



دانشگاه تربیت مدرس

دانشکده مهندسی برق و کامپیوتر

فعالیت سوم کلاسی:

گزارش مقالات پایه

عنوان مقاله:

An overview of blockchain smart contract execution mechanism

مرواری بر مکانیسم اجرای قرارداد هوشمند بلاک چین

استاد محترم درس:

سرکار خانم دکتر مریم لطفی

دانشجو: علی اردشیر

شماره دانشجویی: ۶۰۳۱۶۳۱۶۳۶۰۴

تاریخ تحويل: ۱۴۰۳/۱۲/۲۹

فهرست مطالب

۳	مقدمه
۴	پیش‌زینی
۵	اعتمادپذیری بالا
۶	فناوری‌های مرتبط با قراردادهای هوشمند
۷	تولید، استقرار و اجرای قراردادهای هوشمند
۸	زبان‌های برنامه‌نویسی قراردادهای هوشمند
۹	محیط اجرای قراردادهای هوشمند
۱۰	منابع داده برای قراردادهای هوشمند
۱۱	امنیت قراردادهای هوشمند
۱۱	مکانیزم اجرای قراردادهای هوشمند در پلتفرم‌های بلاکچین
۱۲	مروری بر پلتفرم‌های بلاکچین
۱۵	مقایسه زبان‌های قرارداد هوشمند
۱۶	مقایسه اجرای قراردادهای هوشمند
۱۶	بهینه‌سازی اجرای قراردادهای هوشمند
۱۷	اجرای همزمان مبتنی بر معماری
۱۷	اجرای همزمان مبتنی بر تفکیک نقش‌های نودها
۱۸	اجرای همزمان مبتنی بر شاردنگ
۱۸	اجرای همزمان مبتنی بر DAG
۱۹	حل تعارض‌های همزمان در بلاکچین
۲۰	اجرای زنجیره جانبی در بلاکچین
۲۱	قراردادهای هوشمند در خدمت صنعت
۲۱	نتیجه‌گیری

مقدمه

قراردادهای هوشمند، که نخستین بار توسط نیک زابو مفهومسازی شدند، به عنوان تعهدات دیجیتالی تعریف می‌شوند که بر اساس شرایط از پیش تعیین شده، به صورت خودکار اجرا می‌شوند و هدف آن‌ها خودکارسازی توافقات قانونی است. در ابتدا، نبود محیط‌های اجرایی امن مانع برای پیشرفت آن‌ها محسوب می‌شد، اما ظهور فناوری بلاکچین این مشکل را برطرف کرد و زیرساخت لازم برای اجرای مطمئن قراردادهای هوشمند را فراهم آورد. فناوری بلاکچین که ریشه در سیستم‌های ارز دیجیتال دارد، به عنوان پایه‌ای امن و قابل اعتماد برای اجرای قراردادهای هوشمند عمل می‌کند.

ظهور اتریوم نقطه عطف مهمی بود که ارتباط قراردادهای هوشمند و بلاکچین را تقویت کرد. قراردادهای هوشمند قابلیت برنامه‌پذیری را به بلاکچین اضافه می‌کنند و با ویژگی‌هایی مانند اجرای خودکار و تأیید چندطرفه، امکان پیاده‌سازی کاربردهای متنوعی در دنیای واقعی را فراهم می‌سازند. این قراردادها دامنه کاربرد بلاکچین را فراتر از ارزهای دیجیتال گسترش داده و به حوزه‌هایی که نیازمند اعتماد و همکاری هستند، وارد کرده‌اند.

قراردادهای هوشمند در سیستم‌های بلاکچین مدرن بسیار متنوع و انعطاف‌پذیر هستند و به زبان‌های برنامه‌نویسی مختلف نوشته می‌شوند تا منطق پیچیده و عملکردهای متنوع را اجرا کنند. این قراردادها به عنوان برنامه‌های مبتنی بر دفتر کل توزیع شده عمل کرده و امکان تبادل اطلاعات، انتقال ارزش و مدیریت دارایی‌ها را فراهم می‌کنند.

قراردادهای هوشمند می‌توانند جایگزین یا مکمل قراردادهای قانونی شوند، زیرا دارای قابلیت‌های خوداحرازگری و اجرای خودکار هستند. با وجود پیشرفت‌ها، چالش‌هایی در عملکرد بلاکچین همچنان پابرجا هستند و پژوهش‌های بیشتری برای بهبود کارایی قراردادهای هوشمند موردنیاز است.

پیش‌زمینه

فناوری بلاکچین که ریشه در بیت‌کوین دارد، ترکیبی از رمزنگاری، اجماع توزیع شده، شبکه‌های همتا^۱ و قراردادهای هوشمند است که یک الگوی نوین محاسباتی را ایجاد کرده است. این فناوری از ساختار داده‌ای مبتنی بر زنجیره بلوکی برای ذخیره‌سازی و تأیید داده‌ها، الگوریتم‌های اجماع توزیع شده برای تولید داده‌ها و روش‌های رمزنگاری برای تأمین امنیت اطلاعات استفاده می‌کند.

بلاکچین از ساختار داده ساده‌ای متشکل از یک سربرگ بلوک و بدنه بلوک بهره می‌برد که در آن ساختارهای زنجیره‌ای^۲، الگوریتم‌های هش^۳، درخت‌های مرکل^۴ و برچسب‌های زمانی^۵ نقش کلیدی دارند (شکل ۱). این فناوری در مقایسه با پایگاه‌های داده مرکز سنتی، ویژگی‌هایی مانند عدم تمرکز، یکپارچگی تغییرناپذیر داده‌ها، قابلیت ردیابی و سطح اعتماد بالا را ارائه می‌دهد.

در یک شبکه بلاکچینی غیرمت مرکز، هر شرکت‌کننده می‌تواند یک گره را برای ذخیره داده‌ها ارائه دهد، که این امر اشتراک‌گذاری اطلاعات به صورت توزیع شده و میان چندین طرف را امکان‌پذیر می‌سازد. بهره‌گیری از اصول رمزنگاری باعث می‌شود که سوابق ذخیره شده در زنجیره غیرقابل دستکاری باشند، زیرا هر بلوک شامل برچسب زمانی بلوک قبلی است و داده‌ها به صورت توالی زمانی مرتب می‌شوند.

