



Bretton Woods *digital* Gold

Whitepaper
Fassung 26.09.2023-EN

Inhaltsverzeichnis

Einführung	3
Gold für die Preisstabilität des Tokens	3
Projektidee	4
Dienstleistungen	6
Technische Details	8
Token-Implementierung	9
BWGToken-Vertrag	10
StorageOperator-Vertrag	11
TokenTransferOperator-Vertrag	11
HotwalletTransferer Vertrag	12
StorageOperator-Umstrukturierung	13
Abzug von Goldlagerkosten (Batch-Verfahren)	14
Rollenbasierte Zugangskontrolle	16
Gnosis Safe	17
BWG Token Vertrag Übersicht	18
BWG-Token-Ereignisse	21
StorageOperator-Vertragsübersicht	22
StorageOperator Ereignisse	25
TokenTransferOperator Ereignisse	26
HotWalletTransferer Vertragsübersicht	26
HotWalletTransferer Ereignisse	27
Berichte über die Berichterstattung	27
Haftungsausschluss	28

Einführung

Der Bretton Woods *digital* Gold Token (BWG) kombiniert die Blockchain-Technologie mit dem Edelmetall Gold. Der fungible Token ist zu 100% durch Gold gedeckt, das in einem Schweizer Zolffreilager gelagert wird.

Diese Bindung an das Gold bzw. den Goldpreis macht das BWG Token preisbeständig.

Gold für die Preisstabilität des Tokens

Historisch gesehen hat sich Gold als finanzielle Absicherung in unsicheren Zeiten bewährt. Die Preisstabilität des Edelmetalls lässt sich hervorragend als Inflationsschutz und zur Portfoliodiversifizierung nutzen.

Zukunftssichere Technologie

Der Token basiert auf dem Ethereum-Token-Standard ERC-777, der eine Weiterentwicklung des beliebten ERC-20-Standards ist und im Gegensatz zum ERC-20-Token verbesserte Funktionalitäten und volle Abwärtskompatibilität bietet. Durch die Veröffentlichung auf der Binance Smart Chain kann der BWG Token im Vergleich zu den meisten anderen Token kosten- und energieeffizienter gehandelt werden.

Gründe für den Bretton Woods *digital* Gold Token BWG

BWG Token , als tokenisiertes Gold, bietet die Möglichkeit, beliebig kleine Investitionen zu tätigen. Der Handel funktioniert in Echtzeit und kann weltweit zu jeder Tageszeit durchgeführt werden. Zugleich ist der Token ein Schritt in eine nachhaltige Zukunft. Nicht nur zur Verwahrung von Vermögenswerten, sondern auch zur Absicherung gegen instabile Währungen ist das BWG Token eine hervorragende Möglichkeit, seine Investitionen und Vermögenswerte abzusichern.

Mit dem BWG einen Schritt in die Zukunft wagen

Wir unterstützen Sie bei der sicheren Verwahrung des physischen Goldes und sorgen dafür, dass Sie jederzeit digital auf Ihr Gold zugreifen können. Durch die Abbildung auf der Binance Smart Chain ist es energieeffizienter und ressourcenschonender als herkömmliche Systeme und andere Blockchain-Netze - machen Sie mit uns den ersten Schritt in eine nachhaltige Zukunft.

Projektidee

Seit Jahrtausenden hat Gold für die Menschen eine besondere Anziehungskraft und Bedeutung. Neben der Verwendung als Schmuck oder Luxusgut dient es auch als Wertanlage.

Gerade Krisen haben gezeigt, dass Anleger in unsicheren Zeiten verstärkt Gold kaufen. Entscheidend ist dabei das Vertrauen in das Edelmetall in Krisenzeiten, denn Gold hat sich in der Vergangenheit als finanzielle Absicherung gegen den Verlust von Vermögenswerten bewährt, aber auch als Absicherung gegen Inflation aufgrund der weltweit stark gestiegenen Energiepreise.^{1,2}

Dabei stellen sich die Anleger die folgenden Fragen:

1. Wo kann ich schnell und einfach Gold kaufen?
2. Ist die Anlage in Gold abgesichert?
3. Wo wird das Gold gelagert?

Die Antwort ist der Bretton Woods *digital* Gold Token. Dieser kombiniert die Preisstabilität von Gold mit den technischen Vorteilen der Blockchain.

Die Idee des BWG

Durch die Kombination des klassischen physischen Assets Gold in Verbindung mit der Blockchain-Technologie ergeben sich verschiedene Vorteile. Die beschriebenen Vorteile von Gold bleiben auch in tokenisierter Form erhalten, jedoch kann z.B. die eingeschränkte Fungibilität aufgehoben werden. Das Ergebnis ist eine einfache und effiziente Möglichkeit, in Gold zu investieren. Die Preisstabilität des Tokens ergibt sich aus der Hinterlegung mit physischem Gold. Bei der Ausgabe des BWG-Tokens entspricht ein Token einem Gramm Gold. Um in den Genuss von Gold zu kommen, wird lediglich eine Krypto-Wallet (wie MetaMask) benötigt. Der manchmal komplexe Prozess, der mit dem Erwerb von physischem Gold verbunden ist, wird stark vereinfacht. Der Käufer des BWG wird Eigentümer des im Zollfreilager deponierten Goldes und muss sich nicht um dessen Verwahrung oder Transport an einen sicheren Ort kümmern. Die Lagerung in einem Schweizer Zollfreilager inklusive Zertifizierung ist im BWG enthalten.

Ein weiterer Vorteil ist, dass Gold aufgrund seines digitalen Charakters in beliebig³ kleine Mengen aufgeteilt und in Echtzeit, grenzüberschreitend und zu jeder Tageszeit gehandelt werden kann. Bei derartigen Investitionen und Transaktionen stoßen die Anleger beim Kauf von physischem Gold in der Regel auf Probleme.

¹ Aye, G.C. et al. (2017): Does gold act as a hedge against inflation in the UK? Evidence from a fractional cointegration approach over 1257 to 2016, Resources Policy, 54, pp. 53-57, doi: 10.1016/j.resourpol.2017.09.001.

² Federal Reserve Bank St. Louis (n.d.): Global price of Energy index (PNRGINDEXM). <https://fred.stlouisfed.org/series/PNRGINDEXM>, Zugriff am 23.02.2022.

³ Beliebig bedeutet in diesem Fall 18 Dezimalstellen. Dies ist der Standardwert, der für das ERC-20-Token verwendet wird. Weitere Informationen unter: [https:// docs.openzeppelin.com/contracts/3.x/ERC-20](https://docs.openzeppelin.com/contracts/3.x/ERC-20).

Die Preisstabilität von Gold kann durch das BWG Token in instabilen Ländern und Entwicklungsländern als attraktives Instrument zur Absicherung gegen Hyperinflation und Instabilität genutzt werden. Das BWG Token bietet sich auch als Alternative zu anderen Stablecoins an. Laut Coinmarketcap sind diese in fast allen Fällen an den US-Dollar gekoppelt⁴. Damit stellt sich die Frage, was im Falle eines Wertverlustes des US-Dollars passiert. Um diesem potenziellen Problem entgegenzuwirken, bietet BWG eine Alternative zur Bindung an den US-Dollar. BWG ist bei einem Wertverlust des US-Dollars nicht betroffen, da es an den Goldpreis gekoppelt ist.

BWG als Investitionsschutz

Die Entstehung von Währungssystemen hat seit dem 19. Jahrhundert mehrere Phasen durchlaufen. Dazu gehörten beispielsweise der Goldstandard, Bretton Woods und das Kingstener Währungssystem, das noch heute existiert.

