

# شبكة الكتل - The Blocknet

## مواصفات التصميم

أرلين كولويك ودان ميتكالف

مع مساهمات من أليكس كوش وفريق Blocknet الأساسي

جدول المحتويات	
3	نظام التحكم بالمراجعات
4	مقدمة
4	لماذا ننشر هذا المستند؟
4	نرحب بالمساهمين
5	الفرصة
5	نبذة عن "بلوكنيت" Blocknet
5	المعوقات
6	الحل
8	التصميم
8	أهداف التصميم
11	الهندسة
15	العناصر الأساسية
20	الخدمات الأساسية
28	عناصر سلسلة الكتل
29	خدمات سلسلة الكتل
44	مراحل المشروع
44	توفير المنتج الفعال الأساسي MVP
45	المرحلة الثانية
45	المرحلة الثالثة
45	المرحلة الرابعة
46	المواصفات التقنية
46	مراحل الرسائل
46	مرجع واجهة برمجة التطبيق API
47	حالات الاستخدام use-cases
المصطلحات	قائمة
52.....	قائمة
المختصرة	المصطلحات
57.....	قائمة

التعليقات	المساهم	رقم النسخة	التاريخ
المخططات الأولية (المفاهيمية)	أرلين كولويك	0.1	27-08-2016
المسوّدة الأولى	أرلين كولويك	0.2	18-05-2017
تعديل الهيكلية لإيضاح المفهوم	أرلين كولويك	0.3	26-10-2017
إزالة التوثيق المتدني المستوى وتحسين التركيز	أرلين كولويك	0.4	20-01-2018
ملاءمة مساحة التصميم مع توجيه سلسلة الكتل (blockchain) (routing)؛ إعادة برمجة مواد عقد الخدمة (service nodes) وتنظيمها	أرلين كولويك	0.5	07-03-2018
إنجاز مسوّدة مواد عقد الخدمة؛ إنجاز مخطط بروتوكول نظام الطلب وإضافة قسم على نظام مطابقة الطلبات (order matching system) وكتابة مخطط بروتوكول لسجل الطلبات	أرلين كولويك	0.6	09-03-2018
إضافة قسم "خدمة السجل"؛ القيام بعدة تعديلات بسيطة	أرلين كولويك	0.7	1-03-2018
تحرير النسخ وتصحيح الأخطاء المطبعية	هاني أبو	0.8	11-03-2018
إضافة عدة حالات استخدام للبنية التحتية لخدمة ما بين السلاسل inter-chain infrastructure	أليكس كوش	0.9	12-03-2018
الموافقة النهائية على النسخة العامة الأولى	أرلين كولويك	1.0	15-03-2018

هذا المستند موجّه إلى جمهور من غير المختصين ومن مطوّري البرمجيات في آن معاً، على افتراض أنّ القارئ العادي يفهم، ولو بصورة عامة، المبادئ الأساسية لسلسلة الكتل (Blockchain - بلوكشين). وسوف يتمّ استعراض تصميم شبكة الكتل (بلوكنيت - Blocknet) وهندستها بالإجمال بشكل غير تقني، وذلك عبر استخدام الرسوم البيانية ومن خلال شرح تدريجي لماهية خدمات "ما بين السلاسل" (inter-chain services)، وبالتالي العناصر التي ينبغي على أيّ بنية تحتية لخدمة ما بين السلاسل (inter-chain infrastructure) توفيرها كحد أدنى. ومع تتابع أقسام هذا المستند، سوف يزداد الحديث عن مواصفات بلوكنيت بشكل خاص، بدلاً من مناقشة الخصائص المفترضة لخدمات ما بين السلاسل عموماً. ومن خلال التوفيق بين جوانب التصميم ومسائل التنفيذ والدمج، نأمل في استعراض مواصفات تصميم بلوكنيت بشكل منهجي.

### لماذا ننشر هذا المستند؟

نحن نؤمن بأنّ ما من طريقة أفضل لتصميم الحلول من التواصل مع أطراف متنوعة واكتساب فهم معمق حول عملنا من جهات نظر مختلفة. كذلك، نحن نعتقد أنّ نشر عملنا على نطاق عام من شأنه أن يعود علينا بفوائد جمة بفضل تفاعل الناس معه ومع برمجياتنا. ثالثاً، من بالغ الأهمية في مشروع بنية تحتية كهذا أن يتمّ التواصل مع الأطراف التي ستستخدم التكنولوجيا التي نطوّرها وتبني عليها.

قد يبدي البعض خشيتهم من احتمال أن تنسخ جهات منافسة عملنا وتتفوق علينا، خاصة أنّ التصميم الذي أعدناه هو الأول من نوعه في السوق ولم يكتسب زخماً كبيراً بعد. إنّنا ندرك أنّ هذا الاحتمال وارد، ولكننا نعتبر أنّ الفائدة التراكمية التي قد يقدمها المهتمّون بمجالي التشفير وهيكلية الشركات إلينا هي مورد لا غنى عنه، وبالتالي نحن نرى أنّ ميزان المخاطر-المردود يبقى لصالحنا.

### نرحّب بالمساهمين

بلوكنيت ليست شركة ولا فريقاً حصرياً، بل بنية تحتية. ونحن نؤمن بأنّ ملكية البنى التحتية يجب أن تكون عامة ومُتاحة للجميع بلا أي قيود.

هذا المستند بصيغته الحالية هو مسودّة وليس تصميمًا نهائيًا. وفي الواقع، لا مكان لمفهوم "التصميم النهائي" ضمن مشروع يتطوّر باستمرار. لذا، نحن نرحّب بجميع من يريد المساهمة في هذا المستند ومناقشته.

وتجدر الإشارة إلى أنّ [برمجية بلوكنيت مفتوحة المصدر](#)، ويمكن لكل راغب أن يساهم في مشروع بلوكنيت.

إنّ الفائدة الأكبر لمشاريع من هذا النوع تتأتى من تنوّع وجهات النظر والمهارات المساهمة في التصميم والتنفيذ. لذلك، نحن نرحّب بالمساهمات مهما كان نوعها.

للتواصل مع بلوكنيت، يُرجى إرسال رسالة على البريد الإلكتروني الآتي: [contact@blocknet.co](mailto:contact@blocknet.co).

## الفرصة

### نبذة عن بلوكنيت

بلوكنيت هي بنية تحتية مخصصة لحقبة "ما بين سلاسل الكتل" المقبلة، وهي حقبة تكنولوجية ناشئة، تتميز بشكل أساسي باستبدال نظام "واجهة برمجة التطبيق" (API) الحالي بـ "نظام العملات" (token ecosystem) لامركزي ويولد الإيرادات بنفسه. وسوف يتم بلوغ هذه المرحلة عندما تتطور التكنولوجيات الممكنة (أي العقود الذكية والتطبيقات اللامركزية "dapps") إلى حد تصبح فيه قادرة على تحقيق التوافق التشغيلي العملي ما بين سلاسل الكتل (inter-blockchain interoperability). وفي تاريخ إعداد هذا المستند، كانت بلوكنيت هي الرائدة تكنولوجياً في مجال توفير بنية تحتية ما بين السلاسل، يمكن استخدامها في التطبيقات اللامركزية والعقود الذكية.

نحن نعتبر أنّ نشوء حقبة ما بين سلاسل الكتل سوف يكون له تداعيات مَحَلَّة بِقِطَاعَيْنِ اثْنَيْنِ، هما "البرمجيات كخدمة" (software-as-a-service) والقدرة على استخدام سلسلة الكتل عملياً (practical blockchain usability).

من ناحية أولى، يضمّ نظام العملات عنصرين أكثر تقدماً من نظام "البرمجيات كخدمة": (أ) توليد الإيرادات من الخدمات الرقمية بشكل خالٍ من المعوقات نسبياً؛ (ب) والاستفادة من خصائص المتانة واللامركزية والأمن التي توفرها تكنولوجيا سلسلة الكتل.

من ناحية ثانية، في ما يتعلّق بتكنولوجيا سلسلة الكتل، لا بد من تحقيق التوافق التشغيلي الواسع النطاق والشامل بين خدمات سلاسل الكتل لكي تتم الاستفادة من الإمكانيات الكاملة التي توفرها هذه التكنولوجيا. ففي غياب التوافق التشغيلي ما بين السلاسل، تواجه الخدمات القائمة على سلاسل الكتل احتمالين: (أ) فإمّا أن تقدّم خدماتها ضمن نطاق قاعدة عملائها المحدودة التي تشغّل عقدها (nodes) فحسب، أو تضخّي بالخصائص الأمنية المميزة التي تتمتع بها سلاسل الكتل عبر توسيع الخدمات لتشمل كيانات مركزية؛ (ب) وتواجه مشاكل دائمة متعلقة بتشبع السلسلة (chain bloat) وما يرتبط بذلك من ضغط في السوق لتوفير ميزات جديدة ضمن السلسلة الواحدة.

فمن خلال إنشاء "إنترنت سلاسل الكتل" (internet of blockchains)، يمكن لبلوكنيت السماح بتوليد الإيرادات بشكل سلس من نظام واجهة برمجة التطبيق، وبالتالي تعزيز تكنولوجيا سلسلة الكتل من خلال تحويل آلاف السلاسل المعزولة إلى نظام العملات.

### المعوقات

تعاني الخدمات القائمة على شبكة الإنترنت من مشكلة دائمة تتمثل في انعدام الأمن في أنظمتها التكنولوجية الأساسية (technology stack). كذلك، تتطلب هذه الخدمات عادةً أن تكون وظائفها وخدماتها مركزية، ما يجبر العملاء على تحمّل عبء الثقة. في المقابل تسمح تكنولوجيا سلسلة الكتل بالاستفادة من الدلائل التشفيرية (cryptographic proof) لتقديم خدمات "خالية من عامل الثقة"، حيث يمكن لأي جهة مشاركة أن تثبت بنفسها من صحة النتيجة، ما يحدّ من درجة الثقة المطلوبة للتعامل مع أطراف أخرى بشكل كبير. ويوسع ذلك من نطاق التعاملات الممكنة إلى حدّ بعيد، ويسمح ببناء الكثير من نماذج الأعمال الجديدة، وقد يوفر ضمانات أمنية واضحة المعالم، ما يخفض التكاليف ويحمي قيمة العلامة التجارية.

ولكن لا يمكن لتكنولوجيا سلسلة الكتل أن تبلغ آفاقها الفعلية بشكل فوري، وذلك لسبب أساسي وهو غياب التوافق التشغيلي في ما بينها. فنتوفر حالياً آلاف سلاسل الكتل، ولكنها تعمل بطريقة تشبه الشبكات المحلية (LANs) المفصولة عن شبكة الإنترنت، كما أنها لم توفر بعد الظروف الملائمة لتحقيق الاختراق الذي سيحدد معالم المرحلة، والذي يتمثل بالتوافق التشغيلي الشامل – على نطاق يشبه كيفية توفير شبكة الإنترنت للبيئة الملائمة لنشوء محرّك البحث "غوغل" وموقع التواصل الاجتماعي "فيسبوك".

بلوكنيت هي بنية تحتية تأسيسية لنظام العملات. فهي توفر توافقاً تشغيلياً فعلياً "من الند إلى الند" (peer-to-peer) بين عقد ضمن سلاسل كتل مختلفة، وذلك من أجل السماح بما يلي:

- تقديم أي نوع من الخدمات الرقمية عملياً من عقدة ضمن أي سلسلة كتل إلى أخرى.
- تمكين أي خدمة قائمة على تكنولوجيا سلسلة الكتل من العمل لا كـ"عملة للتطبيقات" (appcoin) بل كـ"خدمة بروتوكول"، أي جعلها صالحة للاستخدام في أي تطبيق لامركزي آخر لأي هدف كان، بدلاً من أن يكون استخدامها محدوداً بالتطبيق اللامركزي الخاص بمنتجها، ما يزيد بشكل كبير من الأسواق التي يمكن لهذه الخدمة ولوجها ومن تدفق إيراداتها.
- السماح لعملات (tokens) العقود الذكية ليس فقط بتوليد الإيرادات من التطبيقات اللامركزية، بل بأن تصبح "رموز بروتوكول"، ما يجعل من المنطقي وضعها في طبقة أدنى ضمن الأنظمة التكنولوجية الأساسية، حيث تتضاعف فائدتها. فضلاً عن ذلك، قد تتحسن جودة برمجيات الخدمة المعنية بفضل توفر مجموعة واسعة من المطورين المساهمين من مشارب متنوعة، إذ يمكن لهذه الخدمة الاستفادة من الخبرات المجتمعة التي يقدمها هؤلاء، عبر تجنب تشييع السلسلة وتكرار البرمجيات والحد من ساعات العمل المطلوبة وتقديم خدمات إلى كامل سوق سلاسل الكتل، بدلاً من أن يكون نطاقها محدوداً بمستخدمي سلسلة الكتل الخاصة بالخدمة نفسها فحسب.
- السماح للتطبيقات اللامركزية بأن تكون وسائط بسيطة لتنظيم خدمات ما بين السلاسل، بدلاً من أن يضطر المطورون إلى كتابة برمجياتها الصعبة من الألف إلى الياء. فنقتصر عندئذٍ مهام التطوير الأساسية على دمج واجهة برمجة التطبيقات، لا كتابة برمجيات جديدة وآمنة ومتقنة، علماً أنّ المهمة الأخيرة بالغة الصعوبة وتتطلب مهارات متخصصة جداً.
- بناء تطبيقات لامركزية بهيكلية قائمة على الخدمات الصغرى (microservices)، بحيث تقدم كل سلسلة كتل خدمة واحدة تكون مدمجة مع عدة خدمات أخرى، على شكل وحدات ضمن نظام عام، ما يسهل عملية تصميم العناصر وتصحيح الأخطاء البرمجية (bugs) وتحديث البرنامج.
- القدرة على التخطي الفعّال لمسألة اختيار أي من سلاسل الكتل المتاحة يجب تطويرها (وهي مهمة صعبة حالياً) – ليس فقط عند بداية المشروع، بل خلال مراحل أخرى من دورة حياته، عندما يُحتمل أن تصبح الخدمات الصغرى متوفرة بشكل أكثر فعالية في سلسلة كتل أخرى.
- توليد الإيرادات من خدمات ما بين السلاسل (inter-chain) وخدمات السلاسل المتعددة (multi-chain) عبر استخدام العملات القيمة الخاصة بها.
- الاستفادة الكاملة من نماذج الأعمال الجديدة القائمة على اقتصاد التشفير (cryptoeconomy) والتي سمحت بتكنولوجيا سلسلة الكتل بنشأتها. فعلى سبيل المثال، قد تزداد قيمة الأعمال من خلال نموذج "أكثر من مجاني"، ومن السياسة النقدية مباشرة (عرض العملة الأولي ICO، ورسوم التعاملات، واقتصديات الانكماش، ومكافآت الكتل block rewards، وأنظمة الكتل الكبرى الذاتية التمويل superbloc self-funding systems)، كما وأن توفر سوق لواجهات برمجة التطبيقات الخاصة بها والمولدة للإيرادات.

وسوف تحقق بلوكنيت الأهداف المذكورة أعلاه عبر مقارنة هيكلية مبنية على البروتوكول. ويشتمل هذا المستند على توثيق هذه المقاربة.

## التصميم

## أهداف التصميم

سوف يتم العمل على تصميم الميزات الآتية ذات الأولوية التنازلية:

## 1. التوافق التشغيلي (Interoperability)

بلوكنيت هي بشكل أساسي بنية تحتية ما بين سلاسل الكتل. لذا، فإن الهدف المباشر الأبرز لتصميمها هو تحقيق التوافق التشغيلي بين الغالبية الساحقة من تطبيقات سلاسل الكتل الحالية والمستقبلية. فضلاً عن ذلك، سوف تحقق بلوكنيت التوافق التشغيلي مع الكيانات المركزية لكي تكون الخدمات التقليدية القائمة على الخوادم متاحة ضمن نظام العملات.

## 2. اللامركزية

اللامركزية تعني، في جوهرها، عدم سيطرة كيان واحد على باقي الكيانات ضمن نظام ما. فعلى سبيل المثال، لعلّ الإنجاز الأبرز لعملة "بيتكوين" (Bitcoin) المشفرة، بصورة عامة، هو تحقيق اللامركزية المالية، بحيث لا يتحكم كيان واحد منفرد (أ) بقيمة العملة (ب) وبتحويل الأموال (ج) وبحفظ سجلات الحسابات (د) وبالسياسة النقدية.

ولكنّ عملة "بيتكوين" موجودة حالياً ضمن نظام مركزي إلى حدّ بعيد، ما يبطل الكثير من فوائدها عملياً. فما من قيمة كبيرة لتقديم الخدمات اللامركزية ضمن نظام مركزي لسببين: (أ) ذلك متوفر أساساً على شكل نظام واجهة برمجة التطبيقات؛ (ب) وذلك يؤدي إلى إبطال الميزة الأبرز، أي اللامركزية، خلال عملية تقديم الخدمات. فإذا اشترى أحد ما عملة "بيتكوين" عبر نظام تبادل مركزي، لا تعود عملية الشراء هذه "خالية من عامل الثقة"، لأنه على المشتري أن يثق بنظام التبادل الذي يستخدمه، فضلاً عن أنّ عملية الشراء تخضع لجميع عناصر البنية التحتية التقليدية للدفع (الرسوم والتأخيرات المصرفية، ورسوم بوابات الدفع (payment gateways)، ورسوم شركتي "فيزا" و"ماستركارد"، وخطر الاحتيال، ومتطلبات "اعرف عميلك" (KYC)، وضرورة أن يأتى المتعامل عدة وسطاء على ماله ومعلوماته الشخصية، وإلى ما هنالك). لذلك، فلكي تتمكن "بيتكوين" أو غيرها من التكنولوجيات اللامركزية من تحقيق مداها الفعلي، من الضروري إنشاء نظام لامركزي حيث يمكن للكيانات التعامل مع بعضها البعض من دون المساس بالميزات التكنولوجية.

