و ماه‌ها اجرا یک گره کامل طول خواهد کشید، به همین جهت و برای سهولت کار راه دیگر استفاده از API [[174]](#footnote-175) هایی است که سرویس‌دهندگان گره ارائه می‌دهند تا با اتصال به آن‌ها بتوان بدون نیاز به اجرا کردن گره کامل به شبکه متصل شد.

Infura, Alchemy, QuickNodeنمونه ای از این سرویس‌دهندگان گره هستند.

ابزار بعدی چهارچوب Truffle با خدمات کاملی برای توسعه‌دهندگان قرارداد‌های هوشمند است؛ برای توسعه قرارداد هوشمند نیاز به زبان سطح بالایی مثل سالیدیتی داریم، این زبان مانند هر زبان برنامه‌نویسی دیگری از دو بخش کلی نحو[[175]](#footnote-176) و کامپایلر[[176]](#footnote-177) تشکیل شده، قسمت Syntax آن به آسانی از طریق اضافه کردن افزونه ویرایشگر متن[[177]](#footnote-178) مانند Visual Studio code [[178]](#footnote-179) قابل دستیابی است، بخش بعد از کامپایلری به نام Solc استفاده می‌کند، این کامپایلر را می‌توان به صورت مستقیم از طریق Npm [[179]](#footnote-180)نصب نمود ولی استفاده از آن دارای پیچیدگی بوده و پیکربندی[[180]](#footnote-181) و تنظیم زمان‌بری دارد.

برای نوشتن قرارداد‌های امن و کارآمد[[181]](#footnote-182) در سالیدیتی بهتر است از کتابخانه‌هایی که جمعی از برنامه‌نویسان این زبان با پیاده‌سازی توابع مهم با در نظر گرفتن تمامی حملات شناسایی شده ساخته‌اند مانند کتابخانه OpenZeppelin استفاده نمود، بعد از کامپایل کد و بدست آمدن بایت‌کد قرارداد هوشمند باید آن را بر روی شبکه Deploy کرد که انجام این کار نیز پیچیدگی‌ها و تنظیمات زمان‌بر خود را دارد.

ما برای انجام کارهای کامپایل و Deploy نیازمند ابزاری هستیم تا توسعه قراردادهای هوشمند را سریع‌تر و آسان‌تر نماید تا بتوانیم با تمرکز بیشتر بر روی نوشتن خود قرارداد، برنامه ای دقیق‌تر و امن‌تر ایجاد نموده و درگیری کم‌تری با مسائل حاشیه‌ای داشته باشیم، Truffle ابزاری است که به ما کمک می‌کند، Truffle علاوه بر کامپایل و Deploy قرارداد، تنظیمات شبکه و پیکربندی شبکه را نیز انجام می‌دهد.

برای امضا تراکنش‌ ایجاد قرارداد و Deploy قرارداد بر روی شبکه اصلی یا تست از طریق سرویس‌دهنده گره‌‌ای مانند Infura نیاز به نصب Truffle HD Wallet و ایجاد یک کیف پول داریم.

### سمت کاربر

تا به اینجا ما به طور کامل ابزارهای لازم جهت توسعه یک قرارداد هوشمند را مورد بررسی قرار دادیم، در ادامه ابزارهای لازم برای توسعه Dappها می‌پردازیم.

اپلیکیشن غیر متمرکز یک صفحه وب[[182]](#footnote-183) است که می‌تواند با قراداد‌(های) هوشمند ارتباط برقرار کند به این نسل جدید از وب Web3 می‌گویند، در اصل Web3 یک نوع API است که با پروتکل JSON RPC[[183]](#footnote-184) ارتباط با یک گره را برقرار می‌کند.

این نوع صفحات وب از دو قسمت Frontend و قرارداد هوشمند تشکیل شده‌اند، قسمت Frontend آن مشابه صفحات وبی که تا الان بوده است می‌‌باشد و از ابزارهای معروف و کاربردی این قسمت مانند HTML, CSS, Java Script, Bootstrap و چهارچوب‌ها پرکاربردی مثل Vue و React استفاده می‌کند.