Traditionell sind Geldsysteme um einen festen "Anker" herum aufgebaut. Jedes Zahlungsinstrument im Geldsystem ist letztlich an einen festen Betrag dieses Ankers gebunden. Der Anker kann viele Formen annehmen, wie z. B. die Bindung an ein Edelmetall oder eine Fiat-Währung. Während des Goldstandards war dies zum Beispiel Gold. Das bedeutete, dass jede von einer Regierung ausgegebene Währungseinheit in eine Goldeinheit konvertierbar war. Im Rahmen von Bretton Woods hielt dieser Anker sogar das gesamte internationale Währungssystem zusammen.

Alle teilnehmenden Länder stimmten festen Wechselkursen zum US-Dollar zu, und die Federal Reserve verpflichtete im Gegenzug die Zentralbanken aller teilnehmenden Länder, Dollar zu einem festen Kurs von 35 Dollar pro Feinunze in Gold zu tauschen. Derzeit ist der Anker in den meisten Währungssystemen eine von einer Regierung ausgegebene Fiat-Währung. Wachsende Ungleichgewichte in der Weltwirtschaft, Handelskriege, die Volatilität der Finanzmärkte, politische und soziale Spannungen in vielen Regionen der Welt sowie Kriege führen zu einer ständigen Instabilität der Währungen und Märkte. Das deutlichste Beispiel ist die Corona-Pandemie. Seit 2021 steigen die Preise stetig an. Aufgrund der strengen Beschränkungen und der Versorgungsengpässe konnte die Wirtschaft die Nachfrage in letzter Zeit nicht mehr decken. So erlitt der S&P 500 zu Beginn der Pandemie eine starke negative Korrektur.⁵

All diese Probleme führten in vielen Regionen der Welt zu Instabilität und großer Unsicherheit bei den Investoren.

Der Anstieg der Goldnachfrage während der Corona-Pandemie verstärkt die Wahrnehmung, dass Gold für Anleger weiterhin ein Zufluchtsort und Ankerwert ist.⁶

⁴ Coinmarketcap (n.d.): Top stablecoin tokens by market capitalization, <https://coinmarketcap.com/de/view/stablecoin/>, Abgerufen am 23. Februar 2022.

⁵ Der S&P 500 ist ein Aktienindex. Visual Capitalist (2020): Veränderung der Performance des S&P 500 während der COVID-19-Pandemie im Vergleich zu früheren großen Crashes, Stand: August 2020, <https://www.statista.com/statistics/1175227/s-and-p-500-major-crashes-change/>, abgerufen am 23. Februar 2022.

⁶ World Gold Council (n.d.): Statistiken zu Goldangebot und -nachfrage, <https://www.gold.org/goldhub/data/gold-supply-and-demand-statistics>, abgerufen am 23. Februar 2022.

Dienstleistungen

Verwahrung Ihres Goldes



Wir garantieren die Sicherheit des hinterlegten Goldes. Jede Wertmarke wird mit zertifiziertem Gold in einem Zollfreilager in der Schweiz hinterlegt.

Unkomplizierte Verarbeitung



Mit einem einfachen Knopfdruck können Sie Gold¹ kaufen oder verkaufen, ohne sich Gedanken über Lieferung, Lagerung, Absicherung und Bearbeitungsgebühren zu machen.

Zugang zu Ihrem Vermögen



Sie haben jederzeit Zugriff auf Ihre BWG-Token, da die Token in Ihrer eigenen Wallet gespeichert sind. Darüber hinaus können Sie auf unserer Plattform jederzeit verschiedene Informationen zu Ihrer angeschlossenen Wallet und Ihrem Konto einsehen.

Audit des Goldes durch regelmäßige Audits des Goldes



Wir werden regelmäßig Audits durchführen, um die Quantität und Qualität des gelagerten Goldes, mit dem die Token unterlegt sind, zu überprüfen. Wir werden die Prüfberichte auf unseren Websites veröffentlichen.

Rückkauf von BWG-Token



Die Token können vom Kunden auf der Plattform gegen FIAT-Währung zurückverkauft werden. Der Verkaufspreis basiert auf dem aktuellen LME (London Metal Exchange) zuzüglich einer Bearbeitungsgebühr.

¹ Der Verkauf Ihrer BWG-Token auf der BWG-Plattform wird bald verfügbar sein.

Ausgabe und Verkauf von Token

Um zu gewährleisten, dass nur durch Gold gedeckte Token im Umlauf sind, werden nur so viele Token über die Plattform verkauft, wie physisches Gold im Schweizer Zollfreilager gelagert ist. Bis neues Gold erworben, gelagert und zertifiziert ist, kann es daher in Zeiten hoher Nachfrage zu kurzen Verzögerungen beim Verkauf kommen.

Beim Kauf von BWG-Token über die Plattform werden die Token unmittelbar nach Zahlungseingang an die angeschlossene, nicht depotgebundene Wallet des Kunden gesendet. Somit stellt der Kauf von BWG eine Übertragung des Eigentums an dem hinterlegten Gold dar.

Um den gesetzlichen Anforderungen, z. B. im Bereich der Geldwäsche, gerecht zu werden, müssen die Kunden zunächst den KYC-Prozess (Know Your Customer) durchlaufen. Die Kunden müssen gültige Ausweisdokumente hochladen und der KYC-Anbieter prüft, ob der Kunde zum Kauf von Token berechtigt ist.

Die für das Gold anfallenden Lagerkosten werden automatisch per Smart Contract aus der Wallet des Kunden beglichen. Die Lagerkosten belaufen sich auf ca. 0,22% der gehaltenen Token pro Monat.

Die Kosten können je nach den Vertragsbedingungen mit dem Zolllager leicht variieren. Die Plattform erhebt über einen Smart Contract automatisch Gebühren in Höhe von 0,66% pro Quartal von der Wallet des Kunden, um die technische Funktionalität sicherzustellen.

Technische Details

Smart Contract

Smart Contracts sind einfach Programme, die auf einer Blockchain gespeichert sind und ausgeführt werden, wenn vorher festgelegte Bedingungen erfüllt sind. Smart Contracts für Token sind nicht nur für die Erstellung von Token verantwortlich, sondern auch für die Abwicklung von Transaktionen und die Verfolgung des Guthabens der einzelnen Token-Inhaber. Um Token zu erhalten, muss man einige ETH/BNB an den Token Contract senden und erhält im Gegenzug Token zugeteilt.

ERC-20 Token-Standard

ERC-20 ist der technische Standard für fungible Token, die mit der EVM-Blockchain erstellt werden. Der Standard ermöglicht die Implementierung einer Standard-API für Token innerhalb von Smart Contracts. Dieser Standard bietet grundlegende Funktionen für die Übertragung von Token sowie für die Freigabe von Token mit niedrigem Wert, damit sie von einer anderen dritten Partei auf der Blockchain ausgegeben werden können.

ERC-777 Token-Standard

Dieser Standard definiert eine neue Art der Interaktion mit einem Token Contract und ist gleichzeitig rückwärtskompatibel mit ERC-20. Er definiert erweiterte Funktionen für die Interaktion mit Token. Er ermöglicht es Betreibern, Token im Namen einer anderen Adresse, eines Contracts oder eines regulären Kontos zu versenden.

BWG-Token-Contract

Bretton Woods Digital Gold (BWG) ist ein ERC-777-Standard-Token. Dieser Token wird in der EVM-basierten Binance Smart Chain eingesetzt. Der BWG-Token ist durch physisches Gold gedeckt. Dieses Gold haben wir gekauft und sicher gelagert.