## 3. الأمن

تتطلب الخدمات اللامركزية والنقدية درجة عالية من الأمن والدقة التشغيلية (determinacy of operation)، شبيهة ببرمجيات الطيران، لأنه (أ) من غير الممكن بشكل عام تغيير خدمة تعمل على أطراف الشبكة، أي على أجهزة مستخدميها، أو فصلها عن الشبكة، (ب) وإذا تبيّن أنه يمكن سرقة المال من نظام لا يخضع لعمليات تصحيح مركزية، فسيفقد هذا النظام معظم قيمته بسرعة كبيرة. لهذين السببين، تتطلب بلوكنيت أعلى درجة ممكنة من الأمن والدقة التشغيلية.

## 4. تقديم الخدمات بمعزل عن "عامل الثقة"

من الفوائد المرغوب فيها والأكثر شيوعاً للامركزية في سياق سلاسل الكتل أنه من غير الضروري أن يثق المرء بالطرف الذي يتعامل معه لكي يتصرف بصدق خلال عملية التبادل. فعلى سبيل المثال، في حالة "بيتكوين"، ليس من الضروري أن يأتى المرء وسيطاً لتحويل الأموال، أو أن يثق بالمستلم لكي يبلغه بصدق إذا كان قد استلم الدفعة وكم تبلغ هذه الأخيرة، ذلك أنّ التعاملات بالـ"بيتكوين" لا تتطلب وسيطاً أساساً، ويمكن لطرفي التبادل أن يتحققا بشكل مستقل من وضع الدفعة بدرجة عالية من الموثوقية.

## BLOCKNET

وفي حالة تقديم الخدمات ما بين السلاسل، يجب تحقيق مستوى مماثل من عدم الحاجة إلى "عامل الثقة" عند تسديد الدفعات لقاء الخدمات ما بين سلاسل الكتل، وذلك لكي يتم تقديم الخدمة وتسديد الدفعة اللازمة لمقدمها من دون الحاجة إلى ضمان تصرف طرفي عملية التبادل بصدق، مما يتيح المحافظة على هذه الميزة الفريدة الخاصة بأنظمة الدفع القائمة على سلاسل الكتل في سياق ما بين السلاسل.

### 5. الدمج البسيط (لا يحتاج إلى برمجة)

لتحقيق أقصى حد ممكن من التوافق التشغيلي ولتجنب المعوقات، فإنّ الدمج مع بلوكنيت وإمكانية الدخول إلى نظام العملات لن يتطلبا تعديل محفظات الأصول (stock wallets) أو العقد. وتجدر الإشارة إلى أنّ الاستفادة من خدمة طرف ثالث مقدّمة عبر بلوكنيت قد تتطلب بعض البرمجة، ولكن استخدام بلوكنيت نفسها لن يتطلب أي برمجة.

### 6. الدمج اللامركزي

لتحقيق أقصى حدّ ممكن من الأمن ولإنشاء نظام مفتوح شبيه بشبكة الإنترنت، لن يتطلّب الدمج مع بلوكنيت ودخول نظام العملات وساطة أي كيان مركزي (بما في ذلك كياننا نحن). كذلك، لتقديم أو استهلاك الخدمات عبر بلوكنيت، لن يضطر المستهلكون إلى (أ) استخدام سلسلة الكتل الخاصة ببلوكنيت، (ب) أو استخدام أي خدمة محددة، (ج) أو استخدام أي خدمة تتضمن نشاطاً مركزياً. (المقصود بتعبير "مركزي" في هذا السياق هو مجموعة من الاحتمالات التي تتراوح بين رقابة جهة مركزية والمركزية القائمة على سلاسل جانبية تدور في فلك شبكة مركزية. ونشير إلى الحالة الأخيرة بعبارة "مركزية ما بين السلاسل").

وتجدر الإشارة إلى أنّ الاستفادة من خدمة طرف ثالث مقدّمة عبر بلوكنيت قد تتطلب وساطة جهة مركزية، ولكن استخدام بلوكنيت نفسها لن يتطلب ذلك.

### 7. الميزة التركيبية (composability)

سوف يتم بناء بلوكنيت مع أخذ الميزة التركيبية ونظام الوحدات في الاعتبار قدر الإمكان، وفق النمط نفسه أعلاه الذي تم تصوّر الخدمات الصغرى ما بين السلاسل من خلاله. وبشكل خاص، فإنّ المبادئ الأساسية لتصميم الخدمات الصغرى تتمثل في تعزيز الميزة التركيبية، مع الأخذ في الاعتبار أي خدمات سيتم استهلاكها معاً بشكل دائم، وذلك لتجنّب بناء "وحدة مترابطة موزعة" (distributed monolith). ولا يطرأ أي تغيير على هذه الخدمات في نظام العملات.

### 8. القدرة على توليد المال (monetisability)

يُضاف مبدأً أساسياً آخر على الميزة التركيبية في نظام العملات، وهو إمكانية توليد المال من الخدمات بشكل تلقائي. وإذا كان ذلك غير ممكن، فنحن نقترح دمج هذه الخدمة ضمن حزمة واحدة مع واجهة برمجة التطبيقات الخاصة بخدمة مولدة للمال، وإلا لن تتوفر الحوافز للأشخاص الذين يشغلون عقد الخدمة لأنهم لم يتمكنوا من الحصول على مصدر إيرادات منها.

بالإضافة إلى ذلك، من الضروري أن يكون مصدر إيرادات الخدمات آمناً بشكل مضمون من خلال بروتوكول خالٍ من "عامل الثقة" أو حوافز تشفيرية، وإلا من غير المرجح أن يتم حصد القيمة الممكنة. فالقدرة على توليد الإيرادات من خدمة ما مرتبطة برغبة المستهلك على الدفع للحصول على الخدمة التي تقدّمونها بقدر ما هي مرتبطة بما إذا كان المستهلك قادراً على الاستفادة من هذه الخدمة مجاناً عنوةً.

في ضوء ما سبق، لا بد من الإشارة إلى أنّ بلوكنيت سوف تولد الإيرادات من خدماتها الأساسية حيث تكون لذلك جدوى، كما أنها سوف تقدّم بعض الخدمات مجاناً. كذلك، سوف توفر بلوكنيت وسائل متعددة يمكن من خلالها توليد الإيرادات من الخدمات المقدّمة عبرها بشكل آمن.

### 9. سهولة التنقل والحجم الصغير

إنّ احتمالات الاستفادة من نظام العملات متعددة، وهي تشمل تطبيقات الهواتف المحمولة والأجهزة التي تضمّ إنترنت الأشياء (IoT) في مجال التأمين والصحة وسلسلة التوريد والزراعة وأنظمة المعلوماتية البعيدة (telematics) في



## BLOCKNET

السيارات والقطاعات الأمنية<sup>1</sup> فالكثير من حالات الاستخدام التي نتوقعها ستتطلب تطبيقات لامركزية محدودة الحجم، وبالتالي غير قادرة على استضافة سلسلة كتل واحدة حتى. لذلك، سوف تمكن بلوكنيت هذه الأجهزة من الدخول إلى نظام العملات لكي تكتسب ميزات أمنية خاصة بسلاسل الكتل، نعتقد أنها ضرورية جدًا للحد من مكامن الضعف (attack surface) للخدمات القائمة على إنترنت الأشياء. وبشكل خاص، سوف تمكن بلوكنيت التطبيقات الصغيرة الحجم من استهلاك (وتسديد ثمن) خدمات ما بين السلاسل من دون الحاجة إلى استضافة سلسلة كتل محليًا.

---

<sup>1</sup> لمزيد من التفاصيل حول هذا الموضوع، راجع النظرة العامة حول أعمال بلوكنيت لشباط/فبراير 2015، المتوفرة عند الطلب من خلال التواصل معنا عبر [contact@blocknet.co](mailto:contact@blocknet.co).

يمكن تحقيق الغاية الأساسية من التوافق التشغيلي ما بين سلاسل الكتل عبر دمج ثلاثة عناصر أساسية، تعمل معًا لتقديم ثلاث خدمات أساسية، إلى جانب أي عدد كان من خدمات سلاسل الكتل وعناصر سلاسل الكتل. وتهدف هذه المكونات إلى بناء عدد لا يُحصى من خدمات ما بين سلاسل الكتل – أي نظام عملات – يمكن تنظيمها جميعًا ضمن تطبيقات ما بين السلاسل.

ولمساعدة القارئ على فهم هذه المفاهيم الجديدة، سوف يتم استعراضها على شكل رسوم بيانية تشرح العلاقة بين العناصر والخدمات. وسوف تتابع هذه الرسوم البيانية كما يلي:

[الرسم في الصفحة 9 من المستند الإنكليزي]:

Inter-chain network overlay	تراكب الشبكة ما بين السلاسل
Blockchain router	موجّه سلسلة الكتل
P2P data transport	نقل البيانات من الند إلى الند
Inter-chain messaging	التراسل ما بين السلاسل
Decentralized exchange	التبادل اللامركزي
Service lookup	البحث عن الخدمات
Light nodes	العقد الخفيفة
Full nodes	العقد الكاملة
Service nodes	عقد الخدمة

سوف يتم شرح العناصر أولاً ثم الخدمات. ولكن قبل ذلك، سوف يتم استعراض الطبيعة العامة لهندسة ما بين السلاسل.

ما هو شكل هندسة ما بين السلاسل؟

بشكل عام، تشمل هندسة ما بين السلاسل دائمةً شبكتي سلاسل كتل على الأقل، فضلاً عن كيان أو وظيفة أخرى لتحقيق التوافق التشغيلي بينهما. وبما أنّ شبكات سلاسل الكتل هي لامركزية وموزّعة، يجب ألا تكون عناصر التوافق التشغيلي في موقع مركزي فالحفاظ على ميزة اللامركزية، يجب أن تعمل هذه العناصر عبر عقد على أطراف كل شبكة أو أن تتفاعل محلياً مع واجهة هذه العقد.

[الرسم في الصفحة 10 من المستند الإنكليزي]

الرسم رقم 1: شبكتان مثاليتان من الند إلى الند، مع تقديم خدمة لاوساطية (disintermediated) من عقدة إلى أخرى.

وقد اقترحت عدة مشاريع حلولاً كالاتية:

- التكنولوجيا التقليدية: وسيط مركزي (مثلاً: Poloniex.com)؛
- القصوية (maximalists): شبكة لامركزية تعمل كوسيط مركزي (مثلاً: "بينكوين" في سياق السلاسل الجانبية)؛

## BLOCKNET

- البرمجيات الاحتكارية (أي المحفظات والعقود الذكية والإضافات "bolt-ons" على المحفظات) التي تحقق التوافق التشغيلي بين سلاسل الكتل ولكن فقط بين العقد التي تستخدم هذه البرمجيات (أي BTCrelay)؛
- "الحديقة المسورة" (walled garden): بروتوكولات ما بين السلاسل فقط بين بعض حالات سلاسل الكتل الخاصة، ما يجبر المطورين على البناء عليها (مثلاً: Aion).

وتجدر الإشارة إلى أنّ أيًا من أنماط تكنولوجيا سلسلة الكتل أعلاه ليس شاملاً ولا مركزياً في آن معاً. فهي إما لا تسمح بتقديم مجموعة لامتناهية من الخدمات (بما في ذلك الخدمات المقدمة على سلاسل الكتل الموجودة أساساً)، أو تعجز عن تقديم هذه الخدمات من دون فرض سيطرة مركزية تقلص بشكل أو بآخر الميزة اللامركزية لهذه الخدمات.

أما وفقاً لأهداف التصميم الخاصة ببلوكتينيت، فالحل الأمثل يجب أن يكون شاملاً ولا مركزياً في الوقت نفسه. وفي هذا السياق، نحن نلتزم بقاعدة "المبادئ أولاً"، أي أننا نبقى أوفياء لطبيعة نظام ما بين السلاسل نفسه.

### 1. هندسة الشبكة الموزعة

أولاً، لا شك أنّ من المهم جداً أن تكون أي عناصر ما بين السلاسل موجودة على أطراف الشبكات التي تتوافق تشغيلياً معها. فمن شأن ذلك أن يوزع الخدمة عبر كل سلسلة كتل تقدم أو تستهلك الخدمات. ولكن بالإضافة إلى ذلك، يجب أن تقدم عناصر ما بين السلاسل خدمات من أطراف شبكتها هي أيضاً، من دون الحاجة إلى نظام مركزي، وإلا سنضطر إلى العمل كأبي وسيط مركزي.

[الرسم في الصفحة 11 من المستند الإنكليزي]:

الرسم رقم 2: يجب أن تكون خدمات ما بين السلاسل الملائمة موجودة على أطراف شبكتها هي والشبكة(ات) التي تستخدمها.

### 2. الجهات الفاعلة اللامركزية

ثانياً، وهو أمر مرتبط بما سبق، يجب أن يكون تقديم أو استهلاك خدمات ما بين السلاسل فعلاً مستقلاً، أي غير خاضع لتحكم طرف ثالث. من الناحية الهندسية (أي بمعزل عن تصميم البروتوكول)، تتمثل الطريقة المباشرة الأكثر أماناً لتحقيق هذه الاستقلالية في أن تكون عقد عنصر خدمة ما بين السلاسل والشبكة التي تستهلكها أو تقدمها، أو الاثنان معاً، موجودة على الجهاز المحلي نفسه. ويتراوح مدى ضرورة ذلك – ومدى تأثير ذلك على حجم خدمة ما بين السلاسل المعنية – بين الحاجة إلى توفير عقد كاملة أو عقد التحقق المبسط من الدفعات (SPV nodes) أو المصادقة على التعاملات، وصولاً إلى الاستعلام عن موقع إلكتروني يستكشف سلاسل الكتل (blockchain explorer website) أو أي وسيط مركزي (centralized oracle) آخر في التطبيقات ذات المستوى الأمني المنخفض، بالحد الأدنى. وتعتبر الحالة الأخيرة الحد الأقصى للأنظمة التي يشملها مصطلح "ما بين السلاسل".

بالتالي، يصبح النطاق الكامل لمتطلبات الهندسة المحلية واضحاً. وفي جميع الحالات الأخرى غير الحالة القصوى المذكورة أعلاه، من الضروري توفير شكل من أشكال المشاركة المباشرة في كل من شبكة تقديم خدمات وشبكة استهلاك خدمات لدى كلّ جهة فاعلة لكي تكون مشاركتها لامركزية بالفعل. ويمكن تصوير ذلك بيانياً عبر تكرار الرسم السابق:

[الرسم في الصفحة 12 من المستند الإنكليزي]:

الرسم رقم 3: بنية تحتية لامركزية ما بين السلاسل – المساحات الزرقاء تشير إلى مزيج مختلف من استهلاك الخدمات وتقديمها وعقد ما بين السلاسل. (العقد خارج المساحات الزرقاء لا تتلاعب مع خدمات ما بين السلاسل).

### 3. لا محدودية في سلاسل الكتل المُتاحة

رغم أنه ينبغي تقديم خدمات ما بين السلاسل من عقد على إحدى السلاسل، يجب ألا تحدّد البنية التحتية ما بين السلاسل على أيّ سلسلة كتل يمكن توفير خدمة ما، وإلا جُلّ ما سيتم تحقيقه هو نموذج عميل-خادم موزّع، وهو النموذج الذي يشكل في الواقع الهندسة المعيارية للتطبيقات المركزية اليوم. على سبيل المثال، يتطلّب نموذج [Blockstream](#) [للسلاسل الجانبية](#) تفاعل كلّ مستخدم مع سلسلة كتل "بينكويين" لكي يتمكّن من الاستفادة من أيّ خدمة أخرى في سلسلة مختلفة. نحن نشير إلى هذا المأزق بعبارة "مركزية ما بين السلاسل". ولتجنّب الوقوع فيه، يجب أن نتّيح "إنترنت سلاسل الكتل"، إن كانت بالفعل كذلك وقادرة على دعم نظام عملات، إمكانية تقديم الخدمات واستخدامها من أي سلسلة كتل.

ويسمح مبدأ "الحياد تجاه السلاسل" (chain-agnostic) هذا بالحد من متطلبات الدمج ومن حجم التطبيقات. فعلى سبيل المثال، إذا كانت بلوكنيت تتطلّب من كل مستخدم لإحدى خدمات ما بين السلاسل أن يحتفظ بنسخة عن سلسلة الكتل الخاصة ببلوكنيت، بالإضافة ربما إلى سلسلة الكتل الخاصة بمقدّم الخدمة، فستكون فائدتها محدودة نسبياً، كما ستكون المعوقات التي يواجهها المستخدمون كثيرة جداً.

وسيوثّر هذا الجانب من تصميم البنية التحتية ما بين الكتل بشكل أساسي في تقديم الخدمات المولّدة للإيرادات، إذ إنّ المستخدمين في شبكة من الند إلى الند هم غير موثوقين، ومن الضروري أن تكون عملية الدفع وتقديم الخدمة تلقائية؛ وثانياً، يجب أن تتلقّى إحدى العقد دفعة بالعملة (token) الخاصة بها من قبل عقدة تستخدم عملة أخرى، وبالتالي يجب أن يتم تبديل هاتين العملتين، والتبديل اللامركزي يتطلّب درجة عالية من الأمان وبرمجيات عالية الجودة. في المقابل، إذا كان من الضروري تحميل سلسلتي كتل أو حتى ثلاثة وحفظها للاستفادة من الخدمة، من غير المرجح أن يتم اعتماد هذا النظام على نطاق واسع. لهذا السبب، توفر بلوكنيت وسيلة لتجنّب هذا العائق.