برای برقراری ارتباط بین این دو قسمت از کتابخانه‌های جاوااسکریپتی Web3.js یا Ethers.js استفاده می‌شود که Web3.js قدیمی‌تر و مورد استفاده‌تر است، Web3.js مجموعه ای از کتابخانه‌های جاوااسکریپت است که توسط بنیاد اتریوم ساخته شده و امکان فراخوانی توابع تعریف شده در قرارداد هوشمند را از طریق واسط کاربری و با استفاده از API Web3 می‌دهد، همچنین دسترسی به این توابع در Frontend از طریق همین کتابخانه Web3.js و با قرار دادن ABI [[184]](#footnote-185) و آدرس قرارداد هوشمند امکان پذیر است.

از آنجایی که پیکربندی اولیه Web3.js و ارتباط با قرارداد دارای پیچیدگی‌هایی بوده جهت سهولت و سرعت بیشتر توسعه Dappها می‌توان از چهارچوب Drizzle استفاده نمود، این چهارچوب که توسط شرکت Truffle در سال ۲۰۱۸ ساخته شده کمک می‌کند تا با یک پیکربندی اولیه ساده و به کمک متد‌های نوشته شده در آن به آسانی با توابع قرارداد هوشمند در Frontend تعامل داشت و در صورت رخداد تغییری در وضعیت قرارداد هوشمند به صورت آنلاین تغییرات را در Frontend اعمال می‌کند، Drizzle این کار را با نگهداری از وضعیت های قرارداد در ابزار مدیریت وضعیت[[185]](#footnote-186) مانند Vuex انجام می‌دهد.

ذکر این نکته نیز مهم است که پیکربندی اولیه Drizzle با اضافه کردن فایل Json قرارداد انجام می‌گیرد، فایل Json قرارداد حاوی دو بخش است

* 1. مشخصات قرارداد شامل اسم، نحوه خواندن، بایت‌کد و ABI قرارداد
  2. اطلاعات شبکه شامل آدرس قرارداد و هش تراکنش ایجاد قرارداد

ابزار مورد استفاده دیگر متامسک است، این ابزار در اصل یک کیف پول رمز ارزی به صورت یک افزونه مرورگرهای مورد پوشش مانند Chrome بوده و جهت انجام تراکنش‌های قرارداد هوشمند موجود در Dapp توسط کاربر مورد استفاده قرار می‌گیرد به این صورت که وقتی متامسک روی مرورگر نصب می‌شود یک شی از Web3 بر روی مرورگر تزریق می‌کند[[186]](#footnote-187) و از این طریق با ایجاد دسترسی به Web3 API امکان تعامل Dapp با بلاکچین را فراهم می‌کند همچنین این ابزار مسئولیت مدیریت حساب‌های کاربر را نیز بر‌عهده دارد.

متامسک کمک بسیار بزرگی در توسعه و همه‌گیری استفاده از Dappها کرده است، در صورت عدم وجود متامسک دو راه برای انجام تراکنش‌های قرارداد هوشمند توسط کاربران بوده

1. یک راه دریافت کلید خصوصی کاربران جهت امضا تراکنش‌ها توسط Dapp بوده که این راه از لحاظ امنیتی غیر کارآمد است
2. راه دیگر انجام مستقیم تراکنش‌ها بر روی کنسول[[187]](#footnote-188)‌های اجرای گره مانند Geth توسط کاربران است که این راه به علت پیچیدگی‌ها، زمان‌بر بودن و ناآشنا بودن کاربران مبتدی غیر کارآمد است.