Im Gegensatz zu ähnlichen Projekten basiert unsere Deckung nicht nur auf Optionen zum Kauf von Gold. Stattdessen sichert tatsächlich vorhandenes physisches Gold die Token.

Alle Tokeninhaber müssen vierteljährlich Lagerkosten für das Gold bezahlen. Diese Kosten beziehen sich auf das Gold, das die von ihnen gehaltenen Token absichert.

Warum wir uns für den ERC-777-Standard entschieden haben

BWG ist ein goldgedeckter Token, bei dem 1 BWG-Token 1 Gramm Gold entspricht, so dass der Token-Inhaber für die Lagerungskosten aufkommen muss. Der ERC-777-Standard verfügt über eine Standardoperatorfunktionalität; der Standardoperator kann Token im Namen einer anderen Adresse versenden. Wir verwenden den ERC-777-Standardoperator, um die Lagerkosten vierteljährlich von den Konten der Token-Inhaber abzuziehen; der Smart Contract Anpassungen des Abzugszeitraums zulässt.

Token Implementierung

Wir haben die folgenden drei grundlegenden Smart Contracts implementiert, um diesen ERC-777 -Token und die Funktionalität für den Lagerkostenabzugsprozess zu erstellen.

1. BWGToken Contract
2. StorageOperator Contract
3. TokenTransferOperator Contract

Wir haben einen weiteren Smart Contract implementiert, der darauf abzielt, automatisch eine erforderliche Mindestmenge an Token auf der Hotwallet der Bretton Woods Gold Plattform zu halten, indem wir Chainlink-Keeper verwenden. Damit soll die Sicherheit der Hotwallet verbessert werden, indem die Menge der dort befindlichen Mittel zu einem bestimmten Zeitpunkt begrenzt wird.

4. HotwalletTransferer Contract

BWGToken Contract

Der BWG-Token ist mit dem ERC-777-Standard implementiert und kann in EVM-basierten Blockchains eingesetzt werden. Die grundlegenden Merkmale des Vertrags sind:

1. Übertragung von Token von einem Konto auf ein anderes. (ERC-20 Standard)
2. Ermittelt den aktuellen Token-Saldo eines Kontos. (ERC-20-Standard)
3. Ermittelt den Gesamtbestand der im Netzwerk verfügbaren Token. (ERC-20-Standard)
4. Genehmigung, ob eine Menge von Token von einem Konto durch ein Drittkonto ausgegeben werden kann. (ERC-20-Standard)
5. Übertragen von Token auf die Empfänger-Wallet, ohne eine Bestätigung vom Token-Inhaber zu erhalten, indem Sie den Standard-Operator verwenden. (ERC-777-Standard)
6. Wir verwenden den Standardoperator als Smart Contract (anstelle einer regulären Wallet) namens `TokenTransferOperator` und überschreiben die Funktion `revokeOperator`, damit der Token-Inhaber die Standardoperatoren nicht widerrufen kann. Dies ermöglicht es uns, den Mechanismus zum Abzug der Lagerkosten zu sichern. (Benutzerdefinierte Funktion)
7. Alle Token-Inhaber werden als `storageWallet`-Liste im `_beforeTokenTransfer`-Hook gespeichert, um die Lagerkosten abzuziehen. (Benutzerdefinierte Funktion)
8. Verwalten einer `costfreeWallet`-Liste, die keine Abzüge für Lagerkosten zulässt. Die kostenfreien Wallets werden nur für interne Wallets der BrettonWoods *digital* AG verwendet, um den Abzug von Lagerkosten von internen Wallets zu verhindern und somit unnötige Transaktionsgebühren zu vermeiden. (Benutzerdefinierte Funktion)
9. Autorisieren des `StorageOperator`-Contracts. (Benutzerdefinierte Funktion)
10. Wir überschreiben `_callTokensToSend` und `_callTokensReceived` mit einem leeren Body, so dass die Sende- und Empfangshooks nicht ausgeführt werden. (Benutzerdefinierte Implementierung)
11. Die Rollen `admin` und `maintainer` sind berechtigt, diesen Vertrag zu aktualisieren. (Zugangskontrolle)
12. Überschreiben von `revokeRole` und `renounceRole`, um Admin-Benutzer einzuschränken. (Zugriffskontrolle)

StorageOperator Contract

Der StorageOperator-Contract ist ein Art Verwaltungsvertrag, der für die Abrechnung der Lagerkosten der Token auf Basis monatlicher / vierteljährlicher / halbjährlicher Abrechnungen zuständig ist. Die grundlegenden Merkmale des StorageOperator Contracts sind die folgenden.

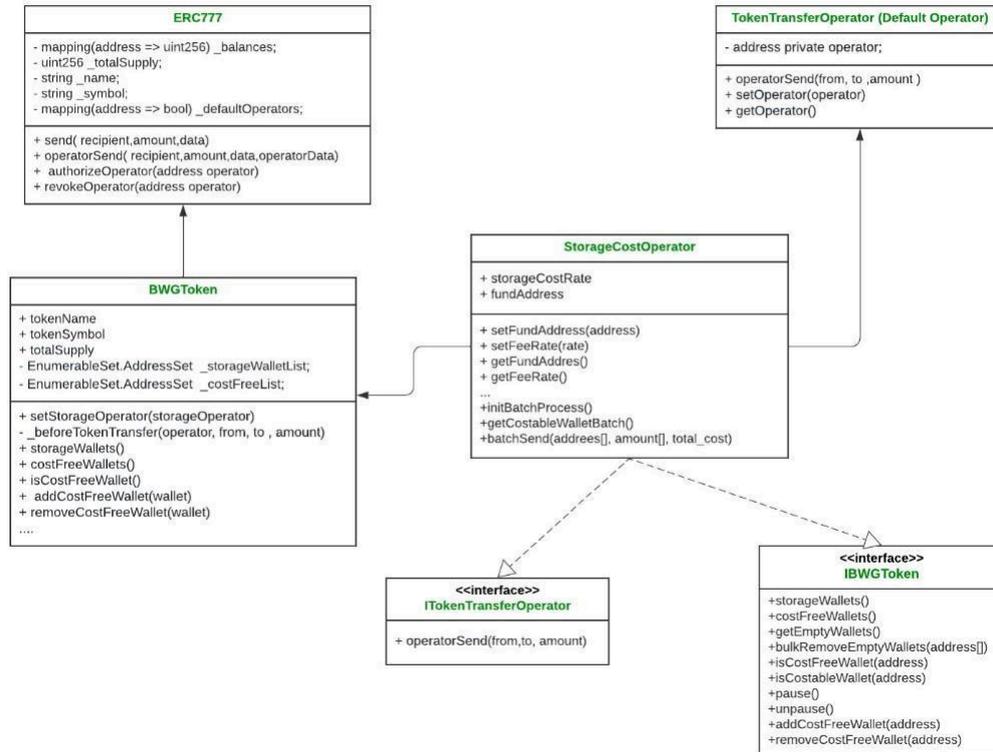
1. Der Contract speichert die Zustände fundAddress, batchSize, storageCostRate, transactionFee, performDay, timeDuration, minTokenBalance und lastPerformTimestamp.
2. Der Contract initiiert batchProcess an einem bestimmten Datum des Monats mit einem bestimmten Monatsintervall.
3. Führt batchSend durch, indem eine Liste von Wallets und berechneten Kosten mit einer anderen Methode namens getCostableWalletBatch erhält. Die batchSend-Methode überträgt Token, um die Lagerkosten zu bezahlen, indem sie die operatorSend-Funktion des TokenTransfer-Operator Contracts verwendet. Dieser Vorgang wird so lange wiederholt, bis das Ende der costableWallets erreicht ist.
4. Wir können den StorageOperator Contract neu bereitstellen, wenn wir die Verwaltungsfunktionen aktualisieren müssen, und dann BWGToken und TokenTransferOperator autorisieren.
5. Die Rollen admin, maintainer und batch-executor sind berechtigt, diesen Contract zu aktualisieren. (Zugriffskontrolle)
6. Überschreiben von revokeRole und renounceRole, um Admin-Benutzer einzuschränken. (Zugriffskontrolle)