### ملخص

تؤدّي الاعتبارات المذكورة أعلاه إلى استخلاص ثلاثة مبادئ توجيهية لتصميم عناصر بلوكنيت:

1. يجب أن تكون خدمات البنى التحتية ما بين السلاسل متوفرة على أطراف كلّ من شبكتها/شبكاتها هي وأي شبكات مخصصة لتقديم واستهلاك الخدمات.
2. من الناحية الهندسية، يمكن تحقيق لامركزية الخدمات بالطريقة الأسهل من خلال تشغيل العناصر اللازمة إما لتقديم أو لاستهلاك خدمة ما على الجهاز المحلي نفسه.
3. يجب أن تحدّد خدمات البنى التحتية ما بين السلاسل من ضرورات الدمج ومن الحجم حيث أمكن.

العناصر الأساسية

تتضمّن بلوكنيت ثلاثة عناصر أساسية، تعمل معًا لتوفر أساس هذه البنية التحتية ذات الهدف العام لخدمات ما بين السلاسل:

- XBridge، تراكب الشبكة ما بين السلاسل؛
- XName، موجهة سلاسل كتل؛
- XChat، تحويل البيانات من الند إلى الند.

وتُصنّف هذه العناصر الثلاثة على أنها "أساسية" لأنه من الطبيعي أن يتطلّب أي حلّ يسعى إلى تحقيق التوافق التشغيلي ما بين السلاسل شكلاً من أشكال التواصل بين عقد موجودة على شبكات فرعية مختلفة، وطريقة لكي تستكشف العقد أين يجب توجيه طلبات الخدمة، وبروتوكول للتواصل من الند إلى الند بعد العثور على عقدة مناسبة.

ولتسهيل عملية تذكر وتصوّر نظام العناصر والخدمات ضمن بلوكنيت للقارئ، سيتم تشكيل رسم بياني تدريجيًا كلما تمّ تقديم أحد العناصر. أمّا الرسم أدناه فيظهر العناصر الأساسية الثلاثة في بلوكنيت فحسب.

[الرسم في الصفحة 14 من المستند الإنكليزي]:

الرسم رقم 4: أولاً: العناصر الأساسية الثلاثة في بلوكنيت – تمامًا كما أنه من الضروري وجود ثلاث جهات متصلة في أطرافها لكي تشكّل مثلثًا، يجب أن تكون هذه العناصر الثلاثة موجودة للتمكن من تقديم خدمات ما بين السلاسل.

Inter-chain network overlay	تراكب الشبكة ما بين السلاسل
Blockchain router	موجهة سلاسل كتل
P2P data transfer	تحويل البيانات من الند إلى الند

**XBridge: تراكب الشبكة ما بين السلاسل**

تضمّن بلوكنيت عنصر XBridge، وهو كناية عن شبكة من الند إلى الند قائمة على جدول دالة تجزئة موزع (DHT). ويتم دمج العقد على هذه الشبكة مع عقد على شبكات أخرى في جهاز محلي، ما يجعل من شبكتنا نمطًا من تراكب الشبكة ما بين السلاسل. ويتيح ذلك البحث والتحديد والتواصل بين العقد على أي شبكة سلسلة كتل.

رسم بياني يظهر سياق XBridge

[الرسم في الصفحة 15 من المستند الإنكليزي]:

الرسم رقم 5: عناصر الشبكة على جهاز محلي

API	واجهة برمجة التطبيق
Node a	العقدة أ
Node b	العقدة ب

للاطلاع على الوثائق الفنية، يُرجى مراجعة قسم المواصفات الفنية.

### التنفيذ

- حاليًا: يتم تنفيذ برمجة تراكب الشبكة في كل من XBridgep2p.exe ومحفظة بلوكنيت. ولا يتم توليد الإيرادات من ذلك.
- في المستقبل: قد تتم تجزئة ذلك إلى وحدات في قاعدة البرمجيات، ولكن من غير المرجح أن يتم إطلاقه كتطبيق منفصل، لأنه سيتوافق تشغيلًا مع عناصر أخرى لتوفير الخدمات الأساسية.

**XName: موجّه سلسلة الكتل**

يتطلب نظام خدمات ما بين السلاسل وسيلة لتوجيه الرسائل إلى سلسلة الكتل الصحيحة، وهو ما سيتم تحقيقه في بلوكنيت من خلال نظام عناوين ما بين السلاسل. لا يزال التوجيه في سلاسل الكتل في مراحله الأولى، وهو يحتاج إلى أن يستكشفه ويتقبله المهتمون في مجال التشفير. ولكن بشكل أساسي، يستلزم التوجيه في سلاسل الكتل معيارًا ما بين السلاسل لتحديد سلاسل الكتل، مثل نظام **MIND الخاص بـUport**، بالإضافة إلى طريقة لحفظ بيانات التوجيه في سجل ووظيفة بحث. ومن المسائل التي ما زالت قيد البحث في هذا الصدد نسبة السعر-مقابل-الموثوقية ( price-to-truthfulness) لنتائج التوجيه. فعلى سبيل المثال، قد تستفيد الخدمات بالشكل الأوسع من السجلات المجانية وتتقبل حدًا معينًا من نتائج البحث السيئة عبر إزالة احتمال عدم الموثوقية بعد البحث، وذلك قبل الدفع وتقديم الخدمة مباشرة. في المقابل، قد تطلب خدمات أخرى نتائج بحث موثوقة بشكل مثبت وتكون مستعدة لتحمل تكاليف طفيفة لقاء ذلك. في ضوء ذلك، تجدر الإشارة إلى أن XName سوف يتبع مقاربة محايدة في ما يتعلق بتصميم خدمة السجل، كما سيلتجى الحل المدروس الذي يوفره حاجات الدمة المتنوعة إذا دعت الحاجة – بما في ذلك إمكانية نشوء سوق تنافسية لخدمات السجل.

لذلك، فمن المنطقي تصميم الموجّه بشكل يحيل مسائل الموثوقية والكلفة الخاصة بـ(أ) البحث و(ب) توظيف بيانات التوجيه إلى خدمات سجل متخصصة – بما في ذلك البحث عن هوية السلاسل (chainIds) الخاصة بخدمات السجل. بالتالي، تكون وظيفة XName إظهار واجهات برمجة التطبيقات الخاصة بخدمات السجل، وتخزين ذاكرة مخبأة عن نتائج التوجيه بعد تقديم الخدمة.

ولتحسين الدائرية – أي الحاجة إلى البحث عن خدمة سجل قبل البحث عن خدمة – عند الإطلاق الأولي، يمكن للعقد أن تعمل بنفسها عبر (أ) البحث في هوية السلاسل ذات البرمجة الثابتة (hardcoded) للعثور على خدمة سجل موثوقة بشكل مثبت، أو (ب) عبر البحث لدى الأنداد عبر XBridge من خلال نداء `getRegistryService` (الحصول على خدمة سجل) مخصص، ثم البحث في كل خدمة سجل تظهر في النتائج، مما يساعد في الاستفادة من ضمانات الموثوقية التي قد توفرها كل خدمة لإعداد لائحة موثوقة ومحلية من خدمات السجل (وغيرها من خدمات ما بين السلاسل). وبالعودة إلى السؤال حول كيفية توظيف خدمات ما بين السلاسل والبحث عنها، يُرجى مراجعة قسمي "البحث عن الخدمات" و"خدمات السجل".

وللاطلاع على الوثائق الفنية، يُرجى مراجعة قسم المواصفات الفنية.

**التنفيذ**

- حاليًا: يتم تنفيذ موجّه سلاسل الكتل في كل من `XBridgep2p.exe` ومحفظة بلوكنيت، إلى جانب الخدمات الأساسية الأخرى.
- في المستقبل: من المرجح أن يتوافق هذا العنصر تشغيليًا دائمًا مع العناصر الأخرى لتقديم الخدمات الأساسية، وبالتالي ليس من الضروري توفيره بشكل مستقل. وقد يتم توفيره بشكل مستقل إذا تم إحرار تقدم كافٍ في مجال توليد الإيرادات من توجيه سلاسل الكتل، إذ إنّ توليد الإيرادات من هذه الخدمة سوف يسمح لها بتغطية تكاليف تشغيلها بشكل ذاتي، كما سيحقق التنافسية بين خدمات التوجيه وسيحفز تطوير هذه الخدمة تكنولوجياً. كذلك، إذا تم توليد الإيرادات من عنصر التوجيه، قد يتم دمج هذا الأخير مع عناصر أخرى ضمن شبكة سلسلة كتل مستقلة.

### XChat: تحويل البيانات من الند إلى الند

يتطلب تقديم الخدمات الرقمية وسيلة لبعث وتلقي الرسائل ومضمون الخدمة نفسها. لذلك، تشمل بلوكنيت عنصر XChat ، وهو نظام تراسل من الند إلى الند مشفّر بحسب مبدأ الطرفين (end-to-end) يتيح إمكانية التراسل الفردي والجماعي. (خدمة الرسائل العامة broadcast messages متوفرة حاليًا من خلال XBridge: تراكب الشبكة ما بين السلاسل).

وتختلف متطلبات التواصل وتقديم الخدمات الرقمية بحسب طبيعة الخدمة المعنية. فالخصوصية وعرض النطاق وفترة الانتظار والإصرار ووجود أو غياب الوطاء هي جميعها من المتغيرات. بالتالي، قد تنمو الخدمة لتشمل عدة تقنيات تحويل بيانات. ولكن في هذه المرحلة المبكرة جدًا، يبدو من الملائم توفير حلّ من الند إلى الند يوفر خصوصية وسرعة عاليتين.

للاطلاع على الوثائق الفنية، يُرجى مراجعة قسم المواصفات الفنية.

### التنفيذ

- حاليًا: يتم تنفيذ تحويل البيانات في كل من XBridgep2p.exe ومحفظة بلوكنيت، إلى جانب الخدمات الأساسية الأخرى.
- في المستقبل: قد تتم تجزئة هذا العنصر إلى وحدات في قاعدة البرمجيات، وإذا كان من الممكن توليد الإيرادات منه كتطبيق منفصل، قد يتم إطلاقه بهذا الشكل.



## الخدمات الأساسية

تتطلب خدمات ما بين السلاسل التي يمكن توليد الإيرادات منها ثلاث خدمات بنى تحتية:

- البحث عن الخدمات: طريقة للعثور على الأنداد لتقديم أو استهلاك خدمة؛
- التراسل ما بين السلاسل: طريقة لتقديم خدمة رقمية؛
- التبادل اللامركزي: طريقة لتوليد الإيرادات من الخدمة.

هذه الخدمات هي نتيجة جمع العناصر الأساسية، وبالتالي يمكن إضافتها على أطراف المثلث في الرسم السابق كما يلي:

[الرسم في الصفحة 18 من المستند الإنكليزي]:

الرسم رقم 5: ثانيًا: الخدمات والعناصر الأساسية الثلاثة لبلوكنيت – بما أن أطراف المثلث هي التي تجمع جوانبه الثلاثة، وضع الخدمات على الأطراف يعبر عن طبيعتها كجمع للعناصر الأساسية.

Inter-chain network overlay	تراكب الشبكة ما بين السلاسل
Blockchain router	موجهة سلاسل كتل
P2P data transfer	تحويل البيانات من الند إلى الند
Service lookup	البحث عن الخدمات
Inter-chain messaging	التراسل ما بين السلاسل
Decentralized exchange	التبادل اللامركزي

## البحث عن الخدمات

خدمة البحث ما بين السلاسل هي نتيجة لجمع عناصر XChat و XName و XBridge، وهي تشمل أي سجل لخدمات ما بين السلاسل على بلوكنيت. وقد يتم التعبير عن هذه الخدمة على شكل واجهة برمجة تطبيقات مع تطوّر بلوكنيت.

فكما هي الحال في شبكة الإنترنت التقليدية، المستخدمون بحاجة إلى البحث عن بعضهم البعض وإيجاد بعضهم البعض لكي يتحقق التوافق التشغيلي بينهم. بالتالي، يجب إيجاد نظام شبيه بنظام أسماء النطاقات (DNS). ولكن بخلاف شبكة الإنترنت التقليدية، يتم تقديم الخدمات من الند إلى الند بشكل عام من أي عقدة على شبكة ما، لا من خوادم يمكن الوصول إليها من خلال عنوان بروتوكول إنترنت (IP) واحد. لذا، يكفي العثور على أي عقدة على شبكة سلسلة كتل معينة من أجل طلب خدمة ما. لذلك، فإن "برمجيات السلاسل" (chain codes) هي الشرط الأساسي، إضافة إلى غيرها من الميزات الأمنية.

ويُفترض أن يتم تقديم الخدمات إلى عقدة محددة دون سواها.

ومن أبرز الميزات التي توفرها خدمة السجل على شبكة عقد غير موثوقة من الند إلى الند أنه يمكن لأي طرف أن يقدم هذه الخدمة (سواء كانت نيته خبيثة أم لا)، بخلاف الظروف التي يكون فيها مقدم الخدمة موثوقًا.

وبناءً على استراتيجية التصميم، قد لا تكون موثوقية البيانات التي ينتجها البحث عن الخدمات مضمونة، ما يستوجب تقديم الضمانات حول موثوقية الخدمة في مرحلة لاحقة، أو قد يتوجب تنفيذ البحث عن الخدمات بطريقة تضمن موثوقية البيانات بشكل مشفر، بصرف النظر عن نية العقدة التي تقدم الخدمة. وتجدر الإشارة إلى أن تصميم بلوكنيت محايد تجاه المقاربة المعتمدة للسجلات.

## خطوات التراسل

في حالة التصميم الأول، تتمثل خطوات البحث عن خدمة في ما يلي:

1. العثور على الأنداد عبر XBridge؛
2. الحصول على هوية السلسلة الخاصة بإحدى الخدمات وتنفيذ وظيفة serviceList عبر XName؛
3. البحث عن أنداد عبر XBridge للعثور عن ند على السلسلة المحددة من خلال هويتها؛
4. الانتقال إلى XChat؛ الحصول على قائمة بالخدمات التي يقدمها الند المحدد؛
5. طلب خدمة (والانتقال إلى مرحلة إثبات مدى موثوقية الخدمة).

أما في حالة التصميم الثاني، قد يتم البحث عن الخدمة بشكل عشوائي (أي من دون مساهمة مميزة من هذا المشروع) عبر استخدام إحدى خوارزميات الإجماع الخاصة بسلاسل الكتل (blockchain consensus algorithm) (مثل إثبات العمل)، وعبر استضافة العقدة التي تبحث عن الخدمة لبيانات هوية السلسلة محلياً والبحث فيها مجاناً ببساطة. ولكن هذه الطريقة تفرض تخزيناً وفترة تشغيل كبيرين على المستخدم، وبالتالي من غير المتوقع أن يتم اعتمادها بشكل نموذجي لاستخدام بلوكنيت. وخارج سياق احتفاظ المستخدمين بسلسلة كتل لسجلات الخدمات، قد يتم استخدام عقد التحقق المبسط من الدفعات بالشكل المذكور في [تقرير "بينكوين" الأساسي الأصلي](#). ولتحقيق قدر أكبر من قابلية التوسع والحد من حجم التطبيق، خلال إعداد هذا المستند كان يتم البحث في أنظمة إثبات بديلة. راجع قسمي "سجل التبادل" و"خدمة السجل" للاطلاع على أنماط التصميم.

رسم بياني سياقي

في ضوء النقاط أعلاه، لا بد من اعتماد الهندسة التالية العالية المستوى:

[الرسم في الصفحة 19 من المستند الإنكليزي]:

الرسم رقم 6: رسم بياني سياقي يظهر العلاقة بين العناصر الأساسية على جهاز محلي يستخدم وظيفة البحث عن خدمة.

Local application	التطبيق المحلي
API	واجهة برمجة التطبيق
P2P data transport	تحويل البيانات من الند إلى الند
Inter-chain network	شبكة ما بين السلاسل
Blockchain router	موجّه سلاسل الكتل

للاطلاع على الوثائق الفنية، يُرجى مراجعة قسم المواصفات الفنية.

## التنفيذ

- حالياً: يتم تنفيذ البحث عن الخدمات حالياً إما كجزء من XBridgep2p.exe أو ضمن محفظة بلوكنيت. وفي مرحلة تطوّر بلوكنيت الحالية، فإنّ التطبيق الوحيد المنفّذ هو التبادل اللامركزي، الذي لا يتطلّب سوى أن تنتقي العقد ثنائيات العملات. بالتالي، يبقى من الضروري بناء خدمة بحث ذات أهداف أوسع.
- في المستقبل: بعد أن يستكشف المهتمون في نظام سلاسل الكتل وظيفة البحث عن الخدمات بشكل أوسع وينشأ توافق حولها، من المخطط تطوير خدمة بحث شاملة. كذلك، لتسهيل استخدام المستهلكين للعناصر الأساسية، قد يتم دمج وظيفة البحث عن الخدمات ضمن واجهة برمجة التطبيقات. ومن المرجح أن تحدد عوامل الوقت

## BLOCKNET

والنمو وطلب المستهلكين ما إذا كان سيتم اعتماد واجهة موحدة لجميع الخدمات الأساسية، أو واجهة مميزة مخصصة للبحث عن الخدمات حصراً.

### التراسل ما بين السلاسل

وظيفة التراسل ما بين السلاسل هي نتيجة لجمع عنصرَي XChat و XBridge، وهي تشتمل على عثور مقدّمي الخدمات والمستهلكين على بعضهما البعض وتواصلهما وإيجاد الخدمة التي يتم تقديمها.