عدم امکان تغییرپذیری داده‌ها در بلاکچین ناشی از هزینه بالای تغییر آن‌هاست؛ برای تغییر اطلاعات، بیش از ۵۱٪ از گره‌های شبکه باید در کنترل یک فرد یا گروه باشد که انجام چنین حمله‌ای از نظر هزینه بسیار سنگین و عمل‌آمیز غیرممکن است.

ساختار ثبت وقایع تغییرناپذیر در بلاکچین، امکان بازیابی و ردیابی تمامی عملیات انجام شده در گذشته را فراهم کرده و ابزار مؤثری برای ممیزی و نظارت‌های قانونی محسوب می‌شود. علاوه بر این، ثبت لحظه‌ای داده‌ها در زنجیره بلوکی، قابلیت ردیابی دقیق داده‌ها در طول زمان را تضمین می‌کند و به هر تراکنش بعد زمانی مشخصی می‌بخشد.

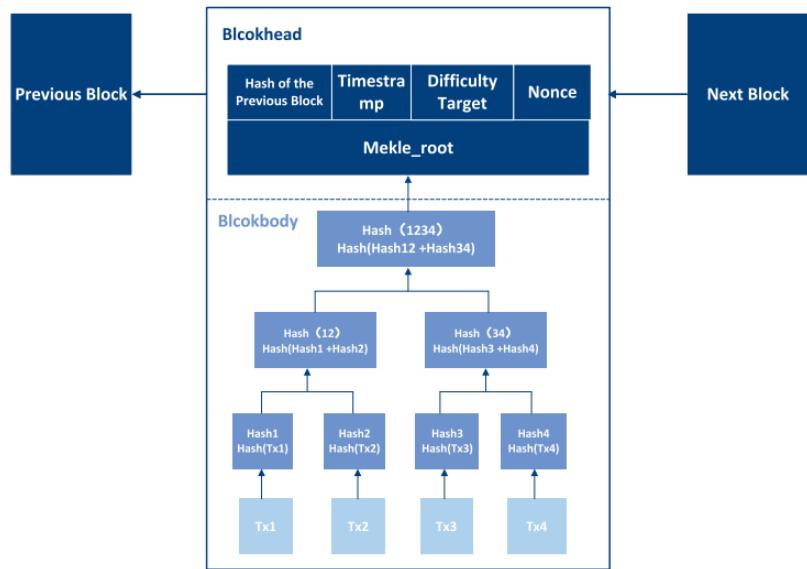
¹ Peer to peer

² Chain structures

³ Hash algorithms

⁴ Merkle trees

⁵ timestamps



شکل ۱ - نمودار شماتیک ساختار داده بلاک

اعتمادپذیری بالا

فناوری بلاکچین یک پایگاه داده امن و غیرمت مرکز است که امکان تراکنش های همتا به همتا را بدون نیاز به واسطه یا اعتماد متقابل بین طرفین فراهم می کند.

هر تراکنش پیش از ثبت شدن، نیازمند احراز هویت فرستنده و تأیید اجماع شبکه است. پس از ثبت، اطلاعات تراکنش غیرقابل تغییر و انکار خواهند بود، که این ویژگی نقش مهمی در افزایش اعتبار داده ها و ایجاد یک سیستم قابل نظارت و شفاف ایفا می کند.

بلاکچین با امکان ثبت رسمی، داوری و همکاری خودکار بر اساس قوانین از پیش تعیین شده، نیاز به واسطه های سنتی را کاهش می دهد. همچنین، این فناوری ایجاد اعتماد در محیط های غیرقابل اعتماد را با هزینه های پایین تر امکان پذیر می سازد.

علاوه بر حوزه مالی، بلاکچین در مدیریت زنجیره تأمین، ردیابی محصولات، اینترنت اشیاء و بسیاری از کاربردهای صنعتی و تجاری مورد استفاده قرار گرفته است. این فناوری به عنوان یک پروتکل پایه برای احراز هویت اعتباری جهانی و انتقال ارزش شناخته می شود و نقشی اساسی در توسعه اقتصاد دیجیتال و سیستم های اعتماد آینده خواهد داشت.

⁶ IOT

فناوری‌های مرتبط با قراردادهای هوشمند

قراردادهای هوشمند به عنوان کدهای برنامه‌نویسی شده‌ای عمل می‌کنند که به صورت خودکار شرایط اجرا را از طریق داده‌های خارجی بررسی کرده و در صورت برآورده شدن شروط، وضعیت قرارداد را به روزرسانی می‌کنند.

اجرای یک قرارداد هوشمند منجر به ایجاد تراکنش‌های جدید در شبکه بلاکچین می‌شود. این تراکنش‌ها می‌توانند توسط هر گره‌ای در شبکه بلاکچین آغاز شوند و روند اجرای قرارداد را به جریان بیندازند.

قطعیت در نتایج اجرای قراردادهای هوشمند یکی از فاکتورهای مهم برای اطمینان از صحت خروجی‌ها و جلوگیری از رفتارهای غیرمنتظره در شبکه است.

چرخه حیات قراردادهای هوشمند شامل سه مرحله اصلی است:

۱. ایجاد قرارداد:

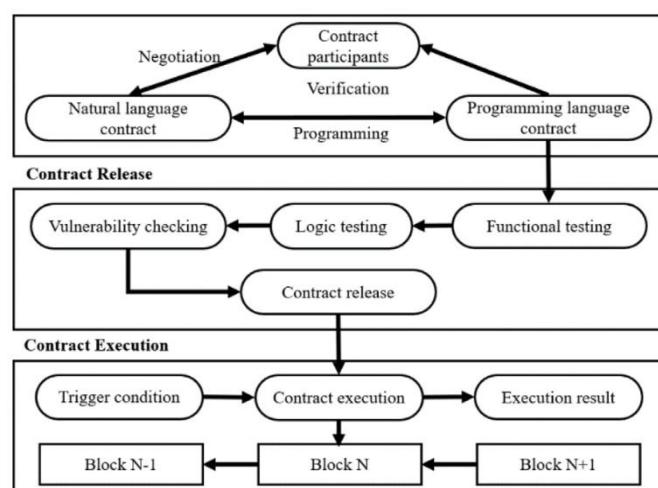
طراحی و برنامه‌نویسی قرارداد بر اساس نیازهای موردنظر.