TokenTransferOperator Contract

Der TokenTransferOperator ist der Standardoperator des BWGToken Contracts und die operatorSend-Methode dieses Contracts kann Token im Namen des Token-Inhabers versenden. Die grundlegenden Merkmale des TokenTransferOperator Contracts sind:

1. StorageOperator Contract autorisieren.
2. Führt operatorSend aus, um Token im Namen des Token-Inhabers zu übertragen. Nur StorageOperator Contracts dürfen auf die operatorSend-Methode zugreifen.
3. Die Rollen admin und maintainer sind berechtigt, diesen Contract zu aktualisieren. (Zugangskontrolle)
4. Überschreiben von revokeRole und renounceRole, um Admin-Benutzer einzuschränken. (Zugriffskontrolle)

BWG Token Contract UML diagram

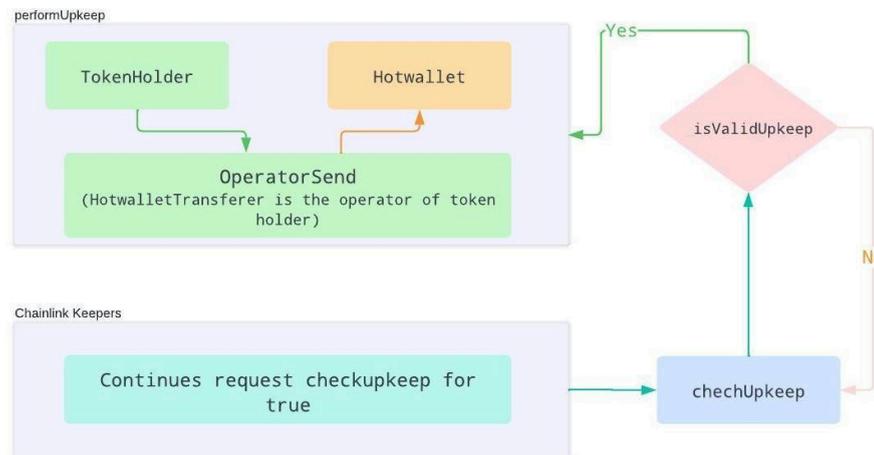


HotwalletTransferer Contract

Der HotwalletTransferer Contract wurde entwickelt, um automatisch die erforderliche Mindestmenge an Token auf der Hotwallet der Bretton Woods Gold-Plattform aufrechtzuerhalten, indem Chainlink-Keeper verwendet werden. Chainlink-Keeper benötigen einige LINK-Token, um diesen Smart Contract zu betreiben. Daher sorgt der Systemadministrator für ein ausreichendes Guthaben an LINK-Token auf dem Konto des Keepers. Jedes Chainlink-Konto hat ein Mindestguthaben an LINK-Token, um sicherzustellen, dass das Konto auch bei einem plötzlichen Anstieg des Guthabens weiterläuft. Wenn Ihr LINK-Guthaben unter diesen Betrag fällt, wird der Upkeep nicht durchgeführt. Die Grundfunktionen des HotwalletTransferer Contracts sind:

1. Automatischer Transfer von Token von einer Herkunftswallet (Token-Halter / Cold-Wallet) zu einer anderen Wallet (Hot-Wallet), wenn der Kontostand der Ziel-Wallet unter einem bestimmten Schwellenwert liegt.
2. Die Chainlink-Keeper sind dafür verantwortlich, diesen Contract zu automatisieren.
3. Die Herkunfts- und Zielwallet können geändert werden.
4. Die Halter von Chainlink Keepern müssen über genügend LINK-Token verfügen, um übertragen zu können.
5. Die Rollen admin und maintainer sind berechtigt, diesen Contract zu aktualisieren. (Zugangskontrolle)
6. Überschreiben von revokeRole und renounceRole, um Admin-Benutzer einzuschränken. (Zugriffskontrolle)

Hotwallet transfer work flow



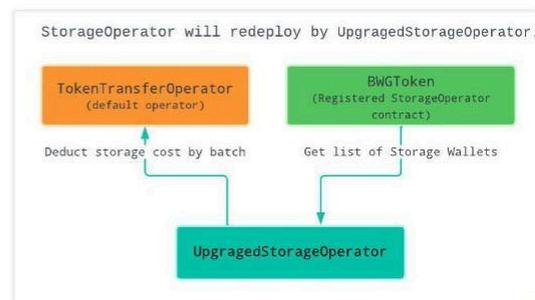
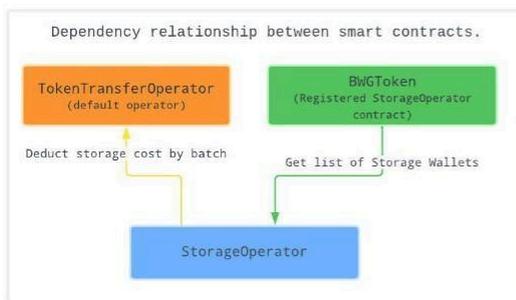
HotWalletTransfer Empfehlung

1. **Wechsel des Token-Inhabers:** Vor dem Wechsel des Token-Inhabers sollte der Systemadministrator den neuen Inhaber für den HotWalletTransferer Contract autorisieren, um einen fehlerfreien Transfer zu gewährleisten. Der performUpkeep schlägt weiterhin fehl und das System verliert LINK-Token, wenn es nicht vor dem Wechsel des Tokeninhabers einen neuen Inhaber autorisiert.
2. **Sicherstellung eines Mindestguthabens an LINK-Token:** Chainlink-Keeper benötigen einige LINK-Token, um diesen Smart Contract zu betreiben, so dass wir sicherstellen einen positiven Saldo aufzuweisen.

StorageOperator Umstrukturierung

Wir haben die Möglichkeit, den StorageOperator Contract neu einzusetzen, wenn wir Verwaltungsfunktionen aktualisieren, z. B. die Einstellungen für den Abzug von Lagerkosten, die von der vorhandenen Implementierung nicht abgedeckt werden kann. Der Beispielcontract "UpgradedStorageOperator" und das Bereitstellungsskript befinden sich im Ordner "redploy_scrips".

Abhängigkeitsverhältnis zwischen Smart Contracts.