ويتم تحديد الخدمة على افتراض أنّ المستهلكين سوف يبحثون بشكل فاعل عن الخدمات، في حين أنّ مقدّمي هذه الخدمات سوف يكونون متاحين بشكل لفاعل.

وتجدر الإشارة إلى أنه على الشبكات العامة من الند إلى الند، لا يمكن افتراض أن يكون كلّ كيان يدعي أنه مقدّم خدمة أو مستهلك حسن النية، وبالتالي، في حالة معظم الخدمات، ينبغي إثبات أنّ مضمون الخدمة موثوق قبل تسديد الدفعة، وأنه لا يمكن للمستهلك تلقّي الخدمة قبل أن يسدّد الدفعة أيضاً. بعبارة أخرى، يجب أن تعمل الخدمات بمعزل عن عامل الثقة وبشكل موحد (atomically).

ومن الطرق المخطط لها لتحقيق غياب "عامل الثقة" في هذا السياق استخدام نظام الإثبات القائم على مبدأ "صفر معرفة" (zero-knowledge). فكما تشير الملاحظات حول [BIP-0199](#)، "تتوفر عدة أنظمة إثبات قائمة على مبدأ صفر-معرفة يمكن استخدامها لضمان استخلاص [دالة هاش الأصلية hash preimage](#) لمعلومات قيمة. فعلى سبيل المثال، يمكن استخدام نظام إثبات قائم على مبدأ صفر-معرفة لإثبات أنّ دالة هاش الأصلية تعمل كمفتاح لفك حل مشفر لأحجية سودوكو. (انظر [ادفع-للعب-سودوكو](#) للاطلاع على مثل ملموس عن بروتوكول مماثل). ولغايات الإيضاح، تجدر الإشارة إلى أنّ توقيعاً رقمياً بسيطاً يتم إنشاؤه لإثبات أنّ مقدّم الخدمة هو بالفعل من يفترضه المستهلك يشكّل نوعاً من أنواع أنظمة الإثبات القائمة على مبدأ صفر-معرفة، إذ إنّ رمز مقدّم الخدمة الخاص يبقى سرياً، ولكن يمكن للمستهلك أن يثبت بنفسه هوية مقدّم الخدمة. ومن المتوقع أن تختلف أنظمة الإثبات المعتمدة بين خدمة وأخرى وفقاً للعناصر المحددة التي يجب إثباتها لتقديم الخدمة بمعزل عن "عامل الثقة".

أما الوحدة (atomicity) في هذا السياق فيتم تحقيقها من خلال التبادل اللامركزي، الذي سيتم شرحه في القسم التالي أدناه.

### خطوات التراسل

إنّ الخطوات المعتادة في خدمة التراسل ما بين السلاسل هي الآتية:

1. العثور على خدمة عبر تراكب الشبكة (network overlay) وطلبها، وفقاً للخطوات المذكورة في القسم السابق؛
2. يوفر مقدّم الخدمة، عبر XChat، مادة لإثبات موثوقية مضمون الخدمة بناءً على مبدأ صفر-معرفة – أي لتقديم إثبات مبني على مجموعة محددة من التأكيدات التي تضمن موثوقية مضمون الخدمة بشكل كافٍ (انظر قسم [سجل التبادل للاطلاع على مثال أولي حول هذا الموضوع](#))؛
3. يقبل المستهلك الخدمة؛
4. ينتقل مقدّم الخدمة إلى مرحلة التبادل اللامركزي للبدء بتقديم الخدمة.

ومن أجل الحد من حجم التطبيق وتبسيط عملية الاستهلاك، قد يتطلّب التراسل ما بين السلاسل واجهة برمجة تطبيقات خاصة. ومن بين حالات الاستخدام التي يبدو فيها هذا الأمر مناسباً، الخدمات المجانية والحالات التي تكون فيها برمجة وظيفة البحث في السلاسل ثابتة، كما في تطبيق تراسل لامركزي مثلاً.

رسم بياني سياقي

في ضوء النقاط أعلاه، لا بد من اعتماد الهندسة التالية العالية المستوى:

[الرسم في الصفحة 21 من المستند الإنكليزي]:

الرسم رقم 7: رسم بياني سياقي يظهر العلاقة بين العناصر الأساسية على جهاز محلي يستخدم خدمة التراسل ما بين السلاسل.

Local application	التطبيق المحلي
API	واجهة برمجة التطبيق
P2P data transport	تحويل البيانات من الند إلى الند
Inter-chain network	شبكة ما بين السلاسل

### التنفيذ

- حاليًا: يمكن للمطورين (أ) دمج XChat ضمن تطبيقاتهم، (ب) والاستفادة من تراكب الشبكة، إمّا من خلال تطبيق XBridge المستقل أو بصيغته المدمجة ضمن محفظة بلوكنيت.
- في المستقبل: سوف يتم وضع العناصر ضمن نظام تركيبي (modularized) وكتابة واجهة برمجة التطبيقات، ويُتحمّل، بالإضافة إلى ذلك، توفير واجهة برمجة لتطبيق يجمع هذين العنصرين ضمن خدمة محددة بشكل واضح.

### التبادل اللامركزي

التبادل اللامركزي هو وسيلة خالية من "عامل الثقة" وموحّدة لتبديل العملات إما بعملات أخرى أو لقاء خدمات. والتبادل اللامركزي هو نتيجة لجمع عنصري XChat وXBridge ومحفظتين أو عقدتين على سلاسل الكتل على الأقل، وذلك كصيغة تنفيذية عامة [ليروتوكول نويل تيرنان للتبادل الموحد](#). وهذا البروتوكول قائم على إنتاج سر، من المطلوب معرفته لصرف الأموال، ولا يمكن تجنّب الكشف عنه عند تسديد الدفعة. بالتالي، إذا صرف منتج هذا السر أموال طرف آخر، يحصل هذا الطرف على السر ويمكنه إذا صرف أموال صاحب السر في المقابل.

ملاحظة: خدمة التبادل اللامركزي مختلفة عن تطبيق التبادل اللامركزي، *Block DX*، الذي يستهلك هذه الخدمة إلى جانب عدة خدمات سلاسل كتل أخرى.

### العمل السابق

تبيّن أن النسخة الأولى من البروتوكول كانت معرّضة لهجمات الابتزاز القائمة على التطويع ([malleability-based extortion attack](#))، وتمّ تصحيحها لاحقًا، ضمن نظام "بينكوين"-ومستنسخاتها (Bitcoin-and-clones)، مع إضافة كود العملية [OP\\_CHECKLOCKTIMEVERIFY](#). ويمكن الاطلاع على تصميم يستخدم هذا الكود في [مدونة](#) كاي كوروكاوا حول الموضوع. أخيرًا، لكي ندعم ليس فقط التبادل الموحد للعملات المشفرة والعملات ولكن أيضًا لأي مضمون رقمي، نضيف إلى هذا البروتوكول "سرًا" هو كناية عن رمز لفك التشفير أو دالة تجزئة (hash function) آمنة. بالتالي، فإنّ كشف هذا السر يتيح الاستهلاك الفوري لمنتج رقمي مشفّر تم تقديمه سابقًا أو إثبات طلب ما للتأكد من الموثوقية. ويسمح ذلك، عندما يقترن باستخدام الإثبات القائم على مبدأ صفر-معرفة المذكور في القسم

السابق، للمستهلك بالتأكد من موثوقية منتج رقمي ما قبل استلامه، كما أنه يحقق استهلاك المنتجات والخدمات المولدة للإيرادات بمعزل عن "عامل الثقة". لذلك، فإن هذا البروتوكول هو للخدمات العامة المولدة للإيرادات.

### خطوات البروتوكول

إنّ خطوات هذا البروتوكول المبدئية هي:

1. ينشئ مقدّم الخدمة رمزاً لفك الشيفرة (للمنتجات الرقمية)، أو توقيعاً رقمياً أو دالة تجزئة تشفيرية لملف ما (للتبث من طلبات التأكد من الموثوقية)، أو رقمًا عشوائياً (لتبادل العملات)، ويستخدمه باعتباره السر الخاص بالبروتوكول.
2. ينشئ مقدّم الخدمة معاملة "مشاركة" (bail-in) قائمة على قاعدة "يمكن للمستهلك صرف هذه المعاملة إذا قدّم (ت) السر، أو إذا وقع كل من المستهلك ومقدّم الخدمة على المعاملة". (إذا كان يتم تقديم منتج رقمي بدلاً من العملات، يُطلب فقط تأمين كمية من الأموال تكفي لتغطية رسم الشبكة).
3. يقيم مقدّم الخدمة آلية احتياطية على شكل معاملة ثانية "إعادة المال"، قائمة على قاعدة "إرسال مخرجات معاملة المشاركة إلى عنوان مقدّم الخدمة، ولكن ليس قبل مدة (x) من الآن".
4. يطلب مقدّم الخدمة من المستهلك توقيع المعاملة لتلبية الشرط الثاني في معاملة المشاركة، ومفاده أنه يمكن استرجاعها إذا وقعها كل من المستهلك ومقدّم الخدمة.
5. يوقع المستهلك معاملة الاسترجاع ويعيدها إلى مقدّم الخدمة.
  1. ملاحظة: يسمح هذا الشرط لمقدّم الخدمة باسترجاع ماله (إذا توفر) بعد مدة زمنية محددة، في حال تراجع المستهلك عن عملية التبادل أو تسبب أي عامل آخر في عدم اكتمال التبادل.
  6. ينشر مقدّم الخدمة معاملة التبادل.
    1. ملاحظة: يمكن للمستهلك "صرف" المعاملة الآن إذا حصل (ت) على السر.
  7. ينشئ المستهلك معاملة "المشاركة" الخاصة به، كما في الخطوة رقم 2 أعلاه، للأموال الموجودة على سلسلة الكتل الخاصة به.
    1. ملاحظة: تتطلب هذه المعاملة كشف السر نفسه للتمكن من صرفها – علماً أنّ مقدّم الخدمة هو الطرف الوحيد الذي يمتلك السر في هذه المرحلة.
    8. ينشئ المستهلك معاملة "الاسترجاع" الخاصة به، كما في الخطوة رقم 3 أعلاه.
    9. يطلب المستهلك من مقدّم الخدمة توقيع معاملة الاسترجاع هذه، كما في الخطوة رقم 4 أعلاه.
    10. يوقع مقدّم الخدمة معاملة الاسترجاع ويعيدها إلى المستهلك.
      1. ملاحظة: يسمح ذلك للمستهلك باسترجاع ماله في حال عجز مقدّم الخدمة عن استلام المال، لأنه إذا لم يستلم مقدّم الخدمة المال، لن يتم الكشف عن السر للمستهلك ولن يتمّ الأخير بالتالي من استهلاك الخدمة أو تلقي المال في عملية التبادل.
      11. ينشر المستهلك معاملة المشاركة الخاصة به.
        1. ملاحظة: يمكن لمقدّم الخدمة في هذه المرحلة صرف أموال المستهلك، لأنه يكون قد حصل على السر.
        2. ملاحظة: عبر صرف أموال المستهلك، يكشف مقدّم الخدمة عن السر، وبالتالي يصبح بإمكان المستهلك الاستفادة من الخدمة.
        3. ملاحظة: إذا فضل مقدّم الخدمة إلغاء عملية التبادل، يمكنه الانتظار حتى انقضاء المدة (x) المذكورة في الخطوة 3 أعلاه ثم نشر معاملة الاسترجاع الخاصة به.
        12. يصرف مقدّم الخدمة معاملة الاسترجاع الخاصة بالمستهلك، ما يؤدي إلى كشف السر.
          1. ملاحظة: إذا لم يصرف مقدّم الخدمة معاملة المشاركة الخاصة به، يمكن للمستهلك نشر معاملة الاسترجاع واسترداد أمواله بعد المدة الزمنية المحددة.

## BLOCKNET

2. ملاحظة: يجب على مقدم الخدمة صرف معاملة المشاركة الخاصة بالمستهلك خلال المدة الزمنية المحددة في معاملة الاسترجاع الخاصة بالأخير، وإلا يكون للمستهلك الحق في استرداد أمواله.

13. يستخدم المستهلك السر إما لصرف معاملة المشاركة الخاصة بمقدم الخدمة أو لفك شيفرة ملف ما أو للتحقق من حقيقة ما.

1. ملاحظة: في حالة تبادل العملات، يجب على المستهلك صرف معاملة المشاركة الخاصة بمقدم الخدمة ضمن المدة الزمنية المحددة في معاملة الاسترجاع، وإلا يكون لمقدم الخدمة الحق في استرداد أمواله.

وتجدر الإشارة إلى أن البروتوكول أعلاه سيشمل خطوات إضافية يتم تحديث عقد الخدمة فيها لكي تبلغ مرحلة المقايضة الموحدة (atomic swap)، وذلك كي يتم تحديث عقد التحقق المبسط من الدفعات والعملاء "الصغار"، الذين يعتمدون على أندايمهم لنقل المعاملات والرسائل، بشكل موثوق.

### خطوات التراسل

يشمل التبادل اللامركزي الخطوات الآتية بالشكل الذي اعتمد فيه ضمن عناصر بلوكنيت:

1. بعد قبول المستهلك لخدمة ما وفقاً للخطوات المذكورة في قسم التراسل ما بين السلاسل، يتخذ مقدم الخدمة الخطوات 1-3 من البروتوكول، ثم يتخذ الخطوة 4 من خلال XChat.
2. يتخذ المستهلك الخطوة 5 من خلال XChat.
3. يتخذ مقدم الخدمة الخطوة 6 من البروتوكول من خلال شبكة سلسلة الكتل المحلية الخاصة به.
4. يتخذ المستهلك الخطوات 7-8، ثم يتخذ الخطوة 9 من خلال XChat.
5. يتخذ مقدم الخدمة الخطوة 10 من البروتوكول من خلال XChat.
6. يتخذ المستهلك ومقدم الخدمة الخطوات 11-13 من البروتوكول من خلال شبكة سلسلة الكتل المحلية الخاصة بكل منهما.

وتجدر الإشارة إلى أن هذه الخطوات تخص خدمة التبادل اللامركزي حصراً. فتطبيق التبادل اللامركزي، Block DX، ومعظم التطبيقات الأخرى، يتطلب عدة خدمات إضافية (أي غير أساسية) لكي يوفر وظيفة التبادل، وخصوصاً نشر الطلبات وملاءمتها واتخاذ إجراءات مكافحة الرسائل غير المرغوب فيها (spam) وإجراءات مكافحة هجمات الحرمان من الخدمات (DoS) وجمع رسوم التبادلات.

### رسم بياني سياقي

في ضوء النقاط أعلاه، تكون العلاقة بين العناصر على جهاز محلي على الشكل التالي:

[الرسم في الصفحة 24 من المستند الإنكليزي]:

الرسم رقم 8: رسم بياني سياقي يظهر العلاقة بين العناصر في جهاز محلي يستخدم خدمة التبادل اللامركزي.

XChat	XChat
API	واجهة برمجة التطبيق
Service provider node	عقدة مقدم الخدمة
Consumer node	عقدة المستهلك

### التنفيذ

## BLOCKNET

- حاليًا: خدمة التبادل اللامركزي مدمجة ضمن محفظة بلوكنيت، وضمن تطبيق XBridgep2p.exe المستقل (ولكن سيستمر تطوير التطبيق الأخير في مرحلة لاحقة). وتتوفر واجهة برمجة للتطبيق تدعم تطبيق التبادل اللامركزي بشكل متزامن.
- في المستقبل: سيتم تجريد (abstraction) الخدمة من مختلف خدمات سلاسل الكتل التي يستهلكها تطبيق Block DX، وسيتم بناء واجهة برمجة تطبيق تجمع بين XChat وخدمات سلاسل الكتل.

عناصر سلسلة الكتل

يمكن للخدمات الأساسية الثلاث، التي تم استعراضها في القسم السابق، أن تتفاعل مع أي من أنواع العناصر التالية على جهاز محلي ما، علمًا أنه يمكن أن تتوفر هذه العناصر على عدة سلاسل كتل:

1. **العقد الكاملة:** عقد ومحفظات "عادية" كاملة المواصفات.
2. **عقد خفيفة:** عقد التحقق المبسط من الدفعات أو حتى العقد الأخف من ذلك (مثل تلك التي تصادق على المعاملات فحسب).
3. **عقد الخدمة:** عقد بمواصفات مميزة تسمح لها بتقديم خدمة معينة بالإضافة إلى وظيفة سلسلة الكتل العادية.

ولا بد من الإشارة إلى أن أنواع العناصر هذه تضيفها الأطراف الثالثة، ولا تتولى بلوكنيت بناءها أو صيانتها. بالرغم من ذلك، تؤدي أنواع العناصر هذه وظيفة أساسية ضمن بلوكنيت، ألا وهي التفاعل مع سلاسل الكتل الخاصة بها. فمن دون هذا التفاعل، لا يمكن لبلوكنيت تحقيق التوافق التشغيلي من دون استنساخ عناصرها الخاصة، وهي مقاربة غير فعالة ولا يمكن اعتمادها بشكل ثابت ومستمر.

وفي ما يلي تصوّر عن الخدمات والعناصر الأساسية الثلاثة، بالإضافة إلى أي أنواع عقد إضافية تستهلك أو تقدّم خدمات ما بين السلاسل:

[الرسم في الصفحة 25 من المستند الإنكليزي]:

الرسم رقم 9: ثالثًا: عناصر سلسلة الكتل بالنسبة إلى الخدمات والعناصر الأساسية الثلاثة. بما أن عناصر سلسلة الكتل تستهلك الخدمات الأساسية، تم تصوير العلاقات بينها على أنها ناشئة من أطراف المثلث.