### سمت سرور

برای پیاده سازی سمت سرور این پروژه جهت ثبت و واکشی اطلاعات مربوط به دوره‌های تهیه و ارائه شده در سایت از زبان Python و چهارچوب جنگو استفاده شده و از پایگاه داده Sqlite3 که به صورت پیش فرض در این چهارچوب وجود دارد استفاده شده است همچنین جهت داشتن یک پنل کاربری[[188]](#footnote-189) گرافیکی و تعامل بهتر با جنگو از چهارچوب Django Rest Framework نیز استفاده شده و جهت هندل کردن سربرگ‌ها برای پاسخ سرور برای همگام کردن با سیاست CORS [[189]](#footnote-190) در جنگو از Django-CORS-Headers استفاده گردیده.

## پیاده‌سازی پروژه

پیاده‌سازی این پروژه بر روی سیستم عامل macOS و دستورات مطرح شده در ادامه مناسب این سیستم عامل است اگرچه این دستورات به علت UNIX Base بودن کرنل[[190]](#footnote-191) macOS با سیستم عامل‌های لینوکسی[[191]](#footnote-192) یکسان است، بنابراین پیشنهاد می‌شود در صورت پیاده‌سازی این پروژه بر روی سیستم عامل ویندوز[[192]](#footnote-193) از WSL [[193]](#footnote-194) استفاده شود.

تمام مراحل نصب، پیاده‌سازی و اجرا این پروژه به طور کامل و طی ۱۵ قدم و به ترتیب در ذیل بیان شده :

1. **نصب متامسک :** با مراجعه به سایت رسمی متامسک می‌توان افزونه متامسک را به مرورگرهای تحت پوشش موجود در سایت اضافه کرد که ما از مرورگر Chrome استفاده کرده‌ایم همچنین لازم است بعد از اضافه نمودن افزونه متامسک کیف پول جدید ایجاد کرد ویا در صورت داشتن کیف پول آن را به متامسک متصل نمود
2. **نصب Nodejs :** بهتر است برای نصب Nodejs ابتدا NVM [[194]](#footnote-195) را نصب کرد تا در صورت نیاز به هر نسخه از Nodejs بتوان به راحتی جابجا شد برای نصب Nodejs از طریق NVM این دستورات اجرا می شود ۱. nvm install node ۲.nvm install 16.15.0 ۳. nvm alias default 16.15.0
3. **نصب NPM :** به همراه نصب Nodejs اخرین نسخه NPM نیز نصب می‌شود جهت تغییر به نسخه مورد نیاز این دستور اجرا می‌شود npm install -g npm@7.6.0
4. **نصب Truffle :** npm install -g truffle
5. **نصب Ganache :** npm install -g ganache-cli
6. **نصب Python و pip :** بر روی سیستم عامل‌های UNIX Base به طور پیش فرض این دو نصب هستند، تنها باید به نسخه نصب شده آن‌ها توجه شود
7. **Clone کردن پروژه :** ابتدا باید git را روی سیستم نصب داشت سپس دستور ذیل را اجرا کرد git clone <https://github.com/ImanS007/smartlearn.git>
8. **نصب نیازمندی‌های سمت سرور :**  ابتدا باید وارد ماشین مجازی پایتون شد برای این کار ابتدا باید وارد فایل اصلی پروژه شد با دستور cd smartlearn سپس دستور ذیل را وارد کرد source smartlearn-env/bin/activate و وارد ماشین مجازی پایتون شد سپس این دستورات را اجرا می‌کنیم : ۱. pip install django ۲. pip install djangorestframework ۳. pip install django-cors-headers
9. **اجرای Backend :** در همان صفحه ترمینال[[195]](#footnote-196) که ماشین مجازی پایتون برقرار است این دستورات را اجرا می‌کنیم : ۱. cd backend ۲. python manage.py runserver
10. **نصب نیازمندی‌های سمت کاربر:** صفحه ترمینال جدیدی باز می‌کنیم سپس این دستورات را اجرا می‌کنیم۱. cd smartlearn ۲. cd smartlearn-front ۳. sudo npm install
11. **نصب نیازمندی‌های سمت بلاکچین :** صفحه ترمینال جدیدی باز می‌کنیم سپس این دستورات را اجرا می‌کنیم۱. cd smartlearn ۲. cd smartcontract ۳. sudo npm install
12. **اجرای Ganache :** در صفحه ترمینال مرحله ۱۱ این دستور را اجرا می‌کنیم : ganche-cli
13. **اتصال شبکه محلی Ganache و متامسک :** ابتدا باید شبکه محلی Localhost 8545 را درصورت وجود در لیست شبکه‌ها انتخاب نمود ویا یک شبکه با مشخصات ذیل اضافه نمود