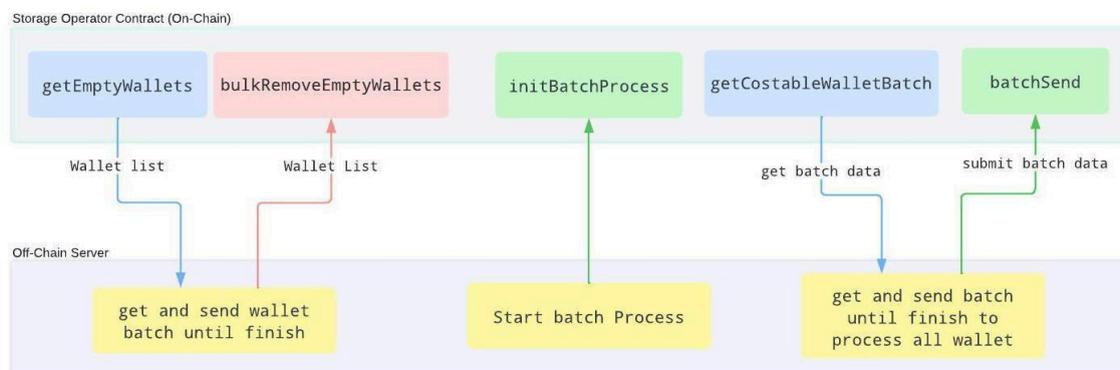


Abzug von Goldlagerkosten (Batch-Verfahren)

Der StorageOperator Contract hat die Fähigkeit, Lagerkosten einzuziehen, indem er den TokenTransferOperator Contract mit der batchSend-Methode verwendet. Zunächst starten Off-Chain-Server den Batch-Prozess, indem sie eine initBatchProcess-Anforderung anfordern. Dann wird die batchSend-Methode ausgelöst, indem Wallets und Beträge von der getCostableWallet- Batch-Methode gesammelt werden. Dieser Prozess wird bis zum letzten Index der storageWallet-Liste fortgesetzt.

Die folgenden Vorgänge müssen während des Batch-Prozesses durchgeführt werden:

1. Abzug der Lagerkosten von allen Token-Inhabern, wenn diese nicht in der costfreeWallet sind.
2. Unterbrechung der Tokenübertragung nach Beginn der initBatchProcess-Anforderung und Aufhebung der Unterbrechung nach deren Beendigung.
3. Der batch-executor kann den Batch-Prozess ausführen (initBatchProcess und batchSend).
4. Der batch-executor kann die Liste der leeren Wallets aus den storageWallets entfernen.
5. Der maintainer kann während des Batch-Prozesses keinen StorageOperator Contract Zustand aktualisieren.
6. Sammeln des Restsaldos nach dem Abzug, wenn er zu gering ist oder unter dem Mindestschwellenwert für den Tokensaldo liegt.



Kommunikation zwischen Off-Chain-Server und StorageOperator während eines Batch-Prozesses.

Wenn ein Batch Process durchgeführt wird

Standardmäßig wird der Batch-Prozess alle drei Monate (vierteljährlich) mit einem festen Monat wie (Januar, April, Juli, Oktober) ausgeführt. Er hängt nicht vom Einsatzdatum des Contracts ab. Obwohl `timeDuration` konfigurierbar ist, kann es monatlich, vierteljährlich, halbjährlich und jährlich eingestellt werden. Wenn `timeDuration` auf monatlich eingestellt ist, wird der Batch-Prozess jeden Monat zum angegebenen Datum ausgeführt.

Mechanismus zum Schutz vor Dust (kleine Menge an BWG-Token)

Für den Abzug der Lagergebühren haben wir einen statischen Wert namens `minTokenBalance` im Smart Contract. Dieser Wert wird auf einen vernünftigen Schwellenwert gesetzt, um zu verhindern, dass kleine Mengen von BWG-Token in einer Adresse liegen und Lagergebühren generieren, während ihr Wert zu gering ist, um für irgendetwas effektiv genutzt zu werden.

Das bedeutet, dass wir während des Abzugs von Lagergebühren von einem Nutzer alle verfügbaren Token von dieser Wallet abziehen, wenn der verbleibende Betrag nach einem Abzug geringer ist als der Mindestschwellenwert für das Token-Guthaben.

Leere Wallets aus der Walletsliste entfernen

`StorageOperator` hat die Möglichkeit, mit Hilfe des `BWGToken Contract` eine leere Wallet aus der Liste der Lagerwallets zu entfernen. Dies kann vor oder nach dem Batch-Prozess geschehen. Zunächst sammelt der On-Chain-Server eine Liste des ersten verfügbaren Batch leerer Wallets aus der `getEmptyWallets`-Methode und sendet sie dann an `bulkRemoveEmptyWallets`.

Umgang mit Costfree Wallets

Die `costfree` (kostenfreie) Wallet ist eine spezielle Art von Wallet, bei der die darin enthaltenen Token nicht für die Lagerkostenabzüge berücksichtigt werden. Der `maintainer` kann `costfree` Wallets hinzufügen oder entfernen. Die `costfree` Wallets werden nur für interne Wallets der *BrettonWoods digital AG* verwendet, um Lagerkostenabzüge von internen Wallets und damit unnötige Transaktionsgebühren zu vermeiden. Die *Bretton Woods digital AG* muss die Lagerkosten ohnehin direkt mit den Lagerpartnern abrechnen und braucht dafür keine Token aus einer Firmen-Wallet zu sammeln.

Rollenbasierter Zugriff

Die Zugriffskontrolle bietet einen allgemeinen rollenbasierten Zugriffskontrollmechanismus. Es können mehrere hierarchische Rollen erstellt und zugewiesen werden, jede für mehrere Konten. Jede Rolle kann nur die Aktionen (Ausführungsfunktionen) ausführen, die ihr zugewiesen wurden. Eine Rolle kann mehreren Wallets zugewiesen werden. Ein Administrator kann andere Rollen gewähren oder entziehen.

Für den BWGToken Contract haben wir zum Beispiel zwei verschiedene Rollen eingeführt:

Eine heißt admin (für zusätzliche Sicherheit auf gnosis safe multisig wallet eingestellt) und die andere heißt maintainer. Der Administrator kann jeder Wallet jede Rolle zuweisen oder entziehen. Admin ist eine gnosis safe Wallet, die die ihr zugewiesenen Aktionen ausführen kann, wobei sie die Zustimmung mehrerer Besitzer benötigt. Auf der Grundlage von Smart Contracts werden verschiedene Rollen erstellt und für verschiedene Aufgaben zugewiesen.

- Der BWGToken Contract hat die Rollen admin und maintainer.
- Der StorageOperator Contract hat die Rollen admin, maintainer und batch-executor.
- Der TokenTransferOperator Contract hat die Rollen admin und maintainer.
- Der HotWalletTransferer Contract hat die Rollen admin und maintainer.

Admin: Ein admin kann anderen Rollen, einschließlich der Rolle des admin, Rechte gewähren oder entziehen. Aber er kann seine eigene Rolle nicht widerrufen oder aufgeben. Im Allgemeinen kann ein admin eine Cold Wallet oder eine Gnosis Safe Multisig Wallet sein.

Maintainer: Der maintainer kann bestimmte Zustände des Smart Contracts hinzufügen/aktualisieren/entfernen. Das kann eine einzelne Wallet sein.

Batch-executor: Der batch-executor kann einen monatlichen Batch-Prozess durchführen. Es sollte eine individuelle Wallet sein, damit ein automatisiertes Skript den Batch-Prozess in einem bestimmten Zeitintervall abwickeln kann.

Die Liste der Operationen kann je nach Benutzerrolle durchgeführt werden.