Inter-chain network overlay	تراكب الشبكة ما بين السلاسل
Blockchain router	موجّهة سلاسل الكتل
P2P data transfer	تحويل البيانات من الند إلى الند
Service lookup	البحث عن الخدمات
Inter-chain messaging	التراسل ما بين السلاسل
Decentralized exchange	التبادل اللامركزي
Full nodes	العقد الكاملة
Light nodes	العقد الخفيفة
Service nodes	عقد الخدمة

خدمات سلسلة الكتل

تعزز خدمات سلسلة الكتل الخدمات الأساسية العامة لكي تلبي حالات استخدام محدّدة. ونظرًا إلى أنه ما من حدّ أقصى لعدد خدمات ما بين السلاسل التي يمكن توفيرها على بلوكنيت، كذلك ما من عدد أقصى لخدمات سلاسل الكتل التي يمكن للأطراف الثالثة إنشاؤها.

ويتطلّب التطبيق اللامركزي الأول على بلوكنيت، Block DX، عدة خدمات سلاسل كتل. وسيتم الحديث عن هذه الخدمات لتوثيقها ولشرح أحد جوانب خدمات سلاسل الكتل.

نظرة عامة عن خدمات سلاسل الكتل التي يوفرها تطبيق Block DX



لكي يحقق تطبيق ما وظيفة التبادل، يجب أن يوفر خدمات أخرى بالإضافة إلى التبادل الموحد بين طرفين. وفي الواقع، يجب أن تتوفر الوظائف الأربع الآتية في أي عملية تبادل (سواء أكانت مركزية أم لامركزية):

- تخزين رأس المال؛
- نشر الطلب (order broadcast)؛
- ملاءمة الطلب (order matching)؛
- الدفع.

وبما أنّ تطبيق Block DX يوفر تبادلًا لامركزيًا، فلا بد من أن تكون هذه الوظائف الأربع لامركزية هي الأخرى (بالإضافة إلى جوانب أوسع نطاقًا متصلة باللامركزية، مثل الحفاظ على برمجة مفتوحة المصدر وقدرة أيّ كان على إدراج أمواله). ومن خلال المقايضات الموحدة، تسمح خدمة التبادل اللامركزي بتنفيذ وظيفتي تخزين رأس المال والدفع بشكل لامركزي أيضًا بشكل بديهي. ولكنّ المقايضات الموحدة لا ترقى إلى مستوى التبادل بحد ذاتها، بل يجب أيضًا توفير وظيفتي نشر الطلب وملاءمته بشكل لامركزي. وهذا ما تحقّقه خدمات سلاسل الكتل:

- يتم نشر الطلب بشكل لامركزي من خلال وظيفة البحث عن الخدمات، التي تجري بأكملها من الند إلى الند.
- تتم ملاءمة الطلب محليًا على التطبيق اللامركزي الخاص بكل طرف في المبادلة، علمًا أنّ هذا التطبيق يجمع ما بين العناصر الأساسية وأحد عناصر سلسلة الكتل أو أكثر.

وتجدر الإشارة إلى أنّ الجانب التصميمي الأبرز لوظيفتي نشر الطلب وملاءمته هو، كسائر الأنظمة التي تعمل من الند إلى الند، إمكانية تعرّضهما لهجمات الحرمان من الخدمات. وبطريقة شبيهة لكيفية توفير "بيتكوين" لنظام نقدي رقمي يحلّ مسألة الجنرال البيزنطي (Byzantine Generals' problem)، يجب أن يقدم الحل الخاص بوظيفتي نشر الطلب وملاءمته ضمانات لجودة الطلب بشكل موثّق. ثانيًا، من الاعتبارات الأخرى التي يجب دراستها في نظام العملات كيفية توليد الإيرادات من خدمة ما، لأنه في نظام لامركزي، إن لم تكن نماذج الأعمال سليمة من الناحية التشغيلية-الاقتصادية، فذلك يعني أنها غير سليمة إطلاقًا. وفي ما يلي نقاش تمهيدي يؤطر هذه المسائل ويستعرض الحل الذي توفره بلوكنيت لها.

### توفير دفاتر طلبات لامركزية

ملاحظة: سوف تفصّل الأجزاء التالية نظام الطلب المقترح ضمن بلوكنيت. وفي حين أنه تتم معاينة أنظمة أخرى، إلا أننا نستعرض النظام الحالي بهدف شرح البيئة التصميمية التي تعمل في ظلها دفاتر الطلبات اللامركزية، كما ولحثّ المهتمين على تقديم التعليقات والمساهمات حول هذا المجال الجديد.

إنّ دفتر الطلبات هو أشبه بلوحة إعلانات يمكن للأطراف المتبادلة أن تضيف السيولة عليها عبر تقديم العروض أو طلب الأسعار، أو أن تسحب السيولة منها عبر قبول العروض أو طلبات الأسعار. وينتج عن تقديم أو قبول الطلبات على الدفاتر التزام بسداد الحسابات عند حدوث ملاءمة بين مقدّم عرض ومقدّم طلب. وفي سياق لامركزي، يصبح دفتر الطلبات هذا أشبه بلوحة إعلانية عامة، يحق لأي كان تقديم طلب عليها، دون أن تكون لأي جهة سلطة على الأطراف التي تقدّم الطلبات، ودون الحاجة إلى أن تكون الأطراف المتبادلة موثوقة.

بالتالي، يتطلّب دفتر الطلبات اللامركزي (أ) وسيلة لضمان تقديم الطلبات الموثوقة من خلاله دون سواها (أي أنه يجب أن يزيل هذا الدفتر الطلبات غير المرغوب فيها)، (ب) احترام الأطراف المتبادلة لالتزاماتها عند حدوث ملاءمة (أي أنه يجب أن يمنع هذا الدفتر هجمات الحرمان من الخدمة). المفارقة هي أنّ سلسلة الكتل ليست تكنولوجيا جيدة لاستخدامها كدفتر طلبات، وذلك لسببين: أولاً، يجب أن تعمل دفاتر الطلبات بسرعة فائقة – أي عليها توفير تجربة فورية للمستخدمين – ولكنّ سلاسل الكتل تتطلّب أن تكون المعاملة (النتيبت من الموثوقية عمومًا) بعمق عدة كتل في السلسلة قبل أن يتم استيفاء درجة الموثوقية المطلوبة. ثانيًا، بما أنّ سلاسل الكتل تتطلّب عملية تقصّي (mining) أو مطالبة (staking) للنتيبت من الموثوقية، فإنّ ذلك يمنح المتقّصين (miners) قدرة استثنائية للاطلاع على معلومات الطلبات، وبالتالي

للاحتيال بقصد الربح (frontrunning)، ما يعطي الأفضلية لطلباتهم هم إذا تقصوا من الكتلة التالية. لذلك، يجب اعتماد نظام آخر لتحقيق اللامركزية في الطلبات.

ولا بد من ذكر ميزة أخيرة لدفاتر الطلبات، وهي أنّ ما من أموال تُرسل أو تتعرض للخطر خلال نشر الطلب أو ملامته، ذلك أنّ الطلبات هي كناية عن التزامات بالتسديد المستقبلي، ما يعني أنّ الأطراف التي تستخدم نظام الطلب تتعرض لدرجة خطر أقل بكثير مما لو كانت تعتمد نظام سداد أو دفع. ويعطي ذلك ميزة لتصميم نظام الطلبات، إذ إنه ليس من الضروري قبول "غرامة الأداء" (performance penalty) المقيدة والتي تصبح إلزامية في حال استخدام سلسلة كتل؛ فنظام الطلبات قد يعمل بنجاح مع تحمّل درجة معينة من عدم الموثوقية. بالتالي، تتمثل شروط التصميم، بالحد الأدنى، في منع الأنشطة غير المشروعة من أن تتعدى بعض الأفعال المنفردة. ويتميّز الحلّ الذي تقدّمه بلوكتيت، بشكل خاص، بالخصائص الآتية:

1. يدعم نظام UTXO للتثبت، أي أنّ الأموال التي تُسدّد خلال طلب ما قابلة للصرف؛
2. يمنع الطلبات غير المرغوب بها من الانتشار، من خلال رسم على التبادل شبيه بنظام إثبات العمل "hashcash"، الذي يكون أثره طفيفاً على المتبادلين الموثوقية، ولكنه يجعل من المكلف جدّاً تقديم الطلبات غير المرغوب بها؛
3. يمنع هجمات الحرمان من الخدمة التي تطل الطلبات من الانتشار، بالطريقة نفسه المذكورة في البند (ب) أعلاه؛
4. يجعل التآمر بين عقد الخدمة (service node collusion) غير مجدٍ من الناحية الاقتصادية (انظر القسم حول خصائص عقد الخدمة المميزة)؛
5. يدعم ملاءمة الطلب الجزئية، ما يسمح للأطراف المتبادلة باستهلاك طلب ما من دون أو يودّي التخلّي عن مبادلة واحدة إلى إيقاف الطلب برمته إلى حين التمكن من صرف معاملة الاسترجاع (refund transaction). وقد يساهم ذلك في تعزيز استراتيجية التبادل التي تتجنّب "تكلفة الفرصة البديلة" (opportunity cost) التي تنشأ عندما يتخلّى أحد الأطراف عن مبادلة واحدة عالية القيمة، وذلك من خلال توزيع رأس المال على عدة طلبات بنطاق أسعار متنوّعة.

### ما هو شكل الطلبات اللامركزية؟ - التنظيم الذاتي (self-sovereignty)

نحن نعتبر أنّ اللامركزية، في ما يخصه الطلبات، تعني الحفاظ على التنظيم الذاتي بين الأطراف المشاركة في التبادل اللامركزي. فتسود الكثير من المفاهيم الخاطئة حول اللامركزية، وغالباً ما يتم الخلط بينها وبين "التوزيع" ومفاهيم مرتبطة بالتوصل إلى إجماع. ولكنّ اللامركزية، كما يشرح فيتاليك بوتيرين (Vitalik Buterin) بوضوح، تتمحور بشكل أساسي حول التحكم: فالنظام اللامركزي لا يضع أي طرف تحت تحكّم طرف آخر، إلا عندما تشتمل المسألة على مورد مشترك، وفي تلك الحالة تشارك جميع الأطراف بشكل متساوٍ. أمّا التوزيع فيعني تقسيم مهمة أو دور ما بين عدة أطراف، بصرف النظر عمّا إذا كان طرف واحد يتحكّم بالعملية برمتها. وأخيراً، يُقصد بـ"الإجماع" التوصل إلى توافق حول حقيقة أو فعل ما، بصرف النظر عمّا إذا كانت هذه الحقيقة أو الفعل موزعين أو لامركزيين. ولكن ذلك لا يعني، مثلاً، أنّ تحقيق اللامركزية في نظام "بيتكوين" لا يستلزم التوصل إلى إجماع حول المعاملات المشروعة أو لا يتطلب توزيع العمل على أطراف الشبكة. فهذان العاملان ضروريان ولا شك، ولكنهما ليسا كافيين لتحقيق اللامركزية في نظام "بيتكوين".

من ناحية أخرى، تشبه اللامركزية في العملات لامركزية الطلبات، حيث يتوفر التنظيم الذاتي في الأولى: فتماماً كما يستطيع كل من يمتلك "بيتكوين" أن يرسل الأموال بمعزل عن تحكّم أي طرف ثالث، كذلك يمكن لأي طرف متبادل أن يضيف السيولة إلى دفتر طلبات أو يسحبها منه؛ وتماماً كما يمكن لمستخدمي "بيتكوين" التثبت بأنفسهم من أنّ مبلغاً محدداً قد أرسل بالفعل ومن وقت إرساله، كذلك يمكن لأي طرف متبادل أن يتحقّق بنفسه من موثوقية الطلبات من رسائل قبول الطلبات. ولكن تشبيه الطلبات بالعملات المشفرة يقف عند هذا الحد؛ فالفرق هو أنّ الطلبات هي عروض إرسال أموال، مقابل استهلاك طرف آخر لها في المستقبل، وهي مبادلة تتضمن عملية ملاءمة. ونظراً إلى أنّ الملاءمة تتطلب توقيع

## BLOCKNET

الأطراف المشاركة على رسالة قبول الطلب، وإلى أنه ليس من المطلوب مشاركة أي طرف آخر للنتيبت من أن ملاءمة الطلب قد حصلت، يمكن بالتالي للأنداد على شبكة ما أن يطلعوا على وضع الطلبات بشكل ذاتي التنظيم بالكامل، من دون الحاجة إلى خوارزميات إجماع، كإثبات العمل مثلاً.

إذاً، ما نقرحه هو تصميم ذاتي التنظيم بالكامل لنظام الطلب، يشمل ثلاث أطراف. وفي ما يلي استعراض غير مفصل لهذا التصميم، الذي سيتم تطويره مع تبلور النقاش:

- قبل نشر أي طلب، يرسل أحد المتبادلين الطلب ومعاملة رسم مكافحة الطلبات غير المرغوب بها ( anti-spam) بشكل خاص إلى عقدة خدمة. بعد ذلك، يمكن لعقدة الخدمة نشر هذه المعاملة، ما يؤدي إلى صرف الأموال على شكل رسم مخصص للشبكة، وناي طرحه احتمال خسارة مرسل الطلب للمال إذا كانت عقدة الخدمة غير موثوقة. وبعد التنتيبت من معاملة الرسم، توقع عقدة الخدمة على الطلب، ويستخدم مرسلو الطلب المفترضين ذلك للتحقق من صحة الطلب.
- بعد نشر طلب ما، يمكن للمتبادلين التنتيبت، بأنفسهم، (أ) من أن الطلب مدعم بأموال حقيقية، (ب) ومن أن الطلب موقع من عقدة خدمة (ما يعني أنه تم دفع رسم التبادل).
- عندما يحاول طرف متبادل أو أكثر قبول طلب ما، يختار مرسل الطلب أحدهم (يقبل أول محاولة موثوقة)، ثم يختار، بنفسه، الطرف المقابل.
- عندما يتم اختيار الطرف المقابل، يكون من مصلحة مرسل الطلب ألا يعتمد الطرف المقابل إلى تنفيذ هجمة حرمان من الخدمة، ولذلك ينتظر مرسل الطلب تأكيد عقدة الخدمة أنه تم سداد رسم الخدمة.
- بعد أن تنتشر عقدة الخدمة رسالة القبول الموقعة، يتم تحديث باقي دفاتر الطلبات في السوق عبر إزالة هذا الطلب منها.

بالتالي، يعمل دفتر الطلبات الخاص بتطبيق Block DX كآلة حالة (state machine) لامركزية. ويصور الرسم البياني التالي آلة الحالة هذه على مستوى مرتفع (الأجزاء التالية من هذا المستند تتمحور حول تفاصيل معينة من هذا الرسم البياني):

[الرسم في الصفحة 29 من المستند الإنكليزي]

الرسم رقم 10: آلة حالة لامركزية للطلبات

Orders: decentralized state machine Notes: - Order books exist on trader nodes (each node maintains an independent order book) - Basic optimization: minimise the number of broadcast messages required per trade (e.g. don't broadcast removals from order book) - Note: this is not decentralized consensus, it is decentralized (self-sovereign) actions on a p2p network	الطلبات: آلة الحالة اللامركزية ملاحظات: - توجد دفاتر الطلب على عقد التبادل (كل عقدة تحتفظ بدفتر مستقل) - تحسين بسيط: الحد من عدد الرسائل المنشورة المطلوبة لكل تبادل (مثلاً: عدم نشر الإزالة من دفتر الطلبات) - ملاحظة: لا يشكل ذلك إجماعاً لامركزياً، بل هو عبارة عن أفعال لامركزية (ذاتية التنظيم) على شبكة من الند إلى الند
Place order Active Conditions for a peer to determine "active status": - Collateral txid paid (order signed by service node) - Input coins confirmation count >= conf setting in DX - UTXO check succeeds	وضع الطلب فاعل شروط تحديد الأنداد أن وضع الطلب "فاعل": - دفع المعاملة الجانبية (توقيع الطلب من قبل عقدة الخدمة) - مبلغ الأموال المدخلة المؤكدة يفوق أو يعادل إعداد "conf" في تطبيق DX - يتم التنتيبت من UTXO بنجاح
Assume cancelled	يفترض أن الطلب ألغي

## BLOCKNET

Presume market maker is offline	يُفترض أن مرسل الطلب خارج الشبكة
Acceptance broadcast received?	تلقي القبول المنشور؟
Accepted (by other)	قبول (من قبل طرف آخر)
Valid order accepted Broadcast received	تلقي قبول الطلب الموثوق والمنشور
Market maker responds?	يستجيب مرسل الطلب؟
Accepted (by self)	قبول (ذاتي)
Both parties sign order accepted message	يوقع الطرفان على رسالة قبول الطلب
Accept order?	قبول الطلب؟
Cancel order?	إلغاء الطلب؟
cancelled	ألغي
Market maker broadcasts cancellation	ينشر مرسل الطلب رسالة الإلغاء
Close wallet?	إقفال المحفظة؟
Cancelled	ألغي
Market maker broadcasts cancellation	ينشر مرسل الطلب رسالة الإلغاء
Service node poll: wallet offline?	استفتاء عقد الخدمة: المحفظة خارج الشبكة؟
Market maker's order(s) cancelled?	ألغي(ت) طلب(ات) مرسل الطلب؟
Cancelled	ألغي
Service node broadcasts trade fee tx	تنشر عقدة الخدمة معاملة رسم التبادل
Trade fee paid?	تم سداد رسم التبادل؟
Match made	حصول ملاءمة
Service node signs order accepted message; both service node and market maker broadcast	توقع عقدة الخدمة رسالة قبول الطلب؛ ينشر كل من عقدة الخدمة ومرسل الطلب الرسالة
Remove from order book	الإزالة من دفتر الطلبات
Service node discards trade fee tx	تتجاهل عقدة الخدمة معاملة رسم التبادل
Pending	قيد الانتظار
Coin exchange process	عملية تبادل الأموال
Yes	نعم
No	لا

### فصل الأدوار: عقد الخدمة

يشمل نظام الطلب الخاص ببلوكنيت ثلاثة أطراف، بدلاً من أن يقتصر على طرفي التبادل، وهو أمر قد يبدو مفاجئاً. ولكن ذلك أتى نتيجة التفكير في أثر الرسم المفروض على تجربة المستخدمين. فمثلاً، يمكن اعتماد حل لامركزي بسيط إذا فرض على المستخدمين تسديد رسم تبادل قبل تقديم طلباتهم، ولكن ذلك يؤدي إلى سداد المستخدمين للرسم حتى لو ألغوا طلبهم، وحتى لو تخلى الطرف الآخر عن عملية التبادل. أما لتحقيق الهدف المرجو (والذي يتمثل بفرض رسم تبادل على المستخدمين فقط في حال اكتمال التبادل أو إلغائه)، وفي الوقت نفسه لضمان دفع الرسوم مسبقاً، فتم اعتماد نظام ثلاثي الأطراف. وما يسمح بتحقيق هذا النظام هو فصل حوافز تقديم السيولة وسحبها والتأكد من أن الطرف الآخر قد سدد الرسم اللازم وجني رسوم التبادل، من دون (أ) نشر معاملة رسم تبادل مزورة، (ب) وتزامناً مع عدم قبض رسوم التبادل إلا بعد قبول الطلب. وتجدر الإشارة مجدداً إلى أن الهدف الأساسي من ذلك هو ضمان التنظيم الذاتي، الذي يصبح ممكناً من خلال فصل وجمع حوافز كل طرف بطرق معينة. لا شك أن هذا النظام بالغ التعقيد، وسوف نعود إلى شرح كل جانب منه على حدة بهدف تبسيط المعلومات؛ أما في هذه المرحلة من النقاش، يكفي القول إن على عقد الخدمة طلب رسوم تبادل، ولكن من دون أن يتم سداد هذه الرسوم بالفعل إذا ألغى الطرف الآخر العملية قبل قبول الطلب (انظر مخطط البروتوكول المفصل أدناه للاطلاع على الأدوار وتتابعها بشكل دقيق في نظام الطلب).