**New RPC URL :** http://localhost:8545 **Chain ID :** 1337

سپس گزینه فراموشی رمز[[196]](#footnote-197) را انتخاب نمود و بعدMnemonic ایجاد شده توسط Ganache را کپی[[197]](#footnote-198) کرد و در صفحه بازیابی کیف پول[[198]](#footnote-199) paste کرد

1. **کامپایل و Deploy قرارداد :** صفحه ترمینال جدیدی باز می‌کنیم سپس این دستورات را اجرا می‌کنیم۱. cd smartlearn ۲. cd smartcontract ۳. truffle migrate
2. **اجرای Frontend :** در صفحه ترمینال مرحله ۱۰ این دستور را اجرا می‌کنیم: npm run serve

# جمع‌بندی و نتیجه‌گیری

## مقدمه

در این پایان‌نامه قصد داشتیم یک شناخت کامل اما کلی نسبت به سازوکار بلاکچین و به خصوص درک مبسوطی از بلاکچین اتریوم به عنوان پرچم‌دار بلاکچینی با امکان ایجاد قرارداد هوشمند بدست آوریم، با ویژگی‌های این تکنولوژی جدید و البته قوی آشنا شدیم و دریافتیم که رویکرد سیستم‌های غیرمتمرکز تا چه اندازه مهم بوده و می‌توانند به ساخت دنیایی آزاد و امن کمک کنند همچنین نگاهی به پروژه های انجام گرفته به کمک قراردادهای هوشمند در زمینه آموزش داشتیم و در انتها با تحلیل و بررسی پلتفرم smartlearn به عنوان یک Dapp با امکانات تعبیه شده در آن را آشنا شدیم و توانستیم با درک کاملی از ابزارهای لازم جهت ساخت Dappها پلتفرم smartlearn را پیاده‌سازی و اجرا نماییم، در این فصل به عنوان فصل پایانی قصد داریم ایده‌هایی جهت ادامه مسیر توسعه این پلتفرم را مطرح کنیم.

## پیشنهادها

همانطور که امکان ارسال پیشنهاد در خود پلتفرم smartlearn تعبیه شده، رویکرد ما نسبت به ادامه مسیر توسعه این پلتفرم انعطاف‌پذیر بوده و تمامی پیشنهادات برای ایده گرفتن و بهتر کردن عمل‌کرد این پلتفرم با آغوشی باز پذیرا هستیم اما همانطور که در فصل مقدمه اهداف این پلتفرم مشخص شد خط مشی و چهارچوب ثابتی برای اعمال پیشنهادات وجود دارد که یکی از مهم‌ترین آن‌ها هدف تحقق آرمان آموزش برای همه می‌باشد.

در جهت تحقق این آرمان تلاش ما این است تا با ایجاد سازوکار های مالی پایا نیازهای مالی این پلتفرم را پوشش دهیم و طبق این خواسته ایده‌هایی برای ادامه مسیر این پلتفرم وجود دارد :

1. ارائه API به سایت های ارائه دهنده دوره‌های اموزشی برای انجام پرداخت‌های کاربرانشان از طریق پلتفرم smartlearn
2. ایجاد یک سازوکار پویا و قابل اعتماد با ایجاد یک قرارداد هوشمند دیگر جهت انجام امور ترجمه و تولید دوره‌های ترجمه شده و رایگان