Rolle	Typ	Rolle gewähren durch	Verzicht auf die Rolle	Rolle widerrufen durch	Smart Contract
admin	Gnosis Safe Multisig-Wallet	admin, maintainer und batch executor	– Anderer admin – Maintainer und batch executor	Nein	BWGToken, StorageOperator, TokenTransferOperator HotWalletTransferer
maintainer	Omdovodieööe Waölet	Nein	Nein	Ja	BWGToken, StorageOperator, TokenTransferOperator HotWalletTransferer
batch-executor	Individuelle Wallet	Nein	Nein	Ja	StorageOperator

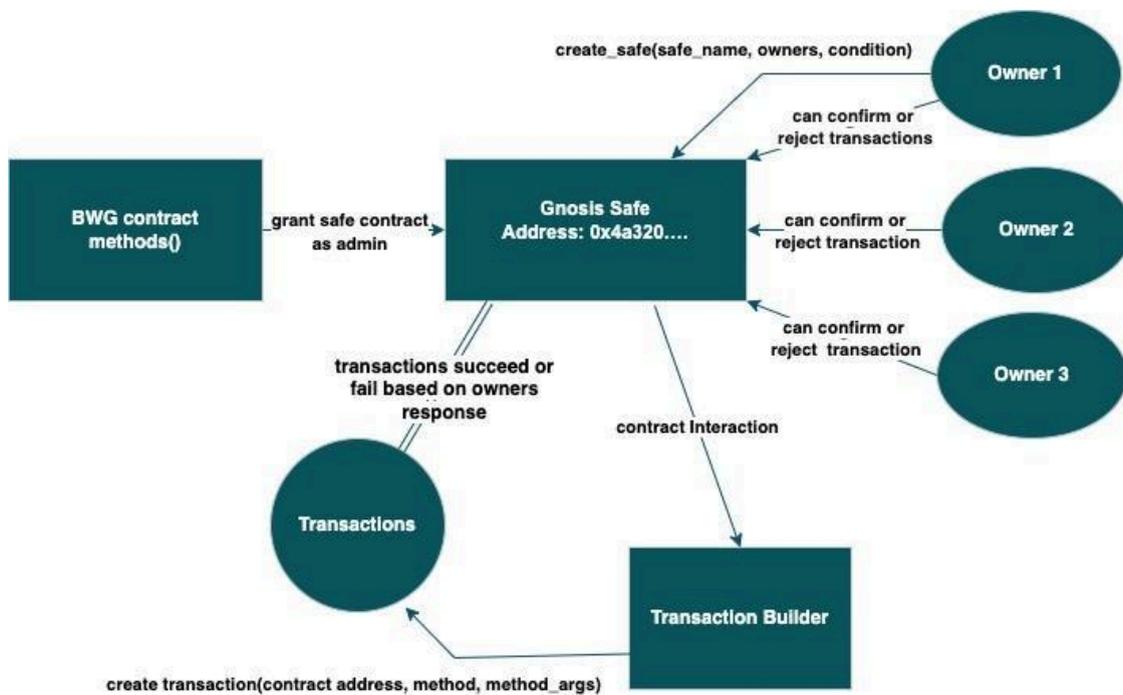
Gnosis Safe

Gnosis Safe ist eine Smart Contract Wallet, die auf Ethereum läuft und eine Mindestanzahl von Personen benötigt, um eine Transaktion zu genehmigen, bevor sie stattfinden kann.

Wie funktioniert das?

Gnosis Safe ist eine Multi-Signatur Smart Contract Wallet, die es Nutzern ermöglicht, eine Liste von Eigentümern/Unterzeichnern und eine Mindestanzahl von Unterzeichnern zu definieren, die zur Bestätigung einer Transaktion erforderlich sind. Sobald der Schwellenwert an Eigentümerkonten eine Transaktion bestätigt hat, kann die Safe-Transaktion ausgeführt werden.

Multisig (z.B. 2 von 3 Eigentümern) Transaktion



BWG-Token Contract Übersicht

Name der Methode	Anfrage Typ	Parameter	Rolle	Kommentare
allowance	Read/Lesen	Adressbesitzer, Adressabgeber	Nein	Gibt die verbleibende Anzahl von Token zurück, die der Ausgeber im Namen des Besitzers durch transferFrom ausgeben darf. Dieser Wert ist standardmäßig Null.
balanceOf	Read/Lesen	Adresskonto	Nein	Gibt die Anzahl der Token zurück, die sich im Besitz des Kontos befinden.
decimals	Read/Lesen		Nein	Gibt die Anzahl der Dezimalstellen zurück, die verwendet wurden, um die Benutzerdarstellung zu erhalten.
defaultOperators	Read/Lesen		Nein	Gibt die Liste der Standardoperatoren zurück. Diese Konten sind Operatoren für alle Token-Inhaber, auch wenn authorizeOperator nie für sie aufgerufen wurde.
granularity	Read/Lesen		Nein	Gibt den kleinsten Teil des Tokens zurück, der nicht teilbar ist. Das bedeutet, dass alle Token-Operationen (Erstellung, Bewegung und Zerstörung) Beträge haben müssen, die ein Vielfaches dieser Zahl sind
Name	Read/Lesen		Nein	Gibt den Namen des Tokens zurück.
symbol	Read/Lesen		Nein	Gibt das Symbol des Tokens zurück.
totalSupply	Read/Lesen		Nein	Gibt den Gesamtvorrat des Tokens zurück.
costfreeWallets	Read/Lesen		Nein (storageOperator)	Gibt die Liste der costfreeWallets des Tokens zurück.
storageWallets	Read/Lesen		Nein (storageOperator)	Gibt die Liste der storageWallets bei Aufruf von storageOperator zurück, ansonsten leere Liste.
isCostableWallet	Read/Lesen		Nein	Gibt true zurück, wenn der Token-Inhaber nicht in der costfree-Liste steht, sonst false.
isCostfreeWallet	Read/Lesen		Nein	Gibt true zurück, wenn die „Wallet in die costfree Liste aufgenommen wurde, sonst false.
getBatchSize	Read/Lesen		Nein	Die externe Methode wird verwendet, um die aktuelle Batch-Größe zu ermitteln.
getEmptyWallets	Read/Lesen		Nein (storageOperator)	Die Methode getEmptyWallets wird verwendet, um alle verfügbaren Wallets mit einem leeren Guthaben zu finden.
addCostfreeWallet	Write/Schreiben	address wallet	Nein (storageOperator)	Die Methode wird verwendet, um die Wallet zur Liste der kostenfreien Wallets hinzuzufügen und ein AddCostFreeWallet Event auszulösen. Sendet ein

				AddCostFreeWallet Event aus.
removeCostfreeWallet	Write/Schreiben	address wallet	Nein (storageOperator)	Die Methode wird verwendet, um die Wallet aus der Liste der kostenfreien Wallets zu entfernen und das Event RemoveCostFreeWallet auszulösen. Sendet ein Event "RemoveCostFreeWallet" aus.
bulkRemoveEmptyWallets	Write/Schreiben	address [] wallet	Nein (storageOperator)	Die Methode wird verwendet, um Wallets mit Nullsaldo aus der Liste storageWallet zu entfernen.

Name der Methode	Anfrage Typ	Parameter	Rolle	Kommentare
setStorageOperator	Write/Schreiben	Address storage_operator_address	maintainer	Die Methode wird verwendet, um die Adressen von Lageroperator Contracts zu aktualisieren, damit gültige Contacts Zugriff auf die Liste der Wallets erhalten.
Gibt ein StorageOperator-Event aus."	Read/Lesen	Address account	Nein	Gibt die Anzahl der Token zurück, die sich im Besitz des Kontos befinden.
setBatchSize	Write/Schreiben	uint256	maintainer	Die externe Methode wird zur Aktualisierung der Batch-Größe verwendet.
Gibt ein BatchSize Event aus."	Read/Lesen		Nein	Gibt die Liste der Standardoperatoren zurück. Diese Konten sind Operatoren für alle Token-Inhaber, auch wenn authorizeOperator nie für sie aufgerufen wurde.
bulkSend	Write/Schreiben	address[] wallets, uint256[] costs	Nein	Die Methode wird verwendet, um Token in einer einzigen Transaktion an mehrere Wallets zu senden
pause	Write/Schreiben		Nein (storageOperator)	Die Methode wird verwendet, um die Übertragung von Token zu unterbrechen. Gibt ein Pause Event aus. Anforderungen: der Vertrag darf nicht pausiert werden.
unpause	Write/Schreiben		Nein (storageOperator)	Die Methode wird verwendet, um die Pause bei der Tokenübertragung aufzuheben Gibt ein Unpaused Event aus. Anforderungen: Der Contract muss pausiert werden.
burn	Write/Schreiben	uint256 amount, bytes data	Nein	Zerstört Token vom Konto des Callers, wodurch der Gesamtbestand verringert wird. Gibt ein Burned Event aus. Anforderungen: Der Caller muss mindestens eine bestimmte Anzahl von Token haben."