### خصائص عقد الخدمة المميزة

من أجل توفير آلية فعّالة لقبول رسوم التبادل ولكن من دون صرفها (رسوم مكافحة الطلبات غير المرغوب بها وهجمات الحرمان من الخدمة) إذا تم إلغاء الطلب قبل قبوله، تتميز عقد الخدمة بعدة خصائص فريدة:

## 1. قليلة العدد ويسهل التعرف إليها

نظرًا إلى أن الأطراف المتبادلة تعتمد على رسوم التبادل كوسيلة لمنع الطلبات غير المرغوب بها وهجمات الحرمان من الخدمة، من الضروري أن تكون هذه الأطراف قادرة على التثبت بسهولة من سداد الرسوم، وهو أمر يتطلب القدرة على التمييز بين الطلبات (ورسائل قبول الطلبات) الموقعة من قبل عقد الخدمة وبين توقعات غيرها من العقد.

لتحقيق ذلك، نحن نطلب أن تكون عقد الخدمة حاملة لـ 5000 BLOCK. بهذه الطريقة، يمكن لأي طرف أن يبحث بسهولة في سلسلة الكتل عن block UTXOS 5000 وأن يعدّ لائحة بعقد الخدمة. بعد ذلك، إذا تحقّق عنوان ما يتضمّن 5000 BLOCK من التوقيع على أحد الطلبات، يعني ذلك أنه تم توقيعه من قبل عقدة خدمة.

## 2. الاحتيال مكلف

يجب على عقد الخدمة أن تحمل BLOCK لـ 1000 كتلة على الأقل قبل أن تتمكن من جني رسوم التبادل ومكافآت الكتل. وإذا عمدت إحدى عقد الخدمة إلى الاحتيال – أي إذا نشرت رسم تبادل رغم إلغاء أحد المستخدمين لطلبه بطريقة مشروعة – يمكن للطرف المعني أن يقدم إثبات-مطالبة على القائمة السوداء (blacklist proof-claim) ضد عقدة الخدمة على الشبكة. ولتجنب إراجها على القائمة السوداء، يجب نقل الـ BLOCK إلى عنوان جديد، ما يعني خسارة عقدة الخدمة المعنية للكثير من رسوم التبادل ومكافآت الكتل، فضلًا عن اضطرارها إلى انتظار 1000 كتلة.

## 3. الاحتيال لا ينتج أي أرباح

بالإضافة إلى ارتفاع "تكلفة الفرصة البديلة" نتيجة الاحتيال، لا يمكن لأي عقدة خدمة أن تحقق أرباحًا مباشرة أو مضمونة من خلال صرف رسم تبادل بشكل غير مشروع، لأن رسوم التبادل تُمنح إلى المشارك الفائز التالي (next winning staker)، الذي يتم تحديده بناءً على قاعدة الاحتمالية، أي أنه النتيجة لا تكون محددة سلفًا. بالتالي، ما من حوافز أو أرباح محتملة تدفع عقد الخدمة إلى ممارسة الاحتيال.

## 4. دعم العملات المشفرة والتوافق التشغيلي بين العملات وبلوكنيت

لكي تقوم بالتثبت من طلب أو رسالة قبول، على عقد الخدمة أن تتأكد من أن العنوان (أو العناوين) في الطلب يتضمّن مبلغًا كافيًا من المال لتمويل الطلب. بالتالي، فإنّ المتطلبات من حيث التجهيزات كثيرة، إذ يجب على عقد الخدمة أن تحتفظ بمحفظات العقد الكاملة (full node wallets) التي تتوافق تشغيليًا مع بلوكنيت. ويمكن هذا الشرط عقد الخدمة من تحويل الرسائل والمعاملات إلى عقد التحقق المبسط من الدفعات والعقد الخفيفة عند الضرورة.

## 5. لا إمكانية للممارسة التبادل من الداخل (insider trading)

بما أنه يمكن لجميع عقد الخدمة التوقيع على الطلبات، ليس من المجدي أن يفصل مالك إحدى عقد الخدمة طلبات طرف معين دون سواه (مثل عقدة التبادل الخاصة به)، لأنه إذا تأخر في التوقيع على الطلبات (أو رسائل القبول) العائدة الآخرين، يمكن لهؤلاء الحصول على خدمة جيدة بسهولة من أي عقدة خدمة أخرى.

نتيجة للخصائص المذكورة أعلاه، تمتلك الأطراف المتبادلة: (أ) "دافعًا جيدًا" للوثوق بادعاء عقد الخدمة، (ب) والقدرة على إيفاد عقدة الخدمة ليس فقط رسم التبادل، بل إيرادات أخرى تفوق هذا الرسم بكثير، إذا مارست الاحتيال. (ج) وكما ذكر أعلاه، فإنّ درجة الخطر المنخفضة التي يواجهها المتبادلون خلال عملية الطلب، فضلًا عن "غرامة الأداء" الشديدة المفروضة في حال طلب درجة أعلى من الموثوقية، لا يشجعان الأطراف على طلب درجة أعلى من الموثوقية حول الطلبات.

## بروتوكول نظام الطلب

## BLOCKNET

إذا افترضنا أنّ النقاش السابق عن الطبيعة العامة لدفتر الطلبات اللامركزي وخصائصه صحيح، فإنّ مخطّط البروتوكول أدناه هو الحل الطبيعي له. ويُرجى الأخذ في الاعتبار أنّ هذا الحل ما زال قيد التطوير من قبل المساهمين في بلوكنيت وقد يخضع لتحسينات إضافية.

### (1) يحضّر مرسل الطلب رسم التبادل

1. يريد مرسل الطلب شراء  $xLTC$  بـ  $yBTC$
2. يحسب مرسل الطلب رسم المعاملة (يُدفع بـ  $priceBLOCK$ ):  $y * 0.05\% / priceBLOCK$ 
  - أ. ملاحظة:  $priceBLOCK$  هو متوسط سعر شراء  $BLOCK$  في السوق لحظة تقديم الطلب.
  - ب. ملاحظة: يجب شراء  $BLOCK$  قبل تقديم الطلب (يوصى بذلك لتسريع العملية)، عند إطلاق تطبيق  $DX$  مثلاً – ولكن فقط عندما ينخفض مخزون  $BLOCK$  في التطبيق إلى ما دون  $BLOCK$  واحد على سبيل المثال. والخطوة (أ) أعلاه تهدف إلى احتساب كمية  $BLOCK$  التي يجب صرفها ببساطة.
3. ينشئ مرسل الطلب  $txspamfee$  (معاملة رسم مكافحة الطلبات غير المرغوب بها) من دون نشرها، علماً أنّ هذا الرسم يُفرض على مرسل الطلبات بهدف منع الطلبات غير المرغوب بها:
  - أ. معاملة  $BLOCK$
  - ب. مع تحديد رسم الشبكة برسم التبادل الصحيح (0.05%)
  - ج. وتحديد المخرجات كعنوان مرسل الطلب.

### (2) يقدّم المرسل طلبه

1. يقدّم المرسل طلب شراء  $xLTC$  بـ  $yBTC$ . الحقول:
  - أ. قيمة  $y$
  - ب. قيمة  $x$
  - ج. عنوان (عناوين)  $BTC$  الذي يمولّ الطلب
  - د. عنوان  $LTC$  الذي سيتلقى الأموال
  - هـ. تفاصيل التواصل مع المرسل على  $XChat$  (العنوان، المفتاح العام  $pubkey$ )
  - و. بيانات أخرى مفيدة: تاريخ انتهاء الصلاحية، إلخ.
  - ز. هوية المعاملة ( $txid$ ) الخاصة بمعاملة رسم مكافحة الطلبات غير المرغوب بها ( $txspamfee$ )
2. يبعث المرسل كلاً من الطلب والـ  $txspamfee$  إلى عقدة الخدمة عبر  $XChat$
3. تتحقّق عقدة الخدمة من الرسم مقارنة بالطلب:
  - أ. تتوافق هوية معاملة  $txspamfee$  مع حقل هوية معاملة  $txspamfee$  الخاص بالطلب
  - ب. رسم الشبكة  $txspamfee$  يبلغ 0.05% من  $y$(1) ملاحظة: يجب على عقدة الخدمة أن تتأكد من سعر  $BLOCK$  على تطبيق  $DX$  لاحتساب الرسم الصحيح؛ ولكي يتم أخذ التقلّب في الاعتبار، يجب أن يكون الرسم ضمن 15% من القيمة الصحيحة
  - ج. يتضمّن عنوان  $BLOCK$   $txspamfee$  ما يكفي من الـ  $BLOCK$  لتغطية قيمة الرسم
  - د. لا معاملات ضمن "مخزن الذاكرة" لصرف عنوان الـ  $BLOCK$  نفسه
4. توقع عقدة الخدمة على الطلب.
5. ينشر كل من عقدة الخدمة والمرسل الطلب والتوقيع.
6. تحتفظ عقدة الخدمة بمعاملة  $txspamfee$  في الذاكرة ولا تنشرها فوراً.

### (3) يحضّر المتلقي رسم التبادل

1. يريد المتلقي شراء  $yBTC$  بـ  $xLTC$

## BLOCKNET

1. ملاحظة: من وجهة نظر تجربة المستخدم (UX)، من المرجح أن يقدم المتلقي طلب سوق أو طلبًا بسعر محدد بكميات مختلفة؛ ويفترض هذا الاحتمال وجود بروتوكول ملائمة ( matching protocol) للجمع بين الكميات المحددة بين الأطراف، وبالتالي لإنجاز الطلبات جزئيًا.
2. يحتسب المتلقي معاملة الرسم (تُدفع بـBLOCK price:  $x * 0.2\%$ ) ملاحظة: BLOCK price هو متوسط سعر شراء BLOCK في السوق لحظة تقديم الطلب.
  1. ملاحظة: BLOCK price هو متوسط سعر شراء BLOCK في السوق لحظة تقديم الطلب.
  2. ملاحظة: يجب شراء BLOCK قبل تقديم الطلب (يوصى بذلك لتسريع العملية)، عند إطلاق تطبيق DX مثلًا – ولكن فقط عندما ينخفض مخزون BLOCK في التطبيق إلى ما دون BLOCK واحد على سبيل المثال. والخطوة (أ) أعلاه تهدف إلى احتساب كمية BLOCK التي يجب صرفها ببساطة.
3. ينشئ المتلقي txDOSfee (معاملة رسم مكافحة هجمات الحرمان من الخدمة) من دون نشرها:
  1. معاملة BLOCK
  2. مع تحديد رسم الشبكة برسم التبادل الصحيح (0.2%)
  3. وتحديد المخرجات كعنوان المتلقي

### (4) يقبل المتلقي الطلب

1. يبعث المتلقي رسالة قبول الطلب إلى مرسل الطلب عبر XChat. الحقول:
  1. عنوان LTC الذي يمول الطلب
  2. عنوان BTC لتلقي الأموال
  3. هوية المعاملة الخاصة بـ txDOSfee
  4. التوقيع باستخدام المفتاح الخاص لعنوان BTC
2. يتحقق المتلقي من رسالة القبول:
  1. الرسالة موقعة بشكل موثوق من خلال المفتاح الخاص التابع لمرسل الطلب والذي يتوافق مع عنوان BTC الخاص بالمرسل
  2. يتضمن عنوان BTC رصيدًا كافيًا
  3. لا معاملات في مخزن الذاكرة تُصرف من عنوان BTC
- (1) ملاحظة حول الثقة: رغم أن مرسل الطلب يستطيع التحقق من رسالة القبول، يجب ألا يتابع العملية إلى حين تحقق عقدة الخدمة من txDOSfee، وإلا قد ينفذ المتلقي هجمة حرمان من الخدمة مجانًا.
3. يوقع مرسل الطلب رسالة قبول الطلب
4. تتحقق عقدة الخدمة من رسالة القبول:
  - (1) ملاحظة حول الثقة: على مرسل الطلب التحقق من رسالة القبول قبل أن تقوم عقدة الخدمة بذلك، وإلا تمكنت عقدة الخدمة من (أ) الحصول على معلومات حول فائدة الشراء بسعر محدد قبل العثور على طرف للتعامل معه، (ب) ومن تفضيل التحقق من إحدى رسائل القبول دون سواها (ج) أو عرقلة/انتقاء رسائل القبول بناءً على معايير اعتباطية.
1. هوية معاملة txDOSfee مماثلة لحقل هوية معاملة txDOSfee في رسالة القبول
2. رسم txDOSfee على الشبكة يبلغ 0.2% من x
- (1) ملاحظة: يجب على عقدة الخدمة أن تتأكد من سعر BLOCK على تطبيق DX لاحتساب الرسم الصحيح؛ ولكي يتم أخذ التقلب في الاعتبار، يجب أن يكون الرسم ضمن 15% من القيمة الصحيحة
3. يتضمن عنوان BLOCK txDOSfee ما يكفي من الـBLOCK لتغطية قيمة الرسم
4. لا معاملات ضمن "مخزن الذاكرة" لأصرف عنوان BLOCK نفسه

6. توقع عقدة الخدمة على الطلب.
7. ينشر كل من عقدة الخدمة والمتلقي رسالة القبول.
8. يحلّ دفتر الطلبات الخاص بكل طرف الرسالة، الموقعة من قبل المرسل وعقدة الخدمة، كتغيير لحالة الطلب من "مفتوح" إلى "منجز"، ويزيل الطلب من الدفتر.
- 5) الانتقال إلى "المشاركة" في إعداد المعاملة
1. بحسب بروتوكول التبادل الموحد.

### شروط نشر معاملات رسوم التبادل

لا تنتشر عقد الخدمة معاملة رسم التبادل إذا:

- انتهت فترة صلاحية الطلب.
- أغلق الدافع التطبيق قبل قبول أي طرف للطلب.
- ألغى الطلب قبل قبوله (الخطوة 4-خ أعلاه).
- لا ينشر الطرف الآخر معاملة المشاركة (bail-in transaction).

### نظام ملائمة الطلبات

تتطلب بلوكنيت نظامًا لامركيًا من أجل (أ) تحويل أنواع الطلبات المعتادة على التبادلات (طلبات السوق وطلبات السعر المحدد، إلخ) إلى حالات استهلاك السيولة (liquidity consumption events)، (ب) وملائمة كميات إحدى العملات بكميات عملة أخرى بأسعار محددة. باعتماد آلة حالة الطلبات (order state machine) المبنية أعلاه – حيث تتخذ الأطراف المتبادلة أدوارًا ذاتية التنظيم كمرسلين أو متلقين – تشتمل ملائمة الطلبات على طلب المتلقين استهلاك السيولة من المرسلين. وفي حالة الطلبات ذات السعر المحدد، تشتمل ملائمة الطلبات على وجود أو غياب طلبات تصل إلى عتبة معينة، وتهدف إلى تحديد دور كل طرف تبادل كمرسل أو متلقي.

وتجدر الإشارة إلى أنّ سلوك طلب السوق هو كالتالي: استهلاك الطلبات الموجودة على الدفاتر، بدءًا من الطلب ذي السعر الأفضل، ثم الانتقال إلى الطلب الذي يليه، وهكذا دواليك، إلى حين استهلاك طلب السوق بالكامل. وإذا استهلك طلب السوق جميع الطلبات الموجودة على الدفتر من أن يتم استهلاكه هو بالكامل، يتم إلغاء المبالغ المتبقية على الطلب.