۲. استقرار قرارداد:

انتشار قرارداد بر روی بلاکچین برای اجرا در محیطی غیرمتبرکز.

۳. اجرای قرارداد:

فعال شدن قرارداد هوشمند و انجام عملیات بر اساس شرایط از پیش تعیین شده.



شکل-۲- چرخه حیات قرارداد هوشمند

تولید، استقرار و اجرای قراردادهای هوشمند

تولید قراردادهای هوشمند

فرآیند تولید قراردادهای هوشمند شامل چندین مرحله اساسی است:

- **تعیین منطق تجاری:** طرفین قرارداد ابتدا منطق تجاری موردنظر را مشخص کرده و آن را در قالب یک قرارداد به زبان طبیعی تنظیم می‌کنند.

- **ترجمه به زبان برنامه‌نویسی:** پس از امضای قرارداد، متن آن به کد قرارداد هوشمند در یک زبان برنامه‌نویسی سازگار با بلاکچین تبدیل می‌شود تا قابلیت اجرا روی گره‌های توزیع شده را داشته باشد.

- **اعتبارسنجی قرارداد:** قبل از ورود به مرحله استقرار، بررسی و تأیید هر دو نسخه (متن طبیعی و نسخه برنامه‌نویسی شده) ضروری است تا از تطابق آنها و درستی عملکرد قرارداد اطمینان حاصل شود.

استقرار قراردادهای هوشمند

استقرار قراردادهای هوشمند شامل دو مرحله کلیدی است:

- **آزمایش قرارداد:**

از آنجایی که قراردادهای هوشمند پس از استقرار غیرقابل تغییر هستند، باید تحت آزمایش‌های دقیق قرار گیرند. این آزمایش‌ها شامل بررسی عملکرد، تست منطق قرارداد و شناسایی آسیب‌پذیری‌ها می‌شود.

- **انتشار قرارداد:**

تنها قراردادهایی که تمامی آزمون‌ها را با موفقیت پشت سر گذاشته‌اند، در پلتفرم بلاکچین مستقر می‌شوند. قراردادها بر اساس سیاست‌های تعیین‌شده منتشر شده و برای اجرا روی گره‌های شبکه آماده می‌شوند.

اجرای قراردادهای هوشمند

قراردادهای هوشمند زمانی اجرا می‌شوند که شرایط از پیش تعیین‌شده در تراکنش‌ها محقق شود. فرآیند اجرا شامل مراحل زیر است:

• شرایط فعالسازی و اجراء:

هنگامی که یک تراکنش شرط فعالسازی تعیین شده در قرارداد را برآورده کند، اجرای قرارداد روی گرههای توزیع شده آغاز می‌شود.

• بسته‌بندی نتایج:

خروجی‌های حاصل از اجرای قرارداد در قالب تراکنش‌های جدید بسته‌بندی می‌شوند تا در یک بلوک قرار گیرند.

• ادغام در بلاکچین:

پس از دستیابی به اجماع در شبکه، بلوک حاوی نتایج اجرا به دفتر کل بلاکچین افزوده می‌شود و اطلاعات قرارداد به‌طور دائمی ثبت می‌گردد.

زبان‌های برنامه‌نویسی قراردادهای هوشمند

زبان برنامه‌نویسی قراردادهای هوشمند نقش مهمی در پیاده‌سازی و توسعه کاربردهای بلاکچینی ایفا می‌کند. انتخاب زبان مناسب به ساختار شبکه، امنیت، و کارایی قراردادهای هوشمند بستگی دارد.

• زبان اسکریپتنویسی بیت‌کوین

بیت‌کوین از یک زبان پشتهدای⁷ برای تأیید تراکنش‌ها استفاده می‌کند. این زبان از کنترل جریان پیچیده پشتیبانی نمی‌کند و در مقایسه با زبان‌های همه‌منظوره محدودیت‌های عملکردی زیادی دارد.

• زبان‌های برنامه‌نویسی در اتریوم

اتریوم از Solidity و Vyper⁸ برای توسعه قراردادهای هوشمند پشتیبانی می‌کند. Solidity محبوب‌ترین زبان است که یک زبان سطح بالا، شی‌گرا و مبتنی بر قرارداد محسوب می‌شود و به‌طور خاص برای توسعه برنامه‌های غیرمت مرکز (Dapp) طراحی شده است. در Solidity، مدیریت خطأ و امنیت کد اهمیت بالایی دارد زیرا قراردادهای مستقر روی بلاکچین غیرقابل تغییر هستند. مکانیسم گس در اتریوم از مشکلاتی مانند حلقه‌های نامحدود جلوگیری می‌کند زیرا هر عملیات نیازمند پرداخت کارمزد است. Python، مشابه Vyper، با حذف ویژگی‌های پیچیده، امنیت را افزایش داده و فرآیند ممیزی قراردادها را آسان‌تر می‌کند.

⁷ Stack-based

⁸ Gas

Yul یک زبان واسط است که به عنوان یک مرحله میانی برای کامپایل از زبان‌های سطح بالا استفاده می‌شود و باعث افزایش انعطاف‌پذیری و خوانایی کد می‌شود.

محیط اجرای قراردادهای هوشمند

اجرای قراردادهای هوشمند در شبکه بلاکچین نیازمند محیط‌های محاسباتی ویژه و ایزووله شده است تا امنیت، کارایی و قابلیت اطمینان را تضمین کند. در این بخش، انواع مختلف محیط‌های اجرایی قراردادهای هوشمند بررسی شده‌اند.