Name der Methode	Anfrage Typ	Parameter	Rolle	Kommentare
send.	Write/Schreiben	address recipient, uint256 amount, bytes data	Nein	Überträgt den Betrag der Token vom Konto des Callers an den Empfänger. Sendet ein Event. Anforderungen: Der Caller muss mindestens über eine bestimmte Anzahl von Token verfügen. Der Empfänger kann nicht die Null Adresse sein.
operatorSend	Write/Schreiben	address sender, address recipient, uint256 amount, bytes data, bytes operatorData	Nein	Verschiebt die Menge der Token vom Sender zum Empfänger. Der Aufrufer muss ein Operator des Absenders sein. Sendet ein Event aus. Anforderungen: Der Absender kann nicht die Null Adresse sein. Der Absender muss mindestens eine bestimmte Anzahl von Token haben. Der Anrufer muss ein Operator für den Absender sein.
authorizeOperator	Write/Schreiben	address operator	Nein	Macht ein Konto zu einem Operator des Callers.. Gibt ein AuthorizedOperator-Ereignis aus. Anforderungen: Operator kann nicht die Caller-Adresse sein.
transfer	Write/Schreiben	address recipient, uint256 amount	Nein	Verschiebt die Tokenmenge vom Konto des Callers zum Empfänger. Gibt einen booleschen Wert zurück, der angibt, ob der Vorgang erfolgreich war. Sendet ein Transfer Event.
approve	Write/Schreiben	address spender, uint256 amount	Nein	Legt amount als Freibetrag des Senders über die Token des Callers fest. Gibt einen booleschen Wert zurück, der angibt, ob der Vorgang erfolgreich war.
transferFrom	Write/Schreiben	address sender, address recipient, uint256 amount	Nein	Verschiebt den Betrag der Token vom Absender zum Empfänger unter Verwendung des Vergütungsmechanismus. Der Betrag wird dann von der Vergütung des Callers abgezogen. Gibt einen booleschen Wert zurück, der angibt, ob der Vorgang erfolgreich war. Sendet ein Übertragungs-Event
revokeRole	Write/Schreiben	bytes32 role, address account	admin	Durch die Überschreibung der Zugriffskontrolle revokeRole können sie alle Rollen außer der Adminrolle widerrufen.
renounceRole	Write/Schreiben	bytes32 role, address account	admin	Override Access control renounceRole ermöglicht es ihnen, auf alle Rollen außer der Adminrolle zu verzichten.



BWG Token Events

Benutzerdefinierte Ereignisse:

StorageOperator(address wallet)

AddCostFreeWallet(address wallet)

RemoveCostFreeWallet(address wallet)

RemoveCostableWallet(address wallet)

BatchSize(uint256 batchSize)

ERC-777 Events:

Sent(address operator, address from, address to, uint256 amount, bytes data, bytes operatorData)

Minted(address operator, address to, uint256 amount, bytes data, bytes operatorData)

Burned(address operator, address from, uint256 amount, bytes data, bytes operatorData)

AuthorizedOperator(address operator, address tokenHolder)

RevokedOperator(address operator, address tokenHolder)

ERC-20 Events:

Transfer(address from, address to, uint256 value)

Approval(address owner, address spender, uint256 value)

Pausable

Paused(address account)

Unpaused(address account)

StorageOperator Contract Übersicht

Name der Methode	Anfrage Typ	Parameter	Rolle	Kommentare
storageWalletCount	Lesen		Nein	Gibt die Anzahl der Wallets von Token-Inhaber zurück.
costFreeWallets	Lesen		Nein	Gibt die Liste der Wallets der Token-Inhaber zurück, die keine Lagerkosten zahlen.
costFreeWalletCount	Lesen		Nein	Gibt die Anzahl der Wallets der Token-Inhaber zurück, die keine Lagerkosten zahlen.
getBatchCursor	Lesen		Nein	Methode, die verwendet wird, um den lastIndex des aktuellen Batch-Cursors während des Batch-Prozesses zu ermitteln.
getBatchSize	Lesen		Nein	Methode, die zur Ermittlung der Chargengröße des Batch-Prozesses verwendet wird.
getCostableWalletBatch	Lesen		Nein	Externe Methode, die verwendet wird, um alle Informationen über den nächsten Batch zu erhalten. Gibt zurück: address[] wallets, uint256[] costs, uint256 totalCost, bool isCompleted **Valid batch request required.
getFundAddress	Lesen		Nein	Gibt aktualisierte funcaddress zurück.
getLastPerformTimestamp	Lesen		Nein	Rückgabe des zuletzt aktualisierten Zeitstempels der Batch-Prozessausführung.
getMinTokenBalance	Lesen		Nein	Rückgabe des zuletzt aktualisierten Wertes von minTokenBalance .
getPerformDay	Lesen		Nein	Gibt Tag des Monats des Batch-Prozesses zurück.
getPerformTimeDuration	Lesen		Nein	Gibt Interval des Batch-Prozesses zurück. Wie 1: Monatlich 3: Vierteljährlich 6: Halbjährlich 12: Jährlich
getStorageCostRate	Lesen		Nein	Rückgabe der StorageCostRate des Batch-Prozesses.
getTransactionFee	Lesen		Nein	Gibt die durchschnittliche Überweisungsgebühr jedes Users während des Batch-Prozesses abgezogen.
getEmptyWallets	Lesen		Nein	Externe Methode, die verwendet wird, um das erste verfügbare leere Wallet-Array zu erhalten



Name der Methode	Anfrage Typ	Parameter	Rolle	Kommentare
getStorageOverview	Lesen		Nein	Gibt alle alle Informationen über den anstehenden Batch-Prozess zurück. uint256 totalStorageCost, uint256 totalWallet, uint256 totalWalletUnderThreshold, uint256 totalCostUnderThreshold, uint256 date, uint256 duration, uint256 lastPerformTime, uint256 costRate
pause	Lesen		Maintainer	Die Methode wird verwendet, um Token-Transfers zu pausieren Sendet ein Pausenevent. Anforderungen: <ul style="list-style-type: none">der Contract darf nicht pausiert sein.
unpause	Schreiben		Maintainer	Die Methode wird verwendet, um pausierte Token-Transfers nicht mehr zu pausieren. Gibt ein Unpaused-Event aus. Anforderungen: <ul style="list-style-type: none">der Contract muss pausiert sein.
bulkRemoveEmptyWallets	Schreiben	address[] calldata wallets	BatchExecutor	Externe Methode, die zum Entfernen aus der Storage Wallet verwendet wird. **Valid batch request required
initBatchProcess	Schreiben	uint256 walletMaxlimit	BatchExecutor	Externe Methode, mit der der Batch-Prozess eingeleitet wird. Sendet ein InitBatchProcess-Event **Valid batch request required
batchSend	Schreiben	address[] calldata wallets, uint256[] calldata costs, uint256 currentIndex, uint256 totalCost	BatchExecutor	Externe Methode zur Durchführung von Batch-Prozessen zur Erhebung von Lagergebühren Gibt ein StorageCostSummary-Event aus. **Valid batch request required
setStorageCostRate	Schreiben	uint256 storageCostRate	Maintainer	Externe Methode wird verwendet, um storageCostRate einzustellen. Die Fließkommazahl darf nicht als Rate festgelegt werden, so dass die Rate immer das 100-fache der ursprünglichen Rate beträgt. Gibt ein StorageCostRate-Event aus.
setTransactionFee	Schreiben	uint256 transactionFee	Maintainer	Die externe Methode wird zur Aktualisierung von transactionFee verwendet. Sendet ein TransactionFee-Event.