فهرست مرجع‏ها

|  |  |
| --- | --- |
| [1] | E. Grant, Ancient Egypt to Plato, New York: Cambridge University Press, 2007. |
| [2] | D. S. Halacy, Charles Babbage, Father of the Computer, Crowell-Collier Press, 1970. |
| [3] | J. Evans, "nodes," [Online]. Available: https://nodes.com. |
| [4] | M. Rajarshi , "medium," 25 5 2020. [Online]. Available: https://medium.com/bitbees/what-the-heck-is-utxo-ca68f2651819. |
| [5] | Canhui Chen, Xu Chen, Jiangshan Yu, Weigang Wu , Di Wu , "Impact of Temporary Fork on the Evolution of Mining Pools in Blockchain Networks: An Evolutionary Game Analysis," *IEEE Transactions on Network Science and Engineering,* 2020. |
| [6] | P. contribution, "ethereum.org," [Online]. Available: https://ethereum.org/en/developers/docs/evm/. |
| [7] | T. McCallum, "medium," 11 2 2018. [Online]. Available: https://medium.com/cybermiles/diving-into-ethereums-world-state-c893102030ed. |
| [8] | G. Wood, "ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER," 2014. |
| [9] | Mialaret Gaston،Vial Jean, Histoire Mondiale de l'éducation, Paris: PUF, 1981. |
| [10] | David Jonassen, Michael J. Spector, Marcy Driscoll, M. David Merrill, Jeroen van Merrienboer, Marcy P. Driscoll, Handbook of Research on Educational Communications and Technology, New York: Routledge, 2007. |
| [11] | Alammary A, Alhazmi S, Almasri M, Gillani S, "Blockchain-Based Applications in Education: A Systematic Review," *Applied Sciences,* 2019. |
| [12] | Ta Nguyen Binh Duong, Joel Yang Tian Jun, "EtherLearn: Decentralizing learning via blockchain," *IEEE,* 2022. |
| [13] | G. E. D. a. F. A. Paganini, A course in robust control theory: A convex approach, New York, NY: Springer, 2000. |
| [14] | T. Mendyk-Krajewska and Z. Mazur, "Problem of network security threats," *3rd International Conference on Human System Interaction,* pp. 436 - 443, 2010. |
| [15] | A. Galves, "Bezier Curve and Surface Fitting of 3D Point Clouds Through Genetic Algorithms, Functional Networks and Least-Squares Approximation," 2007. |
| [16] | Fibikova, Lenka, Müller, Roland and Pohlmann, Norbert, "A Simplified Approach for Classifying Applications," *ISSE 2010 Conference on Securing Electronic Business Processes: Highlights of the Information Security Solutions Europe,* pp. 39-49, 2010. |
| [17] | R. Etges and K. McNeil, "Understanding data classification based on business and security," *ISACA Information Systems Control Journal,* vol. 5, 2006. |
| [18] | E. Bergström and R.-M. Åhlfeldt, "Information classification issues," *Nordic Conference on Secure IT Systems,* pp. 27-41, Springer International Publishing, 2014. |
| [19] | J. L. Bayuk, "The utility of security standards," *44th Annual IEEE International Carnahan Conference on Security Technology,* pp. 1-6, 2010. |

**Abstract**



**K. N. Toosi University of Technology**

**Faculty of Computer Engineering**

**A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science (B.Sc.)**

**in Computer Engineering - Software**

[Thesis Title]

**By:**

[student name]

**Supervisor:**

[supervisor name(s)]

**Advisor:**

[advisor name(s)]