• روش‌های پیاده‌سازی محیط اجرا

محیط‌های اجرای قراردادهای هوشمند معمولاً به سه دسته تقسیم می‌شوند:

۱. محیط اجرای مبتنی بر پشته (Stack-Based Execution)
۲. محیط اجرای ایزووله شده (Sandboxed Execution Environment)
۳. محیط اجرای مبتنی بر سخت‌افزار مورد اعتماد (Trusted Hardware Enclave Execution)

محیط اجرای مبتنی بر پشته

- سبک، سریع و امن است.
- بیت‌کوین از این روش برای اجرای تراکنش‌ها استفاده می‌کند.
- از اسکریپت‌های قفل و بازکردن (Locking & Unlocking Scripts) برای تأیید تراکنش‌ها بهره می‌برد.

محیط‌های اجرای ایزووله شده

شامل دو دسته‌ی اصلی است:

- ماشین‌های مجازی مانند EVM در اتریوم و JVM
- کانتینرها مانند Docker

این محیط‌ها اجرای کد قرارداد را در یک محیط ایزووله شده انجام می‌دهند تا از دسترسی غیرمجاز و تداخل منابع جلوگیری شود.

اجرای مبتنی بر سخت افزار مورد اعتماد

از فناوری هایی مانند Intel SGX و ARM TrustZone برای ایزوله سازی و محافظت از کد و داده های قرارداد هوشمند استفاده می شود.

منابع داده برای قراردادهای هوشمند

اوراکل های بلاکچین به عنوان واسطه هایی میان بلاکچین و منابع داده دنیای واقعی عمل می کنند و نقش مهمی در ارتباط اپلیکیشن های غیر مرکز (DApps) با اطلاعات خارجی دارند. در این بخش، ویژگی ها و چالش های استفاده از اوراکل ها برای تامین داده ها توضیح داده شده است:

نقش اوراکل ها در قراردادهای هوشمند

اوراکل ها به عنوان پل ارتباطی بین شبکه های بلاکچین و منابع داده خارجی عمل می کنند. آنها از داده های خارج از زنجیره (Off-chain) برای فعال سازی قراردادهای هوشمند استفاده کرده و اطلاعات را به صورت امن به بلاکچین منتقل می کنند. اوراکل ها می توانند از اطلاعات بیرونی برای برقراری ارتباط با قراردادهای هوشمند در بلاکچین استفاده کنند.

انواع اوراکل ها

• اوراکل های مرکزی:

مانند Oracle که از یک مکانیزم واحد برای جمع آوری داده ها استفاده می کنند.

• اوراکل های غیر مرکز:

مانند Chainlink و شبکه DoS که از چندین پروتکل مختلف برای فراهم کردن داده های مورد نیاز استفاده می کنند.

مزایا و چالش های استفاده از اوراکل ها

• مزایا:

اوراکل ها امنیت داده ها را در حین انتقال از منابع خارجی به بلاکچین افزایش می دهند.

• چالش‌ها:

داده‌هایی که اوراکل‌ها از منابع خارجی به بلاکچین منتقل می‌کنند ممکن است قابل دستکاری باشند. برای اطمینان از اعتبار داده‌ها، اوراکل‌ها باید مکانیزم‌هایی مانند الگوریتم‌های مبتنی بر شهرت^۹ یا استراتژی‌های تأیید داده‌ها را به کار ببرند.

امنیت قراردادهای هوشمند

اگرچه قراردادهای هوشمند به دلیل قرارگیری روی بلاکچین ذاتاً تغییرناپذیر هستند، اما همچنان در معرض تهدیدات امنیتی قرار دارند. یکی از معروف‌ترین نمونه‌های این آسیب‌پذیری، حادثه "The DAO" در ژوئن ۲۰۱۶ بود که نشان داد حتی در سیستم‌های غیرمت مرکز نیز امکان سوءاستفاده وجود دارد. در این حادثه، یک مهاجم از آسیب‌پذیری بازگشتی^{۱۰} استفاده کرد و حدود ۶۰ میلیون دلار اتر را سوت کرد.

با وجود اقدامات اصلاحی پس از حادثه DAO، مشکلات امنیتی قراردادهای هوشمند همچنان ادامه دارند. گزارش‌های مختلف نشان داده‌اند که حملات سایبری به قراردادهای هوشمند منجر به ضررهای مالی قابل توجهی در سطح جهانی شده‌اند.

امنیت قراردادهای هوشمند باید در تمامی مراحل توسعه، استقرار و اجرا مورد بررسی قرار گیرد. پژوهش‌های تیم SharkTeam نشان داده‌اند که بسیاری از حملات موفق به قراردادهای هوشمند، ناشی از ضعف‌های منطقی در کد قراردادها بوده است. با توجه به غیرقابل تغییر بودن قراردادهای هوشمند پس از استقرار، لازم است تست‌های امنیتی دقیق و مکانیزم‌های مقابله با آسیب‌پذیری‌ها قبل از انتشار انجام شوند.

مکانیزم اجرای قراردادهای هوشمند در پلتفرم‌های بلاکچینی

در این بخش، مکانیزم اجرای قراردادهای هوشمند در پلتفرم‌های بلاکچینی مورد بررسی قرار گرفته و تفاوت‌های بین بیت‌کوین، اتریوم، فریک، کوردا و EOS از جنبه‌های مختلف مقایسه شده است.

ابعاد مقایسه

مقایسه این پلتفرم‌ها در هفت جنبه کلیدی انجام شده است:

۱. زبان برنامه‌نویسی مورد استفاده برای توسعه قراردادهای هوشمند

⁹ Reputation-based

¹⁰ Reentrancy

۲. توانایی تورینگ کامل (Turing Completeness) و قابلیت اجرای عملیات پیچیده
۳. تعداد نودهای اجرایی که در پردازش و اجرای قراردادها مشارکت دارند
۴. اتصال اجرای قرارداد با فرآیند تولید بلاک و تأثیر آن بر کارایی سیستم
۵. مدل دادهای قراردادهای هوشمند و نحوه ذخیره‌سازی اطلاعات
۶. محیط اجرای قراردادها، از جمله ماشین‌های مجازی یا محیط‌های ایزوله شده
۷. پشتیبانی از اجرای همزمان قراردادهای هوشمند برای پردازش موازی

این تحلیل، تفاوت‌های اساسی در شیوه اجرای قراردادهای هوشمند در این پلتفرم‌ها را نشان داده و میزان مناسب بودن هر پلتفرم برای کاربردهای تجاری مختلف را مشخص می‌کند. نتایج این مقایسه به صورت جدول در بخش مربوطه ارائه شده است تا امکان مقایسه شفاف و دقیق بین مکانیزم‌های اجرایی این پلتفرم‌ها فراهم شود.