Name der Methode	Anfrage Typ	Parameter	Rolle	Kommentare
setFundAddress	Schreiben	address fundAddress	Maintainer	Diese externe Methode wird zur Aktualisierung der fundAddress verwendet, an die die monatlichen Lagerkosten überwiesen werden. Gibt ein FundAddress-Event aus.
setLastPerformTimestamp	Schreiben	uint256 timestamp	Maintainer	Die externe Methode wird zur Aktualisierung von lastPerformTimestamp verwendet. <ul style="list-style-type: none"> Zum Ändern von lastPerformTimestamp ist die die Rolle maintainer erforderlich. newTimestamp sollte nicht größer sein als der Zeitstempel des Blocks. Darf während des Batch-Prozesses nicht ausgeführt werden. Sendet ein LastPerformTimestamp-Event.
setMinTokenBalance	Schreiben	uint256 minTokenBalance	Maintainer	Die externe Methode wird zur Aktualisierung von minTokenBalance verwendet. <ul style="list-style-type: none"> Maintainer Rolle erforderlich, um minTokenBalance zu ändern. Darf während des Batch-Prozesses nicht ausgeführt werden. Gibt ein MinTokenBalance-Event aus.
setPerformTimeDuration	Schreiben	uint256 duration	Maintainer	Die externe Methode wird verwendet, um das Intervall des Batch-Prozesses zu aktualisieren. Sendet ein PerformTimeDuration-Event.
setPerformDay	Schreiben	uint256 dayOfMonth	Maintainer	Die externe Methode wird zur Aktualisierung von performDay (Tag des Monats) verwendet. Sendet ein PerformDay-Event aus.
setBatchSize	Schreiben	uint256 batchSize	Maintainer	Die externe Methode wird verwendet, um die Transaktionsgröße batchSize zu setzen. Gibt ein BatchSize-Event aus.
addCostFreeWallet	Schreiben	address wallet	Maintainer	Die Methode addCostFreeWallet wird verwendet, um eine kostenfreie Wallet zur Liste hinzuzufügen. <ul style="list-style-type: none"> Maintainer Rolle erforderlich, um minTokenBalance zu ändern. Darf während des Batch-Prozesses nicht ausgeführt werden.
removeCostFreeWallet	Schreiben	address wallet	Maintainer	Die Methode removeCostFreeWallet wird verwendet, um eine Wallet aus der Liste der kostenpflichtigen Wallets zu entfernen. <ul style="list-style-type: none"> Zum Ändern von minTokenBalance ist die Erlaubnis des Maintainers erforderlich. Darf während des Batch-Prozesses nicht ausgeführt werden.
revokeRole	Schreiben	bytes32 role, address account	Admin	Überschreiben der Zugriffskontrolle revokeRole ermöglicht es ihnen, alle Rollen außer der Adminrolle zu widerrufen



renounceRole	Schreiben	address fundAddress	Admin	Überschreiben der Zugriffskontrolle renounceRole ermöglicht es Ihnen, auf alle Rollen außer der Admin-Rolle zu verzichten.
--------------	-----------	------------------------	-------	--

StorageOperator Events

InitBatchProcess(uint256 maxWallet)

LastPerformTimestamp(uint256 timestamp)

MinTokenBalance(uint256 minBalance)

PerformTimeDuration(uint256 duration)

PerformDay(uint256 day)

StorageCostRate(uint256 storageCostRate)

FundAddress(address fundAddress)

BatchSize(uint256 batchSize)

TransactionFee(uint256 fee)

StorageCostSummary(address fundAddress, uint256 storageCostRate, uint256 totalFee)

Name der Methode	Anfrage Typ	Parameter	Rolle	Kommentare
getOperator	Lesen		Nein	Gibt die Adresse des Lagerhaltungskosten Operator Contracts zurück.
setOperator	Lesen	address operator	Ja	Die externe Methode wird zur Einstellung des Operators verwendet. Gibt ein Operator-Event aus.
operatorSend	Lesen	IERC-777 token, address from, address to, uint256 amount	Ja	Der OperatorSend wird verwendet, um Token im Namen von Token-Inhabern zu übertragen.
revokeRole	Schreiben	bytes32 role, address account	Admin	Durch die Überschreibung der Zugriffskontrolle revokeRole können sie alle Rollen außer der Adminrolle widerrufen.
renounceRole	Schreiben	bytes32 role, address account	Admin	Override Access control renounceRole ermöglicht es ihnen, auf alle Rollen außer der Adminrolle zu verzichten.

TokenTransferOperator Events

Operator(address operator)

HotWalletTransferer Contract Überblick

Name der Methode	Anfrage Typ	Parameter	Rolle	Kommentare
checkUpkeep	Lesen		Nein	Die checkUpkeep Methode ist eine externe KeeperCompatibleInterface Methode, mit der geprüft wird, ob upkeepNeeded erforderlich ist.
performUpkeep	Schreiben		Nein	Die performUpkeep Methode ist eine externe KeeperCompatibleInterface Methode, die verwendet wird, um performUpkeep auszuführen, wenn sie von der checkUpkeep Methode true erhält. Gibt ein TransferToHotWallet-Event aus.
setHolderAddress	Schreiben	address newHolder	Maintainer	Die externe Methode wird verwendet, um den Halter des Tokens zu aktualisieren. Gibt ein HolderAddress-Event aus.
setHotWalletAddress	Schreiben	address hotwallet	Maintainer	Die externe Methode wird zur Aktualisierung der Hotwallet-Adresse verwendet. Gibt ein HotWalletAddress-Event aus.
revokeRole	Schreiben	bytes32 role, address account	Admin	Durch die Überschreibung der Zugriffskontrolle revokeRole können alle Rollen außer der Adminrolle widerrufen.
renounceRole	Schreiben	bytes32 role, address account	Admin	Override Access control renounceRole erlaubt es, auf alle Rollen außer der Adminrolle zu verzichten.

HotWalletTransferer Events

TransferToHotWallet(address indexed holder, address indexed hotWallet, uint256 value)

HotWalletAddress(address indexed hotWalletAddress)

HolderAddress(address indexed holder)

TriggerTokensAmount(uint256 triggerTokensAmount)

TokensAmountToAdd(uint256 tokensAmountToAdd)

Coverage Bericht

Datei	Stmts%	Branch	Func%	Zeile%	Uncovered Lines
Contracts	100	97.41	100	100	
BWGToken.sol	100	95.83	100	100	
HotWalletTransferer.sol	100	100	100	100	
StorageOperator.sol	100	97.73	100	100	
TokenTransferOperator.sol	100	100	100	100	

** Bitte verwenden Sie die Nodeversion v16.13.0, um Coverage Berichte zu erhalten. Es funktioniert nicht mit der neuesten Node-Version.