Summer 2022

1. Online Training (E-learn) [↑](#footnote-ref-2)
2. Covid-19 [↑](#footnote-ref-3)
3. Course [↑](#footnote-ref-4)
4. Deadline [↑](#footnote-ref-5)
5. Platform [↑](#footnote-ref-6)
6. Technology [↑](#footnote-ref-7)
7. Blockchain [↑](#footnote-ref-8)
8. Smart Contracts [↑](#footnote-ref-9)
9. Ethereum [↑](#footnote-ref-10)
10. Financial Incentive [↑](#footnote-ref-11)
11. Django [↑](#footnote-ref-12)
12. Metamask [↑](#footnote-ref-13)
13. Solidity [↑](#footnote-ref-14)
14. Online courses [↑](#footnote-ref-15)
15. User [↑](#footnote-ref-16)
16. Anonymous [↑](#footnote-ref-17)
17. Privacy [↑](#footnote-ref-18)
18. Advertising [↑](#footnote-ref-19)
19. Sociology [↑](#footnote-ref-20)
20. Self-motivation [↑](#footnote-ref-21)
21. Prize [↑](#footnote-ref-22)
22. Financial Life Cycle [↑](#footnote-ref-23)
23. Mechanism [↑](#footnote-ref-24)
24. Stable [↑](#footnote-ref-25)
25. Cryptocurrency [↑](#footnote-ref-26)
26. Satoshi Nakamoto [↑](#footnote-ref-27)
27. Vitalik Buterin [↑](#footnote-ref-28)
28. Gavin Wood [↑](#footnote-ref-29)
29. Database [↑](#footnote-ref-30)
30. Backend [↑](#footnote-ref-31)
31. Python [↑](#footnote-ref-32)
32. Framework [↑](#footnote-ref-33)
33. Java Script (JS) [↑](#footnote-ref-34)
34. User stories [↑](#footnote-ref-35)
35. Software Requirements Document [↑](#footnote-ref-36)
36. Information [↑](#footnote-ref-37)
37. Storage [↑](#footnote-ref-38)
38. Data [↑](#footnote-ref-39)
39. Charles Babbage [↑](#footnote-ref-40)
40. fetching [↑](#footnote-ref-41)
41. Centralized Systems [↑](#footnote-ref-42)
42. UnDistributed [↑](#footnote-ref-43)
43. Hack [↑](#footnote-ref-44)
44. Open Source [↑](#footnote-ref-45)
45. Apache [↑](#footnote-ref-46)
46. Decentralize [↑](#footnote-ref-47)
47. Developer [↑](#footnote-ref-48)
48. Cloud Service Provider [↑](#footnote-ref-49)
49. Cloudflare [↑](#footnote-ref-50)
50. Distributed [↑](#footnote-ref-51)
51. Server [↑](#footnote-ref-52)
52. Bitcoin [↑](#footnote-ref-53)
53. Crypto Currency [↑](#footnote-ref-54)
54. Blockchain [↑](#footnote-ref-55)
55. Distributed Ledger [↑](#footnote-ref-56)
56. مجموعه‌ای از عملیات که مانند یک واحد منطقی عمل می‌کنند به طور مثال انتقال پول در بانک که شامل مجموعه عملیات{ کم شدن از یک حساب - انتقال - اضافه شدن به حساب دیگر } مجموعا یک تراکنش محسوب می‌شوند، تراکنش‌ها لزوما مالی نبوده، رای دادن و یا انتقال مالکیت و ... نیز تراکنش به حساب می‌آیند [↑](#footnote-ref-57)
57. Node [↑](#footnote-ref-58)
58. State [↑](#footnote-ref-59)
59. Atomic [↑](#footnote-ref-60)
60. Consistent [↑](#footnote-ref-61)
61. Balance [↑](#footnote-ref-62)
62. Durable [↑](#footnote-ref-63)
63. Distributed Ledger Technology [↑](#footnote-ref-64)
64. Block [↑](#footnote-ref-65)
65. Genesis block [↑](#footnote-ref-66)
66. Stuart Haber [↑](#footnote-ref-67)
67. w. scott stornetta [↑](#footnote-ref-68)
68. Timestamping [↑](#footnote-ref-69)
69. Host [↑](#footnote-ref-70)
70. Fork [↑](#footnote-ref-71)
71. Full node [↑](#footnote-ref-72)
72. Pruned nodes [↑](#footnote-ref-73)