أما سلوك طلب السعر المحدد فهو كالتالي: إذا كان الطلب للبيع وسعر الطلب أدنى (أو إذا كان الطلب للشراء وسعر الطلب أعلى) من أعلى (أدنى) عرض مقابل على دفتر الطلبات، يتم استهلاك الطلبات الموجودة على الدفتر إلى أن يصبح طلب المستخدم هو الأدنى (الأعلى)، أو إلى حين استهلاكه بالكامل. وفي الحالة الأولى، تتم إضافة ما يتبقى من الطلب إلى الدفتر.

ولتجنب مشاكل الصرافة في حالة العملات المشفرة القائمة على قاعدة UTXO – وما تستتبعه من حاجة إلى الانتظار حتى يتم تأكيد الصرافة قبل التمكن من مبادلة باقي الطلب – سوف يسمح عنصر XBridge باعتماد نظام آلي لتوزيع أموال محفظات الأطراف المتبادلة إلى مبالغ صغيرة ومنتظمة في عناوين منفصلة، من أجل تأمين العدد الأقصى من المخرجات التي يتم استهلاكها بالكامل من قبل الطرف الآخر، ولتخفيف التغيرات في المستويات الضئيلة الأهمية. بالإضافة إلى ذلك، سيتم فرض حد أدنى لحجم التبادلات، يعادل حجم المبالغ الموجودة في كل عنوان، وذلك من أجل تجنب إنتاج الصرافة بهدف الاحتيال (malicious creation of change).

### خدمة سجل التبادل



## BLOCKNET

إنَّ سجلَّ تبادل العملات ضروري لإعداد الرسوم البيانية وغيرها من أدوات التحليل التقني، كما ولحصول الأطراف المتبادلة على معلومات حول السوق لكي يبنوا قراراتهم على أساسها. كذلك، فإنَّ موثوقية سجل التبادلات بالغ الأهمية، إذ يمكن لطرف ما كسب أفضلية كبيرة على الأطراف الأخرى إذا عمد إلى تزوير هذا السجل. لذلك، سوف يتم تأمين سجل التبادلات بين أي عملتين بطريقة خالية من "عامل الثقة" كإحدى الخدمات التي توفرها بلوكنيت.

ولا من الإشارة إلى أنَّ مخطط الحل أدناه مثالي بعض الشيء، لغايات التبسيط ولإستعراض الفكرة الرئيسية بدلاً من تفاصيل الحل التقنية؛ فلا شك أنَّ الحل الجاهز للاستخدام سوف يكون أكثر إيجازاً (وذلك بسبب عدم ذكر بيانات سجل التبادل في إحدى التبادلات الموحدة) وسيعتمد نظام إثبات قائم على مبدأ صفر-معرفة (zero-knowledge proof scheme) وأكثر تطوراً، مثل نظام "bulletproofs" مثلاً.

وستتطلب مجموعة البيانات الخالية من "عامل الثقة" (أ) وسيلة ملائمة وموثوقة لإدراج بيانات التبادل فيها، (ب) ووسيلة موثوقة لاستخراج البيانات منها. ويأتي الحل على الشكل التالي:

سيتم إنشاء سلسلة كتل مخصصة لسجل التبادل، ويشتمل عمل عقد هذه السلسلة المتعلق بالتوصل إلى إجماع علي إدراج بيانات معاملات التبادل من سلاسل كتل أخرى في سلسلة الكتل المخصصة لسجل التبادل. وتتضمن البيانات المعتادة التي سيتم إدراجها:

- العملة أ
- العملة ب
- كمية العملة أ
- كمية العملة ب
- سعر العملة أ: العملة ب
- توقيت صرف أول معاملة مشاركة

ويجب أن تكون البيانات التي يتم إدراجها كافية لكي يتمكن مستهلكو البيانات من صياغتها ضمن أشكال عدة. مثلاً، بالنسبة إلى رسوم TradingView البيانية، يجب توفير البيانات التالية:

- مدة "الشمعة اليابانية" (candle)
- توقيت بدء "الشمعة اليابانية"
- السعر الابتدائي
- السعر الختامي
- الحد الأقصى
- الحد الأدنى
- إلى ما هنالك

### إدراج بيانات الإثبات

لإدراج بيانات التبادل، يجب أن تقدّم العقد على سلسلة الكتل الخاصة بسجلّ التبادل بيانات إثبات:

- العملة أ
- العملة ب
- هوية المعاملة الخاصة بمعاملة المشاركة المصروفة في السلسلة أ
- هوية المعاملة الخاصة بمعاملة المشاركة المصروفة في السلسلة ب (أو تحديد غيابها)
- هوية المعاملة الخاصة بمعاملة رسم التبادل لمرسل الطلب
- هوية المعاملة الخاصة بمعاملة رسم التبادل للمتلقّي
- دلالات زمنية لكل المعاملات

○ دلالة زمنية للمعاملة التي قُدمت فيها بيانات الإثبات

بعد ذلك، تتحقق الشبكة من بيانات الإثبات التي تقدمها عقدة ما، عبر البحث في سلاسل الكتل الخاصة بالعملات المستخدمة في عملية التبادل، والتثبت من صحة بيانات الإثبات عبر مقارنتها بسلاسل الكتل الخاصة بالعملات. كذلك، تشارك العقدة في عملية "إزالة البيانات المتكررة"، التي تُشبه عملية "التقصي" من حيث ضرورة أن تحدد الشبكة هوية العقدة الأولى التي يجب عليها تقديم بيانات الإثبات في عملية تبادل ما، وتجاهل بيانات الإثبات التي تقدمها العقدة الأخرى.

### طرق استخراج البيانات

تتوفر عدة طرق لاستخراج بيانات سجل التبادل. فيمكن للمتبادلين اعتماد واحدة من الطرق الآتية:

- تحميل سلسلة الكتل الخاصة بسجل التبادل، وبالتالي استخراج البيانات مجاًناً
- تخزين جميع التبادلات المنجزة في الذاكرة المحلية – ما سيتطلب تشغيل تطبيق Block DX بشكل دائم (عملياً، لا تفيد هذه الطريقة إلا لتحديث الرسوم البيانية بشكل فوري)
- طلب سجل التبادل من عقدة موجودة على سلسلة سجل التبادل، مقابل رسم:
  - لعمليتين محدّتين
  - لمدة محدّدة

### تقديم بيانات سجل التبادل

لتقديم سجل التبادل إلى الأطراف المتبادلة، يجب على العقدة الموجودة على سلسلة كتل سجل التبادل أن توفر:

- "دالة تجزئة" للعملة "أ" والعملة "ب" ورسوم التبادل ضمن النطاق الذي يحدده الطرف المتبادل
- عنوان العقدة

وباستخدام هذه البيانات، يمكن للطرف المتبادل بناء إثبات بسيط قائم على قاعدة "صفر-معرفة" (مع مراعاة الملاحظة أعلاه) حول موثوقية البيانات من خلال طلب سجل التبادل من عقدة موجودة على سلسلة كتل سجل التبادل، والتثبت من أنّ كل دالة تجزئة تم توفيرها تتوافق مع البقية. وفي حال كانت النتائج متطابقة، يعني ذلك أن احتمال أن تكون البيانات غير موثوقة ضئيل، لأنّه ما من حافز يدفع العقد إلى الوثوق ببعضها البعض، وبالتالي هي ليست في موقع جيد للتأمر معاً بهدف الاحتيال. لكن إذا أراد الطرف المتبادل الحصول على إثبات إضافي بأنّ البيانات موثوقة، يمكن له (أ) طلب سجل التبادل من "عقدة نوال" (mode node)، أو تحميل سلسلة الكتل. كذلك، يمكن لكل عقدة أن تراقب ردود غيرها من العقد على الأطراف المتبادلة ومعاينة العقد التي تقدم بيانات غير صحيحة عبر توفير إثبات حول عدم موثوقية هذه البيانات مقارنة بسجل سلسلة الكتل؛ لذا، من السهل للشبكة التوصل إلى إجماع في هذا الصدد حول عدم موثوقية إحدى العقد وإدراجها على القائمة السوداء.

إذا رضي الطرف المتبادل بالإجابة (ات) التي حصل عليها من عقد السجل (history nodes)، يختار العقدة الأولى التي تقدم دالة التجزئة، ويبدأ عملية مبادلة موحّدة معها تتميز بالخصائص التالية:

- معاملة مشاركة لا يمكن صرفها إلا باستخدام
  - مفتاح خاص يعود إلى العنوان الذي قُدمته العقدة
  - وبيانات سجل التبادل الذي يعود إلى دالة التجزئة التي تم تقديمها
- (بعبارة أخرى، تؤدي بيانات سجل التبادل وظيفة السرّ في المبادلة الموحّدة)

بالتالي، إذا صرفت عقدة سجل التبادل معاملة المشاركة، عليها إذاً كشف بيانات سجل التبادل، وعندئذ يتلقاها الطرف المتبادل. كذلك، لا يمكن كشف بيانات سجل التبادل إذا لم يسدّد الطرف المتبادل ثمنها.

وتجدر الإشارة إلى بعض ميزات المخطط الحالي (المبسّط):

## BLOCKNET

- قد يكون حجم مجموعة البيانات المطلوبة محدودًا بالمدى الأقصى المحدد لحقل "السر" (secret field length) في تصميم المعاملة. ويوفر ذلك عائدات إضافية إلى عقد سجل التبادل، لأنّ الأطراف المتبادلة تضطر إلى تقديم أكثر من طلب واحد إذا أرادت الحصول على بيانات التبادل الخاصة بفترة زمنية أطول.
- يمكن تعديل الرسم المفروض على مجموعة البيانات المطلوبة بشكل تفاعلي مقارنة بحجم التبادل، إذ كلما ازداد الوقت اللازم لـ "الحجم الوحدة" (volume per unit time)، قلت الفترة الزمنية لمجموعة البيانات التي يبلغ حجمها الحد الأقصى المحدد.
- يتم تشويش (obfuscation) بيانات سجل التبادل بشكل طفيف لمنع الأطراف المتبادلة الأخرى من اعتراض البيانات المطلوبة بعد أن تكشفها عقدة سجل التبادل: فنظرًا إلى عدم ذلك نوع العملات أو الفترة الزمنية في مجموعة البيانات المقدّمة، لن تستفيد الأطراف المتبادلة الأخرى منها، وسيكون من الصعب والمكلف جدًا تحليلها.
- ويمكن تشويش بيانات سجل التبادل بشكل أشدّ عبر إرسالها بشكل مشفّر من خلال XChat، ويصبح السر في المبادلة الموحدة في هذه الحالة مفتاح فك الشيفرة للحصول على البيانات. ولكن هذه الطريقة تتطلب إثبات صفر-معرفة أكثر تعقيدًا من ذلك المبين في المخطط (انظر أعلاه).

### خدمة السجل

يبدو أنّ خدمة سجل التبادل (trade history service) المذكورة أعلاه قابلة للتعميم لتوفير خدمة سجل (registry service) أوسع لخدمات ما بين السلاسل. وبالطبع، عندما يكون مضمون عملية التبادل منتجًا رقميًا لا عملة، تبقى مرحلتنا الإدخال والبحث (commit and lookup) في توجيه سلاسل الكتل كما هي. أمّا الأمر الإضافي الضروري فهو بحث عقد سجل التبادل في سلسلة الكتل الخاصة بها عن أحدث بيانات تظهر فيها هويات السلسلة، وجمع هذه الأخيرة في لائحة مع "هويات الخدمة" (serviceIDs) الخاصة بها. ويتم تقديم هذه اللائحة، عوضًا عن سجل التبادل، باستخدام البروتوكول المبيّن في القسم السابق.

## مراحل المشروع

يستعرض هذا الجزء، بشكل عام، كيفية تطور المشروع على المدى البعيد، وهو يهدف إلى قياس النطاق العام، لا إلى توفير مجموعة التزامات حول مراحل التطوير الأساسية. وسيتم نشر خرائط الطرق على المدى الأقرب والمراحل الأساسية بحسب الحاجة.

### توفير المنتج الفعّال الأساسي (MVP)

- عميل/عقدة موحدات (monolithic)
  - موجة سلسلة الكتل
  - بروتوكول XChat
  - آلية توليد الإيرادات من الخدمة (monetisation)
  - آلية توزيع رسوم التبادل
- التطبيق اللامركزي للتبادل اللامركزي
  - واجهة المستخدم الأمامية (frontend UI)
  - طلبات السوق والسعر المحدد والإيقاف
  - دفتر الطلبات
  - سجل الطلبات
  - حدود المستخدم المفتوحة
  - دمج رسوم TradingView البيانية
- تستخدم معلومات التعريف (credentials) الخاصة بواجهة برمجة التطبيق لكل حساب مستخدم
  - برنامج التثبيت (setup wizard): واجهة برمجة التطبيق الخاصة بالمحافظة الآلية وإعدادات واجهة برمجة التطبيق الخاصة بالرسوم البيانية
  - التحكم بالمخاطر (بشأن عدد التأكيدات المقبولة)
    - الاختيار من دفتر الطلبات
    - يسمح لكم بتقديم الطلبات في وقت أقصر
    - التحكم بالصرافة:
      - قبول الطلب؛ إعادة الصرافة
      - لا يمكن استخدام الصرافة لـx دقائق
      - تدخل في خانة المخاطر المناسبة (risk bracket)
      - يتم تحديث خانة المخاطر الخاصة بكل طلب بحسب عمر العملة

### المرحلة الثانية

- تقسيم xbridgep2p إلى وحدات
  - وحدة موجة سلسلة الكتل
  - وحدة XChat
  - وحدة تبادل العملات
  - عميل التبادل اللامركزي
- واجهات برمجة التطبيقات لكل الوحدات
- دعم مضمون البيانات (data payload) في بروتوكول التبادل

## BLOCKNET

- توافق تشغيلي سهل بين بروتوكول التبادل وبروتوكول xchat للتحويل (xchat transport protocol) (الذي يتم التحكم به من خلال التطبيق اللامركزي الخاص بكم)

### المرحلة الثالثة

- دعم أنواع إضافية من الطلبات: وقف الخسارة المتدرج (trailing stop)، طلب " عرض العملة فقط" ) (OCO
- دعم ترك الطلبات في دفتر بعد إغلاق التطبيق (حفظ الطلبات في سلسلة الكتل)

### المرحلة الرابعة

- تحسين البروتوكول: سوق مشتقات للمبادلات (من الند إلى الند للإقراض الهامشي)
- تحسين البروتوكول: أسواق مشتقات شاملة

## المواصفات التقنية

لتسهيل عملية الصيانة، وللحفاظ على مصدر واحد للإثبات للتوثيق المتدني المستوى، تم نقل هذا القسم إلى GitHub.

مراحل الرسائل

سوف يتم توفيقها. انظر <https://github.com/BlocknetDX/blocknet-docs>

مرجع واجهة برمجة التطبيق

انظر <https://github.com/BlocknetDX/BlockDX/blob/master/doc/dx/dxapi.md>

## حالات الاستخدام

لا شك أنّ تصميم البنية التحتية لنظام غير موجود بعد مهمة صعبة. فالسؤال الأكثر شيوعاً هو "ما الهدف من هذا التصميم؟"، والإجابة التي تقدّمها هي من قبيل "الهدف هو كل ما يمكنه الاستفادة من نظام عملات" – أي معظم الأشياء. ولتقديم أمثلة ملموسة عن هذه الإجابة، نورد في ما يلي لائحة قصيرة عن حالات استخدام بلوكنيت:

### 1. التبادل اللامركزي

يشكّل التبادل اللامركزي لل عملات المشفّرة إحدى خدمات بلوكنيت الأساسية، إذ إنّه ضروري لتوليد الإيرادات من أي خدمة أخرى.

ويشكل التبادل اللامركزي، المقرون بتطبيق لامركزي ذي واجهة استخدام سهلة، المنتج الأوّل من بلوكنيت الموجّه للمستهلكين، إذ إنّه يلبي حاجة المهتمّين بمجال التشفير الفعلية إلى تكنولوجيا التبادل اللامركزي.

يوذّي تقسّي عمليات القرصنة والاحتيال والفسل والسرقة في تبادلات العملات اللامركزية إلى [سرقة "بينكوين" واحدة من بين كل 16](#). لذا، من شأن التبادل اللامركزي الفعّال أن يؤمّن التبادل الآمن والموثوقة للعملات ضمن هذا النظام الحديث الولادة.

### 2. موجّه سلسلة الكتل

إنّ توجيه سلاسل الكتل من الخدمات الأساسية لبلوكنيت أيضاً، إذ من الضروري توجيه التعاملات ما بين السلاسل إلى وجهاتها الصحيحة. ولكن توجيه سلاسل الكتل مهم كخدمة مستقلة أيضاً، يمكن لأيّ عقدة استخدامها لتقديم أو استهلاك خدمات ما بين السلاسل. ويقدم موجّه بلوكنيت الأوّلي، أي XBridge، خدمة مجانية حالياً، وقد يبقى الحال كذلك إلى تاريخ غير محدد.