Plantform	Bitcoin	Ethereum 1.0	Hyperldeger Fabric	Corda	EOS
Programming language	OP_RETURN	Solidity	Go	Java	C++
Turing completeness	Incomplete	Complete	Complete	Incomplete	Complete
Execution nodes	All	All	1 or N	1 or N	1 or N
Execution and block production	Couple	Couple	Decouple	Decouple	Decouple
Data model	UTXO	Account	Account	UTXO	Account
Execution environment	Stack Engine	EVM	Docker	JVM	WebAssembly
Concurrent execution	Not	Not	Partially	Partially	Partially

جدول ۱- مکانیزم اجرای قراردادهای هوشمند در پلتفرم‌های بلاکچینی

مروری بر پلتفرم‌های بلاکچین

این بخش یک نمای کلی جامع از پلتفرم‌های بلاکچین مانند بیت‌کوین، اتریوم و هایپرلجر فبیریک ارائه می‌دهد و ویژگی‌ها و قابلیت‌های منحصر به‌فرد آن‌ها را بر جسته می‌کند:

بیت‌کوین

- تراکنش‌های سیستم بیت‌کوین بر ایجاد، تأیید و ثبت تراکنش‌های بیت‌کوین در یک دفتر کل جهانی متتمرکز هستند.

- تراکنش‌های بیت‌کوین سوابق عمومی در بلاکچین هستند و اطلاعات مربوط به انتقال ارزش را در بر می‌گیرند.
- بیت‌کوین فاقد حساب‌ها یا موجودی‌های سنتی است؛ در عوض، دارایی کاربران به صورت خروجی‌های تراکنش خرج‌نشده (UTXO) در بلاک‌ها ذخیره می‌شود.
- هر تراکنش از یک UTXO قبلی استفاده کرده و نودها برای اعتبارسنجی، خروجی‌های خرج‌نشده را ردیابی می‌کنند.
- تراکنش‌ها شامل ورودی‌ها، خروجی‌ها و یک مقدار هش هستند که در ایجاد درخت مرکل نقش مهمی دارند.
- تراکنش‌ها توسط نودها تأیید می‌شوند و در صورت موفقیت، به شبکه اضافه می‌شوند؛ در غیر این صورت، رد می‌شوند.
- ماینرها تراکنش‌ها را در قالب بلاک‌ها بسته‌بندی می‌کنند و اولین ماینری که مقدار هش معتبر را محاسبه کند، بلاک را ثبت می‌کند.
- سایر نودها بلاک را اعتبارسنجی می‌کنند؛ در صورتی که به بلاک‌های بعدی متصل شود، تراکنش‌ها تأیید بیشتری دریافت می‌کنند.

اتریوم

- اتریوم علاوه بر تراکنش‌ها، از قراردادهای هوشمند و برنامه‌های غیرمت مرکز (DApps) نیز پشتیبانی می‌کند.
- این شبکه در ابتدا مانند بیت‌کوین از الگوریتم اثبات کار (PoW) استفاده می‌کرد، اما در نسخه اتریوم ۲.۰ به اثبات سهام (PoS) منتقل شد که باعث افزایش توان عملیاتی و کاهش مصرف انرژی شد.
- در اتریوم ۲.۰، اعتبارسنجها اتر را به عنوان وثیقه ذخیره کرده و مؤلفه‌های نرم‌افزاری مستقل برای اجرا، اجماع و اعتبارسنجی را اجرا می‌کنند.
- تولید بلاک در اتریوم ۲.۰ در بازه‌های زمانی ثابت (slots) و دوره‌های مشخص (epochs) انجام می‌شود و اعتبارسنجها به صورت غیررقابتی، بلاک‌ها را پیشنهاد داده و اعتبارسنجی می‌کنند.
- اعتبارسنجها با رأی گیری (attestations) درباره اعتبار بلاک‌ها تصمیم‌گیری می‌کنند و این فرآیند منجر به توجیه (justification) و نهایی‌سازی (finality) بلاک‌ها می‌شود.

هایپرلجر فبریک

- هایپرلجر فبریک که توسط بنیاد لینوکس نگهداری می‌شود، یک راهکار بلاکچینی انعطاف‌پذیر و ایمن برای کسب‌وکارها ارائه می‌دهد.

- در هایپرلجر فبریک به عنوان قراردادهای هوشمند عمل کرده و منطق تجاری تراکنش‌ها را تعریف می‌کند.
- معماری مازولار این پلتفرم امکان سفارشی‌سازی را برای برآوردن نیازهای مختلف تجاری فراهم می‌کند.
- شبکه از نودهای کلاینت، نودهای همتا(Peer)، و نودهای سفارش‌دهنده(Orderer) برای پردازش تراکنش‌ها تشکیل شده است.
- نودهای تأییدکننده(Endorser) تراکنش‌ها را به صورت مستقل اعتبارسنجی کرده و مجموعه‌های خواندن و نوشتен(Read-Write Sets) را بدون تغییر داده‌های دفتر کل ایجاد می‌کنند.
- نودهای سفارش‌دهنده(Orderer) تراکنش‌ها را با استفاده از پروتکل‌های اجماع بسته‌بندی کرده و ترتیب کلی آن‌ها را در شبکه تضمین می‌کنند.
- نودهای همتا(Peer Nodes) بلاک‌های دریافتی را بررسی کرده و فقط بلاک‌های معتبر را در دفتر کل به روزرسانی می‌کنند.