73. Archival nodes [↑](#footnote-ref-74)
74. در گره هرس شده به حداقل یک ترابایت حافظه پر سرعت نیاز داریم و این گره حدود ۱۲۸ بلوک آخر را در خود ذخیره می‌کند ولی در گره بایگانی به حداقل دوازده ترابایت حافظه پر سرعت نیاز بوده و تمام شبکه بلاکچین را در خود ذخیره می‌کند [↑](#footnote-ref-75)
75. Mining node [↑](#footnote-ref-76)
76. Proof of work [↑](#footnote-ref-77)
77. Lightweight node [↑](#footnote-ref-78)
78. Broadcast [↑](#footnote-ref-79)
79. Body [↑](#footnote-ref-80)
80. Hash [↑](#footnote-ref-81)
81. Algorithm [↑](#footnote-ref-82)
82. Data structure [↑](#footnote-ref-83)
83. Merkle tree [↑](#footnote-ref-84)
84. Satoshi (1 satoshi = 0.00000001 BTC) [↑](#footnote-ref-85)
85. Transaction Fees [↑](#footnote-ref-86)
86. Unspent Transaction Output (UTXO) [↑](#footnote-ref-87)
87. double spending [↑](#footnote-ref-88)
88. Coinbase [↑](#footnote-ref-89)
89. BTC ( The Cryptocurrency of The Bitcoin Blockchain) [↑](#footnote-ref-90)
90. Digital signature [↑](#footnote-ref-91)
91. Asymmetric cryptography [↑](#footnote-ref-92)
92. non-repudiation [↑](#footnote-ref-93)
93. Validation [↑](#footnote-ref-94)
94. Encryption [↑](#footnote-ref-95)
95. Decryption [↑](#footnote-ref-96)
96. Network Difficulty [↑](#footnote-ref-97)
97. Nonce [↑](#footnote-ref-98)
98. Unix Timestamp ( From January 1st, 1970 ) [↑](#footnote-ref-99)
99. The Bitcoin Version Number [↑](#footnote-ref-100)
100. Proof of work [↑](#footnote-ref-101)
101. Consensus Mechanism [↑](#footnote-ref-102)
102. Byzantine fault tolerance [↑](#footnote-ref-103)
103. Spam [↑](#footnote-ref-104)
104. CPU [↑](#footnote-ref-105)
105. Email [↑](#footnote-ref-106)
106. Character [↑](#footnote-ref-107)
107. 0000f79349c800b2ef5ed40ba485e4abb75158f60f0fe7a962b5bd0fa6ccf1a0 [↑](#footnote-ref-108)
108. Hash rate or Hashing power (به تعداد حدس‌های گره‌ها برای تعیین هش با شرایط خواسته شده در یک ثانیه) [↑](#footnote-ref-109)
109. Temporary Fork [↑](#footnote-ref-110)
110. Longest Chain [↑](#footnote-ref-111)
111. Orphaned Block [↑](#footnote-ref-112)
112. Transaction Pool [↑](#footnote-ref-113)
113. 2\*7\*24\*60 = 20160 minutes [↑](#footnote-ref-114)
114. Virtual machine [↑](#footnote-ref-115)
115. Uninterrupted [↑](#footnote-ref-116)
116. Immutable [↑](#footnote-ref-117)
117. Trie [↑](#footnote-ref-118)
118. Practical Algorithm To Retrieve Information Coded In Alphanumeric (Patricia) [↑](#footnote-ref-119)
119. اتریوم در اصل از نسخه تغییریافته این ساختارداده استفاده می‌کند [↑](#footnote-ref-120)
120. Key/Value [↑](#footnote-ref-121)
121. Distributed State Machine [↑](#footnote-ref-122)
122. State Chain [↑](#footnote-ref-123)
123. Externally-Owned Account(EOA) [↑](#footnote-ref-124)
124. Zero Address (0x0000000000000000000000000000000000000000) [↑](#footnote-ref-125)
125. Contract Account [↑](#footnote-ref-126)
126. Message Call [↑](#footnote-ref-127)
127. Byte [↑](#footnote-ref-128)
128. Nibble [↑](#footnote-ref-129)
129. Random Data [↑](#footnote-ref-130)