### 3. التراسل ما بين السلاسل

سواء تم استخدامه كتطبيق تراسل أو لتحويل البيانات، لا شك أن التراسل ما بين السلاسل خدمة لا غنى عنها في نظام العملات. فتماماً كما التبادل اللامركزي وتوجيه سلاسل الكتل، إنّ التراسل ما بين السلاسل من بين خدمات بلوكنيت الأساسية أيضاً، وهو يسمّى XChat. وهذه الخدمة مشفّرة بحسب مبدأ الطرفين، وهي من الند إلى الند، ويمكن استخدامها لتوصيل المنتجات والرسائل الرقمية بشكل آمن جداً. كذلك، فإن هذه الخدمة مجانية حالياً، وهي متوفرة (في الوقت الحالي) ضمن حزمة واحدة مع موجّه سلاسل الكتل في XBridge.

### 4. تطبيق هواتف ذكية يستفيد من عدة سلاسل

من المرجح أن يتضمّن التطبيق الهاتفي، بسبب حجمه الصغير، عقدة واحدة للتحقّق المبسّط من الدفعات فقط. لذا،

- سوف يستهلك هذا التطبيق خدمات، لا عملات أخرى
- سوف تتضمن خدمات سلاسل الكتل التي يستهلكها عناصر بلوكنيت
- عندما يطلب التطبيق خدمة ما، سوف تنتج الخدمة "سراً"، يكون في الوقت عينه مفتاحاً لفك شيفرة المنتجات الرقمية
- سوف ترسل الخدمة بيانات تمكّن التطبيق من بناء إثبات صفر-معرفة حول موثوقية المنتجات
- سوف تنشئ الخدمة معاملة مشاركة في مبادلة موحّدة

- سوف تصرف الخدمة معاملة المشاركة ثم تبادلها بعملة أخرى، إذا كان ذلك مفضلاً
- يتلقى التطبيق السر بهذه الطريقة ويمكنه استهلاك الخدمة

### 5. خلط العملات (coin mixing) بشكل شبه مثالي

يمكن دمج العملات الخاصة، مثل ZCash أو ZCoin أو Monero ضمن XBridge، كتابة برمجة التبادل الآلي لأي عملة بالعملة الخاصة ثم العكس. وبما أن التبادل اللامركزي لا يتطلب الوثوق بأي طرف ثالث لإيداعه بيانات المستخدم، ونظرًا إلى أنّ المبادلة الموحدة لا تطرح أي مخاطر على الأطراف المتبادلة، فالنتيجة هي خدمة خلط خاصة وشبه مثالية.

### 6. تطبيق سوق لامركزي

تتطلب تطبيقات السوق (marketplace app) عادةً الخدمات التالية: (أ) سمعة العميل ومعلوماته، (ب) ومعالجة الدفعات، (ج) وتخزين الصور، (د) وإدراج العناصر. ويوصى باعتماد هندسة الخدمات الصغرى (microservices architecture) في هذا التطبيق، للأسباب المذكورة أعلاه، ما يسمح باستخدام عدة سلاسل كتل. في هذه الطريقة، يمكن لسلسلة واحدة أن تخزن معلومات العميل المشفرة (انظر البند 13 على هذه اللائحة)، واستخدام XBridge لقبول الدفعات بأي عملة مشفرة، وتخزين الصور على خادم، واستخدام سلسلة ثالثة وبرمجية ضمن المحفظة (in-wallet code) لإدراج العناصر ومكونات واجهة الاستخدام (UI). أما النتيجة فهي تحقيق قابلية التوسع والميزة التركيبية ضمن مجموعة خدمات يسهل تصحيح الأخطاء البرمجية فيها (bugfix) وتحديثها أو استبدالها.

### 7. محوّل الوقود (fuel-converter) لعقود "إثيريوم" (Ethereum) الذكية

من خلال التبادل اللامركزي، يمكن تزويد أي عقد "إثيريوم" (Ethereum) بـ"الغاز" (gas) من أي عملة أخرى.

### 8. عملة ثابتة (stablecoin) لامركزية بالفعل

يمكن لعملة ثابتة أن تحافظ على ثباتها عبر الاستفادة من واقع أنّ سجلات التبادل اللامركزي موجودة على سلسلة. بالتالي، تتوفر مجموعة بيانات يمكن إثبات موثوقيتها لتحديد ما إذا كان يجب إنتاج (mint) أو حرق (burn) العملات (أو تجميدها ووقف تجميدها) بهدف الحفاظ على ثباتها.

### 9. هوية ذاتية التنظيم (self-sovereign ID) ومنظم المعلومات الشخصية

قد تحفظ خدمة المعلومات الشخصية البيانات الوصفية الشخصية المشفرة على سلسلة ما تضمّ نظام أذونات قابل للإلغاء (revocable permissioning system). لذا، فالمستخدمون قادرون على تنظيم معلوماتهم الشخصية ذاتيًا. وانطلاقًا من هذه الميزة، يمكن للمستخدم دمج سلسلة الكتل هذه في أي موقع إلكتروني أو تطبيق يتطلب تسجيل دخول، كما يمكن له بيع بياناته الوصفية بشكل طوعي إلى المعالنين مقابل دفعات صغرى (micropayments)، كما يمكن دعم أنظمة جواز السفر/الهوية. أما التكنولوجيات الناشئة التي من المتوقع أن تستفيد من حالة الاستخدام هذه فهي Bitnation وإطار عمل Coco الخاص بشركة "مايكروسوفت" (Coco Framework).

### 10. حل سلسلة التوريد 2.0



إن بنية بلوكنيت التحتية مصممة بشكل مناسب لكي تشكل العمود الفقري لـ "سلسلة التوريد 2.0". فالأطراف المتبادلة غالباً ما تجد نفسها على سلاسل كتل مختلفة، وهي تحتاج إلى ضمان التوافق التشغيلي في ما بينها، ويمكنها القيام بذلك من خلال خدمات بلوكنيت. بالتالي، سوف تتمكن تطبيقات السلاسل المتعددة (multichain apps) من قراءة البيانات من عدة سلاسل، سواء أكانت مختصة ببيانات الشحن مثل بوليصة الشحن، أو بيانات صناعة المنتجات مثل قائمة المواد، أو البيانات المالية أو غيرها. وعبر مقارنة البيانات الوصفية من عدة مصادر، يمكن لبلوكنيت تمكين الشركات من الحد من نقاط المهاجمة (attack vectors)، مثل تزوير الفواتير (invoice spoofing) وتزوير الشهادات (certification counterfeiting).

### 11. البنية التحتية لـ "إنترنت الأشياء"

يمكن حل مشاكل أمنية لطالما كانت "إنترنت الأشياء" تعاني منها من خلال تكنولوجيا سلسلة الكتل، وتحقيق التوافق التشغيلي بين آلاف سلاسل الكتل على بلوكنيت. كذلك، تتوفر فرص متنوعة لتوليد الإيرادات من البيانات المنفردة (granular monetisation): فعلى سبيل المثال، يمكن تشكيل حزم معاملات على عدة سلاسل في الوقت نفسه باستخدام محفظات التحقق المبسط من الدفعات (SPV). بالتالي، يمكن ترميز (tokenize) دفق البيانات وتحفيز العقد على العثور على الأنماط في البيانات الضخمة لشركة ما.

### 12. خدمة الإعلانات ضمن التطبيق

يمكن للمستخدمين جني العملات من خلال التطبيق الهاتفي عبر عرض الإعلانات المعروضة في التطبيق، وذلك من ضمن الخدمات التي توفرها بلوكنيت. ويمكن استخدام هذه العملات لتحفيز استهلاك خدمات ما بين السلاسل في التطبيق، ما يوفر خدمة "مجانية" للمستخدم ويسمح في الوقت نفسه لمقدمي الخدمة بتوليد الإيرادات.

### 13. حل التخزين اللامركزي من الند إلى الند

يمكن توسيع قاعدة المستخدمين الخاصة بحلول التخزين القائمة على سلاسل الكتل، مثل Storj، وتوليد الإيرادات منها من خلال توفير خدمات ما بين السلاسل.

### 14. منصة لعرض العملة الأولى (ICO) لا تحتاج إلى أدونات

يمكن لأي كان عرض بيع العملات على نظام التبادل اللامركزي، من دون الحاجة إلى إنترنت.

### 15. أداة حالة العمل لإدارة الميزانية الموزعة

تتطلب المشاريع التشفيرية عادةً كحالة عمل كبيرة قائمة على التعهيد الجماعي (ICO)، حيث يتم التفاوض على الميزانية في السوق. غير أن رصيد الحساب الفعلي يتقلب مع تغير قيمة سعر العملة التشفيرية التي أجري بها التعهيد الجماعي. وباستخدام بلوكنيت، يمكن للمطورين إدارة توزيع العملات والحسابات عبر السلاسل. بالإضافة إلى ذلك، يمكن، عبر استخدام العقود الذكية، إدارة الدفعات والاستثمارات بعملات أخرى، وبرمجة خطة الأعمال وتنفيذها بطريقة بشكل آلي عموماً من خلال العقد بشفافية مطلقة.

## BLOCKNET

### 16. الدمج مع أنظمة "تخطيط موارد المؤسسة" (ERP) و"إدارة علاقات العملاء" (CRM) و"إدارة دورة حياة المنتجات" (PLM) عبر الشركات

يحقق الدمج البسيط القائم على واجهة برمجة التطبيق الذي توفره بلوكنيت التوافق التشغيلي إما مباشرة أو بشكل غير مباشر مع سلاسل الكتل الخاصة وتلك الأشبه بـ"اتلافات"، مثل ORACLE و SAP.

### 17. بنية تحتية لإنترنت القيمة

سوف تعمل البنية التحتية ما بين السلاسل الخاصة ببلوكنيت بشكل متزايد مع مرور الوقت لإنشاء "إنترنت القيمة" (Internet of value)، تكون موثوقة وشفافة ومُتاحة بشكل عادل. وإذ تبدأ الدفاتر العامة والفرعية (general and sub-ledgers) للشركات بالتفاعل تدريجياً مع دفاتر الشركات الأخرى من خلال سلاسل الكتل، سوف تصبح شبكة سلاسل الكتل التي تنشأ جراء ذلك انعكاساً لتدفقات القيمة ولقيمة كل نظام. ومن شأن ذلك تحقيق الوعي المتقدم والمعتمق حول القيمة عبر الأنظمة برمتها، مع آثار شديدة وواسعة النطاق تطل النظام المالي.

سلسلة الكتل	بلوكشين	Blockchain
شبكة الكتل	بلوكنيت	Blocknet
حالات الاستخدام		use-cases
توجيه سلسلة الكتل		blockchain routing
عقد الخدمة		service nodes
نظام مطابقة الطلبات		order matching system
البنية التحتية لخدمة ما بين السلاسل		inter-chain infrastructure
"خدمات" ما بين السلاسل		inter-chain services
نظام العملات		token ecosystem
التطبيقات اللامركزية		dapps
التوافق التشغيلي العملي ما بين سلاسل الكتل		inter-blockchain interoperability
البرمجيات كخدمة		software-as-a-service
استخدام سلسلة الكتل عملياً		practical blockchain usability
عقد		nodes
تتبع السلسلة		chain bloat
إنترنت سلاسل الكتل		internet of blockchains
الأنظمة التكنولوجية الأساسية		technology stack
الدلائل التشفيرية		cryptographic proof
الند إلى الند		peer-to-peer
عملة للتطبيقات		appcoin
عملات		tokens
الخدمات الصغرى		microservices
الأخطاء البرمجية		bugs
ما بين السلاسل		inter-chain
خدمات السلاسل المتعددة		multi-chain
اقتصاد التشفير		cryptoeconomy
مكافآت الكتل		block rewards
أنظمة الكتل الكبرى الذاتية التمويل		superblock self-funding systems
التوافق التشغيلي		Interoperability
عملة بيتكوين	بيتكوين	Bitcoin
بوابات الدفع		payment gateways
الدقة التشغيلية		determinacy of operation
محفظات الأصول		stock wallets
الميزة التركيبية		composability
وحدة مترابطة موزعة		distributed monolith
القدرة على توليد المال		monetisability
أنظمة المعلوماتية البعيدة		telematics
مكامن الضعف		attack surface
لاوساطية		disintermediated
القصوية		maximalists
الإضافات		bolt-ons

## BLOCKNET

الحديقة المسورة		walled garden
عقد التحقق المبسط من الدفعات		SPV nodes
موقع إلكتروني يستكشف سلاسل الكتل		blockchain explorer website
وسيط مركزي		centralized oracle
الحياد تجاه السلاسل		chain-agnostic
تراكب الشبكة ما بين السلاسل	inter-chain network overlay	XBridge
موجهة سلاسل كتل	blockchain router	XName
تحويل البيانات من الند إلى الند	p2p data transport	XChat
السعر-مقابل-الموثوقية		price-to-truthfulness
هوية السلاسل		chainIds
البرمجة الثابتة		hardcoded
الحصول على خدمة سجل		getRegistryService
الرسائل العامة		broadcast messages
برمجيات السلاسل		chain codes
خوارزميات الإجماع الخاصة بسلاسل الكتل		blockchain consensus algorithm
بشكل موحد		atomically
صفر معرفة		zero-knowledge
دالة هاش الأصلية		hash preimage
الوحدة		atomicity
تراكب الشبكة		network overlay
نظام تركيبي		modularized
تطبيق التبادل اللامركزي		Block DX
هجمات الابتزاز القائمة على التطويق		malleability-based extortion attack
بيتكوين-ومستنسخاتها		Bitcoin-and-clones
دالة تجزئة		hash function
مشاركة		bail-in
المقايضة الموحدة		atomic swap
الرسائل غير المرغوب فيها		spam
تجريد		abstraction
نشر الطلب		order broadcast
ملاءمة الطلب		order matching
مسألة الجنرال البيزنطي		Byzantine Generals' problem
تقضي		mining
مطالبة		Staking
المنقّصين		Miners
للاحتيال بقصد الربح		Frontrunning
غرامة الأداء		performance penalty
نظام إثبات العمل		hashcash (proof-of-work system)
التآمر بين عقد الخدمة		service node collusion
معاملة الاسترجاع		refund transaction
تكلفة الفرصة البديلة		opportunity cost
التنظيم الذاتي		self-sovereignty

## BLOCKNET

مكافحة الطلبات غير المرغوب بها		anti-spam
كآلة حالة		state machine
مطالبة على القائمة السوداء		blacklist proof-claim
المشارك الفائز التالي		next winning staker
محفظات العقد الكاملة		full node wallets
التبادل من الداخل		insider trading
معاملة رسم مكافحة الطلبات غير المرغوب بها		Txspamfee
المفتاح العام		Pubkey
بروتوكول ملائمة		matching protocol
معاملة رسم مكافحة هجمات الحرمان من الخدمة		txDOSfee
معاملة المشاركة		bail-in transaction
حالات استهلاك السيولة		liquidity consumption events
آلة حالة الطلبات		order state machine
إنتاج الصرافة بهدف الاحتيال		malicious creation of change
نظام إثبات قائم على مبدأ صفر-معرفة		zero-knowledge proof scheme
الشمعة اليابانية		Candle
عقدة منوال		mode node
عقد السجل		history nodes
المدى الأقصى المحدد لحقل السر		secret field length
الحجم الوحدة		volume per unit time
تشويش		Obfuscation
خدمة سجل التبادل		trade history service
دمة سجل		registry service
الإدخال والبحث		commit and lookup
هويات الخدمة		serviceIDs
عميل/عقدة موحدات		monolithic
موجة سلسلة الكتل		blockchain router
آلية توليد الإيرادات من الخدمة		monetisation
واجهة المستخدم الأمامية		frontend UI
معلومات التعريف		credentials
برنامج التثبيت		setup wizard
خانة المخاطر		risk bracket
مضمون البيانات		data payload
وقف الخسارة المتدرج		trailing stop
التراسل ما بين السلاسل		inter-chain messaging
الطرفين		end-to-end
خط العملات		coin mixing
تطبيقات السوق		marketplace app
هندسة الخدمات الصغرى		microservices architecture
برمجية ضمن المحفظة		in-wallet code
تصحيح الأخطاء البرمجية فيها		bugfix

## BLOCKNET

محوّل الوقود		fuel-converter
	إثيريوم	Ethereum
	غاز	gas
عملة ثابتة		stablecoin
إنتاج		mint
حرق		burn
هوية ذاتية التنظيم		self-sovereign ID
نظام أذونات قابل للإلغاء		revocable permissioning system
دفعات صغيرة		micropayments
		Bitnation
		Coco
تطبيقات السلاسل المتعددة		multichain apps
نقاط المهاجمة		attack vectors
تزوير الفواتير		invoice spoofing
تزوير الشهادات		certification counterfeiting
توليد الإيرادات من البيانات المنفردة		granular monetisation
ترميز		tokenize
إنترنت القيمة		Internet of value
الدفاتر العامة والفرعية		general and sub-ledgers

المنتج الفعّال الأساسي	Minimum Viable Product	MVP
واجهة برمجة التطبيق	Application Programming Interface	API
الشبكات المحلية	Local Area Network	LANs
عرض العملة الأولي	Initial Coin Offering	ICO
اعرف عميلك	Know Your Client	KYC
إنترنت الأشياء	Internet of Things	IoT
جدول دالة تجزئة موزع	Distributed Hash Table	DHT
نظام أسماء النطاقات	Domain Name System	DNS
بروتوكول إنترنت	Internet Protocol	IP
إجراءات مكافحة هجمات الحرمان من الخدمات	Denial of Service	DoS
ناتج المعاملة غير المنفقة	Unspent Transaction Output	UTXO
تجربة المستخدم	User Experience	UX
هوية المعاملة	bitcoin transaction ID	txid
عرض العملة فقط	Only Coin Offering	OCO
واجهة المستخدم	user interface	UI
إدارة دورة حياة المنتجات	Product Lifecycle Management	PLM
إدارة علاقات العملاء	Customer Relationship Management	CRM
تخطيط موارد المؤسسة	Enterprise Resource Planning	ERP