کوردا

- به عنوان یک بلاکچین خصوصی برای حفظ حریم خصوصی تراکنش‌ها بین نهادهای شرکت‌کننده عمل می‌کند.
- شامل نودهایی است که نرمافزار کوردا و اپلیکیشن‌های مخصوص(CorDapps) را اجرا می‌کنند.
- از گواهی‌های دیجیتال برای پیوند هویت‌های واقعی به هویت‌های شبکه استفاده می‌کند.
- دارای پایگاه‌های داده توزیع شده است و قادر یک مخزن مرکزی اطلاعات می‌باشد.
- حقایق دفتر کل را با استفاده از States نمایش داده و از طریق مدل UTXO تغییرناپذیری را تضمین می‌کند.
- از مکانیزم‌های اجماع قابل تنظیم مانند Raft و BFT برای نیازهای مختلف پشتیبانی می‌کند.
- از انتقال پیام نقطه‌به‌نقطه برای به روزرسانی دفتر کل استفاده می‌کند.
- از سیستم Flows برای اتوماسیون تراکنش‌ها در پردازش‌های پیچیده و محترمانه بهره می‌برد.

طراحی کوردا به آن امکان مدیریت تراکنش‌های پیچیده و محترمانه، به ویژه در بخش مالی را می‌دهد و ویژگی‌هایی مانند مکانیسم‌های اجماع منحصر به فرد و اجرای خودکار تراکنش‌ها را ارائه می‌کند.

مقایسه زبان‌های قرارداد هوشمند

مقایسه زبان‌های قرارداد هوشمند در پلتفرم‌های مختلف بلاکچین، رویکردهای متفاوتی را در برنامه‌نویسی قراردادهای هوشمند نشان می‌دهد:

بیت‌کوین:

- زبان اسکریپت‌نویسی بیت‌کوین برای انجام تراکنش‌های خاص طراحی شده و نیازی به توانایی کامل تورینگ ندارد.

- عملکردهای اصلی آن طی بیش از یک دهه عملیات پایدار تأیید شده‌اند.

اتریوم:

- از زبان Solidity برای پردازش منطق پیچیده قراردادهای هوشمند و داده‌ها استفاده می‌کند.

هایبرلجر فبریک:

- از زبان‌هایی مانند Go برای اجرای منطق پیچیده تراکنش‌ها و بهبود عملکرد پشتیبانی می‌کند.

کوردا:

- از زبان‌های Java و سایر زبان‌های مبتنی بر JVM برای اسکریپت‌نویسی قراردادها پشتیبانی می‌کند.

- جاوا به دلیل زیرساخت‌های بالغ و پشتیبانی قوی، در کوردا محبوب است.

- کاتلین تعادل بین انعطاف‌پذیری و استحکام را نسبت به جاوا ارائه می‌دهد.

:EOS

- عمدتاً از زبان C++ برای میزبانی برنامه‌های بلاکچینی کارآمد استفاده می‌کند.

- از زبان‌هایی مانند Python و Rust نیز پشتیبانی می‌کند که به WebAssembly کامپایل می‌شوند تا نیازهای متنوع توسعه‌دهنده‌گان را برآورده کند.

انتخاب زبان برنامه‌نویسی قرارداد هوشمند عمدتاً به طراحی و نیازهای پلتفرم بلاکچین بستگی دارد. با پیشرفت فناوری بلاکچین، دامنه زبان‌های برنامه‌نویسی قابل استفاده برای قراردادهای هوشمند نیز گسترش خواهد یافت.

مقایسه اجرای قراردادهای هوشمند

مقایسه مکانیسم‌های اجرای قراردادهای هوشمند نشان‌دهنده تفاوت‌های قابل توجهی است که تحت تأثیر معماری پلتفرم و مکانیزم اجماع قرار دارند:

• بیت‌کوین و اتریوم ۱۰۰ از الگوریتم اثبات کار (PoW) استفاده می‌کنند که امنیت بالایی دارد اما باعث کاهش کارایی و مصرف بالای منابع می‌شود.

• اتریوم ۲۰۰ به اثبات سهام (PoS) منتقل شده و در نظر دارد با معرفی شارдинگ، توان عملیاتی را افزایش دهد.

• اتریوم ۲۰۰ به شاردهایی تقسیم می‌شود که هر کدام دارای مکانیزم اجماع مستقل هستند، این امر امکان پردازش همزمان تراکنش‌ها را فراهم کرده و توان عملیاتی را تا ۱۰۰,۰۰۰ تراکنش در ثانیه افزایش می‌دهد.

• هایپرلجر فربیک و کوردا به عنوان بلاکچین‌های کنسرسیومی، اجرای قرارداد را از تولید بلاک جدا می‌کنند تا توان عملیاتی را از طریق پردازش موازی افزایش دهند.

• کوردا از مدل UTXO برای بهبود کارایی اجرای قراردادهای هوشمند استفاده می‌کند که از طریق مذاکره تراکنش، اجرای موازی قراردادهای هوشمند و شارдинگ، کارایی اجرای قراردادها را افزایش می‌دهد.

بهینه‌سازی اجرای قراردادهای هوشمند

بهینه‌سازی اجرای قراردادهای هوشمند در سیستم‌های بلاکچینی برای مقابله با چالش‌های مقیاس‌پذیری و افزایش ظرفیت کلی توان عملیاتی بسیار حائز اهمیت است. در این بخش به نکات کلیدی زیر پرداخته شده است:

چالش مقیاس‌پذیری:

بزرگ‌ترین مانع فناوری بلاکچین، مقیاس‌پذیری آن است که مانع از گسترش و پذیرش گسترده‌تر این فناوری می‌شود. افزایش ظرفیت توان عملیاتی سیستم، راهکاری ضروری برای حل این مشکل محسوب می‌شود.

رویکردهای افزایش مقیاس پذیری:

در نظریه محاسبات توزیع شده، دو رویکرد اصلی برای افزایش مقیاس پذیری وجود دارد:

۱. مقیاس گذاری عمودی (Vertical Scaling)

۲. مقیاس گذاری افقی (Horizontal Scaling)

مقیاس گذاری افقی، که یک سیستم سریالی را به یک سیستم موازی تبدیل می کند، ترجیح داده می شود زیرا پردازش دستورالعمل ها را با سرعت بیشتری امکان پذیر می سازد.