130. One ether = 1,000,000,000,000,000,000 wei (1018) [↑](#footnote-ref-131)
131. Storage Root [↑](#footnote-ref-132)
132. State Variables [↑](#footnote-ref-133)
133. Code Hash [↑](#footnote-ref-134)
134. Byte code [↑](#footnote-ref-135)
135. Ethereum Virtual Machine (EVM) [↑](#footnote-ref-136)
136. World State Root [↑](#footnote-ref-137)
137. Protocol [↑](#footnote-ref-138)
138. State Machine [↑](#footnote-ref-139)
139. Gas [↑](#footnote-ref-140)
140. Infinite Loop [↑](#footnote-ref-141)
141. Gas Price [↑](#footnote-ref-142)
142. Gas Limit [↑](#footnote-ref-143)
143. Constructor [↑](#footnote-ref-144)
144. Self-executing [↑](#footnote-ref-145)
145. Nick Szabo [↑](#footnote-ref-146)
146. Viper [↑](#footnote-ref-147)
147. Low-level Lisp-like Language [↑](#footnote-ref-148)
148. Compile [↑](#footnote-ref-149)
149. Opcodes [↑](#footnote-ref-150)
150. Total Turing machine [↑](#footnote-ref-151)
151. Deterministic State Transition Function [↑](#footnote-ref-152)
152. Review Article [↑](#footnote-ref-153)
153. Peer-Learning System [↑](#footnote-ref-154)
154. Agile software development [↑](#footnote-ref-155)
155. بخش انتهایی داستانه‌های کاربر اختیاری است [↑](#footnote-ref-156)
156. cryptocurrency [↑](#footnote-ref-157)
157. Ether (رمزارز شبکه اتریم) [↑](#footnote-ref-158)
158. Crypto wallet [↑](#footnote-ref-159)
159. Chrome – Firefox – Brave - Edge [↑](#footnote-ref-160)
160. Button [↑](#footnote-ref-161)
161. Header [↑](#footnote-ref-162)
162. Array [↑](#footnote-ref-163)
163. Box [↑](#footnote-ref-164)
164. Object [↑](#footnote-ref-165)
165. Method [↑](#footnote-ref-166)
166. Optional [↑](#footnote-ref-167)
167. Dapp (Decentralized application) [↑](#footnote-ref-168)
168. Main Network [↑](#footnote-ref-169)
169. Deploy [↑](#footnote-ref-170)
170. Test Network [↑](#footnote-ref-171)
171. Local Network [↑](#footnote-ref-172)
172. Go Ethereum [↑](#footnote-ref-173)
173. Bandwidth [↑](#footnote-ref-174)
174. Application Programming Interface [↑](#footnote-ref-175)
175. Syntax [↑](#footnote-ref-176)
176. Compiler [↑](#footnote-ref-177)
177. Text Editor [↑](#footnote-ref-178)
178. VS code [↑](#footnote-ref-179)
179. Node Package Manager [↑](#footnote-ref-180)
180. Configuration [↑](#footnote-ref-181)
181. Efficient [↑](#footnote-ref-182)
182. Web [↑](#footnote-ref-183)
183. Remote Procedure Call [↑](#footnote-ref-184)
184. Application Binary Interface [↑](#footnote-ref-185)
185. State Management [↑](#footnote-ref-186)
186. Inject [↑](#footnote-ref-187)
187. Console [↑](#footnote-ref-188)
188. User Panel [↑](#footnote-ref-189)
189. Cross-Origin Resource Sharing [↑](#footnote-ref-190)
190. Kernel [↑](#footnote-ref-191)
191. Linux OS (Operation System) [↑](#footnote-ref-192)
192. Windows OS [↑](#footnote-ref-193)
193. Windows Subsystem for Linux [↑](#footnote-ref-194)
194. Node Version Manager [↑](#footnote-ref-195)
195. Terminal [↑](#footnote-ref-196)
196. Forgot Password [↑](#footnote-ref-197)
197. Copy [↑](#footnote-ref-198)
198. Reset Wallet [↑](#footnote-ref-199)