راهکارهای لایه دوم (Layer 2)

محققان در حال بررسی پروتکل های لایه دوم مانند Lightning Network هستند تا بدون تغییر در پروتکل اصلی زنجیره، عملکرد و مقیاس پذیری را افزایش دهند. این راهکارها با هدف کاهش تنگناهای عملکردی و بهبود کارایی سیستم های بلاکچینی توسعه یافته اند.

اجرای همزمان مبتنی بر معماری

این بخش به طبقه بندی اجرای همزمان در فناوری بلاکچین پرداخته و راهبردهای طراحی شده برای حل مشکلات ناشی از تعارضات همزمان را بررسی می کند.

اجرای همزمان مبتنی بر تفکیک نقش های نودها

در بلاکچین های عمومی، ارتباط بین اجرای قراردادهای هوشمند و تولید بلاک منجر به پردازش زائد تراکنش ها می شود. برای حل این مشکل، محققان تلاش کرده اند تا نقش های نودها را تفکیک کرده و پردازش موازی را امکان پذیر کنند، که در نهایت باعث افزایش توان عملیاتی سیستم می شود.

روش های بهینه سازی اجرای همزمان:

۱. تفکیک نقش های نودها

• جداسازی نقش های اجرای قراردادهای هوشمند از فرآیند تولید بلاک

• افزایش کارایی از طریق پردازش همزمان تراکنش ها

۲. بهبودهای معماري

- معرفی ساختارهایی مانند کانال‌ها، گروه‌ها، و پارتیشن‌ها برای تقویت اجرای موازی قراردادهای هوشمند

اجrai همزمان مبتنی بر شارдинگ

مفهوم شارдинگ که اصلتاً از پایگاه‌داده‌ها نشأت گرفته، اکنون در بلاکچین برای افزایش عملکرد و ظرفیت سیستم استفاده می‌شود. این تکنیک با تقسیم شبکه به چندین بخش (شارد)، هر شارد به طور مستقل تراکنش‌ها و داده‌ها را پردازش می‌کند.

شارдинگ یک استراتژی برای بهبود عملکرد و ظرفیت شبکه‌های بلاکچین است که با تقسیم شبکه به بخش‌های مختلف صورت می‌گیرد. در شبکه‌های بلاکچین سنتی مانند Ethereum 1.0، تمامی نودها باید همه تراکنش‌ها را پردازش و ذخیره کنند که منجر به ازدحام و مشکلات عملکردی می‌شود.

شارдинگ بلاکچین شامل تقسیم شبکه به چندین بخش یا "شارد" است که هر شارد مسئول پردازش تراکنش‌ها و داده‌ها در یک دامنه خاص است. هر شارد تاریخچه تراکنش‌ها، وضعیت، و قراردادهای هوشمند خود را نگهداری می‌کند، به طوری که پردازش موازی امکان‌پذیر می‌شود و در عین حال تعاملات بین شاردهای مختلف ممکن است. شارдинگ امکان اجرای همزمان قراردادهای هوشمند را در بخش‌های مختلف شبکه فراهم می‌کند که منجر به کاهش ازدحام و افزایش مقیاس‌پذیری می‌شود.

اجrai همزمان مبتنی بر DAG

فناوری بلاکچین مبتنی بر DAG (گراف جهت‌دار غیرمدور) از ساختار داده‌ای به جای زنجیره‌های سنتی بلاکچین استفاده می‌کند. این رویکرد باعث بهبود عملکرد و افزایش مقیاس‌پذیری می‌شود. تاکنون، چندین پیاده‌سازی از سیستم‌های مبتنی بر DAG توسعه یافته‌اند، از جمله IOTA، NANO، Hashgraph، Byteball، Spectre و چارچوب‌های نوآورانه‌ای که توسط محققان پیشنهاد شده‌اند.

نمونه‌هایی از بلاکچین‌های مبتنی بر DAG :

IOTA •

از ساختار DAG تحت عنوان "Tangle" استفاده می‌کند که امکان پردازش همزمان تراکنش‌ها را فراهم می‌سازد. به طور ویژه برای کاربردهای اینترنت اشیا (IoT) مانند اشتراک‌گذاری داده، پرداخت‌های خرد، و مدیریت زنجیره تأمین طراحی شده است.

Hashgraph •

از یک الگوریتم ناهمگام (Asynchronous) برای حفظ DAG استفاده می‌کند که به نودها اجازه می‌دهد بدون نیاز به فرآیند استخراج، به‌طور غیرهمزمان به اجماع برسند و ترتیب تراکنش‌ها را تعیین کنند.

NANO •

از یک ساختار خاص به نام "Block Lattice" بهره می‌برد که در آن هر کاربر یک بلاکچین اختصاصی برای تراکنش‌های خود دارد. این مدل باعث می‌شود که تراکنش‌ها به صورت غیرهمزمان و با تأییدات سریع پردازش شوند. محدودیت‌های مقیاس‌پذیری بلاکچین‌های سنتی را با ارائه زنجیره‌های حساب مستقل برطرف می‌کند.

Spectre •

یک رویکرد جدید مبتنی بر DAG برای افزایش مقیاس‌پذیری و سرعت تأیید تراکنش‌ها ارائه می‌دهد. از تکنیک‌های رمزنگاری برای بهبود امنیت و عملکرد بهره می‌برد.

Byteball •

از ساختار DAG مبتنی بر واحدها (Units) برای اتصال تراکنش‌ها به صورت متوالی استفاده می‌کند. این روش تضمین می‌کند که تراکنش‌ها به صورت همزمان و غیرقابل تغییر تأیید شوند.

حل تعارض‌های همزمان در بلاکچین

همزمانی در بلاکچین باعث افزایش عملکرد می‌شود، اما در عین حال، به دلیل دسترسی همزمان چندین تراکنش به داده‌های مشترک، ممکن است منجر به تعارضات پردازشی شود. پژوهشگران در حال توسعه روش‌های مختلفی برای مدیریت و کاهش این تعارضات هستند.

برای کاهش تعارضات تراکنش‌ها، چندین روش مورد بررسی قرار گرفته‌اند:

۱. تحلیل استاتیک و دینامیک تراکنش‌ها:

- پیش‌بینی و شناسایی تعارضات احتمالی

- توقف پیش‌دستانه تراکنش‌های متعارض پیش از اجرا

۲. استفاده از تکنیک‌های کنترل همزمانی در پایگاه داده‌ها:

- **کشینگ (Caching)**: ذخیره داده‌های موقت برای کاهش برخوردهای تراکنش‌ها.
- **قفل‌گذاری (Locking)**: جلوگیری از تراکنش‌های همزمان که می‌توانند منجر به ناسازگاری شوند.
- **بازچینی (Reordering)**: تنظیم مجدد ترتیب اجرای تراکنش‌ها برای کاهش تعارضات.

اجرای زنجیره جانبی^{۱۱} در بلاکچین

زنجیره جانبی یک بلاکچین کمکی است که به صورت موازی با زنجیره اصلی اجرا می‌شود. این زنجیره دارای ویژگی‌های مستقل، مکانیسم‌های اجماع و قراردادهای هوشمند مخصوص به خود است، که می‌تواند برای کاربردهای خاص تنظیم شود بدون آنکه عملکرد زنجیره اصلی را تحت تأثیر قرار دهد.

ویژگی‌های کلیدی زنجیره جانبی

۱. تعامل با زنجیره اصلی:

- زنجیره‌های جانبی قابلیت انتقال دارایی بین زنجیره‌ای^{۱۲} را دارند.

- این ویژگی به انعطاف‌پذیری و شخصی‌سازی بیشتر در بلاکچین کمک می‌کند.

۲. چالش‌های زنجیره جانبی:

- قابلیت همکاری بین زنجیره‌ای^{۱۳}: امکان انتقال ایمن اطلاعات و دارایی بین زنجیره اصلی و جانبی.

- امنیت: حفظ امنیت زنجیره‌های جانبی بدون آسیب‌پذیری زنجیره اصلی.

- تمرکز زدایی: اطمینان از غیرمت مرکز بودن زنجیره جانبی بدون ایجاد نقاط ضعف.

هر پلتفرم برای حل این چالش‌ها از استراتژی‌های متفاوتی بهره می‌برد تا نیازهای متنوع کاربران و موارد استفاده مختلف را پوشش دهد.

¹¹ Sidechain Execution

¹² Cross-Chain Asset Transfer

¹³ Interoperability

قراردادهای هوشمند در خدمت صنعت

بلاکچین اعتماد را از طریق داده و ریاضیات برقرار می‌کند و اجرای ایمن قراردادهای هوشمند را ممکن می‌سازد. این فناوری مکانیسم‌های اعتماد اجتماعی را متتحول می‌کند و موجب شفافیت، امنیت و اعتماد در اقتصاد بازار می‌شود. شفافیت و تغییرناپذیری بلاکچین امکان ایجاد اعتماد بین طرفهای ناآشنا را فراهم کرده و همکاری و بهره‌وری را بهبود می‌بخشد. مکانیسم اعتماد بلاکچین می‌تواند فرآیندهای تجاری را بهینه کند و نوآوری در مکانیسم‌های اعتماد اجتماعی را به ارمغان آورد و نحوه ایجاد اعتماد را متتحول کند. با حرکت به سمت "زنگیره اعتماد" در صنعت، بلاکچین می‌تواند اشتراک‌گذاری داده‌های ایمن، شفاف و کارآمد را تضمین کند و موجب افزایش رقابت‌پذیری و بهبود تصمیم‌گیری در صنایع مختلف شود.

اینترنت صنعتی نیازمند پلتفرم‌هایی برای اشتراک‌گذاری داده‌های ایمن و قابل اعتماد است. بلاکچین با این نیازها همخوانی دارد و امکان اشتراک‌گذاری داده‌ها به صورت ایمن، شفاف و کارآمد را فراهم می‌کند و صنعت را به سمت "زنگیره اعتماد" سوق می‌دهد.

این پژوهش بررسی می‌کند که چگونه قراردادهای هوشمند به صنعت در زمینه‌های زیر قدرت می‌بخشند:

• ردیابی محصولات^{۱۴}

• تولید قابل اعتماد^{۱۵}

• مدیریت زنجیره تأمین^{۱۶}

• مالی زنجیره تأمین^{۱۷}

نتیجه‌گیری

نویسندها در این مطالعه بررسی جامعی از پژوهش‌های مرتبط با قراردادهای هوشمند انجام داده‌اند. بررسی‌های پیشین عمدتاً بر پلتفرم‌ها، کاربردها، زبان‌های برنامه‌نویسی و شناسایی آسیب‌پذیری‌ها تمرکز داشته‌اند.

¹⁴ Product Traceability

¹⁵ Trustworthy Manufacturing

¹⁶ Supply Chain Management

¹⁷ Supply Chain Finance

اما یک شکاف مهم پژوهشی در تحلیل جامع چرخه حیات و الگوهای اجرایی قراردادهای هوشمند مشاهده شده است.

مشکلات عملکردی قراردادهای هوشمند، مانع بزرگی برای پذیرش گسترده آن‌ها است و ضرورت بهبود کارایی اجرا را برجسته می‌سازد. این مطالعه بر چرخه حیات قراردادهای هوشمند متمرکز شده است تا چالش‌ها و راهکارهای موجود در فرآیند اجرا را خلاصه کند. پژوهش حاضر بر بهینه‌سازی عملکرد اجرای قراردادهای هوشمند، مدیریت تضادهای همزمانی در اجرا و نقش آن‌ها در توانمندسازی صنعت تأکید دارد. این پژوهش چشم‌اندازی از تحول صنعت به یک صنعت مبتنی بر اعتماد ارائه می‌دهد و بررسی می‌کند که چگونه فناوری بلاکچین می‌تواند به کارگیری قراردادهای هوشمند را تسهیل کند. شکاف‌های پژوهشی فعلی شناسایی شده‌اند و مسیرهای تحقیقاتی آینده پیشنهاد شده‌اند تا محققان، مهندسان، استادی و فعالان صنعتی را در بهره‌گیری از قراردادهای هوشمند برای پیشرفت فناوری راهنمایی کنند.